Quantum Glitchy Period Finding

Searidang Pa and Shelby Kimmel

Department of Computer Science, Middlebury College

INTRODUCTION

Let a discrete function $f: [\mathrm{N}] \rightarrow [\mathrm{W}]$ be a periodic function with period r. If $r < \sqrt{N}$, a quantum algorithm can learn the period r very efficiently. This period finding algorithm is what allows quantum computers to break public key cryptosystems.

What if there are random insertions or deletions (glitches) in the periodic function? Will the quantum period finding algorithm break?

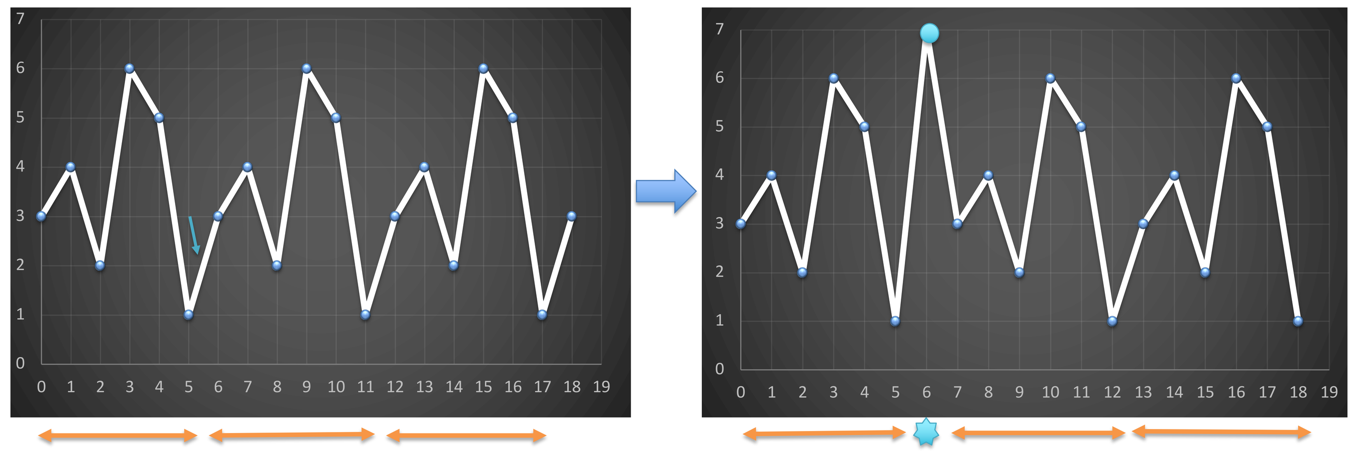

Figure 1: A periodic function before (left) and after one insertion (right).

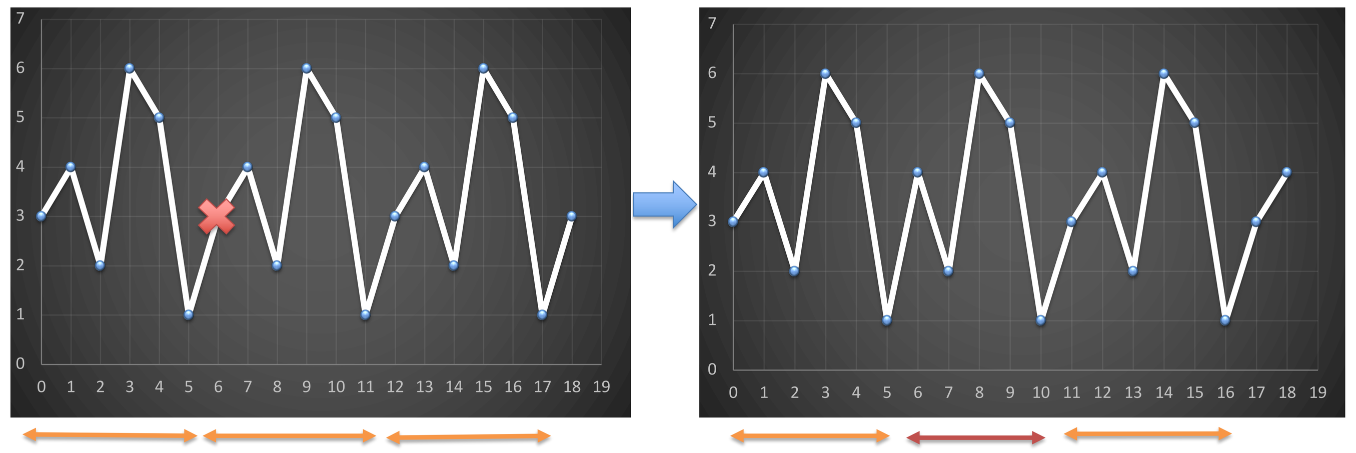

Figure 2: A periodic function before (left) and after one deletion (right).

