

Dynamic quantum circuit compilation

[2310.11021]

Kun Fang

School of Data Science

The Chinese University of Hong Kong, Shenzhen

Joint work with Yinan Li, Ruqi Shi, Munan Zhang



香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen



武漢大學
WUHAN UNIVERSITY

@Shenzhen-Nagoya Workshop on Quantum Science 2024

Main challenges of Quantum Computing

Scalability

- Existing quantum computers are limited by the **number of qubits** available for computation.
- We need to **make efficient use of qubits** when designing and executing quantum algorithms.

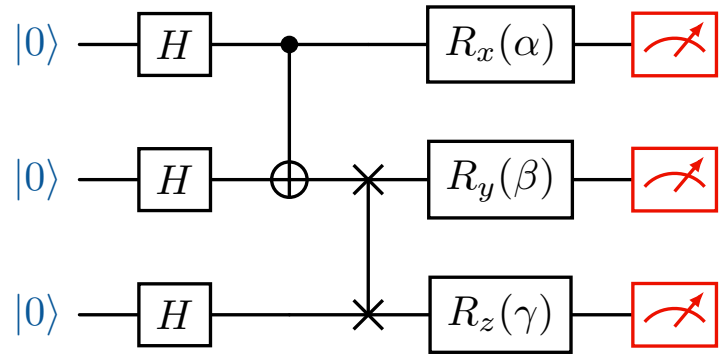
Fidelity

- Current generation of quantum computers are extremely prone to noise.
- **Circuit optimization** and **error correction** needed !

...

Motivation from Hardware

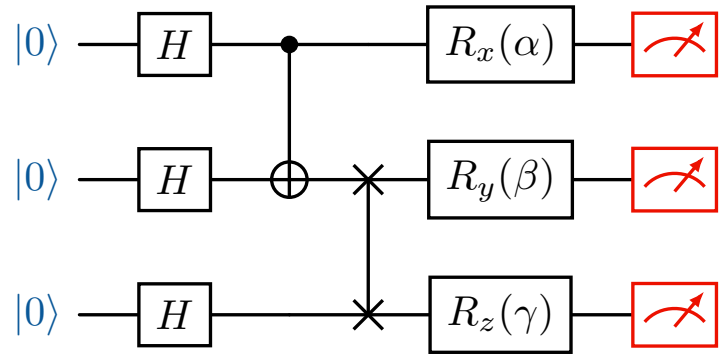
Static Quantum Circuit



- Start with state initialization
- Measurement at the end

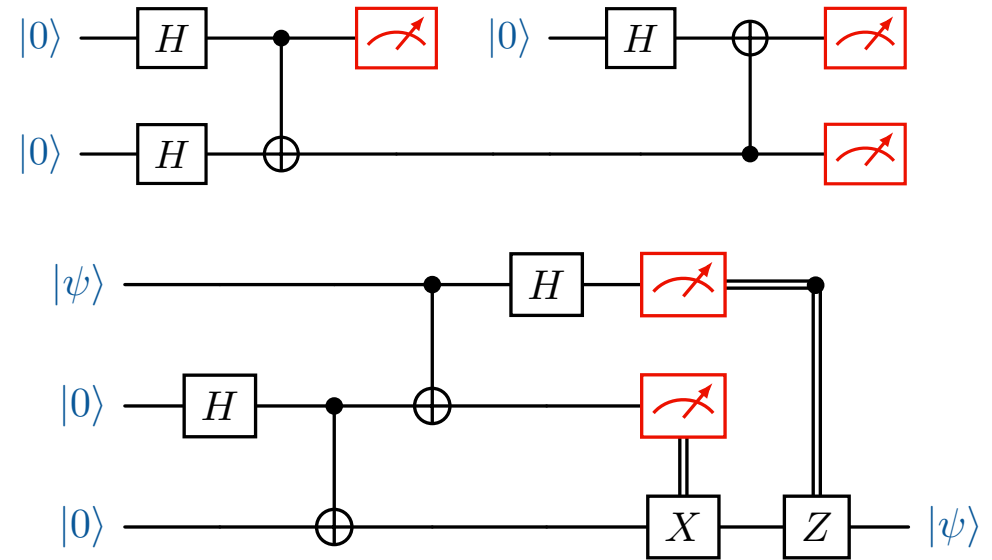
Motivation from Hardware

Static Quantum Circuit



- Start with state initialization
- Measurement at the end

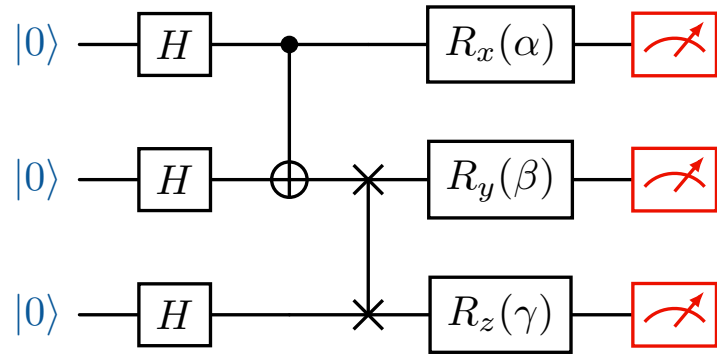
Dynamic Quantum Circuit



- Mid-circuit measurement and reset
- Classically controlled gate

Motivation from Hardware

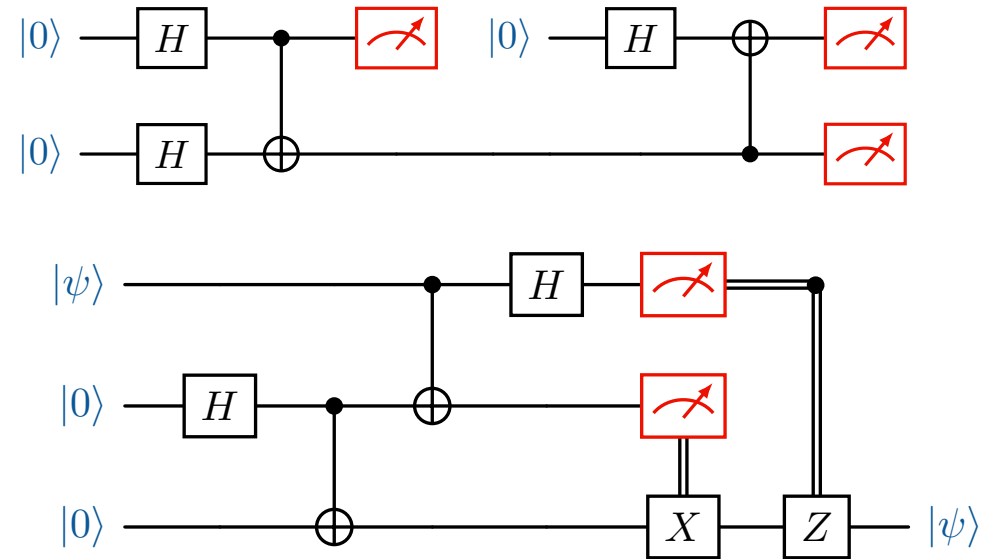
Static Quantum Circuit



- Start with state initialization
- Measurement at the end

Quantum Computation

Dynamic Quantum Circuit



- Mid-circuit measurement and reset
- Classically controlled gate

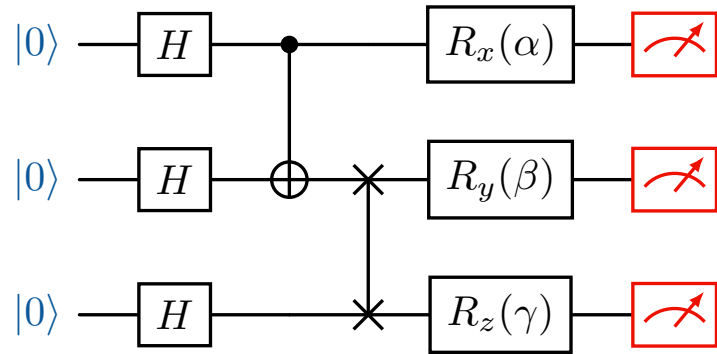
Quantum Computation

+

Classical Computation and Communication

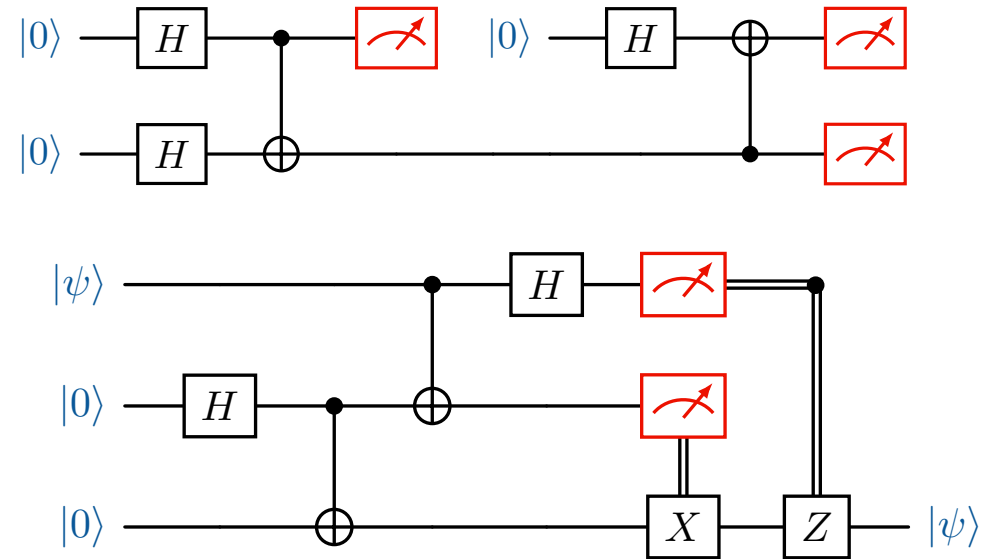
Motivation from Hardware

Static Quantum Circuit



- Start with state initialization
- Measurement at the end

Dynamic Quantum Circuit



- Mid-circuit measurement and reset
- Classically controlled gate

Quantum Error Correction !
(Syndrome + Correction)

Motivation from Hardware

Superconducting Circuit

- Relatively easy for scaling
- Limited qubit connectivity
- Short coherence time

Trapped Ion

- Extremely long coherence time.
- All-to-all qubit connectivity,
- Relatively hard for scaling up.

Motivation from Hardware

Superconducting Circuit

- Relatively easy for scaling
- Limited qubit connectivity
- Short coherence time

Trapped Ion

- Extremely long coherence time.
- All-to-all qubit connectivity,
- Relatively hard for scaling up.

Our main target platform in this work


Hardware Support for Dynamic Quantum Circuit

Exploiting Dynamic Quantum Circuits in a Quantum Algorithm with Superconducting Qubits

A. D. Córcoles, Maika Takita, Ken Inoue, Scott Lekuch, Zlatko K. Mineev, Jerry M. Chow, and Jay M. Gambetta
Phys. Rev. Lett. **127**, 100501 – Published 31 August 2021

IBM, 2021

Demonstration of the trapped-ion quantum CCD computer architecture

[J. M. Pino](#), [J. M. Dreiling](#), [C. Figgatt](#), [J. P. Gaebler](#), [S. A. Moses](#), [M. S. Allman](#), [C. H. Baldwin](#), [M. Foss-Feig](#), [D. Hayes](#) , [K. Mayer](#), [C. Ryan-Anderson](#) & [B. Neyenhuis](#)

[Nature](#) **592**, 209–213 (2021) | [Cite this article](#)

Honeywell, 2021

Suppressing quantum errors by scaling a surface code logical qubit

Superconducting Circuit

[Google Quantum AI](#)

[Nature](#) **614**, 676–681 (2023) | [Cite this article](#)

Google, 2023

Midcircuit Qubit Measurement and Rearrangement in a ^{171}Yb Atomic Array

M. A. Norcia *et al.*

Phys. Rev. X **13**, 041034 (2023) – Published 22 November 2023

Atom Computing, 2023

Hardware Support for Dynamic Quantum Circuit

Beating the break-even point with a discrete-variable-encoded logical qubit

[Zhongchu Ni](#), [Sai Li](#), [Xiaowei Deng](#), [Yanyan Cai](#), [Libo Zhang](#), [Weiting Wang](#), [Zhen-Biao Yang](#), [Haifeng Yu](#), [Fei Yan](#), [Song Liu](#), [Chang-Ling Zou](#), [Luyan Sun](#) ✉, [Shi-Biao Zheng](#) ✉, [Yuan Xu](#) ✉ & [Dapeng Yu](#) ✉

Nature **616**, 56–60 (2023) | [Cite this article](#)

