Cost Metrics for Reversible and Quantum Logic Synthesis

Dmitri Maslov[1]

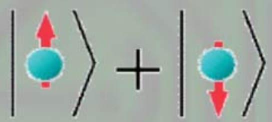
D. Michael Miller[2]

[1]*Dept. of ECE, McGill University*

[2]*Dept. of CS, University of Victoria*

Outline

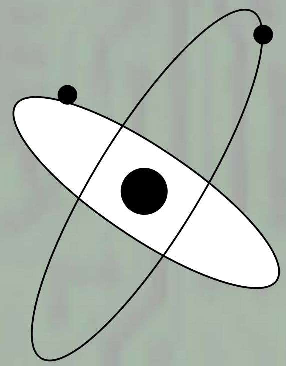
- Introduction (background, motivation)
- Motivation for our research
- Definitions and Problem Statement
- Our solution: Pruned Prioritized Breadth-first Search
- Results and Conclusions

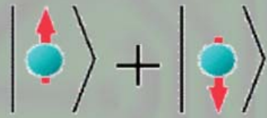


Introduction

Quantum bit could be a state of a single proton in a static magnetic field (magnetic spin). For a fixed proton state of a magnetic spin is known to be probabilistic, in other words, only the measurement tells what was the state.

α|0>+β|1> - state of a proton.
|α| - probability of finding the proton in a lower energy state
|β| - probability of finding the proton in a higher energy state





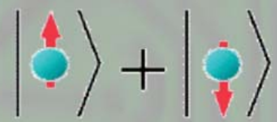
Introduction

Quantum n-bit system is described by a vector of length 2^n with complex coefficients, called amplitudes.

$$\alpha_{00\ldots0}|0,0,\ldots,0\rangle + \alpha_{00\ldots01}|0,0,\ldots,0,1\rangle + \ldots + \alpha_{11\ldots1}|1,1,\ldots,1\rangle$$

Quantum computation is done through multiplication of the state vector by 2^nx2^n unitary matrices.

Rather than working with huge matrices, we consider a **circuit computation** model. This saves space and illustrates what happens better.



Introduction

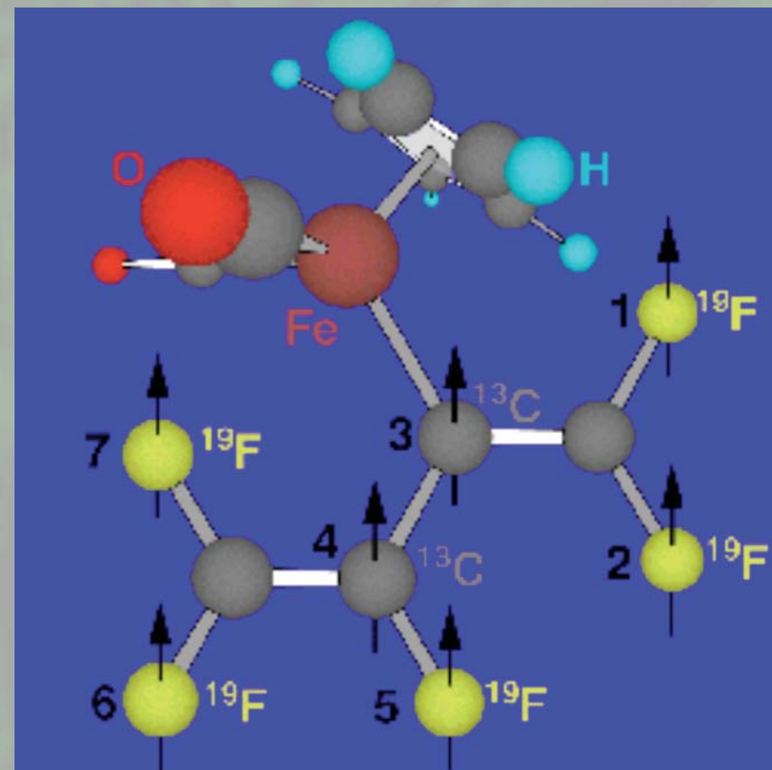
Quantum computation features:

1. **Quantum errors.** At any time state |0> can spontaneously change to the state |1> and vice versa.
2. **Measurement kills the system.**
3. **Copying is impossible.** No fan outs.
4. **Computation lifetime is limited** by approximately 2 sec.
5. **Limited number of basic (elementary) gates.**

5.5. All the computations are **reversible**.

6. **Scaling is difficult.**
7. **Quantum superposition.** Quantum system with n qubits is associated with presence of 2^n complex numbers.
8. **Quantum parallelism.** It is possible to compute a Boolean function on all the possible inputs simultaneously.

Introduction



IBM research group, 2001: 7-qubit quantum processor.

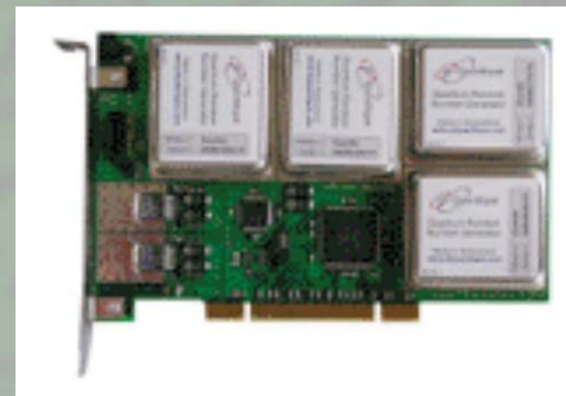
Introduction

Quantum key distribution.

Main features

•First commercial quantum key distribution system

•Key distribution distance: up to 100 km



Quantum random number generator.