Motivation

- To test the robustness of quantum algorithms in the presence of errors
- To study the effect of errors on the behavior of the quantum algorithm.

Hypothesis

Without glitches, the algorithm outputs a "good" value with high probability. Good values enable us to learn the period r.

With glitches, we hypothesize that with high probability we get either a good value or a value close to a good value. We could then learn the period r if we look at all points within a certain range of the output of the algorithm.

Numerical Plots

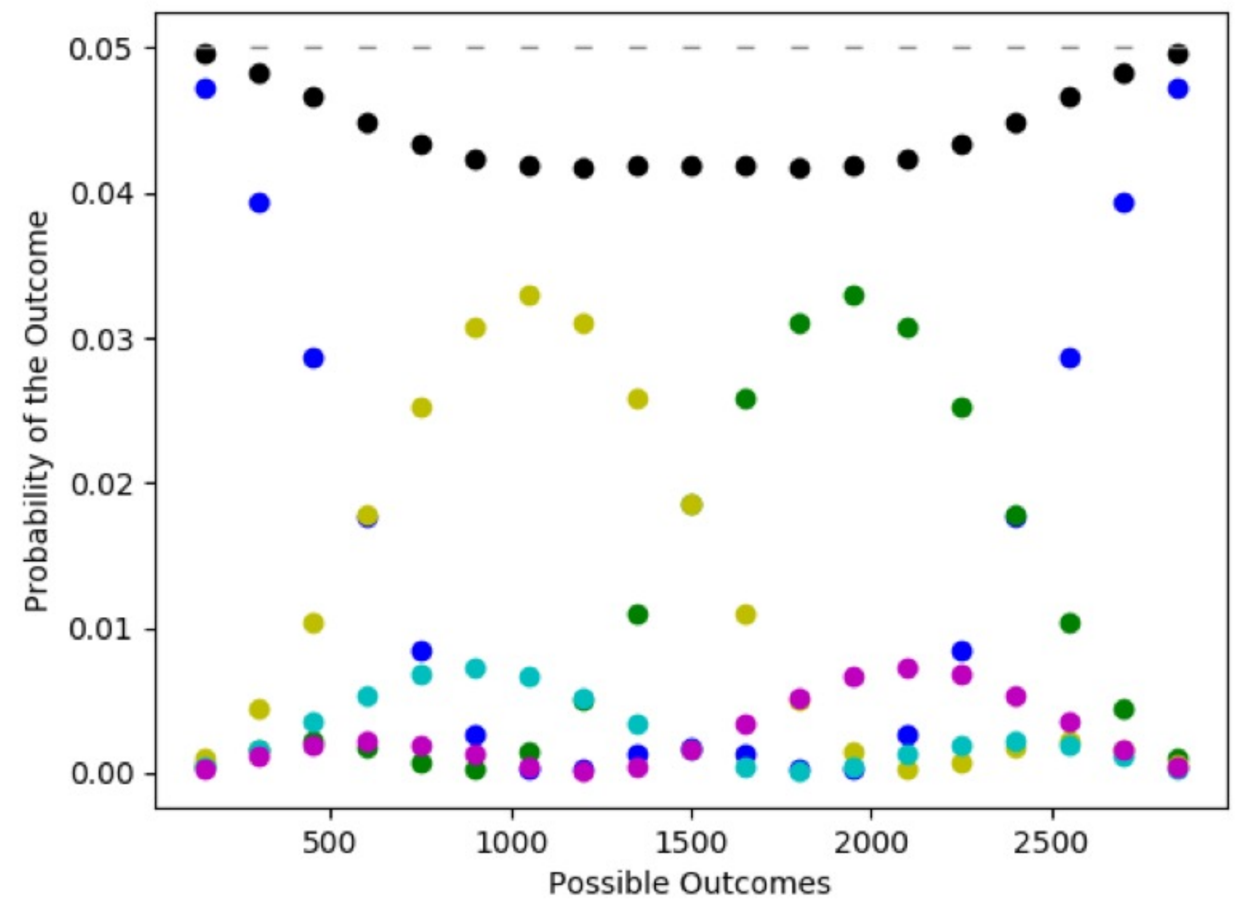

Figure 3: Probabilities of outcomes for N = 3000, r = 20 for 2 glitches. When there are glitches, if the probability of a good value (blue) decreases, the probabilities of nearby outcomes increase.