SUSTech, 2023

High-Fidelity Detection on $^{171}\text{Yb}^+$ Qubit via $^2D_{3/2}$ Shelving

Xueying Mai,^{*} Liyun Zhang,^{*} and Yao Lu[†]

¹*Shenzhen Institute for Quantum Science and Engineering (SIQSE),
Southern University of Science and Technology, Shenzhen, P. R. China.*

²*International Quantum Academy, Shenzhen 518048, China.*

³*Guangdong Provincial Key Laboratory of Quantum Science and Engineering,
Southern University of Science and Technology Shenzhen, 518055, China.*

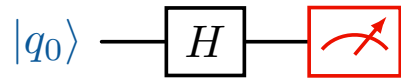
(Dated: March 1, 2024)

SUSTech, 2024

Background - Dynamic quantum circuit compilation

(yield same sampling results)

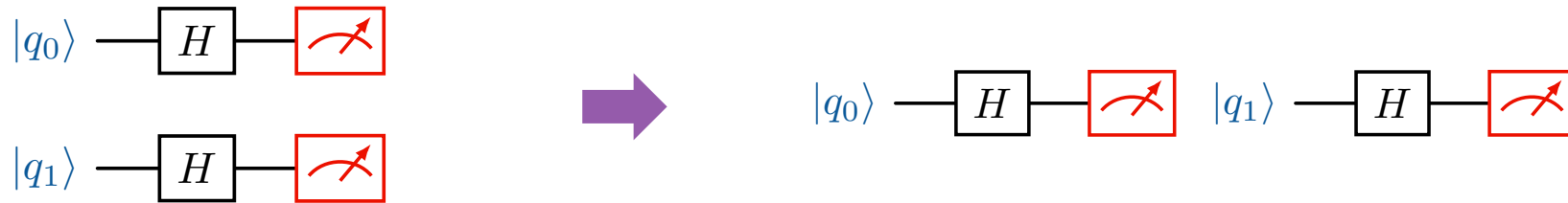
Rewires static quantum circuits into equivalent dynamic circuits using **fewer** qubits through qubit-reuse strategies.



Background - Dynamic quantum circuit compilation

(yield same sampling results)

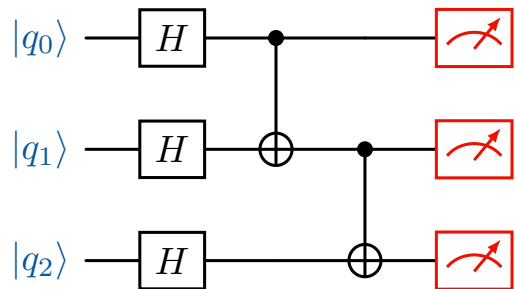
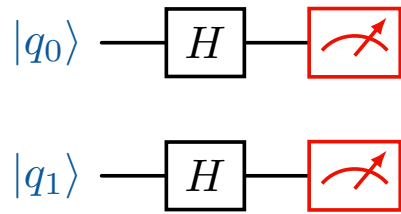
Rewires static quantum circuits into equivalent dynamic circuits using **fewer** qubits through qubit-reuse strategies.



Background - Dynamic quantum circuit compilation

(yield same sampling results)

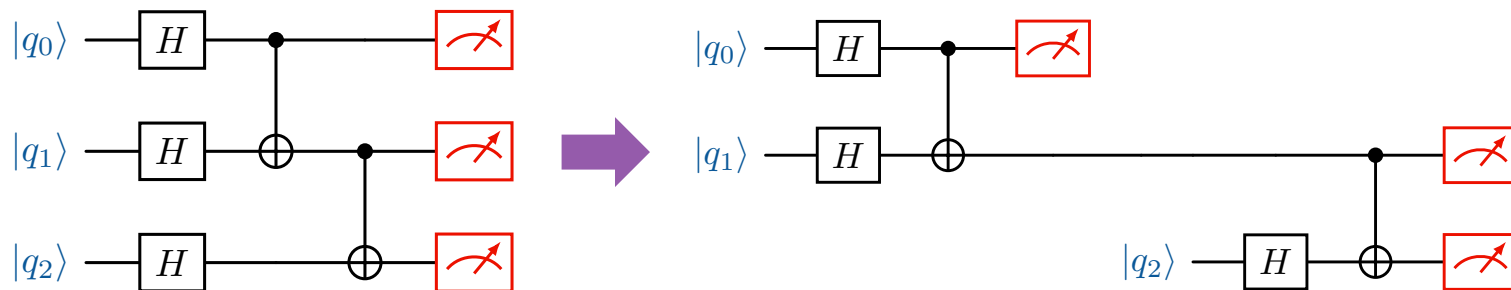
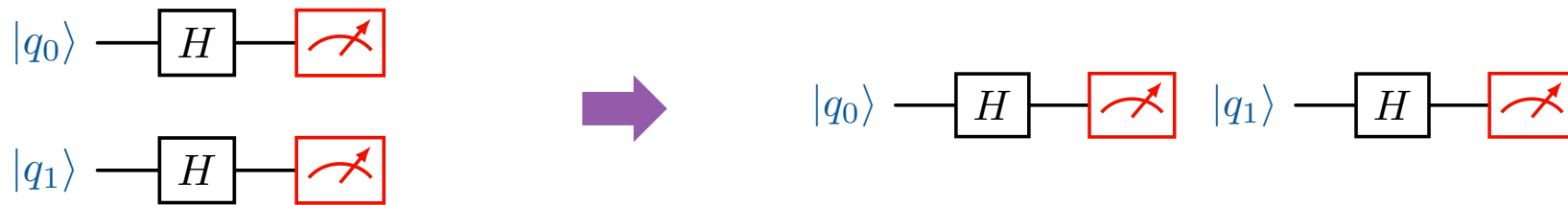
Rewires static quantum circuits into equivalent dynamic circuits using **fewer** qubits through qubit-reuse strategies.



Background - Dynamic quantum circuit compilation

(yield same sampling results)

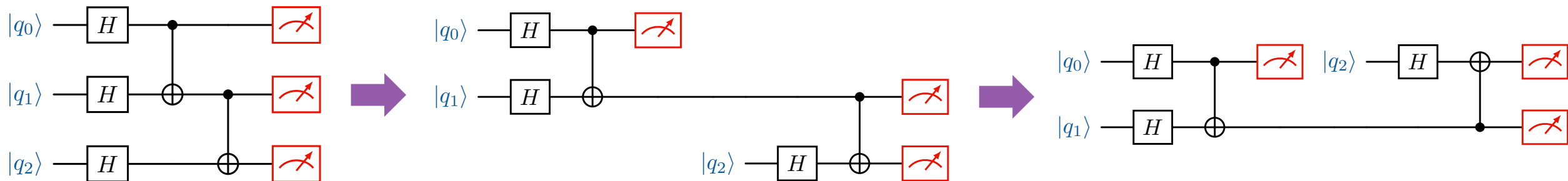
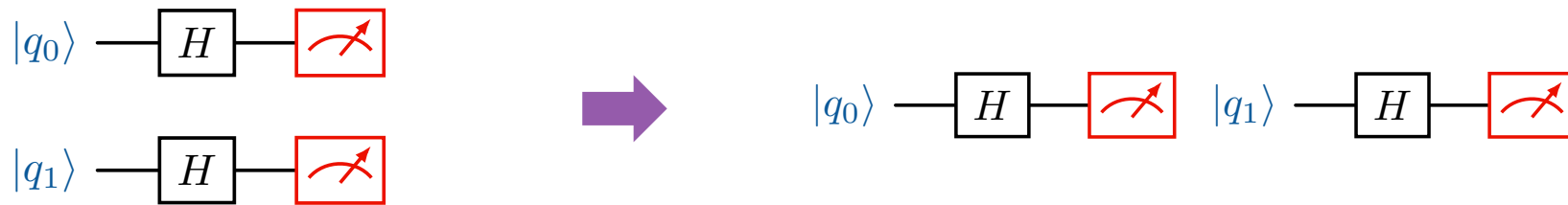
Rewires static quantum circuits into equivalent dynamic circuits using **fewer** qubits through qubit-reuse strategies.



Background - Dynamic quantum circuit compilation

(yield same sampling results)

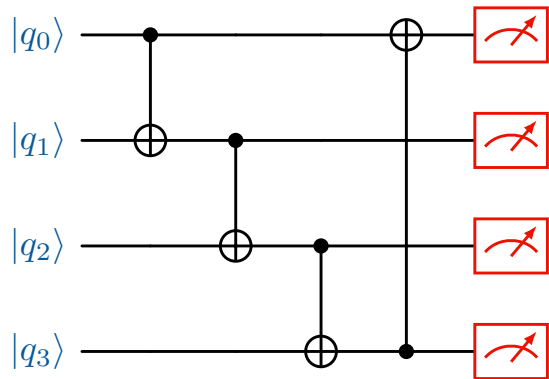
Rewires static quantum circuits into equivalent dynamic circuits using **fewer** qubits through qubit-reuse strategies.



Background - Dynamic quantum circuit compilation

(yield same sampling results)

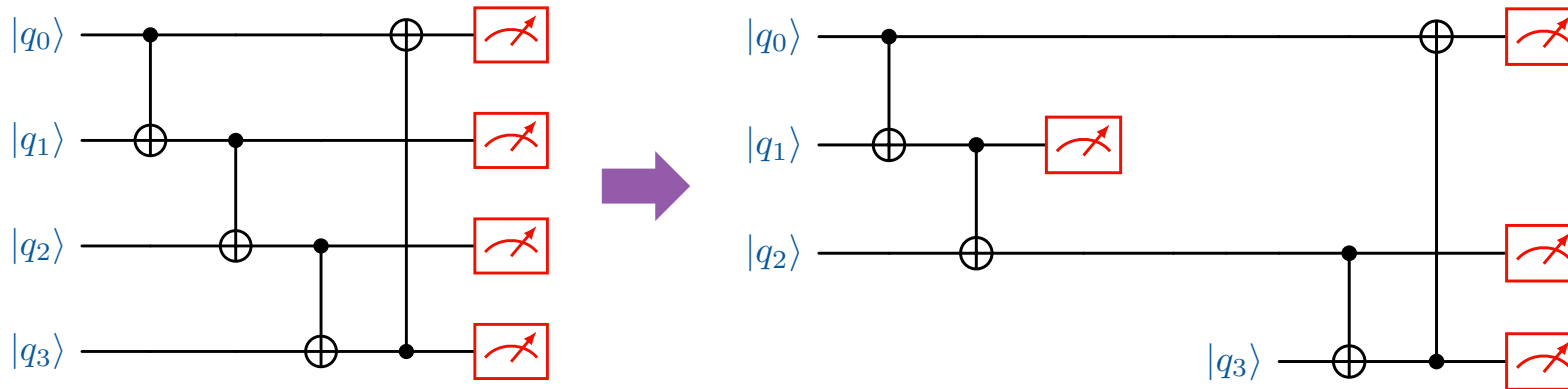
Rewires static quantum circuits into equivalent dynamic circuits using **fewer** qubits through qubit-reuse strategies.



Background - Dynamic quantum circuit compilation

(yield same sampling results)

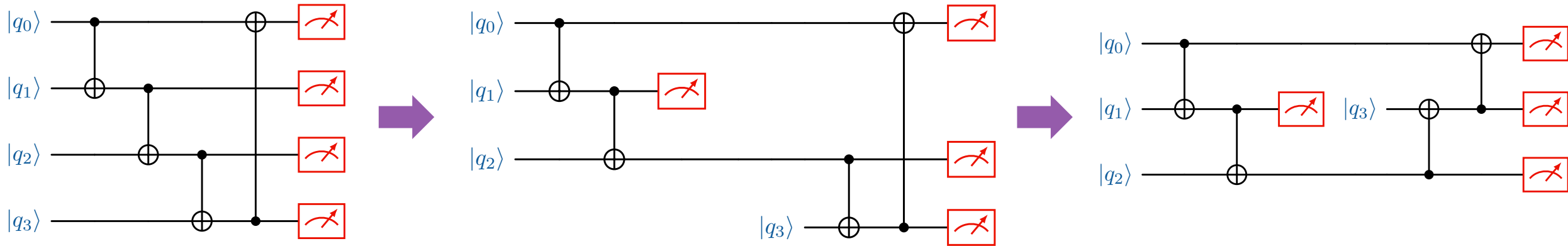
Rewires static quantum circuits into equivalent dynamic circuits using **fewer** qubits through qubit-reuse strategies.