Main features

• PCI card

• random bit rate of up to 16Mbps

www.idquantique.com

Problem Statement

- Find optimal NCV circuits for the 8! 3-variable quantum Boolean (reversible) functions.
- Optimal can be based on gate count or on total gate cost for some costing model.
- Gate count is just a cost model where all gates have cost 1.

Motivation

- NOT, CNOT, controlled-V and controlled-V+ (NCV) gates are elementary and well studied blocks.
- We are interested in the ***direct*** synthesis of ***small*** circuits composed of NCV gates (rather than of libraries with macros).
- Observing optimal circuits for small cases often will shed light on good (if not optimal) synthesis approaches.
- Since we know the optimal results for 3-line Toffoli circuits, it is of interest to know what the optimal NCV circuits might look like.
- It is in its own right a challenging problem (21^{16}=1,430,568,690,241,985,328,321 ~ 10^{21}).

Definitions

- A Boolean function $f:\{0,1\}^n \rightarrow \{0,1\}^n$ is reversible if it maps each input pattern to a unique output pattern (it is a bijection).
- There are $2^n!$ n-variable reversible functions.
- For n=3, this yields 8! = 40,320 functions.

Definitions

- A quantum circuit is a sequence of quantum gates (cascade), linked by "wires"
- The circuit has fixed "width" corresponding to the number of qubits being processed
- Logic design (classical and quantum) attempts to find circuit structures for needed operations that are
  - Functionally correct
  - Independent of physical technology
  - Low-cost, e.g., use the minimum number of qubits or gates

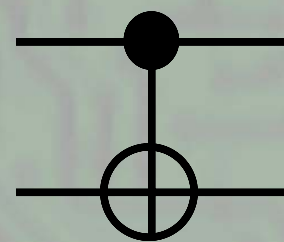
- Quantum logic design is not at all well developed.

Definitions

NOT

In	Out
0	1
1	0

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

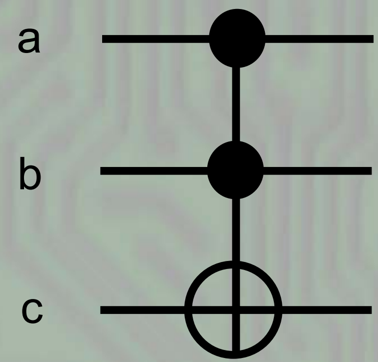


CNOT

In	Out
0 0	0 0
0 1	0 1
1 0	1 1
1 1	1 0

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Definitions

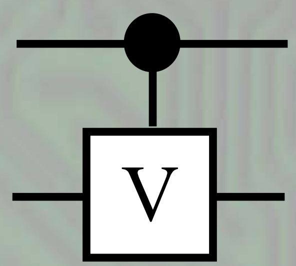


Toffoli

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

In	Out
a b c	a b c
0 0 0	0 0 0
0 0 1	0 0 1
0 1 0	0 1 1
0 1 1	0 1 0
1 0 0	1 0 0
1 0 1	1 0 1
1 1 0	1 1 1
1 1 1	1 1 0

Definitions

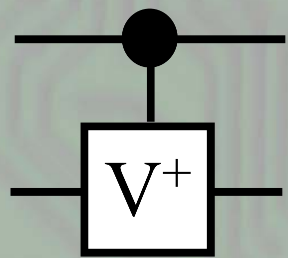


Controlled-V

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2}+\frac{1}{2}i & \frac{1}{2}-\frac{1}{2}i \\ 0 & 0 & \frac{1}{2}-\frac{1}{2}i & \frac{1}{2}+\frac{1}{2}i \end{bmatrix}$$

In	Out
0 0	0 0
0 1	0 1
1 0	1 v0
1 1	1 v1
0 v0	0 v0
0 v1	0 v1
1 v0	1 1
1 v1	1 0

Definitions

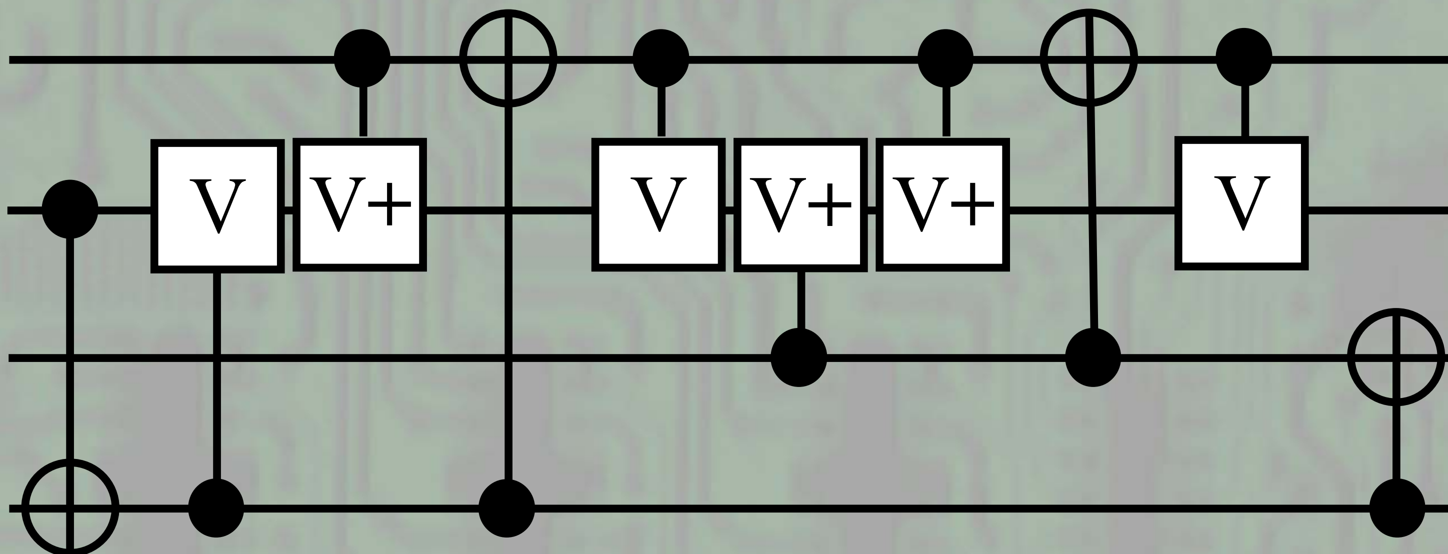


Controlled-V^{+}

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2}-\frac{1}{2}i & \frac{1}{2}+\frac{1}{2}i \\ 0 & 0 & \frac{1}{2}+\frac{1}{2}i & \frac{1}{2}-\frac{1}{2}i \end{bmatrix}$$

