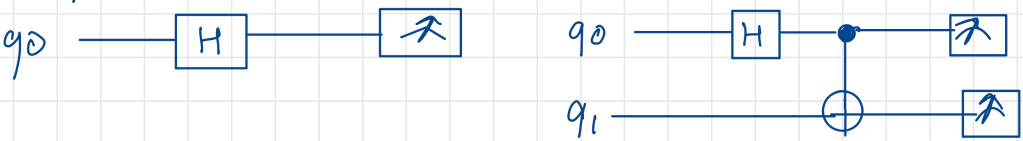

CUTQC

Cut QC: Using small Quantum Computers for large Quantum Circuit Evaluations.

Background

Quantum programs are expressed as circuits that consists of a sequence of single- and multi qubit gate operations.



One wildely used package is IBMs Qiskit, which allows simulations and cloud access to IBMs quantum hardware.

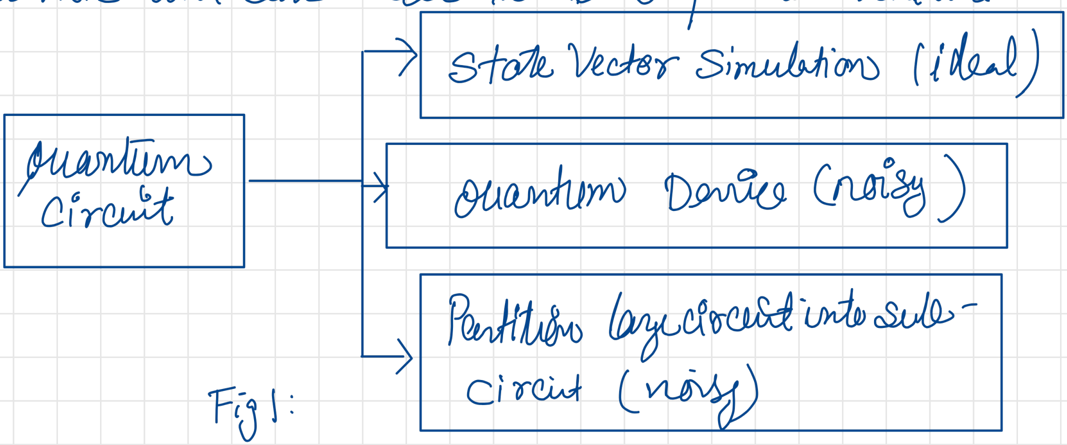


Fig 1:

State vector simulation is typically an idealized noiseless simulation of a quantum circuit. All quantum operations are represented as unitary matrices. $(2^n \times 2^n)$

This simulation sequentially multiply each gate's corresponding unitary matrix with the current state vector.

Circuit Cutting

Unitary Matrix of an arbitrary quantum operation in a quantum circuit can be decomposed into any set of orthonormal matrix basis.

$$A = \frac{Tr(AI)I + Tr(AX)X + Tr(AY)Y + Tr(AZ)Z}{2}$$

($\because$ this one requires access to complex amplitudes, which are not available on a quantum computer)

we further decompose the Pauli matrices into their eigenbasis and organize the terms.

$$A = \frac{A_1 + A_2 + A_3 + A_4}{2}$$

$$A_1 = [Tr(AI) + Tr(AZ)]\, |0\rangle\langle 0|$$

$$A_2 = [Tr(AI) - Tr(AZ)]\, |1\rangle\langle 1|$$

$$A_3 = Tr(AX)\left[2|+\rangle\langle +| - |0\rangle\langle 0| - |1\rangle\langle 1|\right]$$

$$A_4 = Tr(AY)\left[2|+i\rangle\langle +i| - |0\rangle\langle 0| - |1\rangle\langle 1|\right]$$

$Tr(AI)$ ~ Trace operator means measuring the qubit in that basis

$|+\rangle\langle +|$ ~ Density operator means to initialize the qubit in one of the eigen states