Legends:

- Probability of a good value when no glitches present
- The sum of probabilities of all outcomes within l of a good value, where l is the number of glitches
- Probability of a good value when glitches are present
- Probability of an outcome -1 away from a good value
- Probability of an outcome +1 away from a good value
- Probability of an outcome -2 away from a good value
- Probability of an outcome +2 away from a good value
- Probability of an outcome -3 away from a good value
- Probability of an outcome +3 away from a good value

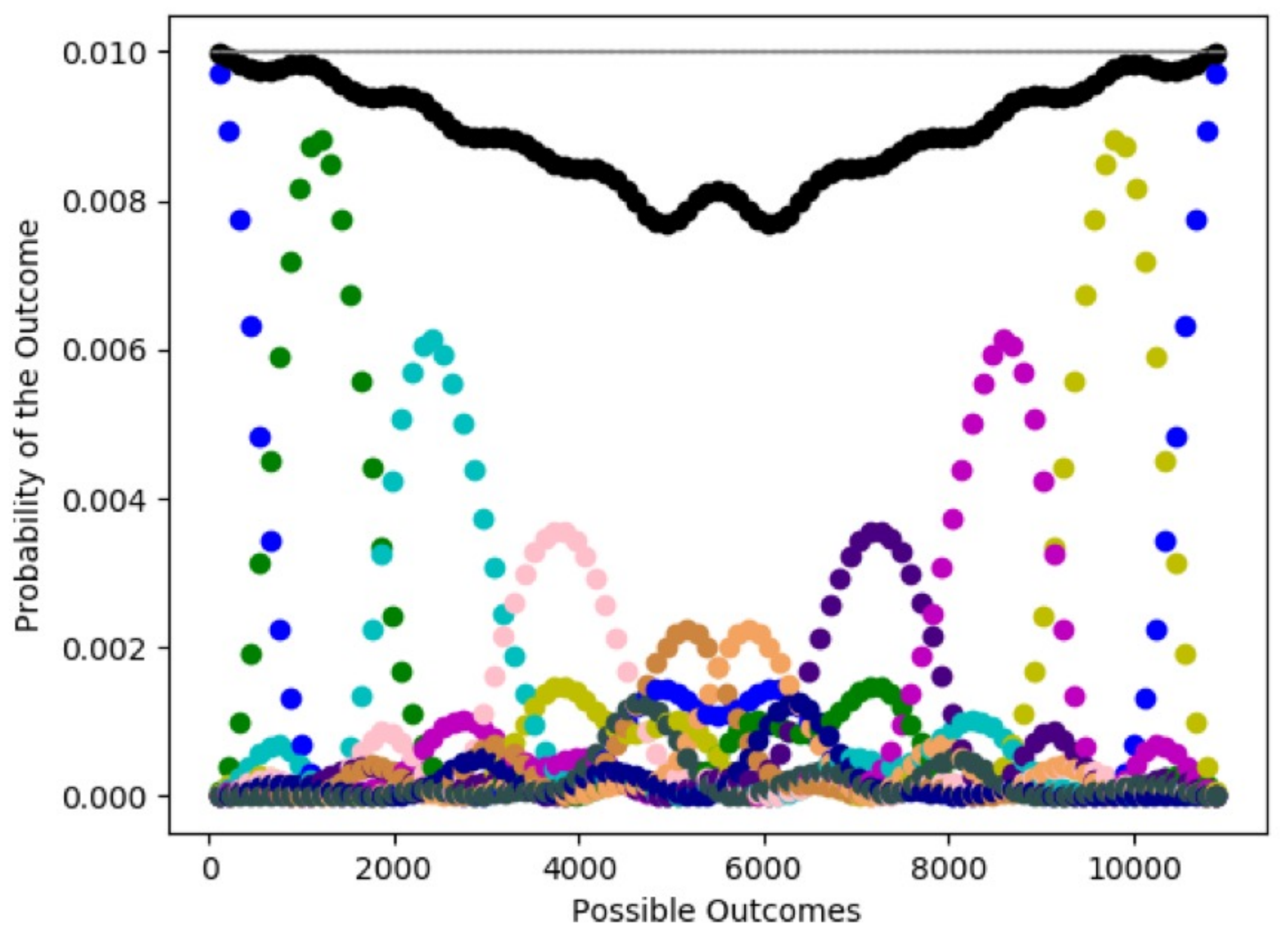

Figure 4: Probability of outcomes for N = 11000, r = 100 for 10 glitches. The sum of all probabilities of the outcomes within the range [-10, 10] of the good values, (black), remains high throughout.