In	Out
0 0	0 0
0 1	0 1
1 0	1 v1
1 1	1 v0
0 v0	0 v0
0 v1	0 v1
1 v0	1 0
1 v1	1 1

Definitions



Example of a quantum circuit (3-bit full adder)

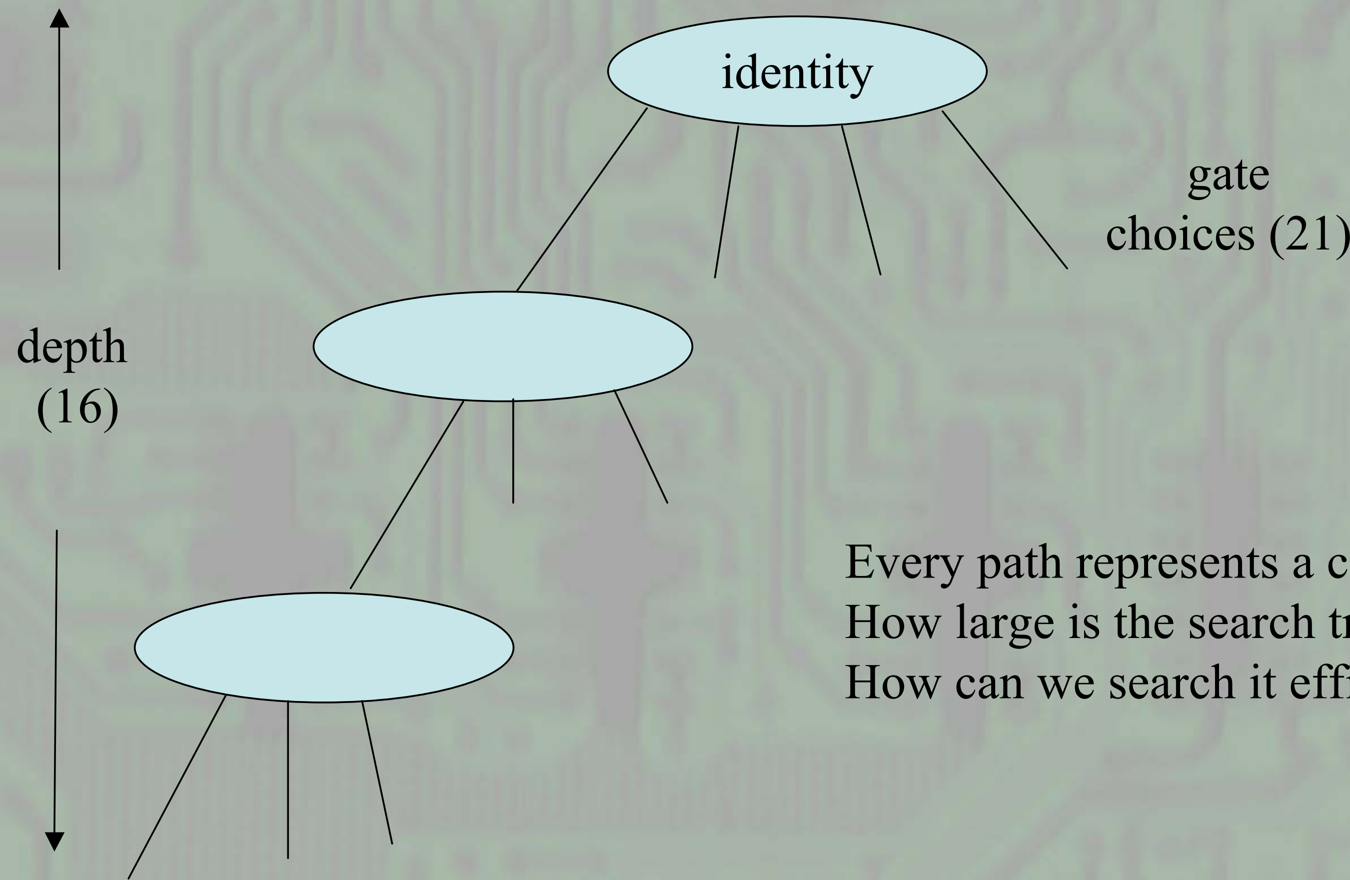
Problem Statement

- Find optimal NCV circuits for the 8! 3-variable quantum Boolean functions.

- Optimal can be based on gate count or on total gate cost for some costing model.

- Gate count is just a cost model where all gates have cost 1.

Our solution



Every path represents a circuit.
How large is the search tree? 10^{21}
How can we search it efficiently?

- works well for ‘small’ trees but pruning is often required for large problems
- should work for gate count cost but what about other cost models?

