5-2011

Improved Complexity of Quantum Oracles for Ternary Grover Algorithm for Graph Coloring

Marek Perkowski
Portland State University, marek.perkowski@pdx.edu

Yushi Wang

Citation Details

Perkowski, Marek and Wang, Yushi, "Improved Complexity of Quantum Oracles for Ternary Grover Algorithm for Graph Coloring" (2011). *Electrical and Computer Engineering Faculty Publications and Presentations*. 204.
https://pdxscholar.library.pdx.edu/ece_fac/204

Improved Complexity of Quantum Oracles for Ternary Grover Algorithm for Graph Coloring

Yushi Wang

Marek Perkowski
mperkows@cecs.pdx.edu

Department of Electrical and Computer Engineering,
Portland State University

Goals of our research

- Mathematicians who create quantum algorithms (Grover, Shor, Phase Estimation, Many Body Dynamics, etc) do not pay attention to how the circuit is designed and to its complexity
- They write: “let be given Unitary matrix U”
- We want to create practical (gate level) quantum algorithms for practical problems , such as graph coloring or SAT

Goals of our research

- Compare binary versus MVL quantum algorithms
  1) Yale Fan – 2007 Oslo
  2) Yushi Wang – 2011 Tuusula
  3) Vamsi Parasa – 2011 Tuusula

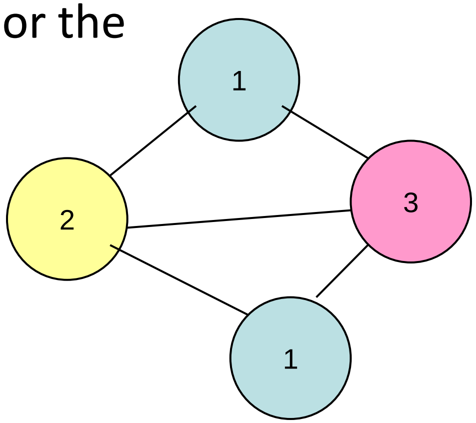
- Compare one-hot and minimum length binary encodings
  1) Sidharth Dhawan – 2011 Tuusula

- Compare various multi-valued encodings.
- Compare various non-minimum length binary encodings.
- Compare various non-minimum length MVL encodings.

- Do these comparisons on realistic quantum costs, on several practical examples

Map and Graph Coloring Problem

- **Map coloring problem:**
  - Given a map with disjoint regions, called countries, color the countries
  - No two countries may be of the same color
  - Minimal number of colors desired.



- **Graph coloring**
  - In a map, replace all countries with nodes, and boundaries with lines
  - Resulting planar graph represents the map.
  - In general, graph is not planar, which means that the chromatic number can be as high as the number of nodes.

Graph coloring problem

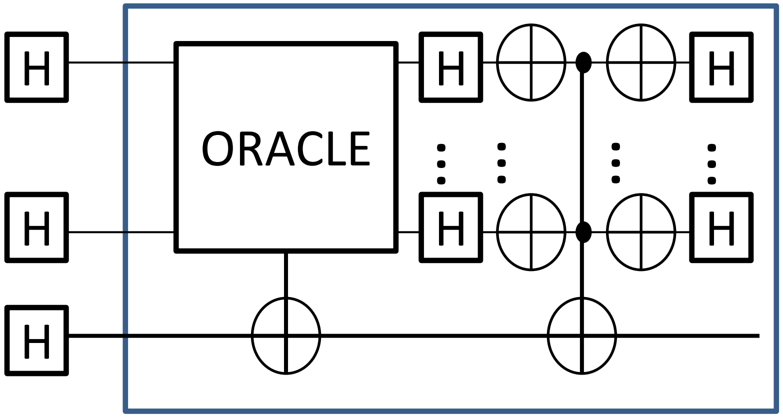
- Represent colors as strings (binary or otherwise)
- Each $node_i$ is given a color, color ($node_i$)
- Adjacent nodes are compared.
  - If color ($node_a$) = color ($node_b$), bad coloring
  - If color ($node_a$) ≠ color ($node_b$), for all adjacent nodes, good coloring

- The Graph Coloring Grover Algorithm has several applications in logic synthesis, scheduling, allocation, planning, robot motion, robot communication, resource allocation, conflict resolution, floor-planning, and many others.

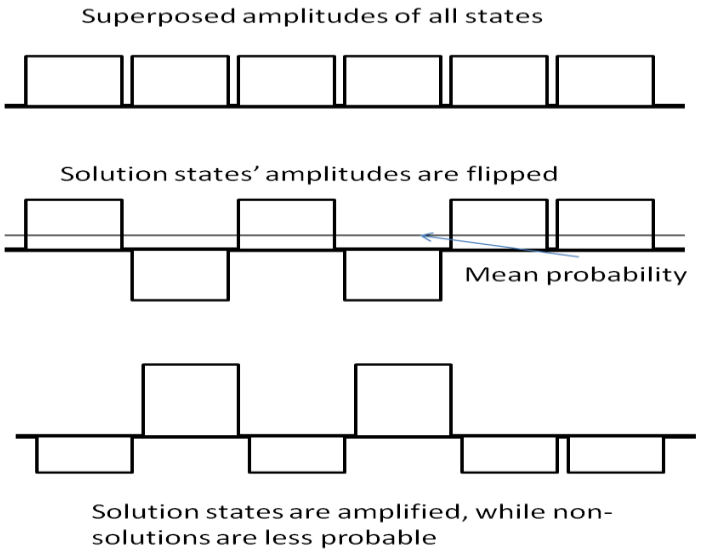
- It can serve as a "generic CSP solver" similar to a general SAT solver.

Quantum Unstructured Search ("Grover")

- **Grover's Algorithm** uses superposition property of quantum information to perform an unstructured search more quickly
- The initial input qubits are superposed to represent all possible solutions
- The **Oracle** operation tags the phase of the solution states in this superposition
- Another circuit then changes the phase information (which is hidden) into amplitude information (which we can detect). inversion-about-the-average operation HZH
- Uses permutative Grover's Oracle based on comparators for edges and a global AND gate.
- This process is iterated √N times (as opposed to N iterations in classical logic) to maximally amplify the states.



Grover Loop (* √N)