Background - Dynamic quantum circuit compilation

(yield same sampling results)

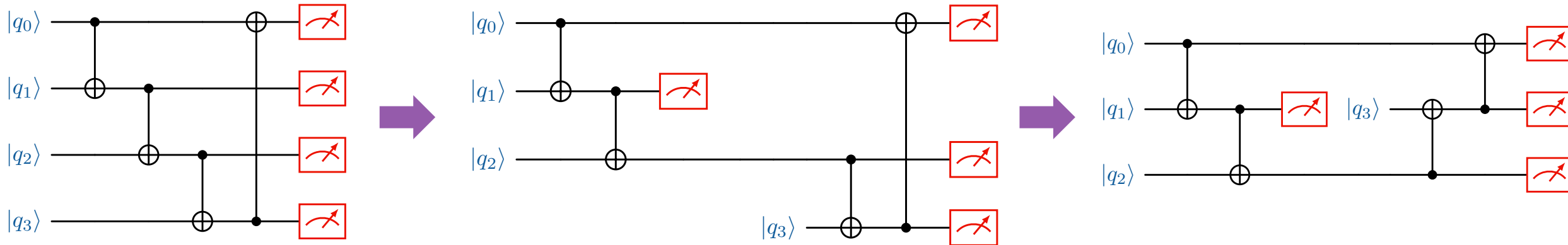
Rewires static quantum circuits into equivalent dynamic circuits using **fewer** qubits through qubit-reuse strategies.



Background - Dynamic quantum circuit compilation

(yield same sampling results)

Rewires static quantum circuits into equivalent dynamic circuits using **fewer** qubits through qubit-reuse strategies.



Hard to compile manually for large circuits.

Automatic Tool Needed!

Take-home Message*

Large-Scale
Quantum Algorithm

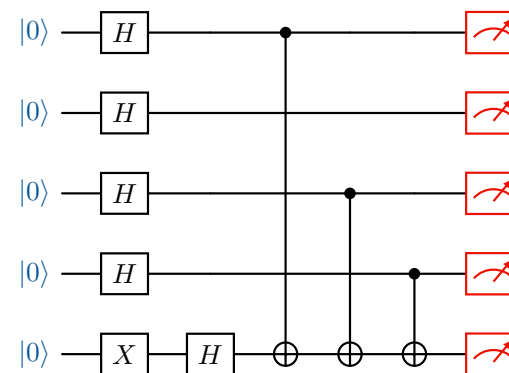
GAP

Quantum Computer
with Limited Resource

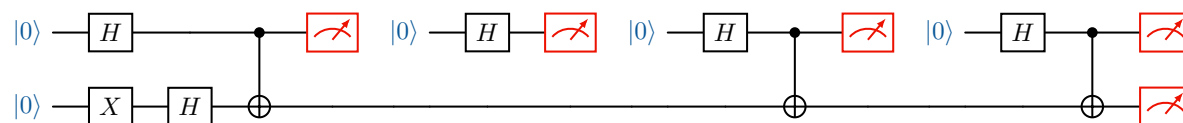
Dynamic
Quantum

Circuit
Compilation

N-qubit Bernstein-Vazirani Algorithm



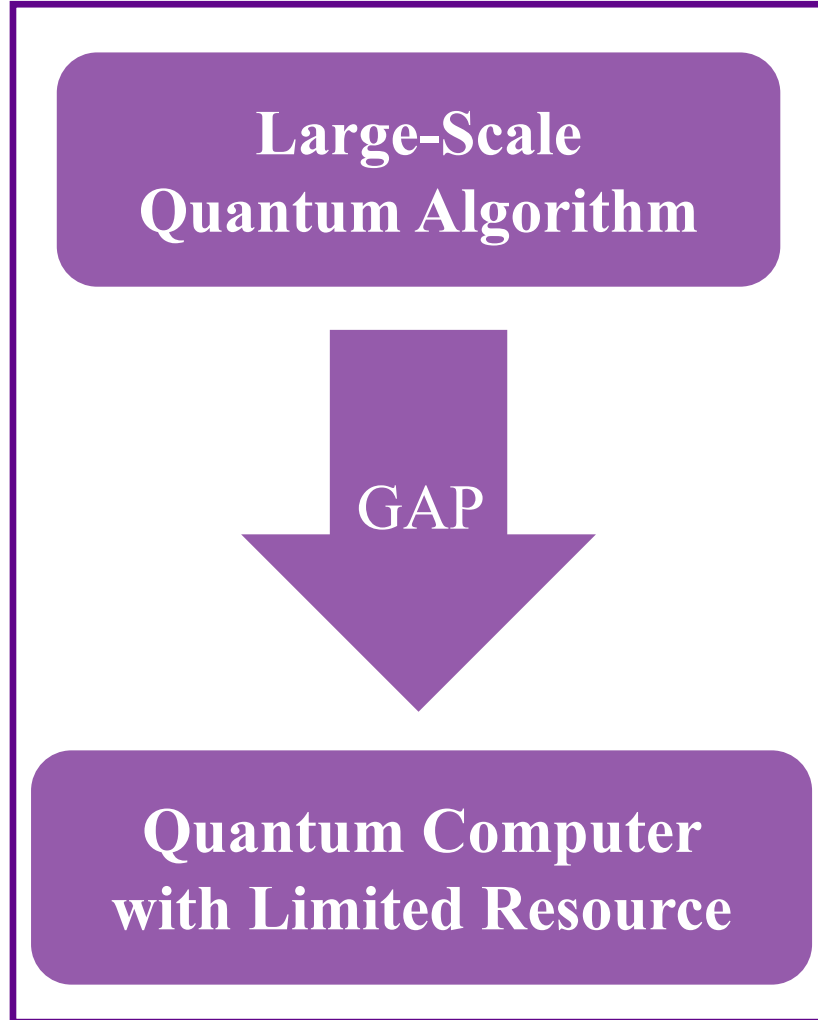
2-qubit with Dynamic Circuit



Advantages

- Qubit Saving
- Swap Reduction
- Improved Fidelity

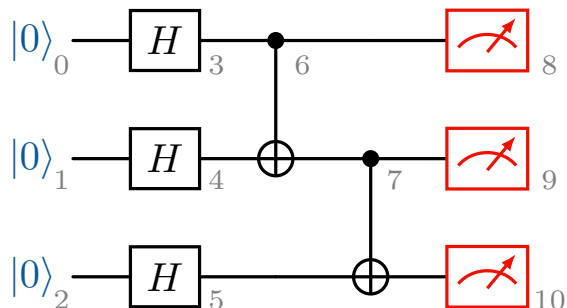
Take-home Message*



- Dynamic circuit compilation **reduces the number of qubits** required for executing quantum algorithms.
- **Bridging the gap** between theoretical quantum algorithms and their practical implementation on quantum computers with limited resources.
- **Complementary** to other circuit optimization techniques and can be **seamlessly integrated** into existing ones.

Quantum Circuit Representations

Circuit Diagram

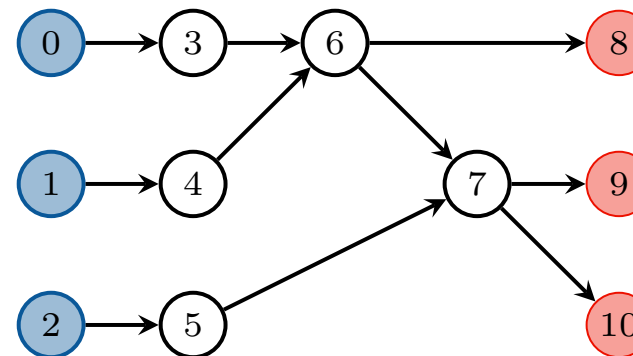


Circuit Instruction

Quantum Circuit Instructions

ID	TYPE	QUBIT	PAR
0	RESET	0	NONE
1	RESET	1	NONE
2	RESET	2	NONE
3	H	0	NONE
4	H	1	NONE
5	H	2	NONE
6	CX	[0, 1]	NONE
7	CX	[1, 2]	NONE
8	MEASURE	0	NONE
9	MEASURE	1	NONE
10	MEASURE	2	NONE

Directed Acyclic Graph (DAG) Representation

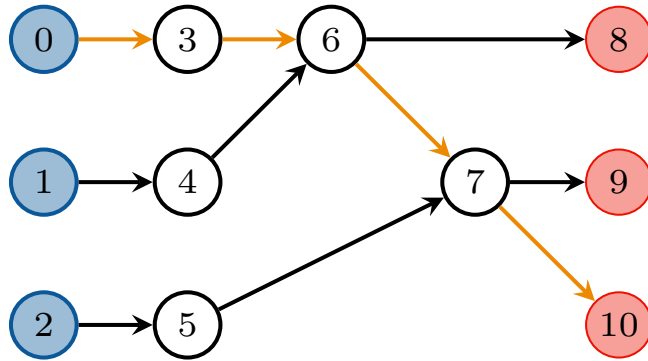


$$G(V = R \cup T \cup I, E)$$

- Root vertices ($r \in R$), indegree $\delta^-(r) = 0$
- Terminal vertices ($t \in T$), outdegree $\delta^+(t) = 0$
- Internal vertices ($i \in I$), $\delta^-(i) \neq 0, \delta^+(i) \neq 0$

Graph Representation Simplified

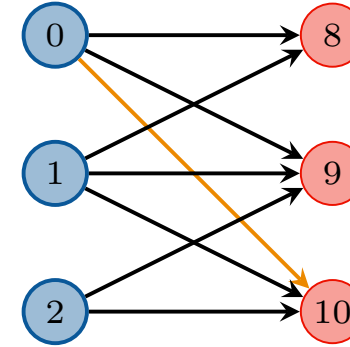
DAG Representation



$$G(V = R \cup T \cup I, E)$$



Simplified DAG

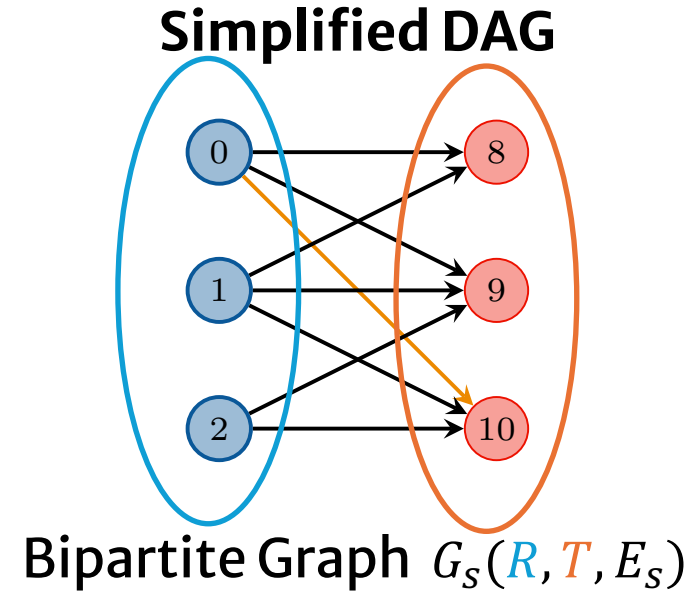
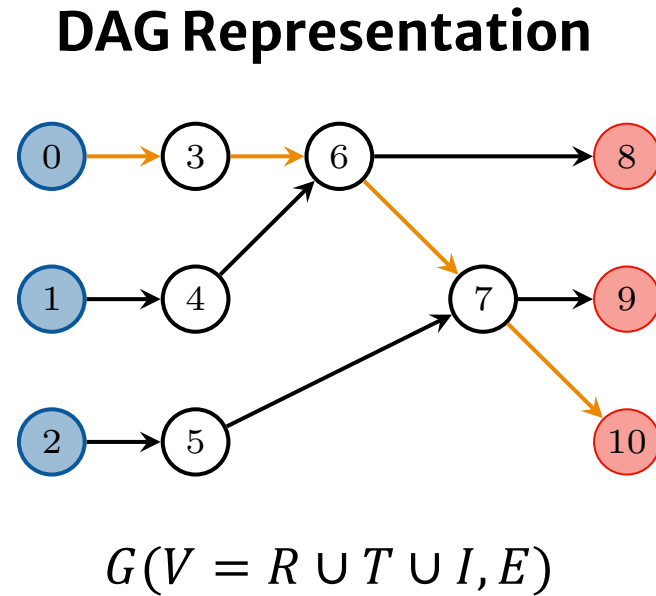


Bipartite Graph $G_S(R, T, E_S)$

Simplify the DAG by using the graph reachability

An edge $(r, t) \in E_S$ if a *directed path* from r to t exists within the DAG representation.

Graph Representation Simplified

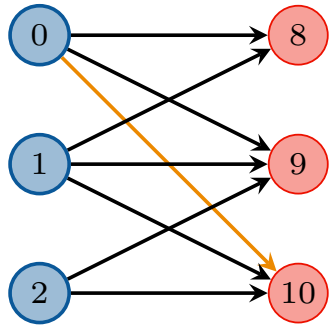


Simplify the DAG by using the graph reachability

An edge $(r, t) \in E_S$ if a *directed path* from r to t exists within the DAG representation.