Our solution

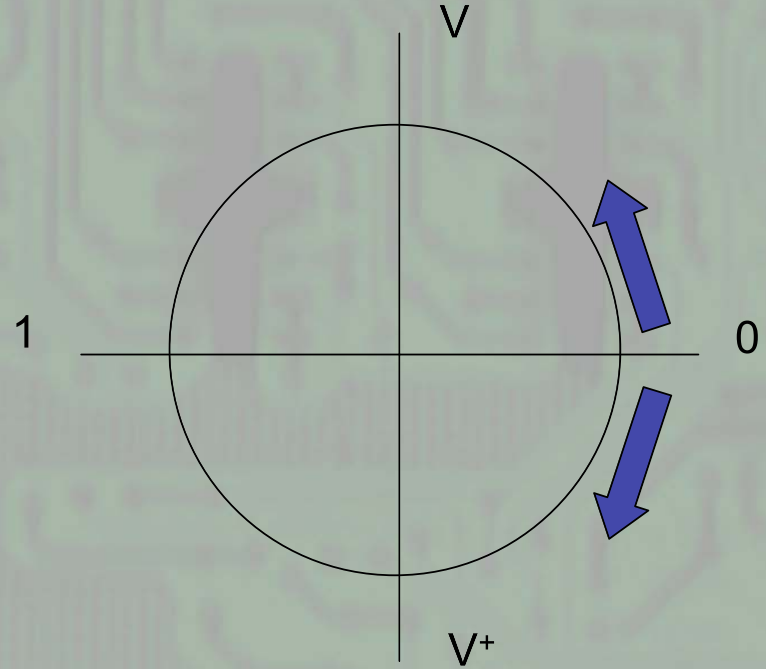
- Issues:
  - How to code the functions accounting for Boolean and quantum values?
  - How to limit the search space?
  - How to search the tree efficiently?
  - How to account for different gate costs?

- Assumption: never use a ‘quantum’ line as a control for a V or V^+ gate.

Our solution

How to code functions?

- The Boolean and quantum values can be treated as follows:



A V gate is a quarter turn counter-clockwise

A V+ gate is a quarter turn clockwise

Our solution

A simple coding is sufficient

0	0 0
V	0 1
1	1 0
V+	1 1

We can think of a quantum function as having a base Boolean function (reversible parent) with a quantum signature added.

a	b	c	A	B	C
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	V
0	1	1	0	1	V+
1	0	0	1	0	V
1	0	1	1	0	V+
1	1	0	1	1	1
1	1	1	1	1	0

a	b	c	A	B	C
0	0	0	00	00	00
0	0	1	00	00	10
0	1	0	00	10	01
0	1	1	00	10	11
1	0	0	10	00	11
1	0	1	10	00	01
1	1	0	10	10	10
1	1	1	10	10	00

Our solution

How to limit the search space?

Theorem. A circuit realizing a Boolean reversible function realizes the same function if controlled-V gates are replaced by controlled-V+ gates and controlled-V+ gates are replaced by controlled-V gates.

Proof: Obvious from circle of values.

Hint 1: during the search it is always enough to use gate controlled-V as the first quantum gate.

Our solution

The number of gate choices is 21:

- 3 NOT
- 6 CNOT
- 6 controlled-V
- 6 controlled-V^{+}

But not all gate choices are applicable in all situations.

Hint 2: Don't follow a gate with another gate with the same control and target – such a pair can always be reduced to one gate regardless of the gate types. <u>*Assumes no gate type is realizable by a lower cost composition of other gates types.*</u>

Our solution

Hint 3: Once an optimal implementation of a function is found, we have also found an optimal implementation for all functions that differ from this one only by their input-output labeling.

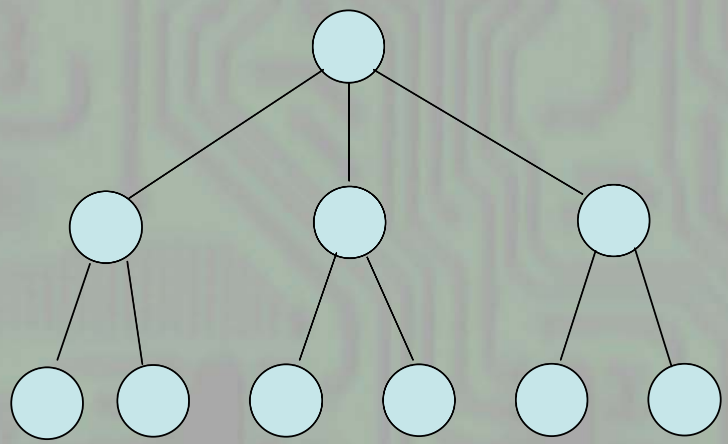
Hint 4: don't consider a circuit (tree node) if we have already found a cheaper realization for that function.

Hint 5 (not used): once $G_1 G_2 \ldots G_k$ is an optimal circuit for a reversible function f, $G_k^{-1} G_{k-1}^{-1} \ldots G_1^{-1}$ is an optimal circuit for f^{-1}.

Our solution

- There are 40,320 3-line Boolean reversible functions.
- We don’t know how many quantum function will have to be considered.
- In the breadth-first search we want to visit the cheaper circuits first. For gate count cost, this is easy and can be done with one queue.
- But for a cost model with different costs for different gate types, multiple queues are required.