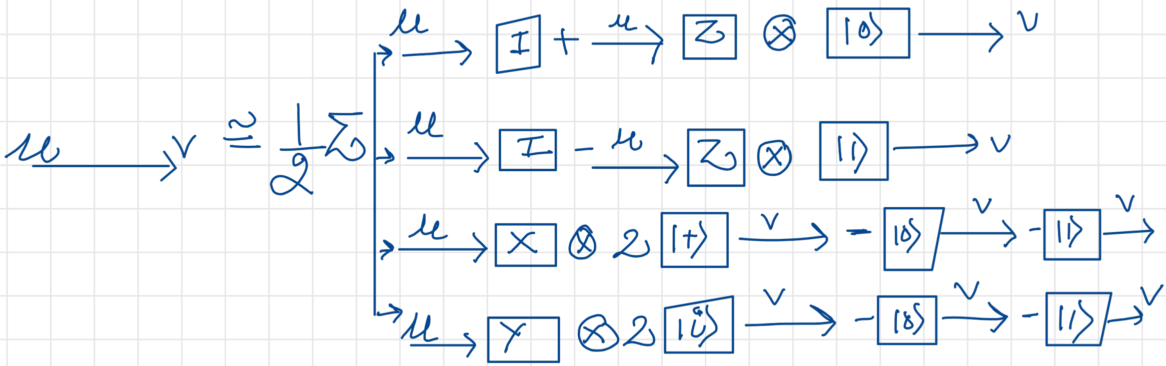
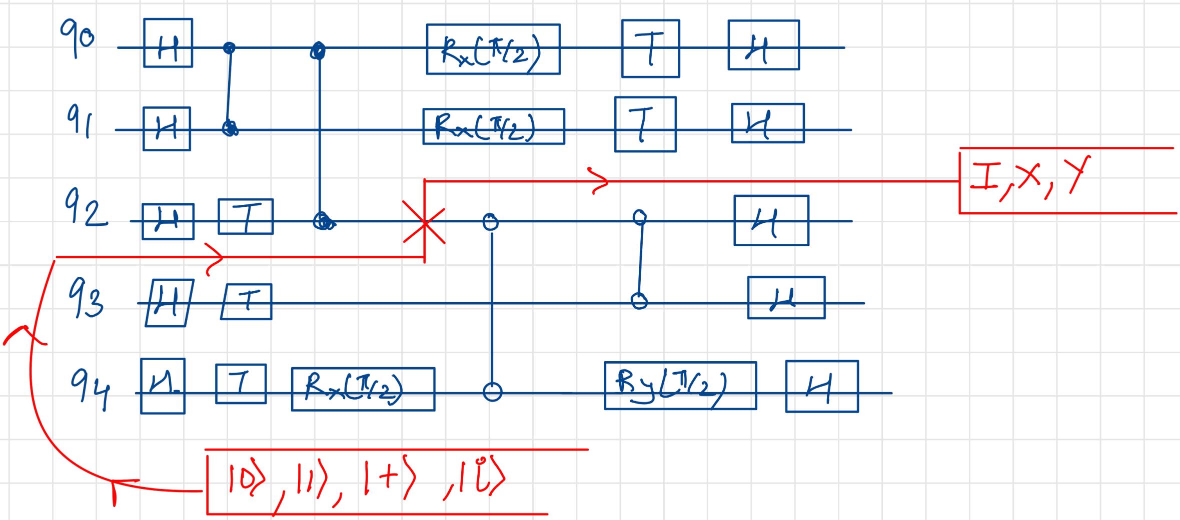


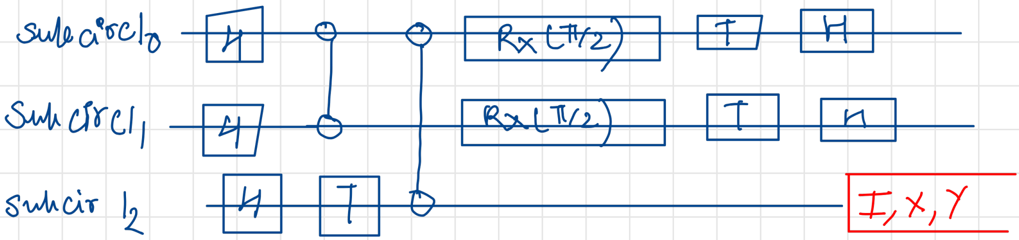
Fig 2: Procedure to cut one qubit wire. The wire between vertices u and v can be cut by summing over four pairs of measurement circuits appended to v. Measurement in I and Z have same physical meaning.