Matrix Representation of Graph

Simplified DAG



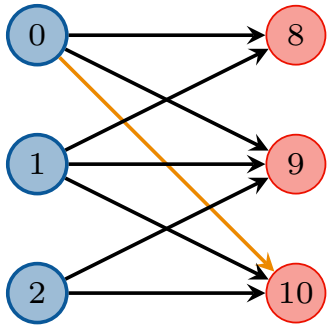
Bipartite Graph $G_s(R, T, E_s)$

Matrix Representation

- $G(V, E)$, Adjacency matrix $A(G)$, $A_{ij} = 1$ if $(v_i, v_j) \in E$
- For a bipartite graph $G_s(R, T, E_s)$, $A(G_s) = \begin{pmatrix} 0 & B \\ 0 & 0 \end{pmatrix}$
- Biadjacency matrix $B(G_s)$, $B_{ij} = 1$ if $(r_i, t_j) \in E$

Matrix Representation of Graph

Simplified DAG



Bipartite Graph $G_S(R, T, E_S)$

Matrix Representation

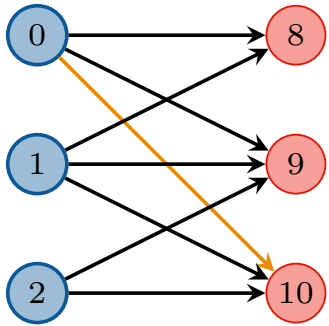
- $G(V, E)$, **Adjacency matrix** $A(G)$, $A_{ij} = 1$ if $(v_i, v_j) \in E$
- For a bipartite graph $G_S(R, T, E_S)$, $A(G_S) = \begin{pmatrix} O & B \\ O & O \end{pmatrix}$
- **Biadjacency matrix** $B(G_S)$, $B_{ij} = 1$ if $(r_i, t_j) \in E$

Biadjacency Matrix

$$\begin{matrix} & t_0 & t_1 & t_2 \\ r_0 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_1 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_2 & \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

Matrix Representation of Graph

Simplified DAG



Bipartite Graph $G_S(R, T, E_S)$

Matrix Representation

- $G(V, E)$, **Adjacency matrix** $A(G)$, $A_{ij} = 1$ if $(v_i, v_j) \in E$
- For a bipartite graph $G_S(R, T, E_S)$, $A(G_S) = \begin{pmatrix} O & B \\ O & O \end{pmatrix}$
- **Biadjacency matrix** $B(G_S)$, $B_{ij} = 1$ if $(r_i, t_j) \in E$

Biadjacency Matrix

$$\begin{matrix} & t_0 & t_1 & t_2 \\ r_0 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_1 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_2 & \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

Lemma *A directed graph is acyclic if and only if its adjacency matrix is nilpotent* ^[1].

Nilpotent matrix $M_{k \times k}$: $\exists 1 \leq l \leq k, M^l = O_k$

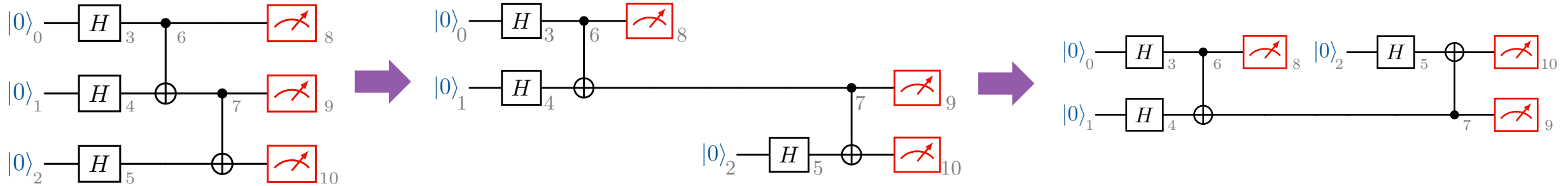
[1] N. Deo, Graph theory with applications to engineering and computer science, Theorem [9-17] (2016).

Circuit Compilation via Graph Manipulation*

First Step: From Circuit to Graph

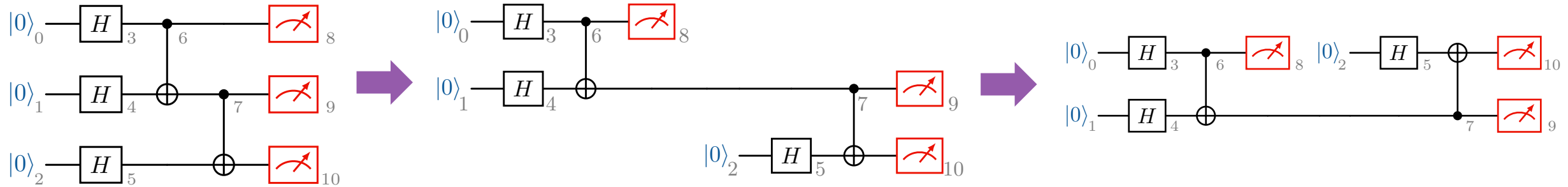
Quantum Circuit Compilation via Graph Manipulation*

Dynamic Circuit Compilation

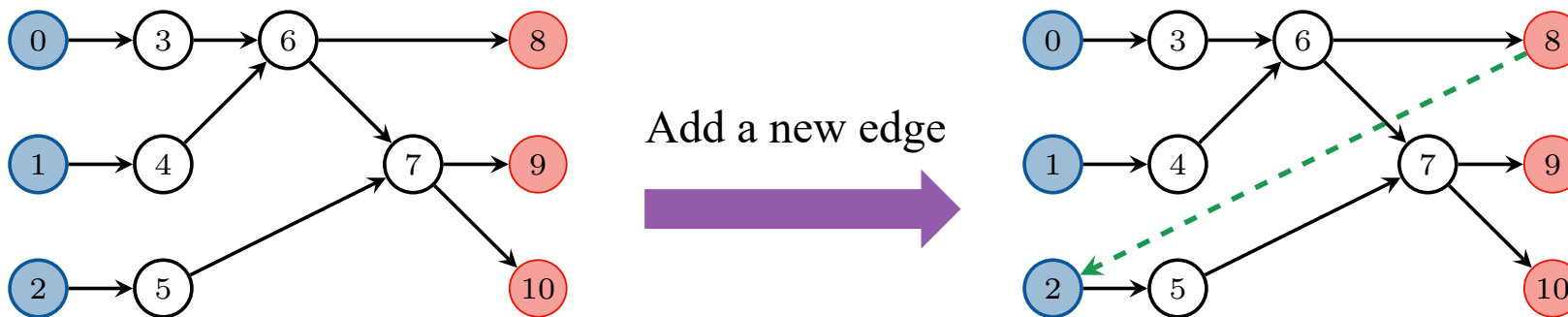


Quantum Circuit Compilation via Graph Manipulation*

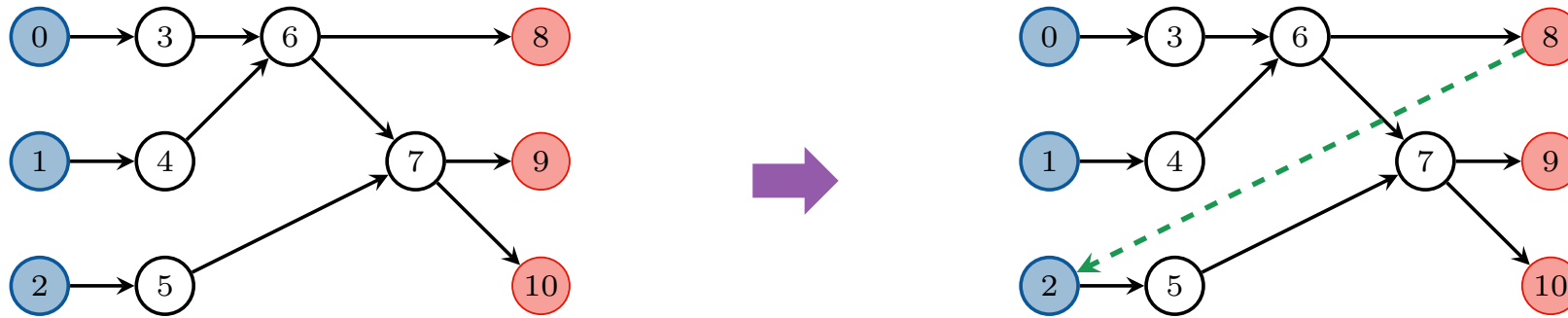
Dynamic Circuit Compilation



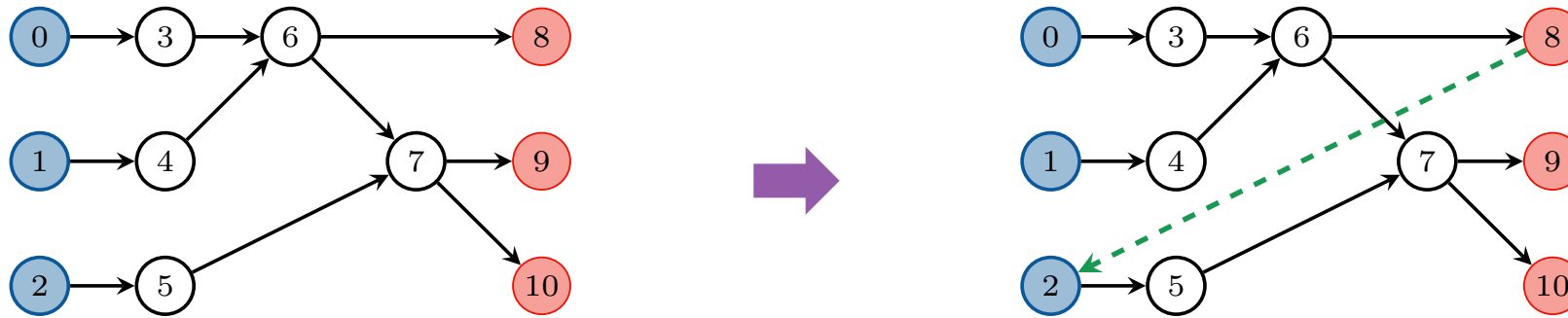
Graph Manipulation



Quantum Circuit Compilation via Graph Manipulation*



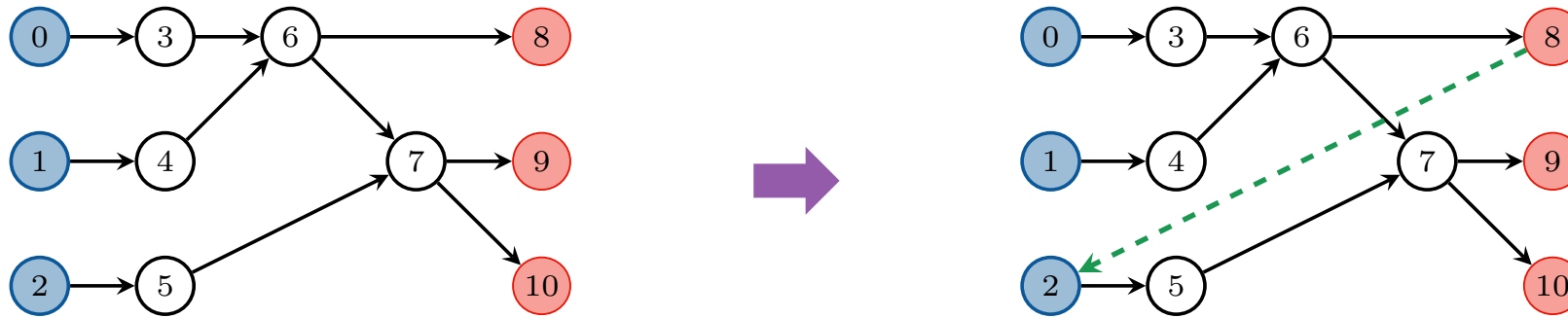
Quantum Circuit Compilation via Graph Manipulation*



Constraints

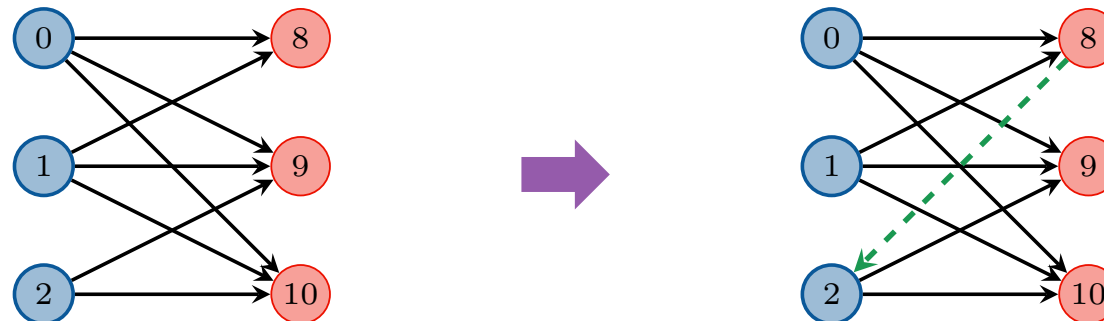
- $\forall (t, r) \in E',$ it has $t \in T$ and $r \in R$;
- $\forall (t, r) \in E',$ it has $\text{outdegree}(t) = 1$ and $\text{indegree}(r) = 1$;
- $G' = (R \cup T, E \cup E')$ is an **acyclic** graph.