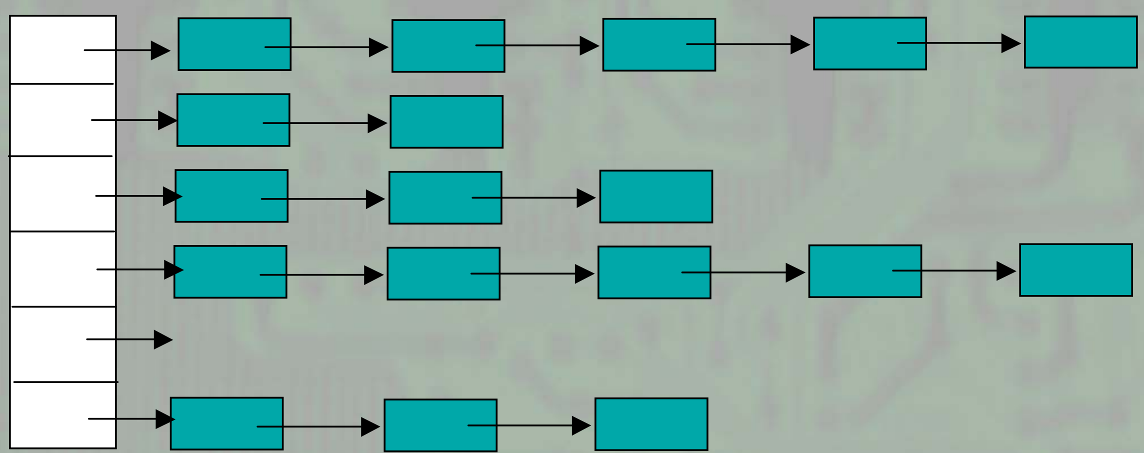
Our solution



NQ = max gate cost +1

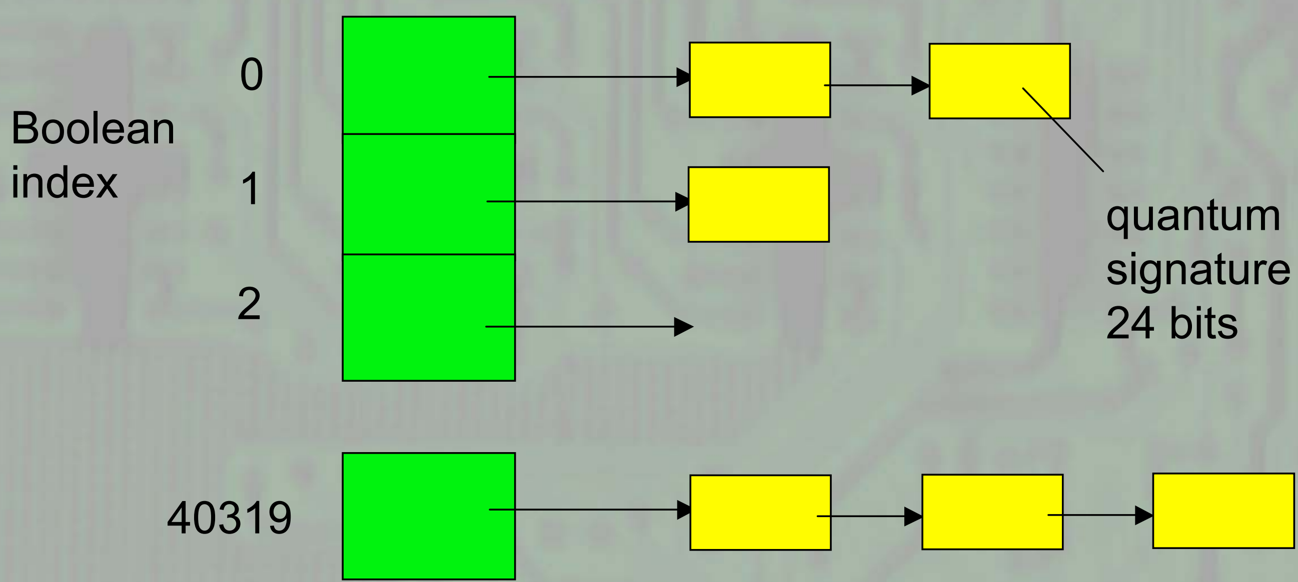
a circuit of cost C is queued in queue C mod NQ

This is a prioritized breadth-first search.



Our solution

- A ***reversible parent*** is readily mapped to an index (integer) and vice versa (see p. 161 in Combinatorial Algorithms, by Reingold, Nievergelt and Deo).



Results

NCV-111 cost model

- average gate count: 10.03
- average cost: 10.03
- Boolean functions queued: 6,828
- Boolean function cost reductions: 0
- Quantum functions queued: 206,410
- Quantum function cost reductions: 0
- user time: 61 seconds on a fairly fast UNIX box