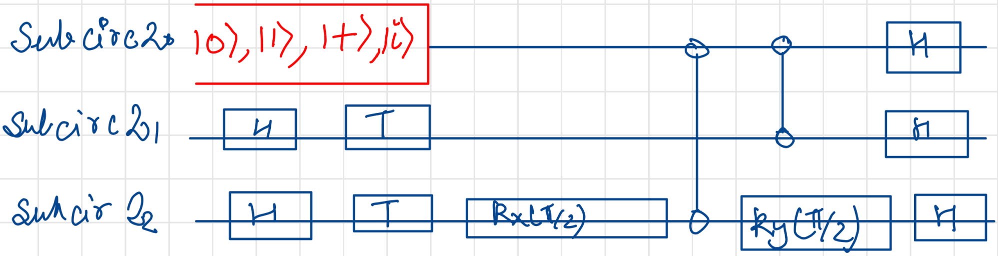
3 upstream : $\boxed{I}$ $\boxed{X}$ $\boxed{Y}$

4 downstream: $|0\rangle, |1\rangle, |+\rangle, |i\rangle$

Circuit Cutting : Example







A 5 qubit circuit is cut into two smaller subcircuits of three qubits each.

How to select the cut locations? (see later)

Also see that subcirc 1_2 does not appear in the final output

∴ Each shot from subcirc 1 needs to be multiplied by a ± 1 contingent on the measurement outcomes of subc 1_2. Specifically

$$\overline{xx0}, \overline{xx1} \longrightarrow +\overline{xx} \qquad M_2 = I$$

$$\overline{xx0} \longrightarrow +\overline{xx}$$

$$\overline{xx1} \longrightarrow -\overline{xx} \quad \text{otherwise}$$

where $\overline{xx}$ is the measurement outcome of the qubits 1_0 and 1_1

Let's see this with an example;

Let the state $|\underbrace{01}_{\text{sub1}}\underbrace{010}_{\text{sub2}}\rangle$

Sub circuit 1 $|01\rangle$ third combo $|0\rangle$ or $|1\rangle$

$$p_{1,1} = p(|1010\rangle|I) + p(|1011\rangle|I) + p(|1010\rangle|Z) - p(|1011\rangle|Z)$$

$$p_{1,2} = p(|1010\rangle I) + p(|1011\rangle I) - p(|1010\rangle Z) + p(|1011\rangle Z)$$

$$p_{1,3} = p(|1010\rangle X) - p(|1011\rangle|X)$$

$$p_{1,4} = p(|1010\rangle Y) - p(|1011\rangle Y)$$

that's why we have

$|1010\rangle, |1011\rangle \rightarrow +\overline{01}$ $M_2 = I$

$|1010\rangle \longrightarrow +01$

$|1011\rangle \rightarrow -01$

$$\left(p(|1010\rangle I) + p(|1011\rangle I\right)$$

Sign will be +ve since $M_2 = I$

Two possibilities

Similarly

$$p(|1010\rangle Z) - p(|1011\rangle Z)$$

similarly $p(|1010\rangle x) - p(|1011\rangle x)$

$p(|1010\rangle Y) - p(|1011\rangle Y)$

Sign will be -ve since $|1010\rangle = +|01\rangle$

$|1011\rangle = -|01\rangle$

Two possibilities

Now for the subcircuit 2

$$p_{2,1} = p(|1010\rangle)|0\rangle \qquad p_{2,2} = p(|1010\rangle)|1\rangle$$

$$p_{2,3} = 2p(|1010\rangle|+\rangle) - p(|1010\rangle|0\rangle) - p(|1010\rangle|1\rangle)$$

$$p_{2,4} = 2p(|1010\rangle|i\rangle) - p(|1010\rangle|0\rangle) - p(|1010\rangle|1\rangle)$$

Building up to full probabilities

The full probability distribution can be reconstructed by taking the relevant outputs of the two smaller subcircuits performing four pairs of Kronecker product and summing them up together.