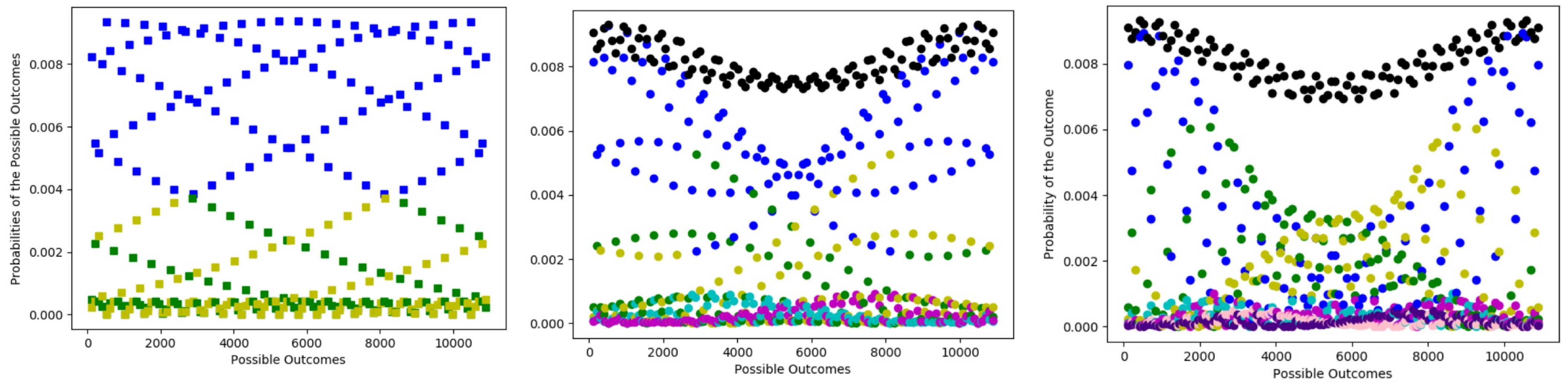

Figure 5: Probability of outcomes for N = 11000, r = 107 for no glitch (left), for 2 glitches (middle), and 3 glitches (right). We see some interesting patterns for the case that r does not divide N.

Progress

We have proved that for the case of one glitch and when r divides N, the probability of getting an outcome in the desirable range is greater than ***0.447*** with respect to the case of no glitch, which means the quantum algorithm would still be successful with high probability with only a constant cost to efficiency.

We are working on the case of multiple glitches, but it involves analyzing this very gruesome expression:

Let $l \in \mathbb{N}$ be the number of glitches. Let $0 < x_1 < x_2 < x_3 < \cdots < x_l < t$. Let $0 < \alpha < 1$. For convenience, let $x_0 = 0$ and $x_{l+1} = t$. We have the following expression

$$A_0 = \sum_{j=0}^{l}(x_j - x_{j-1})^2 + 2\sum_{j>j',1}^{l+1}(x_j - x_{j-1})(x_{j'} - x_{j'-1}) * cos\Big[2\pi\alpha(j-j')\Big].$$

$$A_{hj} = \frac{t}{h\pi} sin\Big[h\pi(x_j - x_{j-1})\Big]$$

$$S_h = 2\sum_{j=1}^{l+1} A_{hj}^2 + 4\sum_{j>j',1}^{l+1} A_{hj}A_{hj'} * cos\Big[2\pi\alpha(j-j')\Big] cos\Big[h\pi(x_{j-1} + x_j - x_{j'-1} - x_j)\Big]$$

$$S = A_0 + \sum_{h=1}^{l} S_h.\ \text{Prove } S \geq 0.5$$

DISCUSSION

While quantum algorithms are often thought to be fragile, this result would imply that certain quantum algorithms can tolerate some errors quite well.

The correctness of the conjecture above would mean that even in the worst case, the cost to the runtime would scale like the number of glitches.

FUTURE WORK

- Expand the proof to the general case of multiple glitches.
- See if the similar approach would also make other quantum algorithm robust in the presence of glitches or errors.