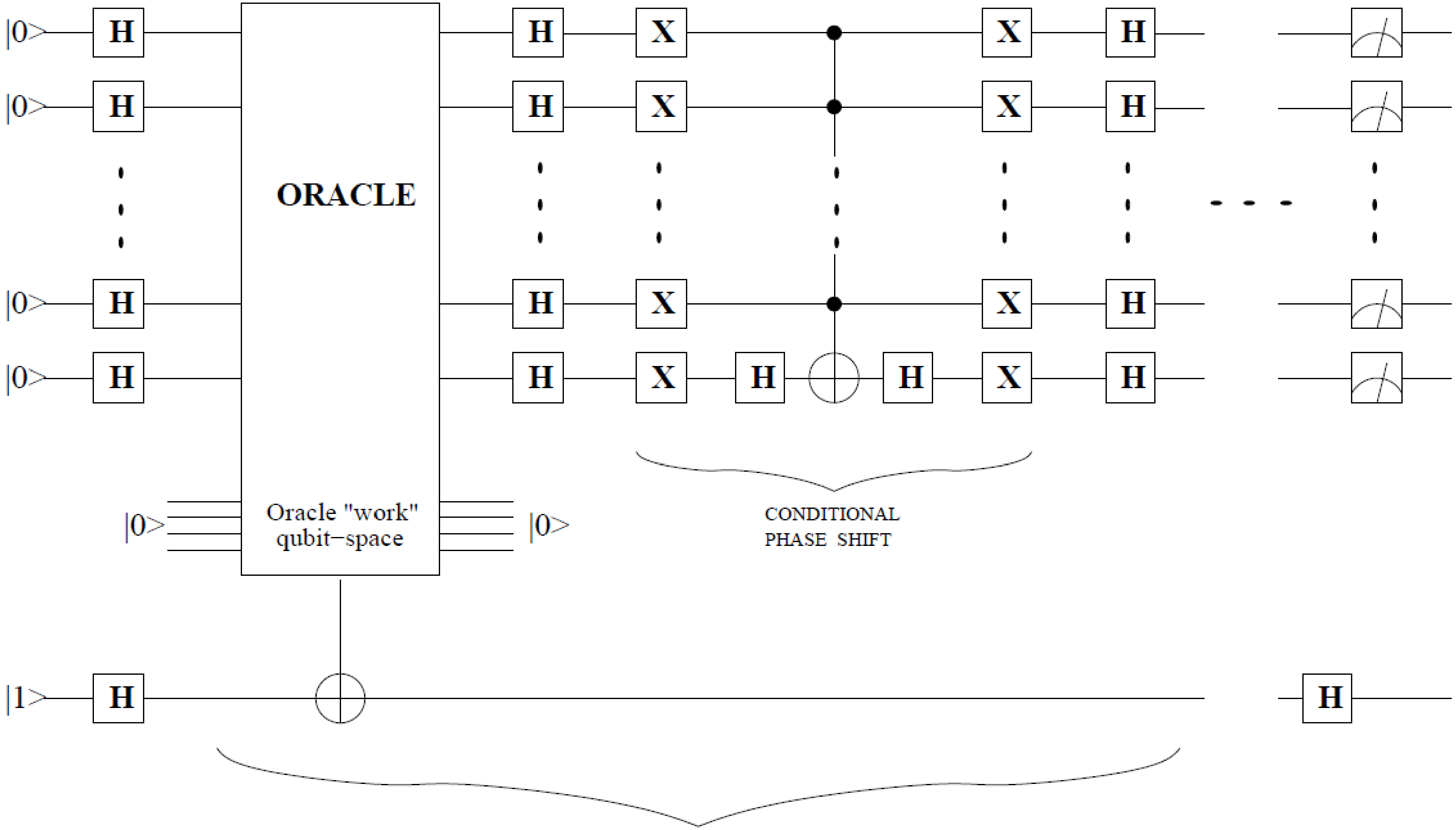

Superposed amplitudes of all states

Solution states' amplitudes are flipped

Mean probability

Solution states are amplified, while non-solutions are less probable

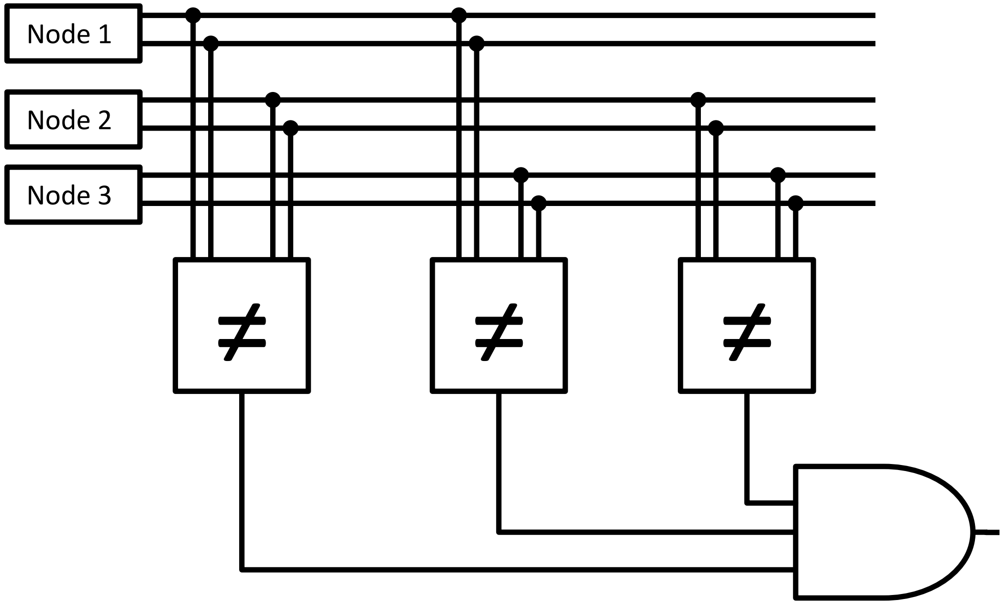
Circuit level representation of Grover



Iterate Grover Loop Square root of N times

Concept of the coloring oracle

- Since there are three nodes, each can take on up to three colors



- Each node is represented with ||log₂(3)|| = 2 qubits so that the total number of possible collective states per node is more than three.

- For example, the states|00> and |01> represent different colors.

- I have to make sure no two nodes that are connected are assigned the same color using bit-by-bit inequality gates.

Ternary Quantum Logic

- **Qutrits** instead of **qubits**; superpositions are now of states
  $|0\rangle$, $|1\rangle$, and $|2\rangle$

- Chrestenson gate is used in place of Hadamard gate:

$$CH = \frac{1}{\sqrt{3}}\begin{bmatrix} 1 & 1 & 1 \\ 1 & d1 & d2 \\ 1 & d2 & d1 \end{bmatrix}$$, where *d*1 = $-(1 + d2) = -\frac{1}{2}(1 - \sqrt{3}i) = e^{\frac{2\pi i}{3}}$

- Controlled-NOT binary gates replaced by single-qutrit reversible functions:

  Controlled(+1), controlled(+2), controled(01),

  Controlled(02), and controlled(12)

9 colors, 4 qubits

0000
0001
0010
0011
0100
0101
0110
0111
1000

9 colors, 2 qutrits

00
01
02
10
11
12
20
21
22

Worst case
9 nodes = 9*8/2=36 edges

36 * 4-qubit comparators
PLUS final Toffoli gate with four inputs in AND

Versus
36 2-qutrit comparators
PLUS final Toffoli gate with 2 inputs in AND

PLUS global Toffoli with 36 INPUTS, this gate can be internally built as binary or ternary, but realizes binary logic AND

Ternary Grover’s

- In 2007, Yale Fan proposed a design for a generalized multi-valued Grover’s Algorithm

- For base n, a vector of quantum Fourier transforms (base n) is applied to the vector of |0⟩ states. (Chrestenson Transform).