$$p(|01010\rangle) = \frac{1}{2}\sum_{i=1}^{4} p_{1,i} \otimes p_{2,i}$$

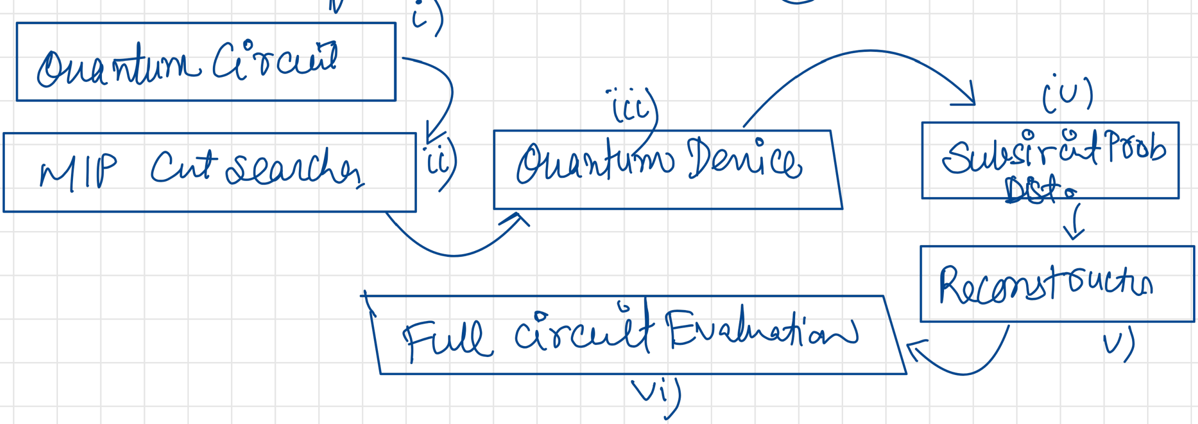
do not take too few shots. (at most 8192)

Challenges

i) Where to place the cut? or How to find the cut location?
For general quantum circuits with n quantum edges, this task faces $O(n!)$ combinatorial search space.

ii) How to scale the classical post-processing?

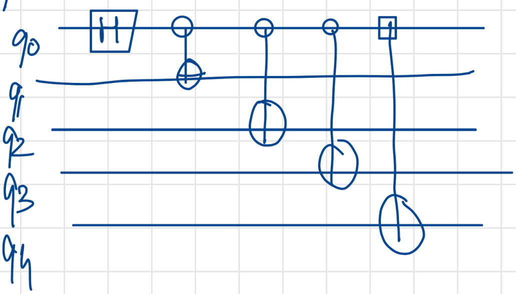
We make use of MIP cut searcher using Gurobi Solver.



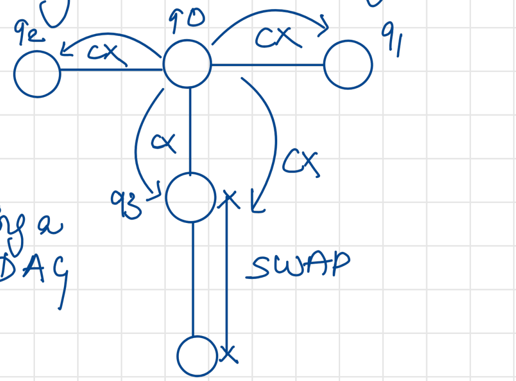
Mixed Integer Programming Cut Searcher.

Assumptions:

i) Input quantum circuit is fully connected i.e all qubits are connected via multiqubit gate either directly or indirectly



Modelling a QC as a DAG



Model parameters

a) input quantum circuit

b) maximum number of qubits allowed per subcircuit D
≈ size of the quantum device available to the user.

c) maximum number of subcircuits allowed n_c

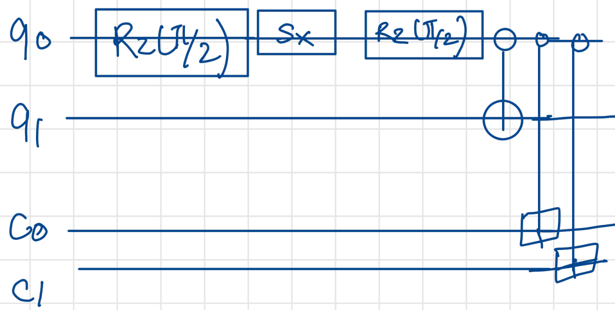
The single qubit gates are ignored during the cut finding process, since they do not affect the connectivity of the circuit.