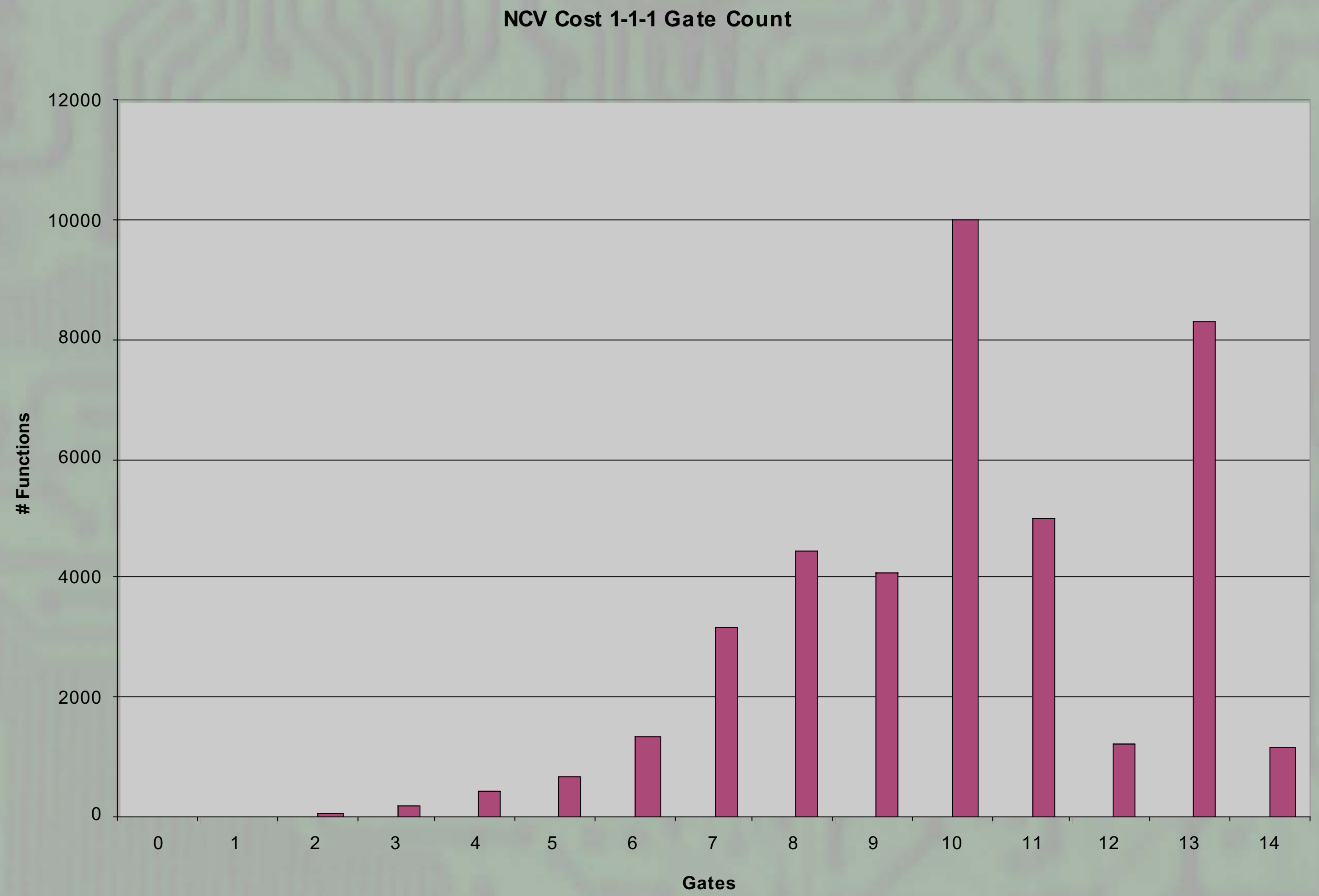
Results

```
    0 :  0 1 2 3 4 5 6 7 : 0 : ;

 5167 :  1 0 3 2 5 4 7 6 : 0 : N(1,0);
11536 :  2 3 0 1 6 7 4 5 : 1 : N(2,0);
23616 :  4 5 6 7 0 1 2 3 : 2 : N(3,0);

  121 :  0 1 3 2 4 5 7 6 : 0 : N(1,2);
 1565 :  0 3 2 1 4 7 6 5 : 1 : N(2,1);
 3109 :  0 5 2 7 4 1 6 3 : 2 : N(3,1);
    7 :  0 1 2 3 5 4 7 6 : 3 : N(1,3);
   16 :  0 1 2 3 6 7 4 5 : 4 : N(2,3);
  592 :  0 1 6 7 4 5 2 3 : 5 : N(3,2);

 5046 :  1 0 2 3 5 4 6 7 : 0 : N(1,0).N(1,2);
10814 :  2 1 0 3 6 5 4 7 : 1 : N(2,0).N(2,1);
21410 :  4 1 6 3 0 5 2 7 : 2 : N(3,0).N(3,1);
 5160 :  1 0 3 2 4 5 6 7 : 3 : N(1,0).N(1,3);
```

Results

```
28024 :  5 3 7 2 4 6 0 1 : 2 :
   N(3,1).V(1,3).N(1,0).V(1,2).N(2,3).VP(1,2).V(3,1)
.VP(3,2).N(2,1).V(3,2).V(2,3).VP(2,1).N(1,3).VP(2,1);

37137 :  7 2 4 3 1 5 6 0 : 0 :
   N(1,2).V(3,1).V(3,2).N(2,1).V(3,2).V(1,2).N(2,3).
VP(1,2).VP(1,3).V(3,1).N(2,0).N(1,2).V(3,1).V(3,2);

38337 :  7 4 1 3 2 5 6 0 : 0 :
   N(1,2).V(3,1).V(3,2).N(2,1).V(3,2).V(1,2).N(2,3).
VP(1,2).VP(1,3).V(3,1).N(2,0).N(1,2).V(3,1).V(3,2);

36209 :  7 1 2 5 4 6 3 0 : 0 :
   V(1,2).N(2,3).V(1,3).VP(1,2).V(2,1).V(2,3).N(3,1)
.VP(2,3).V(1,2).N(3,0).N(3,2).N(2,3).VP(1,2).VP(1,3);

36231 :  7 1 2 6 4 3 5 0 : 0 :
   V(1,2).N(2,3).V(1,3).VP(1,2).V(2,1).V(2,3).N(3,1)
.VP(2,3).V(1,2).N(3,0).N(3,2).N(2,3).VP(1,2).VP(1,3);
```

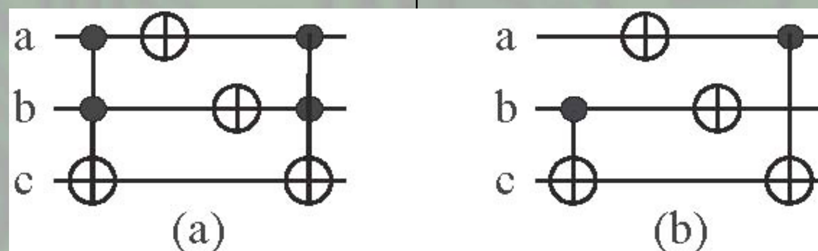
Results



NCV Cost 1-1-1 Gate Count
12000
10000
8000
6000
4000
2000
0
Functions
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Gates