- The HZH has been replaced by diffusion operator

$$D = (1 - \omega^{-1})|\psi\rangle\langle\psi| - I$$

CH Z CH

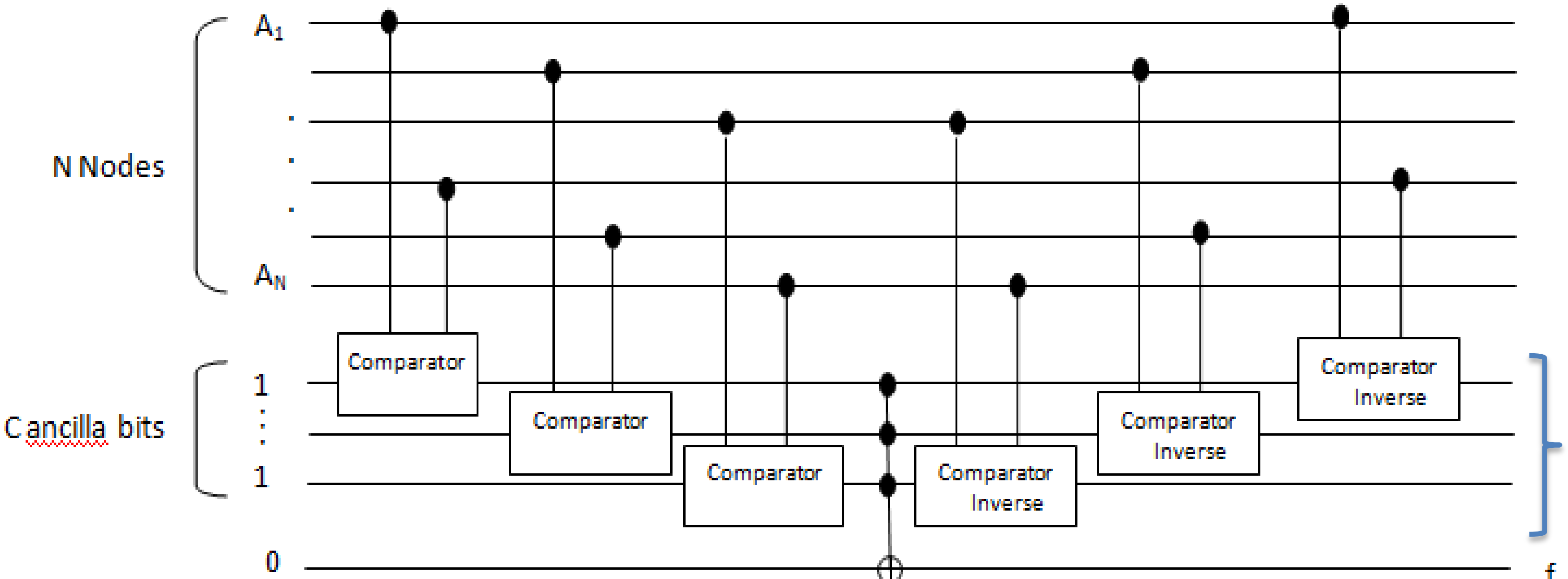
Oracle overview: a general case

- Evaluates a coloring of a map by comparing each pair of connected nodes.

- Each comparison gives either 1 (good coloring of edge) or 0 (bad coloring)

- This is inequality comparator. Output is 1 when colors are different.

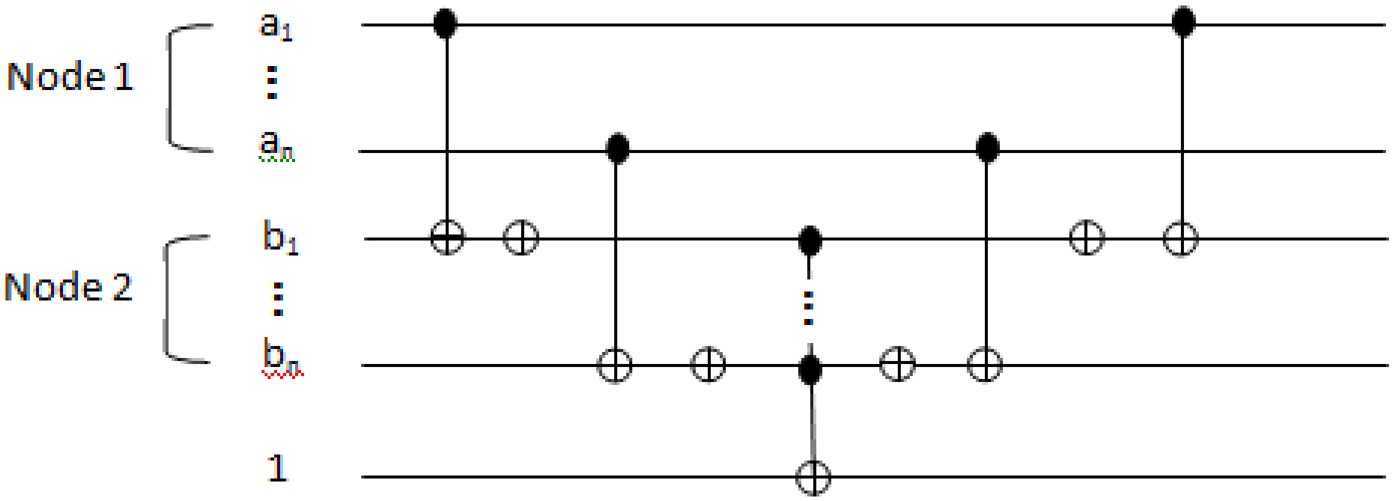
- If all of these comparisons are good, overall output is 1. If the two neighboring nodes have he same color, output = 0.



Can be reused

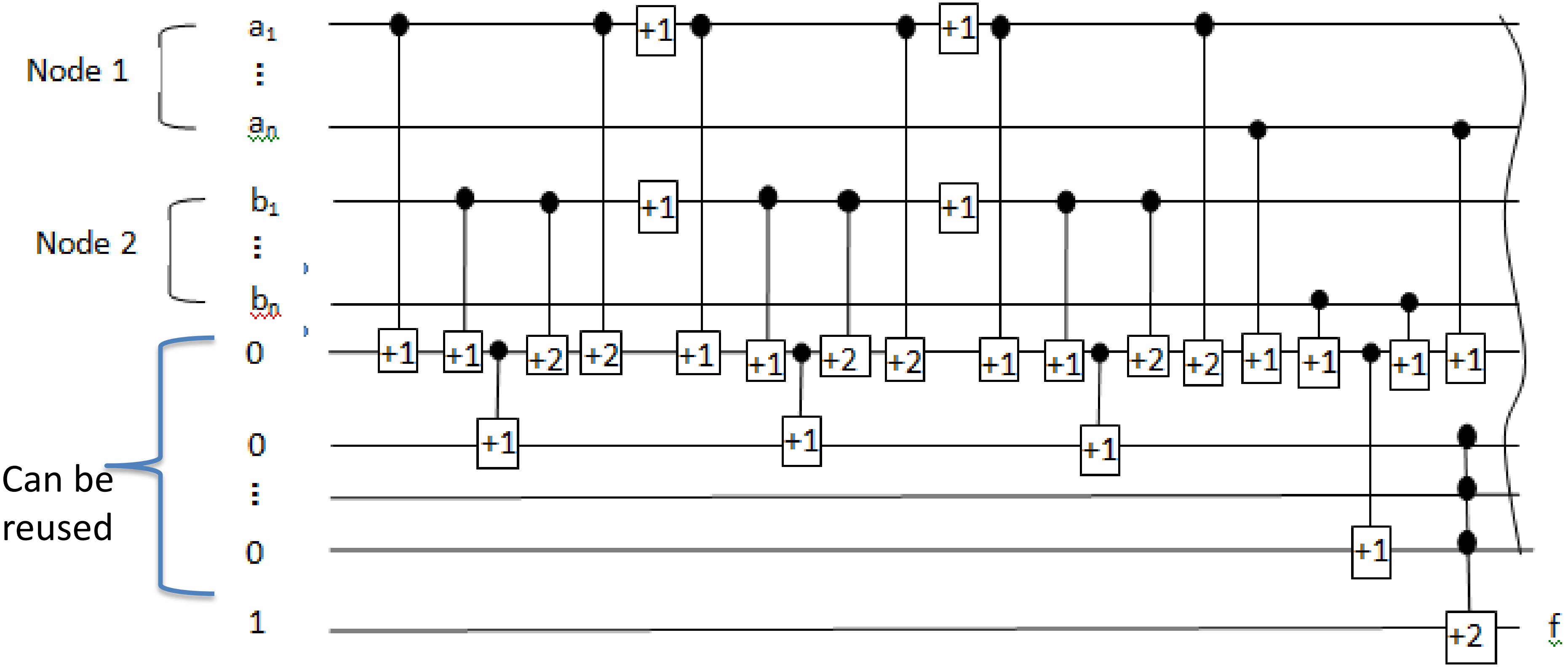
Binary Inequality Comparator circuit

- Compare two strings of digits and operates on an ancilla bit.
- If the two strings are exactly the same, output is flipped from 1 to 0.
- If the two inputs are not exactly the same, output = 1.
- After comparison, the operations are mirrored to return ancilla bit to 1