Quantum Circuit Compilation via Graph Manipulation*



Constraints

- $\forall (t,r) \in E'$, it has $t \in T$ and $r \in R$;
- $\forall (t,r) \in E'$, it has $outdegree(t) = 1$ and $indegree(r) = 1$;
- $G' = (R \cup T, E \cup E')$ is an acyclic graph.



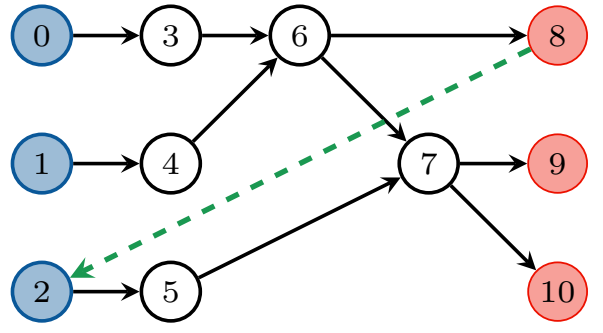
**Valid on
Simplified DAG**

Circuit Compilation via Graph Manipulation*

Last Step: From Graph to Circuit

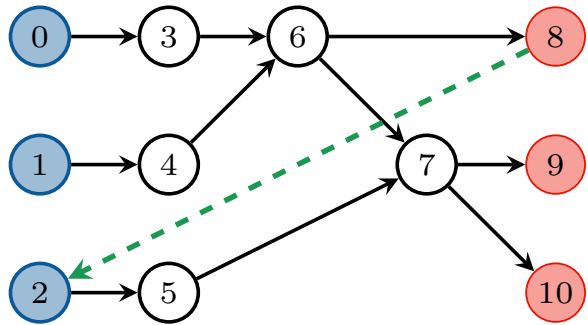
Quantum Circuit Compilation via Graph Manipulation*

DAG with new edges



Quantum Circuit Compilation via Graph Manipulation*

DAG with new edges



Topological



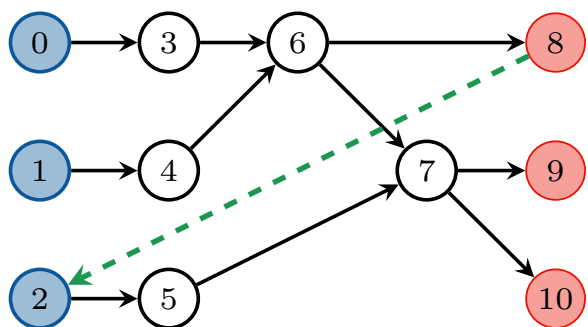
Ordering

Feasible Execution Sequence

0 → 1 → 3 → 4 → 6 → 8 → 2 → 5 → 7 → 10 → 9

Quantum Circuit Compilation via Graph Manipulation*

DAG with new edges



Topological



Ordering

Feasible Execution Sequence

0 → 1 → 3 → 4 → 6 → 8 → 2 → 5 → 7 → 10 → 9

Qubit ↓ Reallocation

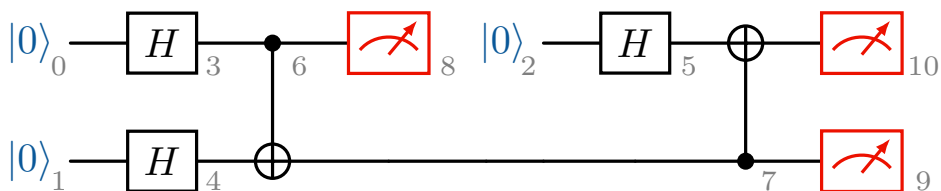


Dynamic Circuit Instructions

Quantum Circuit Instructions			
ID	TYPE	QUBIT	PAR
0	RESET	0	NONE
1	RESET	1	NONE
3	H	0	NONE
4	H	1	NONE
6	CX	[0, 1]	NONE
8	MEASURE	0	NONE
2	RESET	0	NONE
5	H	0	NONE
7	CX	[1, 0]	NONE
10	MEASURE	0	NONE
9	MEASURE	1	NONE

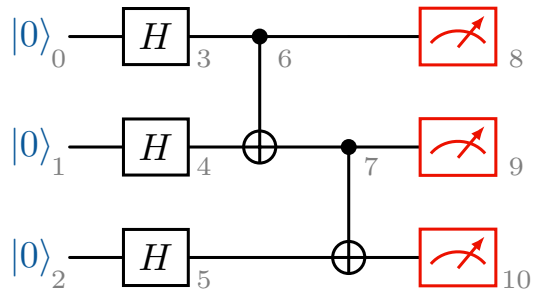
Reallocate 2 to 0

Dynamic Circuit



Compilation Procedures

Static Circuit

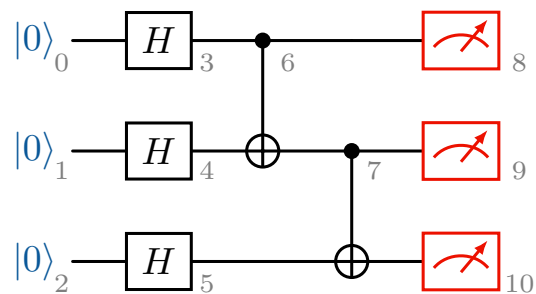


Static Circuit Instruction

Quantum Circuit Instructions			
ID	TYPE	QUBIT	PAR
0	RESET	0	NONE
1	RESET	1	NONE
2	RESET	2	NONE
3	H	0	NONE
4	H	1	NONE
5	H	2	NONE
6	CX	[0, 1]	NONE
7	CX	[1, 2]	NONE
8	MEASURE	0	NONE
9	MEASURE	1	NONE
10	MEASURE	2	NONE

Compilation Procedures

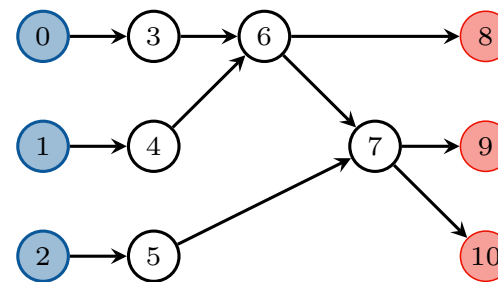
Static Circuit



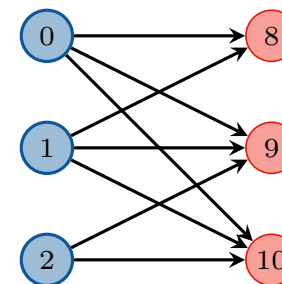
Static Circuit Instruction

ID	TYPE	QUBIT	PAR
0	RESET	0	NONE
1	RESET	1	NONE
2	RESET	2	NONE
3	H	0	NONE
4	H	1	NONE
5	H	2	NONE
6	CX	[0, 1]	NONE
7	CX	[1, 2]	NONE
8	MEASURE	0	NONE
9	MEASURE	1	NONE
10	MEASURE	2	NONE

DAG Representation

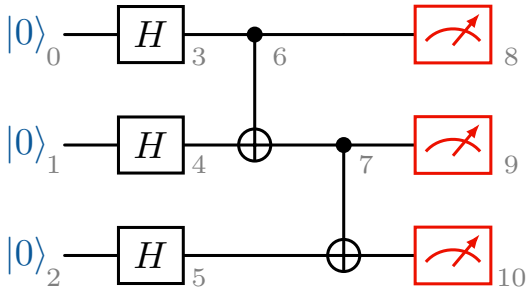


Simplified DAG



Compilation Procedures

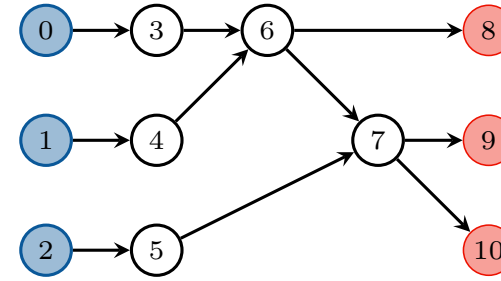
Static Circuit



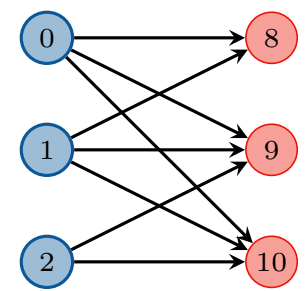
Static Circuit Instruction

ID	TYPE	QUBIT	PAR
0	RESET	0	NONE
1	RESET	1	NONE
2	RESET	2	NONE
3	H	0	NONE
4	H	1	NONE
5	H	2	NONE
6	CX	[0, 1]	NONE
7	CX	[1, 2]	NONE
8	MEASURE	0	NONE
9	MEASURE	1	NONE
10	MEASURE	2	NONE

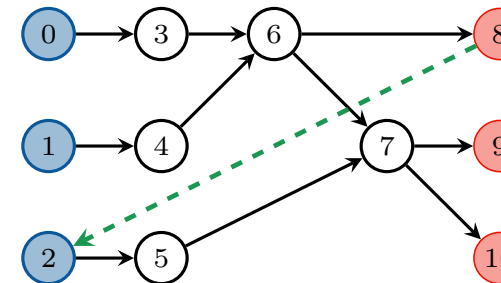
DAG Representation



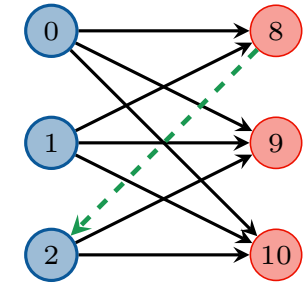
Simplified DAG



DAG with a new edge

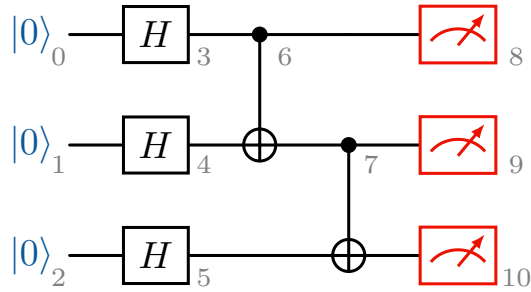


Add a new edge



Compilation Procedures

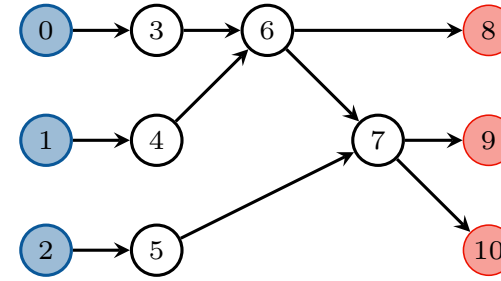
Static Circuit



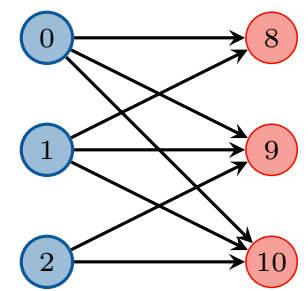
Static Circuit Instruction

Quantum Circuit Instructions			
ID	TYPE	QUBIT	PAR
0	RESET	0	NONE
1	RESET	1	NONE
2	RESET	2	NONE
3	H	0	NONE
4	H	1	NONE
5	H	2	NONE
6	CX	[0, 1]	NONE
7	CX	[1, 2]	NONE
8	MEASURE	0	NONE
9	MEASURE	1	NONE
10	MEASURE	2	NONE