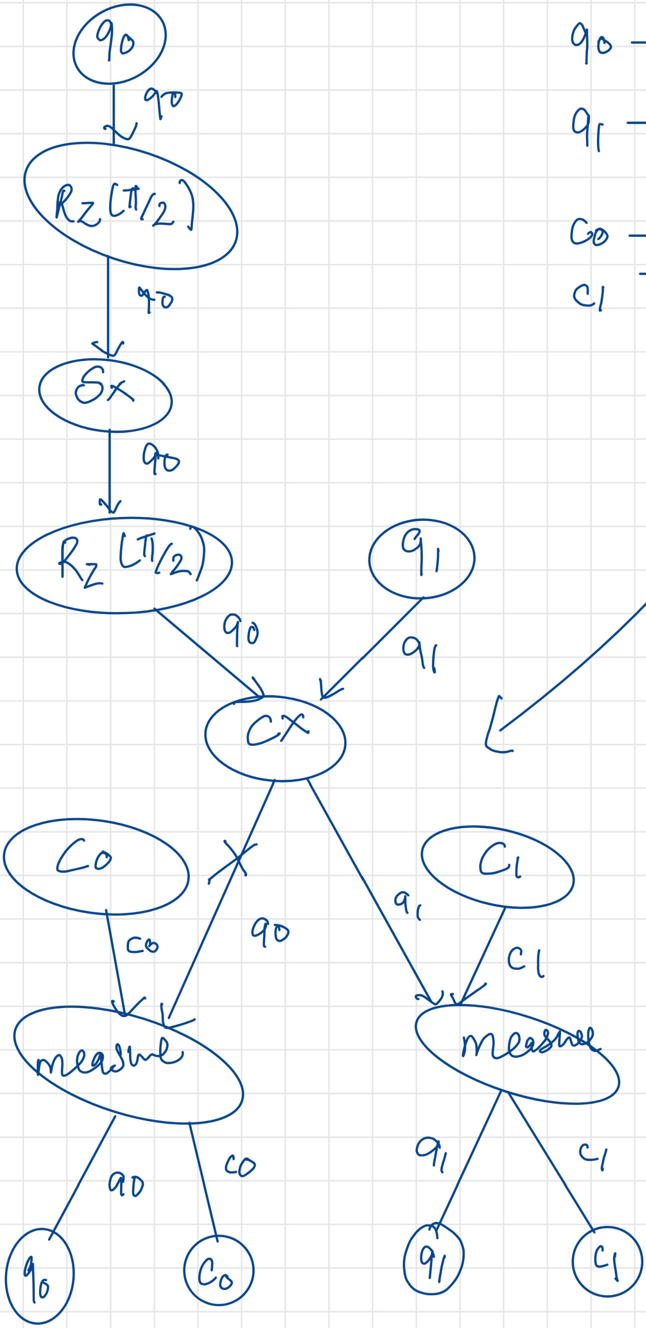
The multiqubit quantum gates are modelled as vertices $V = \{v_1, \dots v_n\}$ and the qubit wire as edges $E = \{(e_a, e_b) : e_a \in V, e_b \in V\}$

Chosing which edge to cut in order to split G into subcircuits $C = \{C_1, \dots C_{n_c}\}$ can be thought as clustering the vertices

To visualize a quantum circuit as a graph. Qiskit does that internally using 'rustworkx'. An Example

Quantum Circuit as a Direct Acyclic Graph

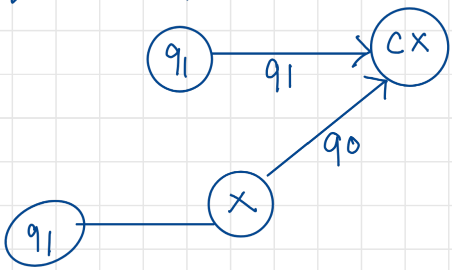




Vertices are the quantum gates and edges are the qubits among them

The MIP cutter assign a parameter 'w' to each vertices 'v' that indicates the number of original input qubits directly connected to it.

$w_v \in \{0,1,2\}$



$w_v = 2$

Since any gate involving more than 2 qubits in can be decomposed into native set.

Constrained Graph Clustering Algorithms

$$y_{v,c} = \begin{cases} 1, & \text{if vertex } v \text{ is in subcircuit } c \quad \forall v \in V, \forall c \in C \\ 0, & \text{otherwise} \end{cases}$$

$$x_{e,c} = \begin{cases} 1, & \text{if edge 'e' is cut by subcircuit } c \quad \forall e \in E \ \forall c \in C \\ 0, & \text{otherwise} \end{cases}$$

The number of qubits required to run a subcircuit is the sum of two parts, namely the number of original input qubits and the number of initialization qubit induced by cutting.

weight factors for the vertices.