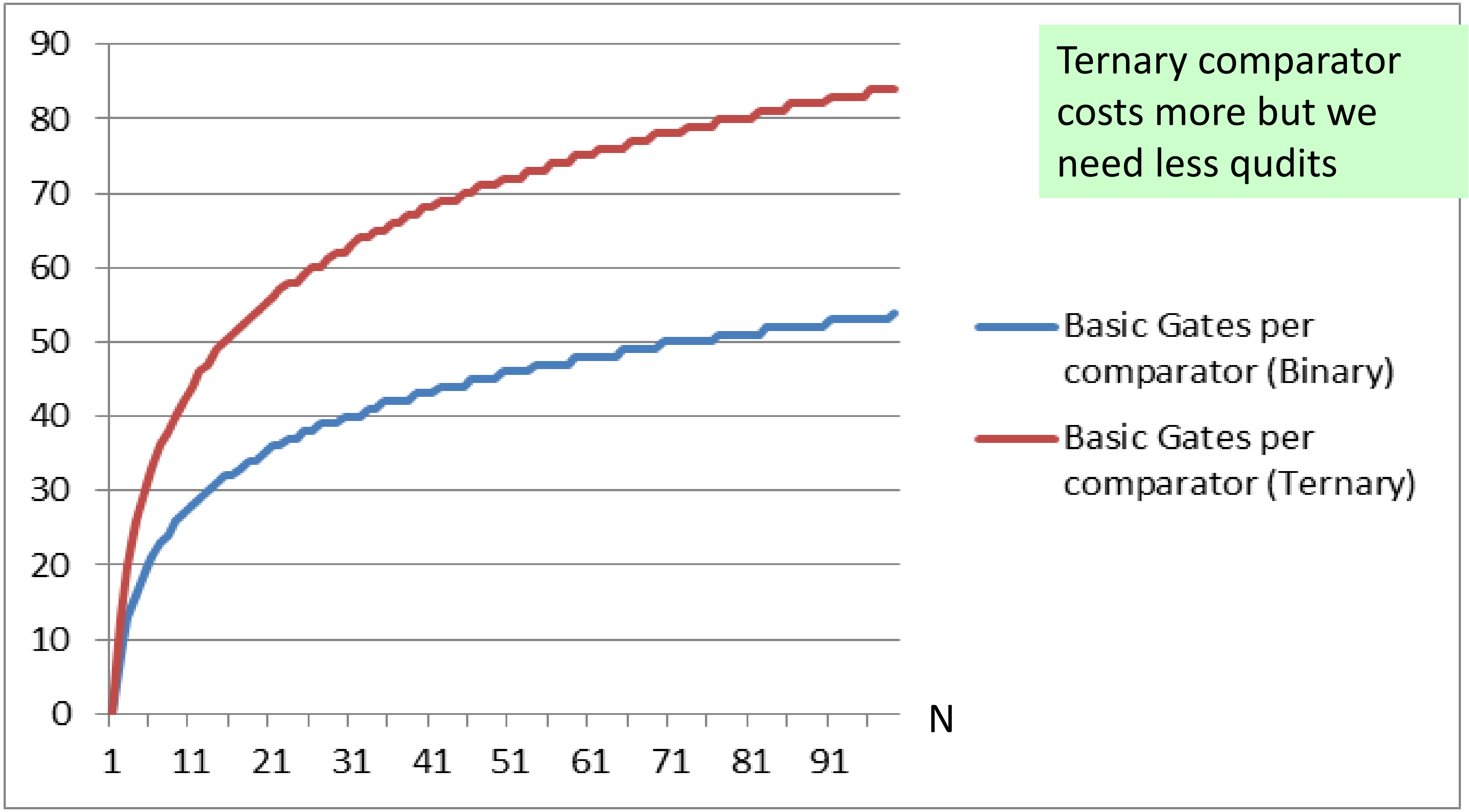
Ternary Quantum Inequality Comparator

- Does the same thing, except in ternary



The curved line at the end of the circuit means mirror.

Costs of binary vs ternary quantum comparators



Ternary comparator costs more but we need less qudits

Figure 7 – Graphic Description of the cost of Comparators, excluding Toffoli.

Costs of binary Toffoli gate with many inputs

- In [3], the cost of the binary *n*-bit Toffoli gate was found to be one of the following:

- For a small number of input gates, such as from 2-5 inputs, the optimal construction of the Toffoli gate currently costs $2^n - 3$ basic quantum operations.

- For larger numbers, the cost grows linearly, at $32n – 96$ gates, plus one garbage bit per input.

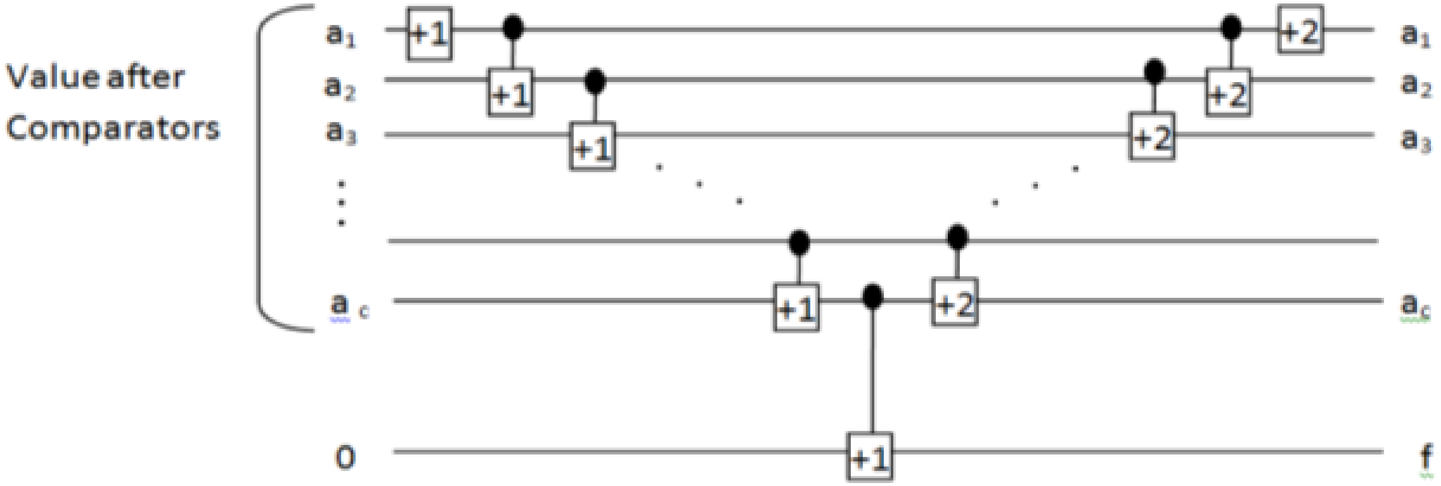
- Thus, the complexity of the Toffoli gate overall grows linearly.

NEW -- Ternary Toffoli

- As shown in previous diagram, a ternary version of the Toffoli gate (multi-controlled NOT gate)

- Proposed multi-controlled NOT gate operates on one ancilla bit and any number of inputs, like the binary Toffoli

- If all inputs equal 1, output is 1. Otherwise, output = 0



The reduced cost of the new Toffoli gate

- Using the same standard cost of 2-qudit gates, the calculated cost of the multi-valued Toffoli gate is significantly lower.

- Using the above diagram, one can construct a multi-valued extension using only $2n + 1$ gates, which is also linear, but at 1/8 of the binary version's cost.

- A graphic comparison of the three is shown in next figure.