DAG Representation

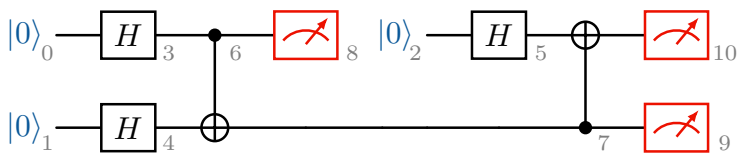


Simplified DAG



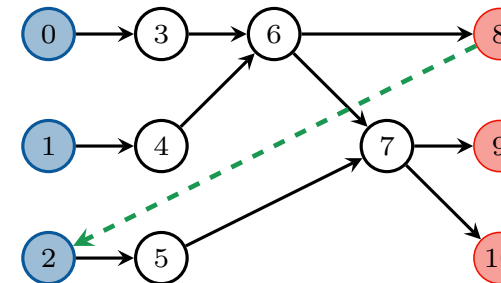
Dynamic Circuit Instruction

Dynamic Circuit

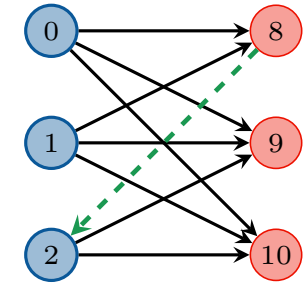


Quantum Circuit Instructions			
ID	TYPE	QUBIT	PAR
0	RESET	0	NONE
1	RESET	1	NONE
3	H	0	NONE
4	H	1	NONE
6	CX	[0, 1]	NONE
8	MEASURE	0	NONE
2	RESET	0	NONE
5	H	0	NONE
7	CX	[1, 0]	NONE
10	MEASURE	0	NONE
9	MEASURE	1	NONE

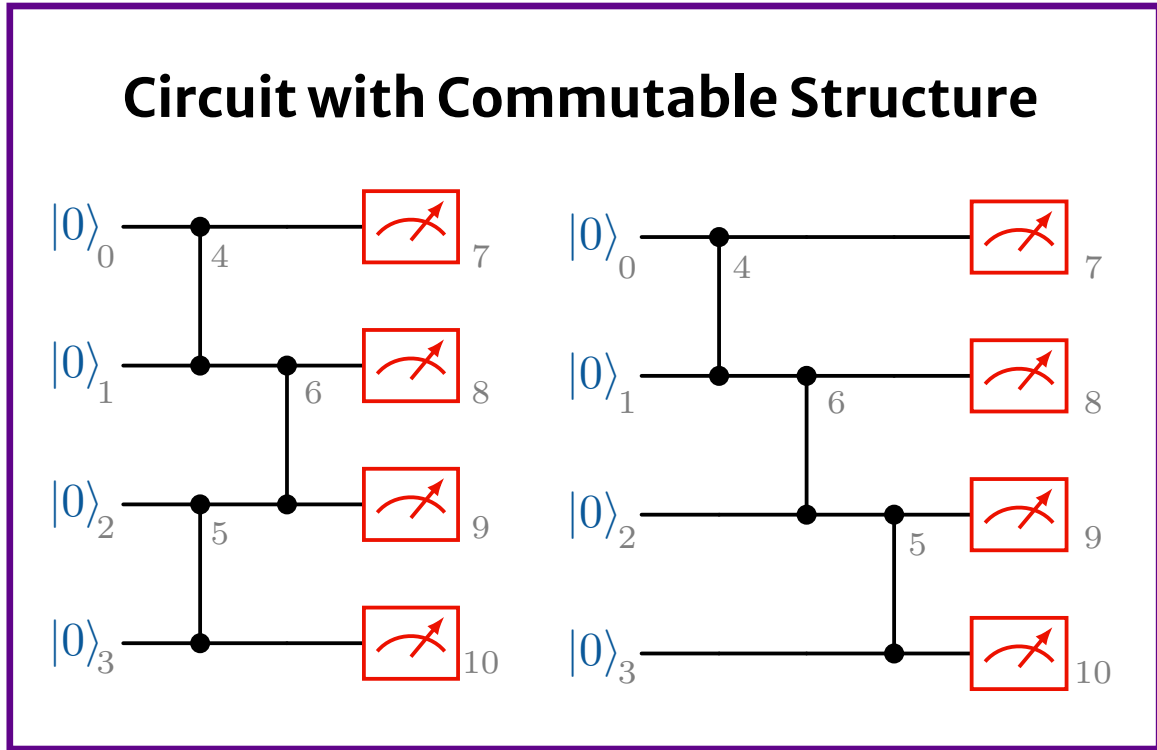
DAG with a new edge



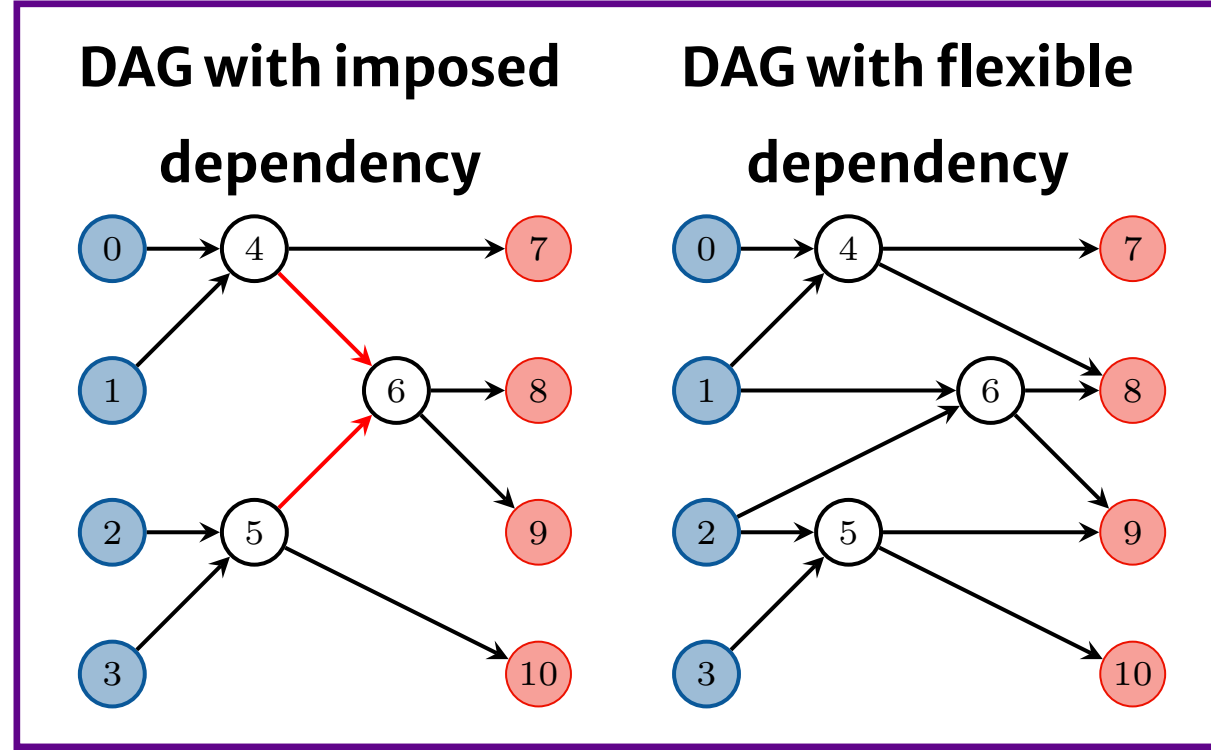
Add a new edge



Quantum Circuit with Commutable Structures

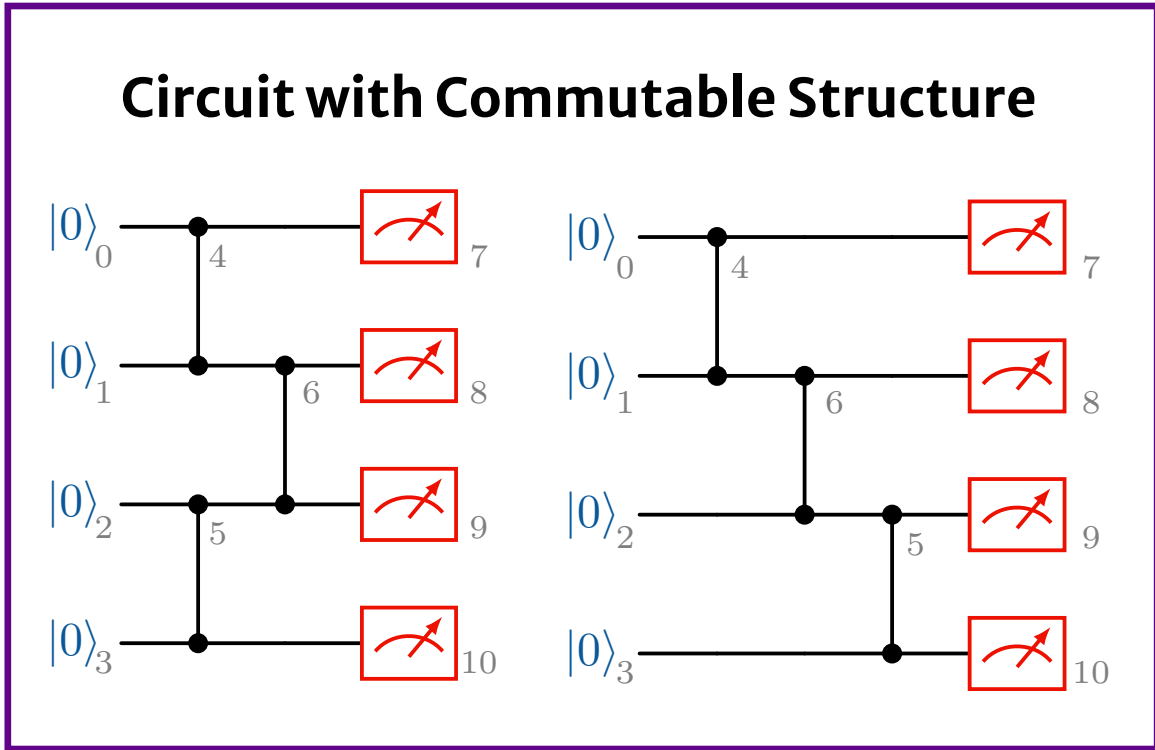


CZ gates are commutable

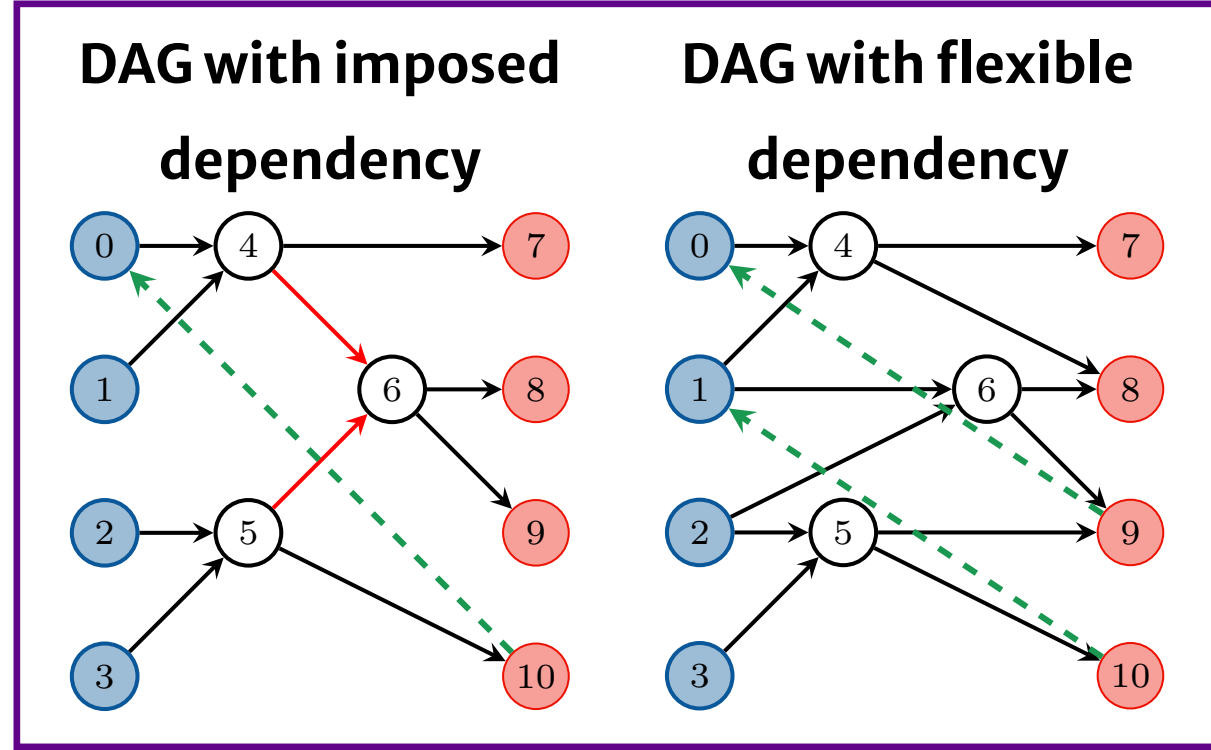


Seems more edges
but actually less restrictive on reachability

Quantum Circuit with Commutable Structures



CZ gates are commutable



More added edges !