Results

Opt. NCT			Opt. NCV
Cost	GC	NCV-111	NCV-111
0	1	1	1
1	12	9	9
2	102	[illegible]	[illegible]
3	625	[illegible]	[illegible]
4	2780	[illegible]	[illegible]
5	8921	[illegible]	[illegible]
6	17049	[illegible]	[illegible]
7	10253	335	3176
8	577	1300	4470
9	0	3037	4122
10	0	3394	10008
11	0	793	5036
12	0	929	1236
13	0	4009	8340
14	0	8318	1180
15	0	4385	0
16	0	255	0
[illegible]	[illegible]	1297	0
[illegible]	[illegible]	4626	0
[illegible]	[illegible]	4804	0
[illegible]	[illegible]	475	0
21	0	106	0
22	0	503	0
23	0	357	0
24	0	4	0
27	0	17	0
28	0	2	0
WA	5.8655	14.0548	10.0319



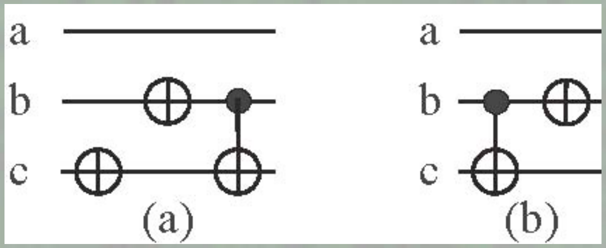
(a) (b)

Conclusion 1: small Toffoli gate count is not an effective illustration of the implementation cost.

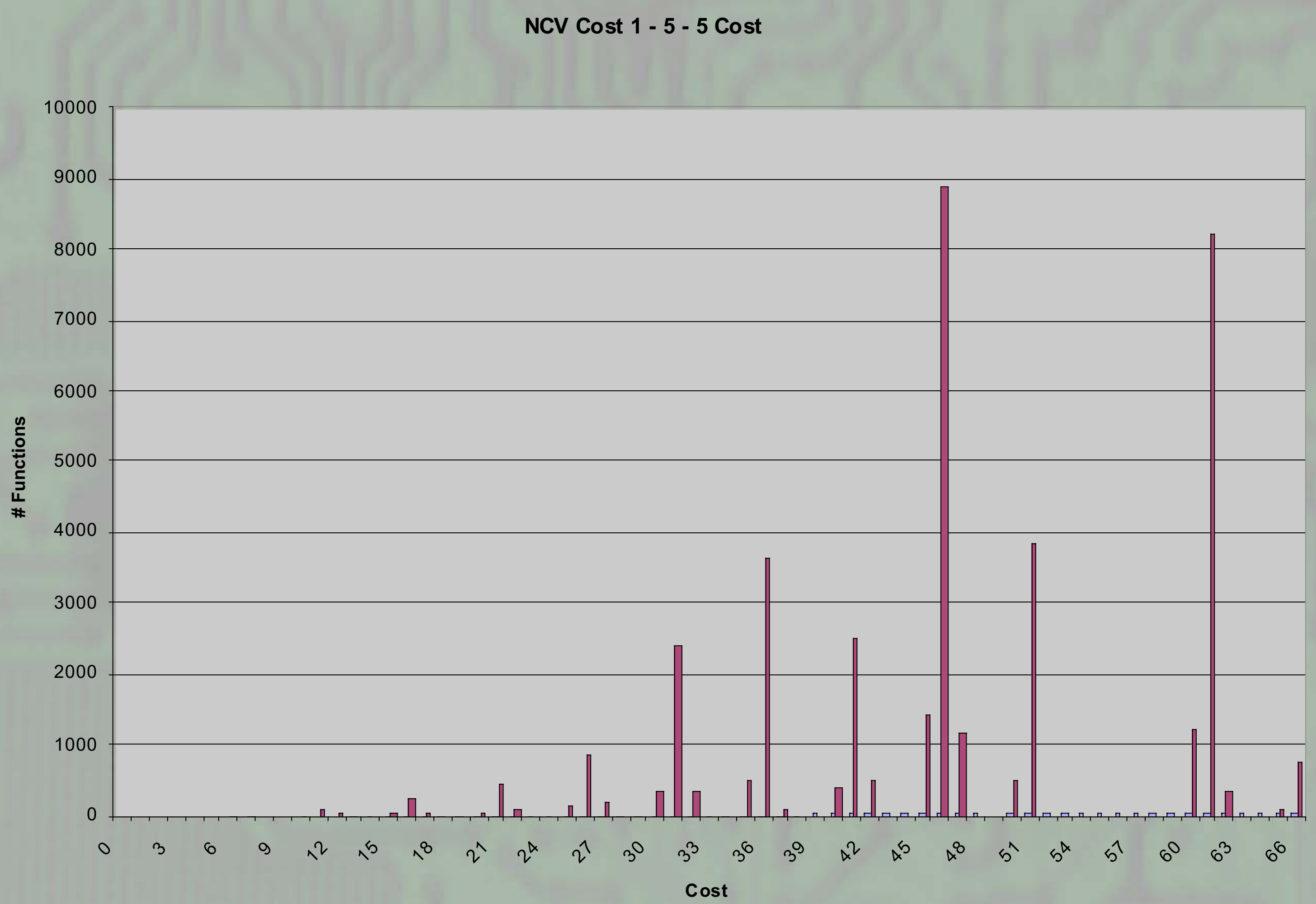
Results

NCV-155 cost model

- average gate count: 10.03
- average cost: 46.35
- Boolean functions queued: 6,878
- Boolean function cost reductions: 50
- quantum functions queued: 232,406
- Quantum function cost reductions: 19,038
- user time: 68 seconds



Results



NCV Cost 1 - 5 - 5 Cost
Functions
Cost
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
3
6
9
12
15
18
21
24
27
30
33
36
39
42
45
48
51
54
57
60
63
66