Costs of Toffoli gates

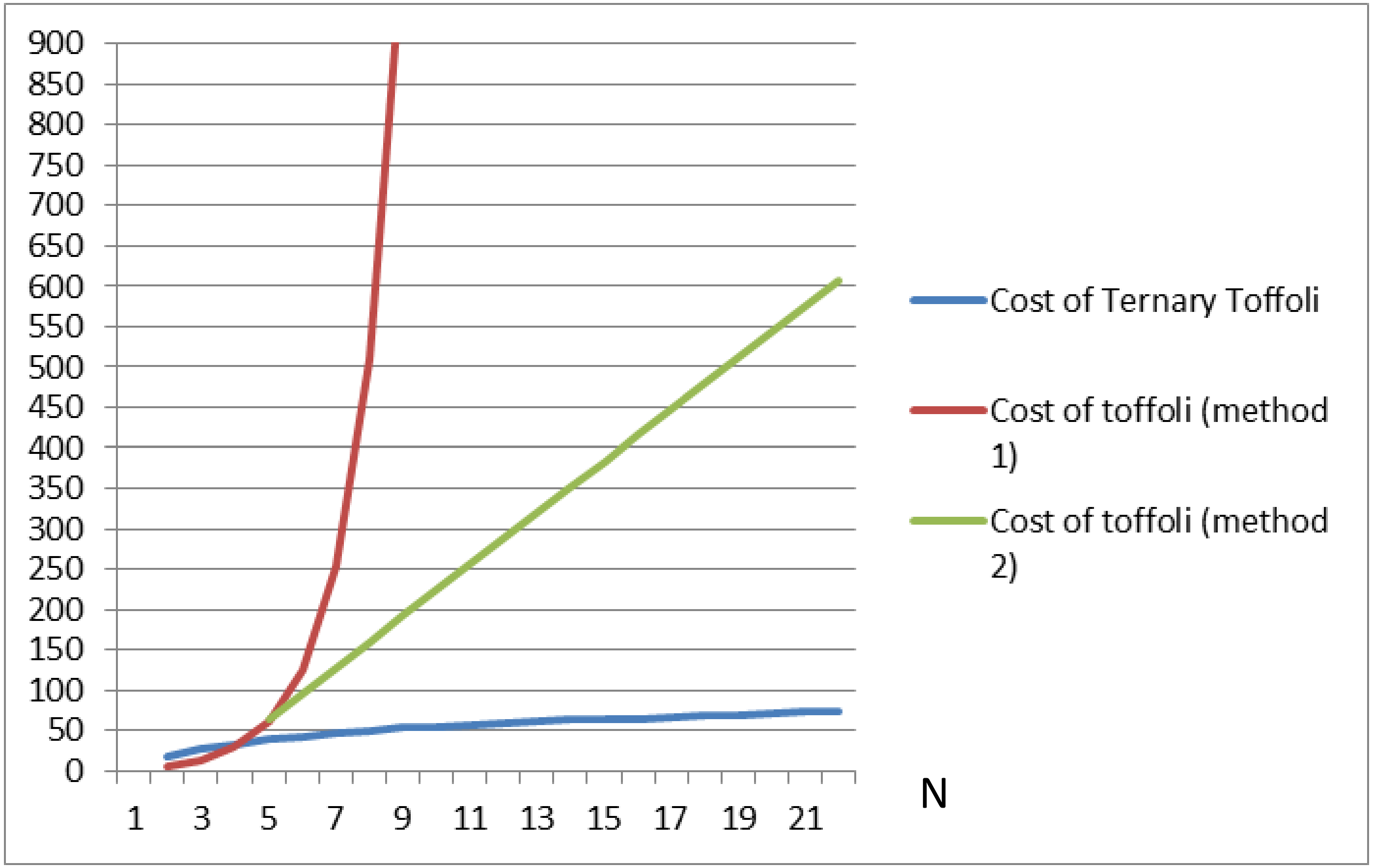


Figure 6 – Graphic Description of the cost of Toffoli gates

ISMVL 2011, 23-25 May 2011, Tuusula, Finland

Final cost comparison of binary and ternary comparators

- Using the formulae for the cost of Toffoli in terms of two-qubit gates, the cost jumps significantly to $4(\log_2 N) + 32(\log_2 N) – 96$ gates, assuming $N \geq 5$.

- For $N < 5$, the cost is $4(\log_2 N) + 2^{(\log 2N)} – 3$ gates.

- Now factoring the cost of Toffoli gates into the equation, the cost becomes much more favorable towards ternary logic, as shown in next Figure.

The final total costs: binary vs ternary

- Using the diagrams of the mirrored comparators, we can see that the cost of the ternary comparator is

➢ 34($\log_3 N$) two-qutrit gates,

➢ 4($\log_3 N$) single-qutrit gates,

➢ and one Toffoli gate with $\log_3 N$ inputs.

- The binary comparator costs 4($\log_2 N$) two-qubit gates plus one $\log_2 N$-input Toffoli gate.

Costs of comparators. With Toffoli

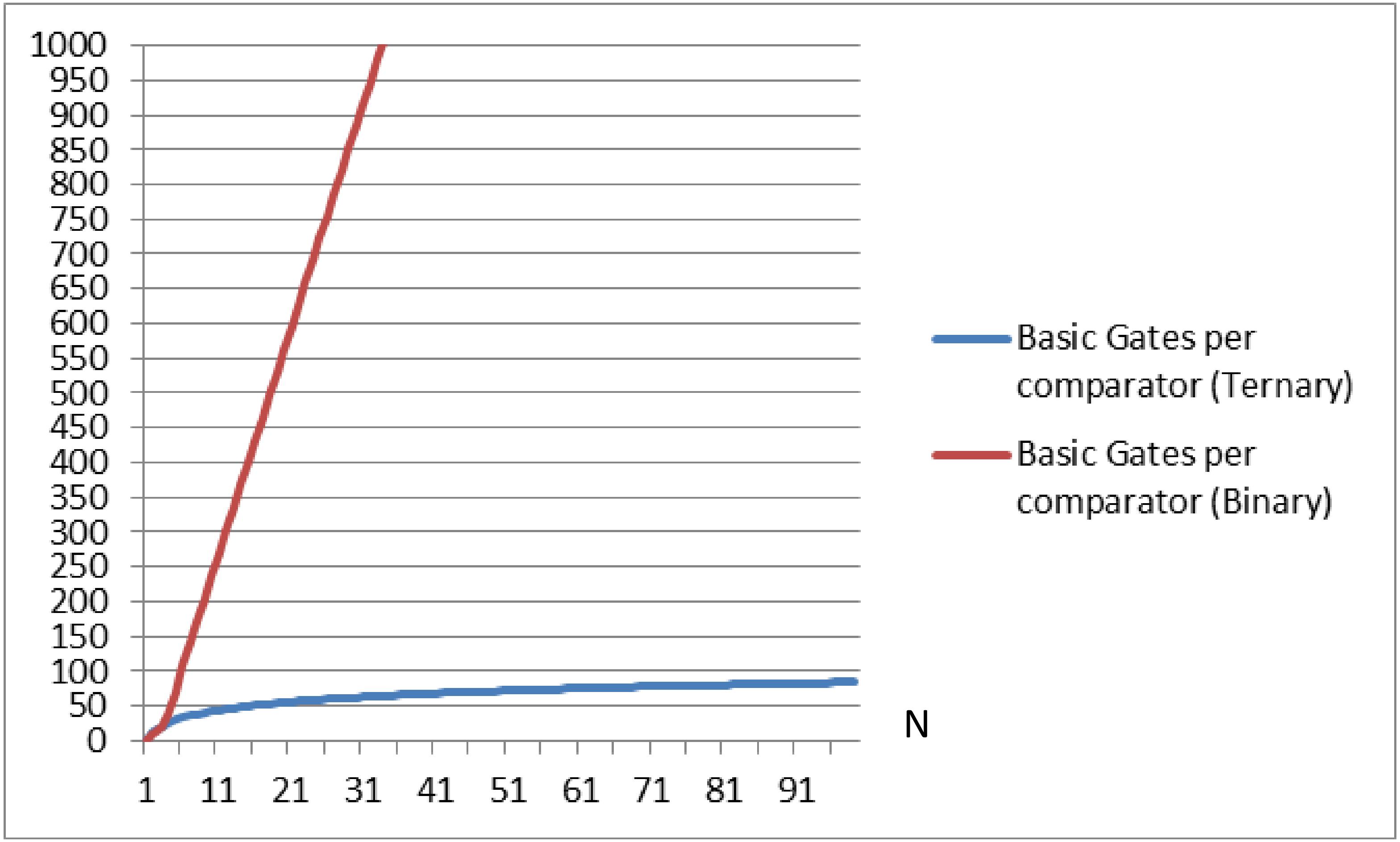
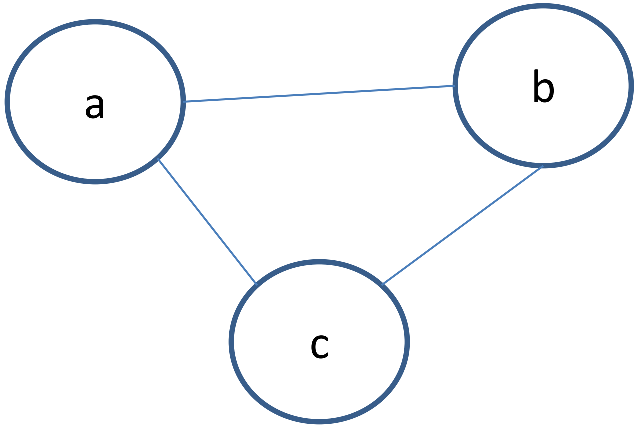


Figure 8 – Graphic Description of the cost of Comparators, including Toffoli.