Quantum Circuit with Commutable Structures

Commutable Structures are a **significant feature** in some **recent quantum applications**.

e.g. Quantum Approximate Optimization Algorithm (QAOA) Circuit

$$U(\vec{\gamma}, \vec{\beta}) = \prod_{i=1}^p U_B(\beta_i) U_C(\gamma_i)$$

Quantum Circuit with Commutable Structures

Commutable Structures are a **significant feature** in some recent quantum applications.

e.g. Quantum Approximate Optimization Algorithm (QAOA) Circuit

$$U(\vec{\gamma}, \vec{\beta}) = \prod_{i=1}^p U_B(\beta_i) \boxed{U_C(\gamma_i)}$$

ZZ gate are commutable

Quantum Circuit with Commutable Structures

Commutable Structures are a **significant feature** in some recent quantum applications.

e.g. Quantum Approximate Optimization Algorithm (QAOA) Circuit

$$U(\vec{\gamma}, \vec{\beta}) = \prod_{i=1}^p U_B(\beta_i) \boxed{U_C(\gamma_i)}$$

ZZ gate are commutable

e.g. Instantaneous Quantum Polynomial (IQP) Circuit

$$\mathcal{C} = H^{\otimes n} D H^{\otimes n}$$

Quantum Circuit with Commutable Structures

Commutable Structures are a **significant feature** in some recent quantum applications.

e.g. Quantum Approximate Optimization Algorithm (QAOA) Circuit

$$U(\vec{\gamma}, \vec{\beta}) = \prod_{i=1}^p U_B(\beta_i) \boxed{U_C(\gamma_i)}$$

ZZ gate are commutable

e.g. Instantaneous Quantum Polynomial (IQP) Circuit

$$C = H^{\otimes n} \boxed{D} H^{\otimes n}$$

D is a block of gates diagonal in the computational basis
e.g. randomly selecting gates from the set $\{\sqrt{CZ}, T\}$

How to determine if a quantum circuit
can be compiled
before the actual compilation?

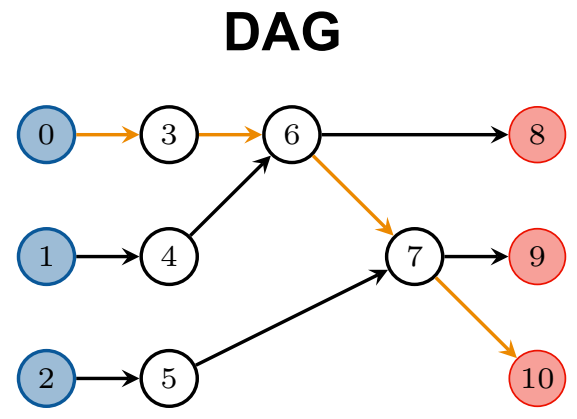
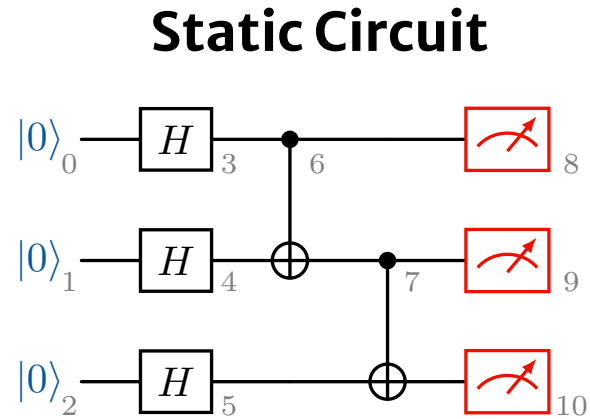
Determine the Reducibility of Quantum Circuits

How to determine if a quantum circuit can be compiled before the actual compilation?

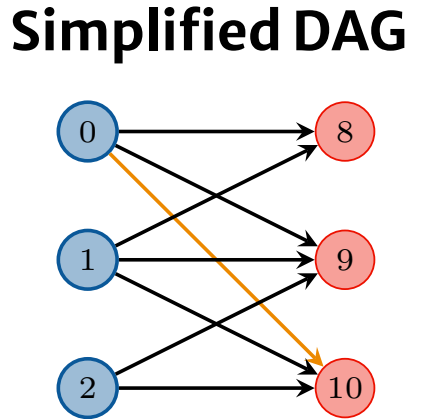
- A quantum circuit is **reducible** if it can be written as an equivalent quantum circuit with fewer qubits.
- Otherwise, it is called **irreducible**.

Determine the Reducibility of Quantum Circuits: Approach I

Approach 1: determine the reducibility from graph

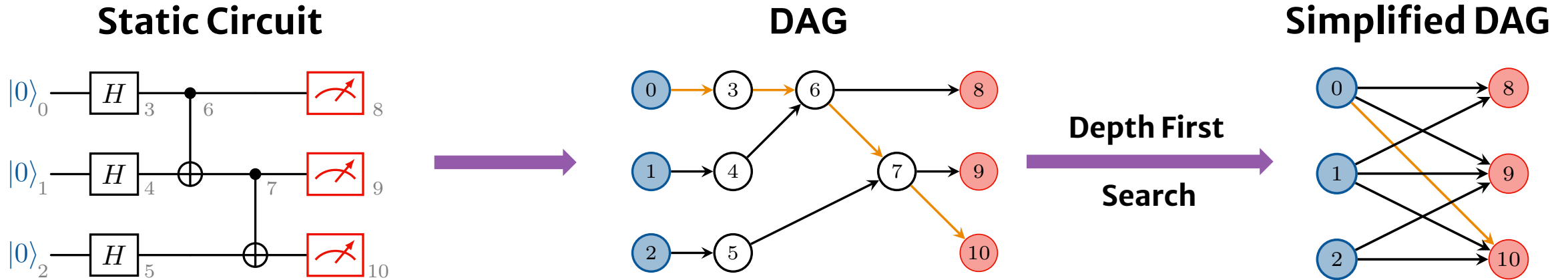


Depth First Search



Determine the Reducibility of Quantum Circuits: Approach I

Approach 1: determine the reducibility from graph



Proposition A static circuit is *irreducible* if and only if its simplified DAG is a *complete bipartite graph*.

Time Complexity: $O(m^2/n + mn)$

m is the number of instructions

n is the number of qubits

Determine the Reducibility of Quantum Circuits: Approach II

Approach 2: determine the reducibility from qubit reachability

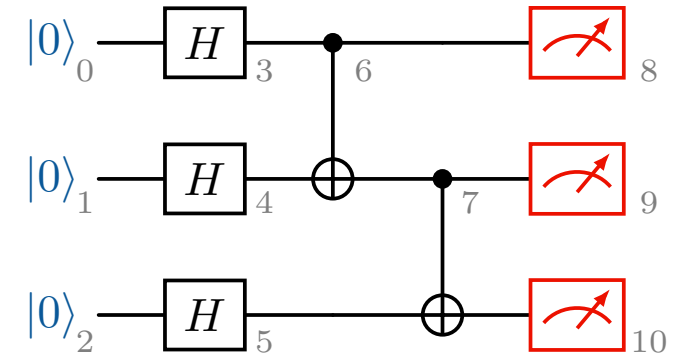
Definition Given an instruction list of an n -qubit quantum circuit acting on the set of qubits $Q = \{q_0, q_1, \dots, q_{n-1}\}$. Then any two-qubit gate introduces two qubit relations:

$$q_i \xrightarrow{k} q_j \quad \text{and} \quad q_j \xrightarrow{k} q_i$$

where q_i and q_j are the qubits upon which the gate operates and k is the order index of the instruction within the list. A qubit q_i reaches q_j (or, equivalently, q_j is reachable from q_i), denoted as $q_i \rightarrow q_j$, if there exists a sequence of qubit relations

$$q_i \xrightarrow{k_1} q_{l_1}, q_{l_1} \xrightarrow{k_2} q_{l_2}, \dots, q_{l_s} \xrightarrow{k_s} q_j$$

such that $k_1 \leq k_2 \leq \dots \leq k_{s+1}$. Moreover, qubit q_i and q_j are mutually reachable if q_i reaches q_j and vice versa.



$$\text{CNOT}[q_0, q_1] \quad \longrightarrow \quad q_0 \xrightarrow{6} q_1 \quad \text{and} \quad q_1 \xrightarrow{6} q_0$$

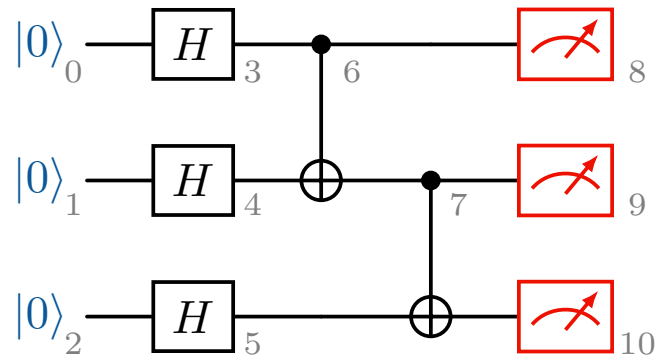
$$\text{CNOT}[q_1, q_2] \quad \longrightarrow \quad q_1 \xrightarrow{7} q_2 \quad \text{and} \quad q_2 \xrightarrow{7} q_1$$

Now we have $q_0 \rightarrow q_2$ through

$$q_0 \xrightarrow{6} q_1, q_1 \xrightarrow{7} q_2, \text{ but no } q_2 \rightarrow q_0$$

Determine the Reducibility of Quantum Circuits: Approach II

Approach 2: determine the reducibility from qubit reachability



CNOT $[q_0, q_1]$ \rightarrow $q_0 \xrightarrow{6} q_1$ and $q_1 \xrightarrow{6} q_0$

CNOT $[q_1, q_2]$ \rightarrow $q_1 \xrightarrow{7} q_2$ and $q_2 \xrightarrow{7} q_1$

$q_0 \rightarrow q_2$ through $q_0 \xrightarrow{6} q_1, q_1 \xrightarrow{7} q_2$, but no $q_2 \rightarrow q_0$

Proposition *A static quantum circuit is irreducible if and only if any two qubits within this quantum circuit are mutually reachable.*

Time Complexity: $O(m^2/n + mn)$

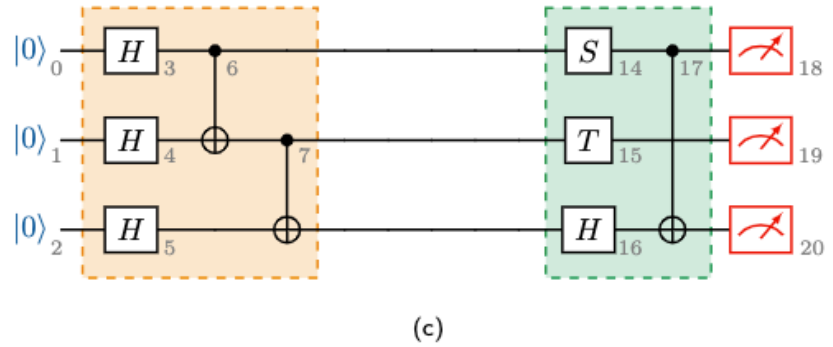
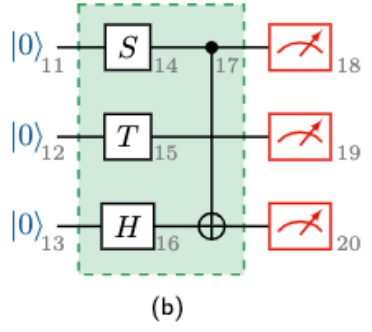
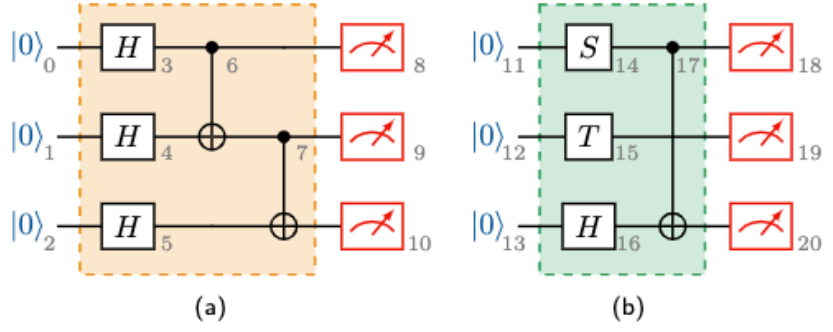
m is the number of instructions