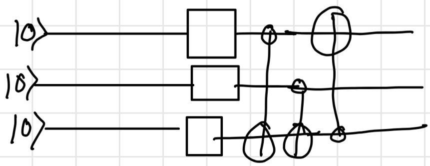
$$\alpha_c \equiv \sum w_v \times y_{v,c} \quad \forall c \in C$$

original input qubits in each subcircuit

Mixed Integer Programming Cut Searcher.

i) Model Parameters

a) input quantum circuit



b) maximum number of qubits allowed per subcircuit (D)

c) maximum number of subcircuits allowed (n_c)

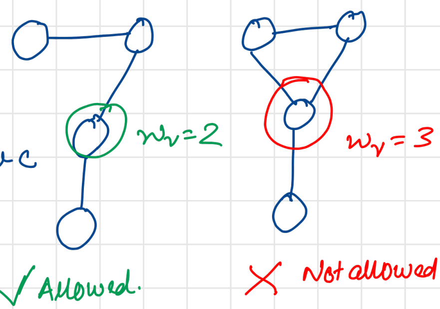
d) maximum number of cuts allowed.

quantum gates are modeled as vertices $V = \{v_1, v_2, \dots v_{n_v}\}$ and the qubit wires are modeled as edges. $E = \{(e_a, e_b) : e_a \in V, e_b \in V\}$

the set of subcircuits $C = \{c_1, c_2 \dots c_{n_c}\}$

we also make use of a parameter w (weight) associated with each vertice that indicates the number of original inputs qubits directly connected to v. $w_v \in \{0, 1, 2\}$

(can be made to accomodate multi-qubit gates)



ii) Variables

$$y_{v,c} = \begin{cases} 1, & \text{if vertex } v \text{ is in sub } c \\ 0 & \text{otherwise} \end{cases}$$

gate

$\forall\, v \in V$

$\forall\, c \in C$

$$x_{e,c} = \begin{cases} 1 & \text{if edge } e \text{ is cut by subcircuit } c \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, e \in E \quad \forall\, c \in C$$

qubit

Q)

The number of qubits required to run a subcircuit is the sum of two parts, i) the number of original input qubits ii) the number of initialization qubits induced after cutting.

$$x_c \equiv \sum_{v \in V} w_v \times y_{v,c} \qquad \forall c \in C$$

this is basically the sum of qubits directly connected to the vertex v.

b) A subcircuit requires initialization qubits when a downstream vertex e_b is in the subcircuit for some edge (e_a, e_b) that is cut. The number of initialization qubits ρ_c is

$$\rho_c \equiv \sum_{e:(e_a,e_b) \in E} x_{e,c} \times y_{e_b,c} \qquad \forall c \in C$$

c) A subcircuit requires measurement qubits when an upstream vertex e_a is in the subcircuit for some edge (e_a, e_b) that is cut.

$$O_c \equiv \sum x_{e,c} \times y_{e_a,c} \quad , \forall c \in C$$

consequently the number of qubits in a subcircuit that contributes to the final measurement of the original uncut circuit is

$$f_c \equiv x_c + \rho_c - O_c \quad , \forall c \in C$$

iii) Constraints

a) Every vertex be assigned to exactly one subcircuit

$$\sum_{c \in C} y_{v,c} = 1 \qquad \forall v \in V$$

b) the qubits (d_c) in a subcircuits (c) be no larger than the input device size D.

$$d_c \equiv x_c + \rho_c \leq D \qquad \forall c \in C$$

c) Edge constraint

$$x_{e,c} = y_{e_a,c} \oplus y_{e_b,c} \quad \forall e = (e_a, e_b) \in E, c \in C$$

This formulation force vertices with smaller indices to be in subcircuits with smaller indices.

iv) Objective function

We seek to minimize the classical post-processing overhead required to reconstruct a circuit from its subcircuit. Hence the objective function is the number of Floating-point multiplications involved.

number of cuts made is

$$K = \frac{1}{2} \sum_{c \in C} \sum_{e \in E} x_{e,c}$$

The objective function hence becomes.

$$L = 4^{K} \sum_{c=2}^{n_c} \sum_{i=1}^{c} 2^{f_i}$$

minimize this

Classical Post Processing

Full Definition

do you want to reconstruct the full 2^n state probability

Dynamic Definition

or not.

this in turn make use of greedy subcircuit order, early termination and parallel processing

vector arithmetic have no data dependency and hence can be easily executed in parallel.

place the smallest subcircuit first hence affect the carryover vector and less FLOPs. (may reduce overhead by 50%)

if one vector of the subcircuit goes 0, the all goes 0, hence good to terminate early.