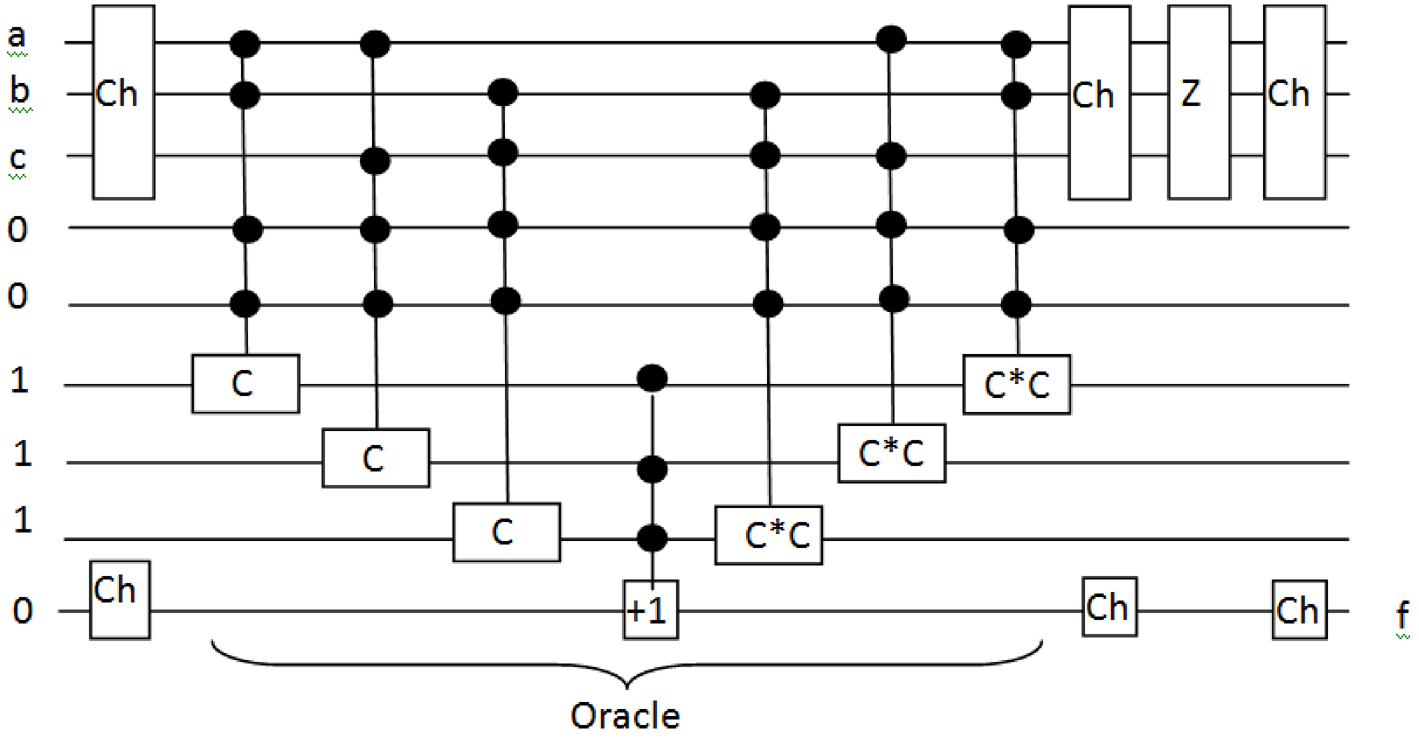
Simulation

- Simulations have been carried out in MATLAB
- Since MATLAB isn't particularly good with large matrices, simulations have been limited to small cases.



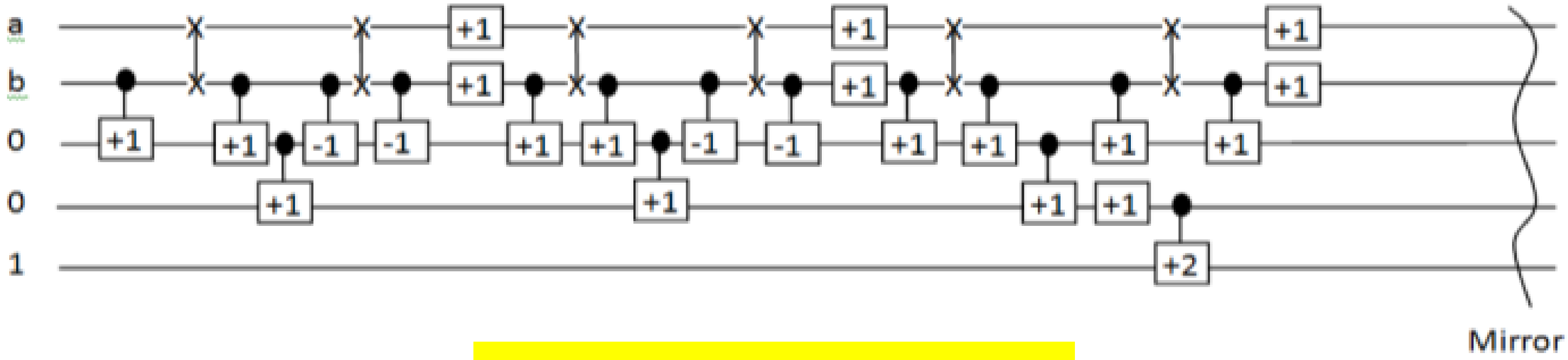
Simulation (cont'd)

- Overall layout of the simulated circuit



Simulation (cont'd)

- MATLAB requires no wire crossovers, so the simulation used a slightly modified version of the ternary comparator circuit