Results

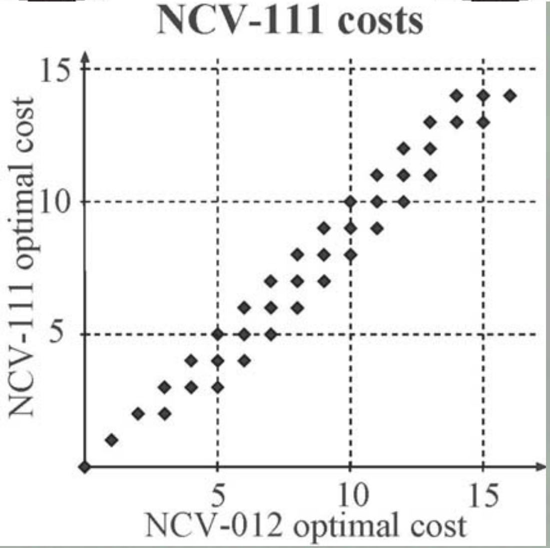
Distribution of controlled-V/controlled-V+ gates.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	2	0	0	0	0	0	0	0	0	0	0
1	9	0	0	0	0	0	0	0	0	0	0
2	51	0	0	0	0	0	0	0	0	0	0
3	187	0	0	0	0	0	0	0	0	0	0
4	393	0	0	24	0	0	0	0	0	0	0
5	474	0	0	240	0	0	0	0	0	0	0
6	215	0	0	1158	0	0	0	0	0	0	0
7	14	0	0	3162	0	0	0	0	0	0	0
8	0	0	0	4110	0	0	360	0	0	0	0
9	0	0	0	714	0	0	3408	0	0	0	0
10	0	0	0	0	0	0	10008	0	0	0	0
11	0	0	0	0	0	0	5036	0	0	0	0
12	0	0	0	0	0	0	4	0	0	1232	0
13	0	0	0	0	0	0	0	0	0	8340	0
14	0	0	0	0	0	0	0	0	0	1180	0
15	0	0	0	0	0	0	0	0	0	0	0

Conclusion 2: number of controlled-V/controlled-V+ gates in optimal implementations is divisible by 3.

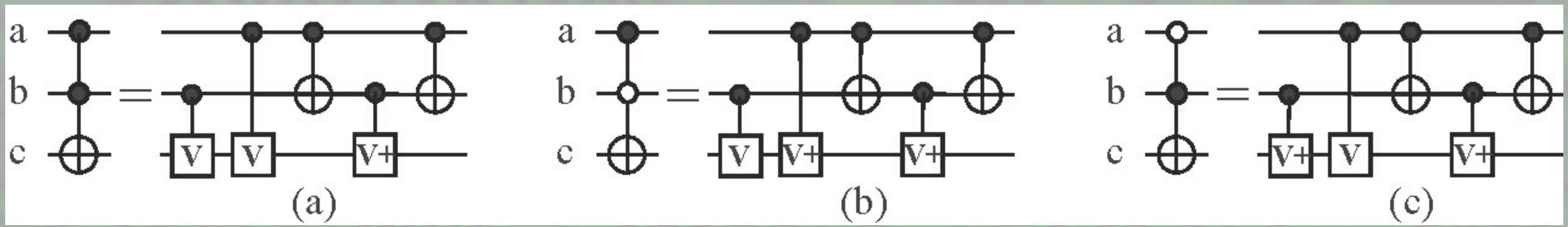
Results

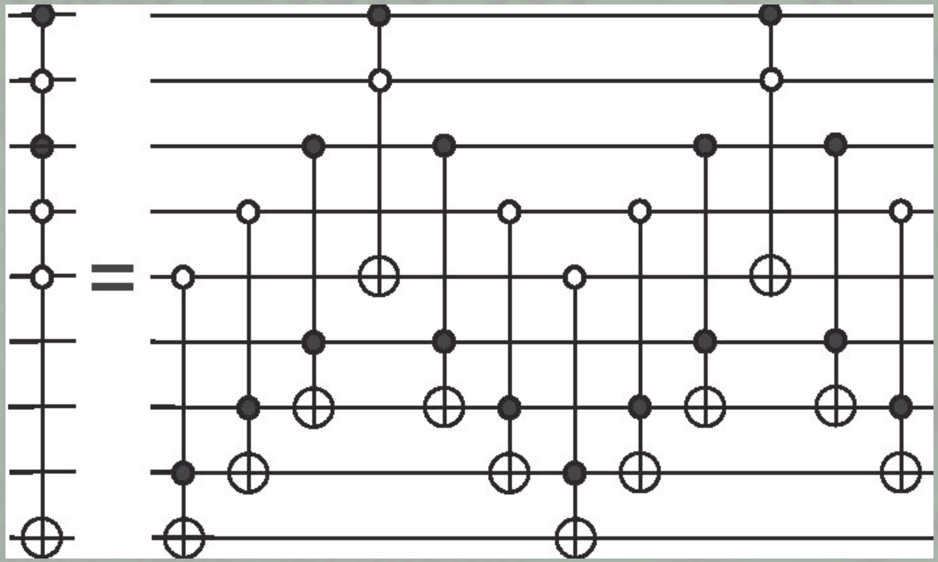
Interchangeability chart.



Conclusion 3: for small functions, it does not matter much in which metric to minimize a circuit. NCV-111 metric, however, seems to be more useful.

Results





Conclusion 4: multiple control Toffoli gates with some but not all negations are no more expensive than Toffoli gates with all positive controls.

Acknowledgements

- Gerhard Dueck, Faculty of Computer Science, University of New Brunswick
- Natural Sciences and Engineering Research Council of Canada
- NB IEEE

Thank you!

Comments, Questions, Critiques?