Linear nearest neighbor model

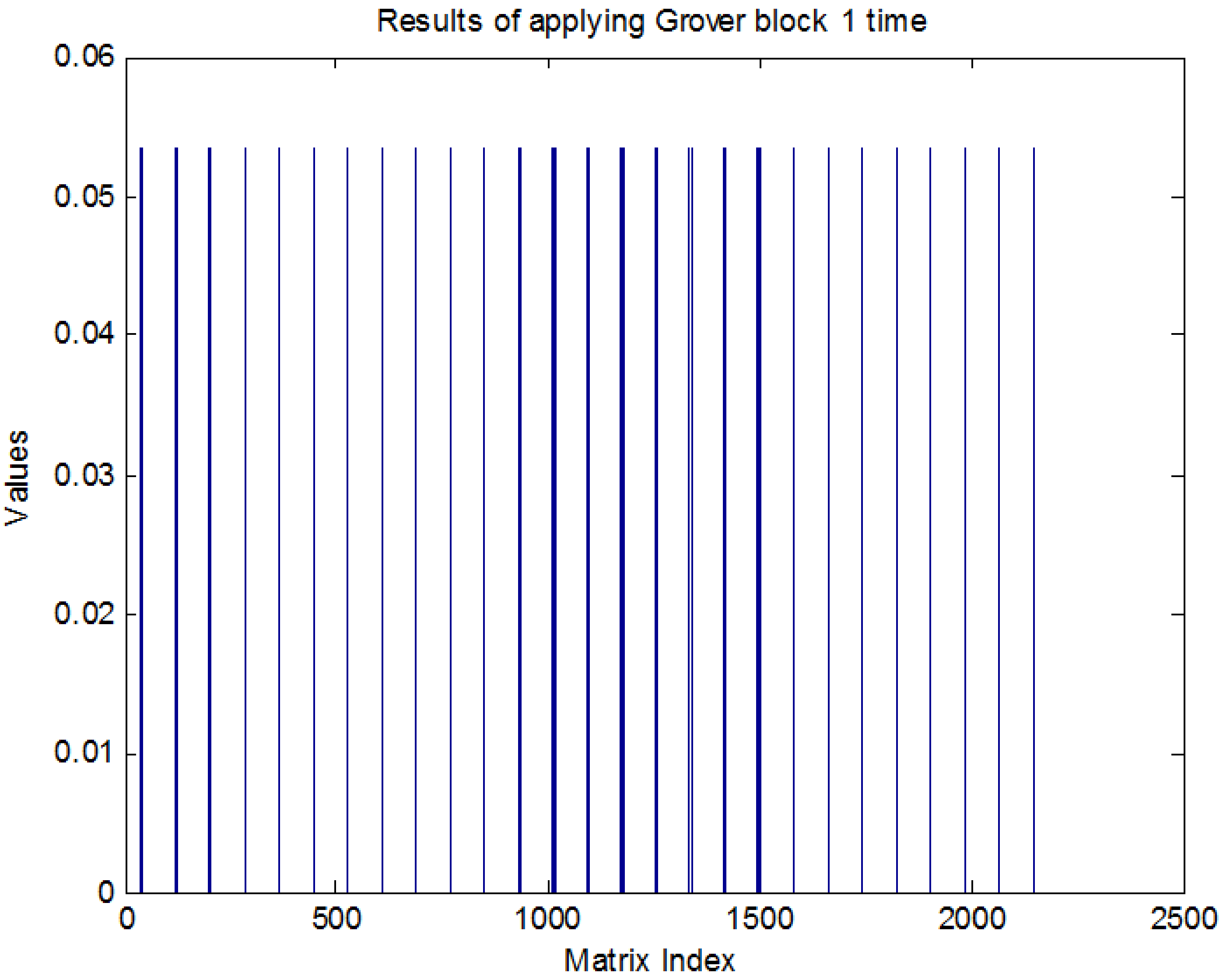
Grover Iterations

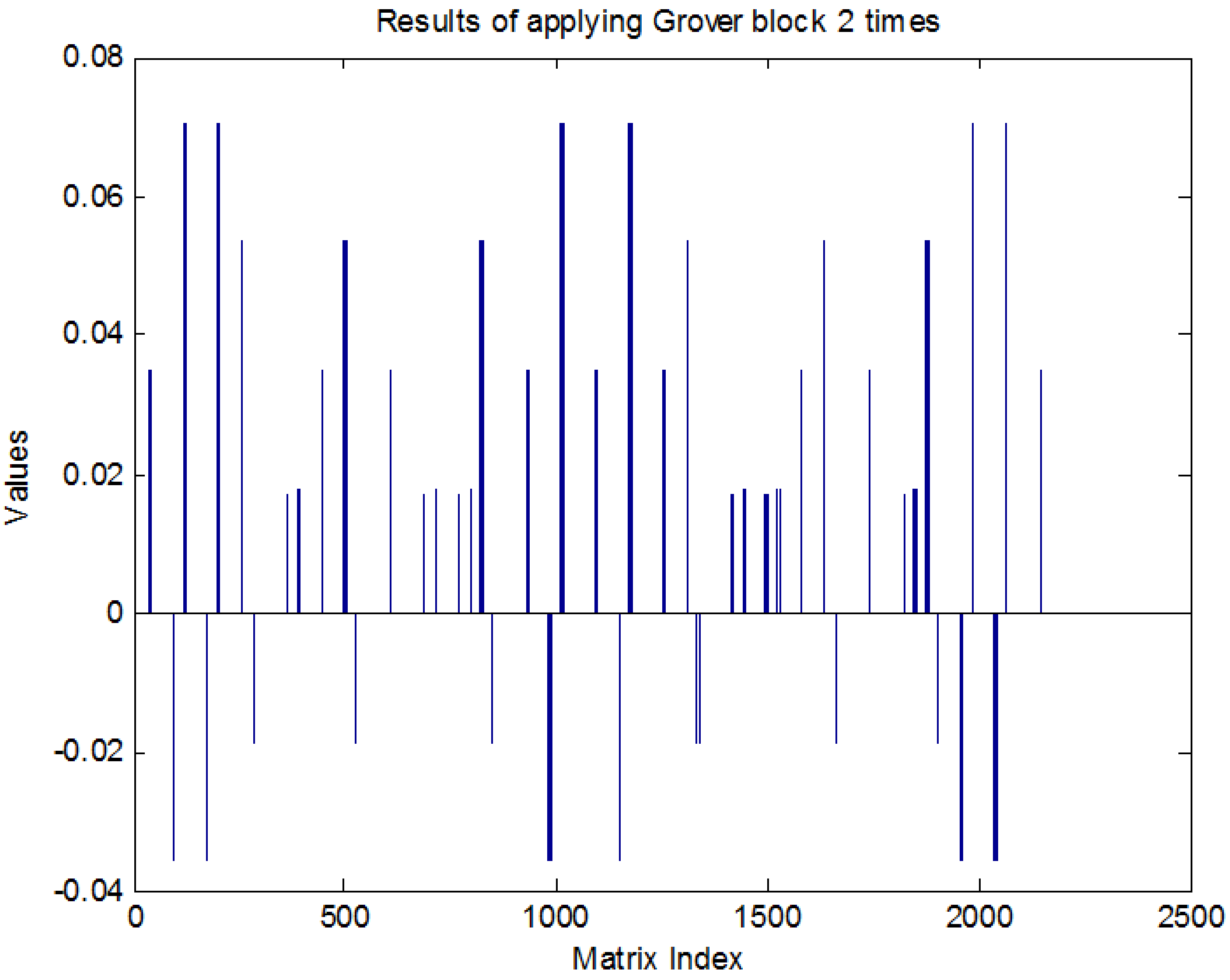
- Grover Algorithm has been shown to require approximately √ *M.* iterations rounded up, where M is the number of “items in database”.

- Since we have 3 nodes, each with 3 states, the total number of items is 3^3, or 27.

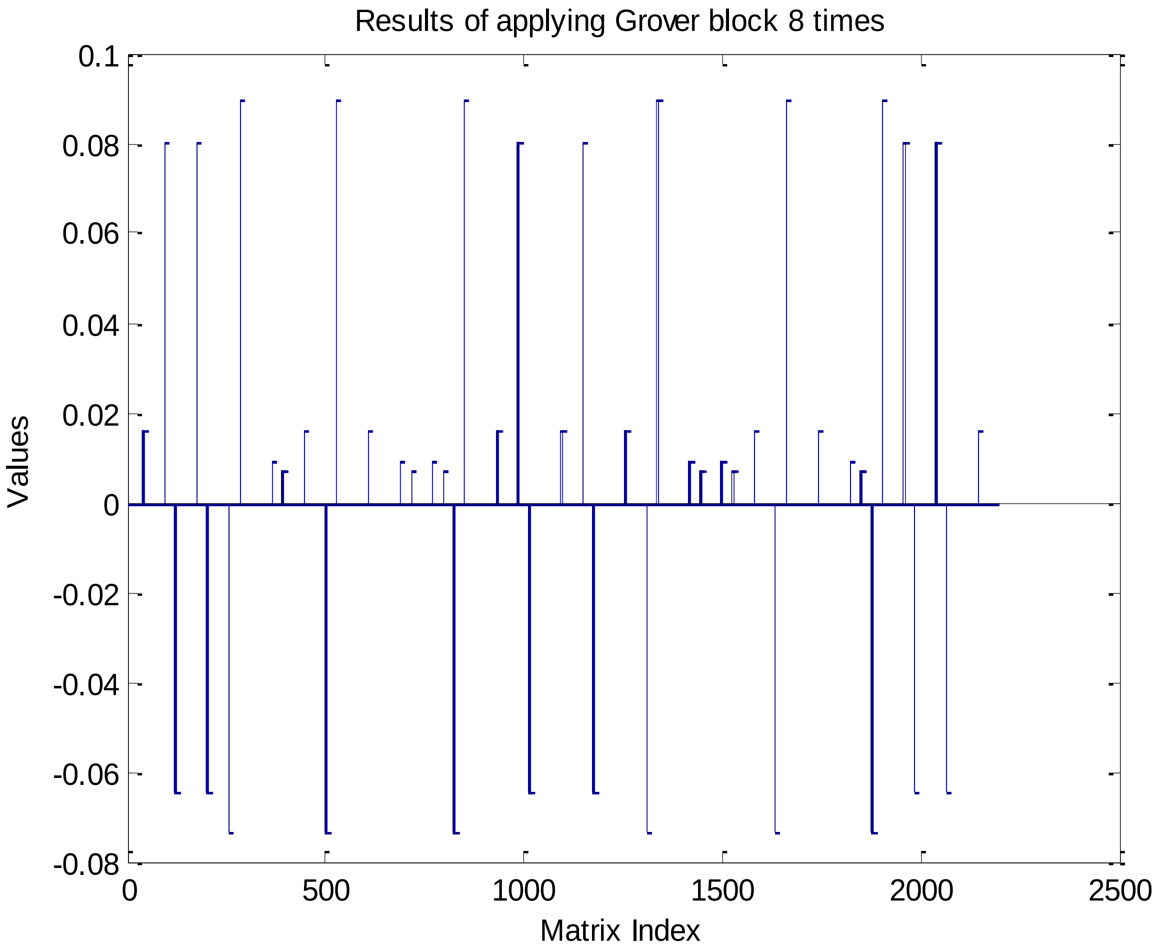
- Thus, our solution algorithm should have √ *27 = 6* iterations of the Grover operator.

- With each successive application of the operator, the probabilities of the “good” states should increase in amplitude, whereas the “bad” states should decrease in amplitude until they reach ~0.


Results of applying Grover block 1 time
Values
Matrix Index
0.06
0.05
0.04
0.03
0.02
0.01
0
0
500
1000
1500
2000
2500


Results of applying Grover block 2 times
Values
0.08
0.06
0.04
0.02
0
-0.02
-0.04
0
500
1000
1500
2000
2500
Matrix Index

After 6 iterations



How many iterations?

- After 2 Grover loops, the highest amplitude of the output was .655,
- whereas after 6 Grover loops, the highest amplitude had increased to .87.

Conclusions 1

- We synthesized a binary quantum algorithm for a graph coloring problem from binary gates

- We synthesized a ternary quantum algorithm to a graph coloring problem from ternary gates

- The ternary oracle can solve NP-complete problems, the same as binary

- We compared the complexities of binary and ternary oracles for solving arbitrary size graph-coloring problems.

- The ternary version of the oracle was shown to be able to perform the same functions as the binary functions equally well, giving the same outputs. (simulation)

- The ternary oracle has a lower cost in wires and garbage bits because of its ability to store more information per qudit

- The ternary oracle has smaller quantum cost.

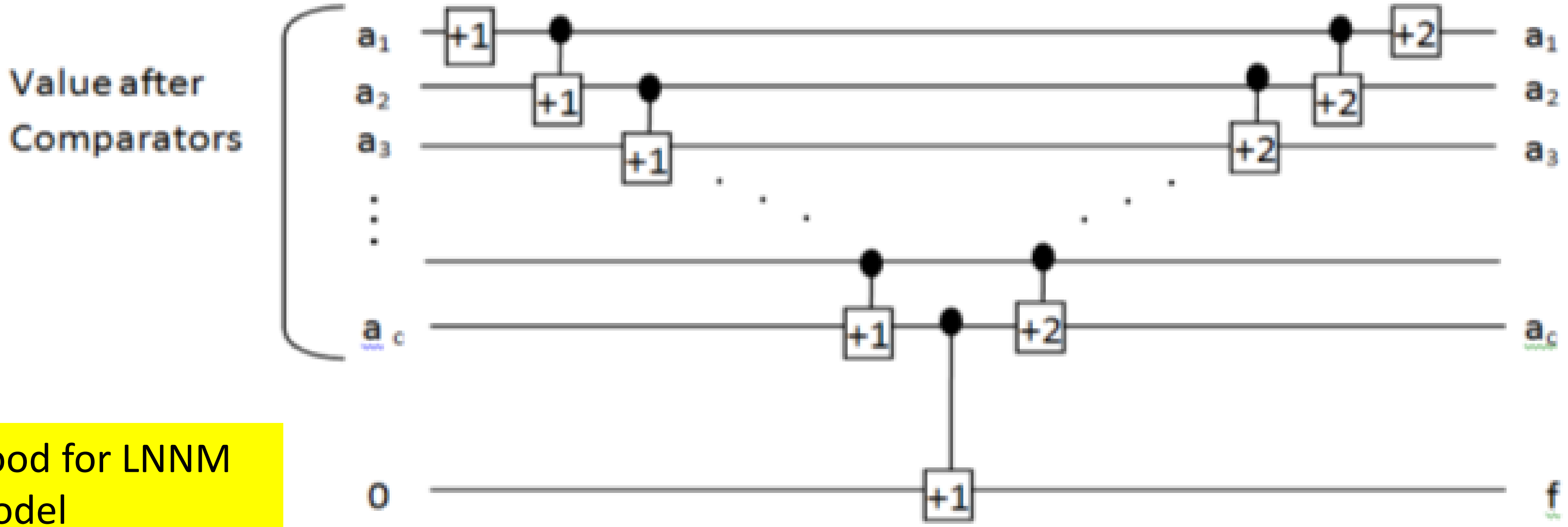
Conclusions 2

- The number of gates required to build the ternary oracle is initially higher than that of the binary oracle for small graphs.
- However, as the size and complexity of the graph grows, oracles in ternary-based logic have greater efficiency relative to binary oracles.
- The multi-valued extensions of binary oracles can be created for comparators, arithmetic circuits and Toffoli gates.
- Thus, any problem, even those requiring strictly binary outputs, can be solved with MV circuits.
- The circuits that are binary input / binary output can be built more efficiently with ternary logic inside.

Invention of a new binary Toffoli gate, ternary inside

- The ternary Toffoli gate, invented in this paper, performs a similar function and will be used as a building block for ternary logic synthesis in the future.

- Our new ternary Toffoli gate has 1/8 the cost of the binary Toffoli gate, as it scaled linearly with the number of inputs with slope two, whereas the binary gate required 16 more 2-qubit gates for each additional input.



Good for LNNM model

Conclusions 4

- Ternary gates cause different rotations than binary gates on particles.

- Therefore, the cost in NMR (Nuclear Magnetic Resonance) pulses of 2-qudit gates in binary and ternary are not the same.

- Even 2-qubit binary gates require different numbers of pulses from each other, such as the Feynman gate, which requires 5 pulses, compared to the V and V^+ gates, which require 3 pulses each.

- Ternary gates, since they operate with 120° rotations, would have still different numbers of pulses, which may affect their efficiency relative to binary gates.

Conclusions 5

- For some future Constrained Satisfaction Problems for Grover, the oracle should have mixed (hybrid) qudits for reduction of quantum cost and processing speed.

- Synthesis methods for hybrid quantum circuits and binary external/multiple-valued internal quantum circuits should be created

- Ternary results will be even better for the Nearest Linear Nearest Neighbor Model, as the ternary comparators are more compact and ternary solution requires less SWAP gates to be added.

  On the other hand, the ternary SWAP gates are more expensive in terms of pulses.