n is the number of qubits

Determine the Reducibility of Quantum Circuits: Approach III

Approach 3: determine the reducibility from Boolean matrix multiplication

Circuit Composition $\mathcal{C} = \mathcal{C}_1 \circ \mathcal{C}_2$



Quantum Circuit Instructions

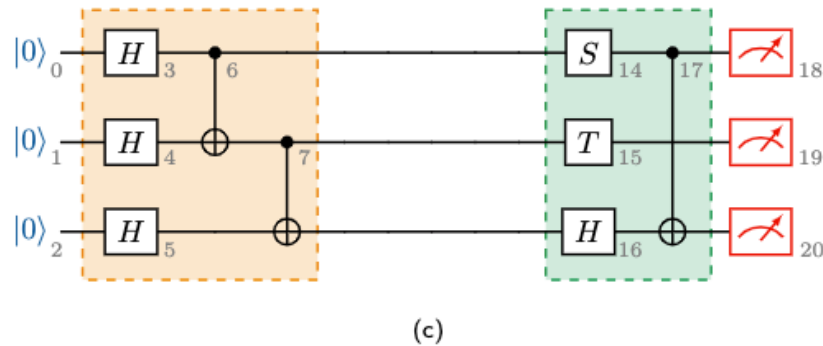
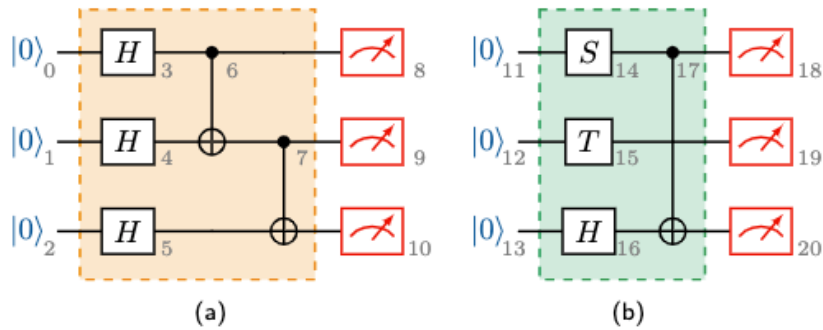
ID	TYPE	QUBIT	PAR
0	RESET	0	NONE
1	RESET	1	NONE
2	RESET	2	NONE
3	H	0	NONE
4	H	1	NONE
5	H	2	NONE
6	CX	[0, 1]	NONE
7	CX	[1, 2]	NONE
14	S	0	NONE
15	T	1	NONE
16	H	2	NONE
17	CX	[0, 2]	NONE
18	MEASURE	0	NONE
19	MEASURE	1	NONE
20	MEASURE	2	NONE

(d)

Determine the Reducibility of Quantum Circuits: Approach III

Approach 3: determine the reducibility from Boolean matrix multiplication

Circuit Composition $\mathcal{C} = \mathcal{C}_1 \circ \mathcal{C}_2$



Quantum Circuit Instructions

ID	TYPE	QUBIT	PAR
0	RESET	0	NONE
1	RESET	1	NONE
2	RESET	2	NONE
3	H	0	NONE
4	H	1	NONE
5	H	2	NONE
6	CX	[0, 1]	NONE
7	CX	[1, 2]	NONE
14	S	0	NONE
15	T	1	NONE
16	H	2	NONE
17	CX	[0, 2]	NONE
18	MEASURE	0	NONE
19	MEASURE	1	NONE
20	MEASURE	2	NONE

Proposition Biadjacency matrix of composite

circuit $B(\mathcal{C}_1 \circ \mathcal{C}_2) = B(\mathcal{C}_1) \odot B(\mathcal{C}_2)$, where

$$(A \odot B)_{ij} := \bigvee_{l=0}^{k-1} (A_{il} \wedge B_{lj}).$$

e.g.

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \odot \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Determine the Reducibility of Quantum Circuits: Approach III

Approach 3: determine the reducibility from Boolean matrix multiplication

- Consider an n -qubit circuit with only one two-qubit gate acting on q_i and q_j , its biadjacency matrix

$$B_{ij} = I_n + E_{ij}^n + E_{ji}^n$$

- **Note that any quantum circuit is a composition of elementary gates.** Then the biadjacency matrix of a circuit can be calculated as a **sequential Boolean product**

$$B(C) = B_1 \odot B_2 \odot \dots \odot B_m$$

Determine the Reducibility of Quantum Circuits: Approach III

Approach 3: determine the reducibility from Boolean matrix multiplication

- Consider an n -qubit circuit with only one two-qubit gate acting on q_i and q_j , its biadjacency matrix

$$B_{ij} = I_n + E_{ij}^n + E_{ji}^n$$

- Note that any quantum circuit is a composition of elementary gates. Then the biadjacency matrix of a circuit can be calculated as a **sequential Boolean product**

$$B(C) = B_1 \odot B_2 \odot \dots \odot B_m$$

Proposition *A static circuit is **irreducible** if and only if its biadjacency matrix is an **all-one matrix**.*

Time Complexity: $O(m^2/n + mn)$

m is the number of instructions

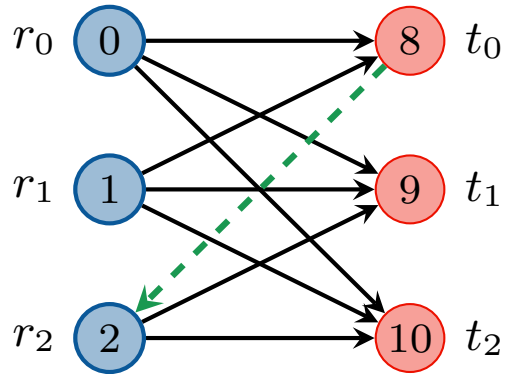
n is the number of qubits

Optimal Circuit Compilation

Optimal Circuit Compilation for Maximizing Qubit Reuse

Candidate Matrix

Simplified DAG



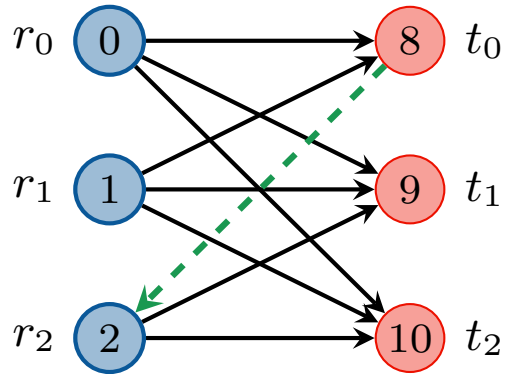
$$G_S(R, T, E_S)$$

- $(r_i, t_j) \notin E_S \rightarrow (t_j, r_i)$ is a candidate edge
- **Candidate Matrix \bar{B}** : $\bar{B}_{ij} = 1$ if (t_i, r_j) is a candidate edge
- $\bar{B} = \neg(B^T)$, B is the biadjacency matrix

Optimal Circuit Compilation for Maximizing Qubit Reuse

Candidate Matrix

Simplified DAG



$$G_S(R, T, E_S)$$

- $(r_i, t_j) \notin E_S \rightarrow (t_j, r_i)$ is a candidate edge
- **Candidate Matrix \bar{B}** : $\bar{B}_{ij} = 1$ if (t_i, r_j) is a candidate edge
- $\bar{B} = \neg(B^T)$, B is the biadjacency matrix

Biadjacency Matrix

$$B(G_S) = \begin{matrix} & t_0 & t_1 & t_2 \\ r_0 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_1 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_2 & \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

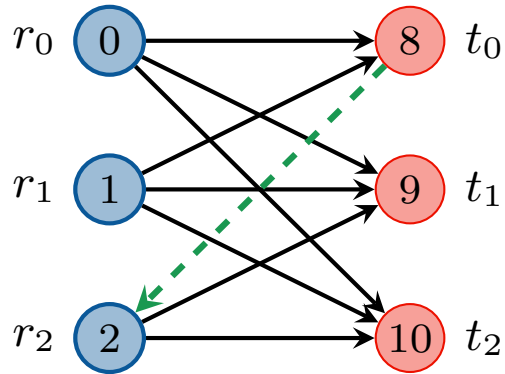
Candidate Matrix

$$\bar{B}(G_S) = \begin{matrix} & r_0 & r_1 & r_2 \\ t_0 & \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \\ t_1 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ t_2 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

Optimal Circuit Compilation for Maximizing Qubit Reuse

Candidate Matrix

Simplified DAG



$$G_S(R, T, E_S)$$

- $(r_i, t_j) \notin E_S \rightarrow (t_j, r_i)$ is a candidate edge
- **Candidate Matrix \bar{B}** : $\bar{B}_{ij} = 1$ if (t_i, r_j) is a candidate edge
- $\bar{B} = \neg(B^\top)$, B is the biadjacency matrix
- All added edges F : $F_{ij} = 1$ if (t_i, r_j) is added to G_S
- **All edges in F should be selected from \bar{B} , i.e. submatrix**

Biadjacency Matrix

$$B(G_S) = \begin{matrix} & t_0 & t_1 & t_2 \\ r_0 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_1 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_2 & \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

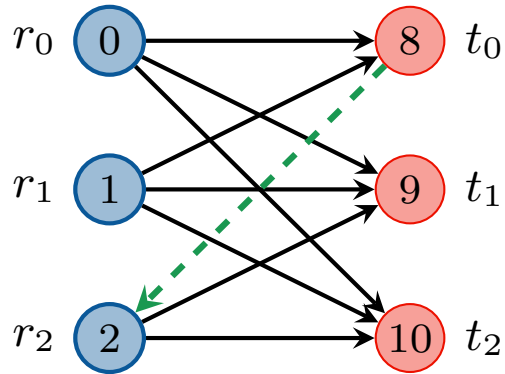
Candidate Matrix

$$\bar{B}(G_S) = \begin{matrix} & r_0 & r_1 & r_2 \\ t_0 & \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \\ t_1 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ t_2 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

Optimal Circuit Compilation for Maximizing Qubit Reuse

Candidate Matrix

Simplified DAG



$$G_S(R, T, E_S)$$

- $(r_i, t_j) \notin E_S \rightarrow (t_j, r_i)$ is a candidate edge
- **Candidate Matrix \bar{B}** : $\bar{B}_{ij} = 1$ if (t_i, r_j) is a candidate edge
- $\bar{B} = \neg(B^T)$, B is the biadjacency matrix
- All added edges F : $F_{ij} = 1$ if (t_i, r_j) is added to G_S
- **All edges in F should be selected from \bar{B} , i.e. submatrix**

Candidate Matrix

$$\bar{B}(G_S) = \begin{matrix} & r_0 & r_1 & r_2 \\ t_0 & \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \\ t_1 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ t_2 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

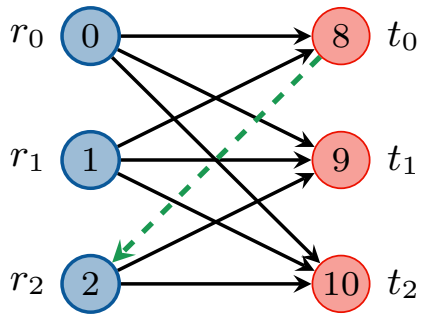
Added edges

$$F = \begin{matrix} & r_0 & r_1 & r_2 \\ t_0 & \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \\ t_1 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ t_2 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

Optimal Circuit Compilation for Maximizing Qubit Reuse

Optimal Quantum Circuit Compilation as a Binary Integer Programming Problem

Simplified DAG



Biadjacency Matrix

$$\begin{matrix} & t_0 & t_1 & t_2 \\ r_0 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_1 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_2 & \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

Candidate Matrix

$$\begin{matrix} & r_0 & r_1 & r_2 \\ t_0 & \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \\ t_1 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ t_2 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

Graph Manipulation

- $\forall (t, r) \in E'$, it has $t \in T$ and $r \in R$;
- $\forall (t, r) \in E'$, it has $\text{outdegree}(t) = 1$ and $\text{indegree}(r) = 1$;
- $G' = (R \cup T, E \cup E')$ is an acyclic graph.

Exact Characterization

$$\alpha := \max \sum_{i,j=0}^{n-1} F_{ij} \quad \text{Maximized added edges}$$

$$\text{s. t. } F_{ij} \leq \bar{B}_{ij}, \forall i, j \in \{0, 1, \dots, n-1\}$$

$$\sum_{j=0}^{n-1} F_{ij} \leq 1, \forall i \in \{0, 1, \dots, n-1\}$$

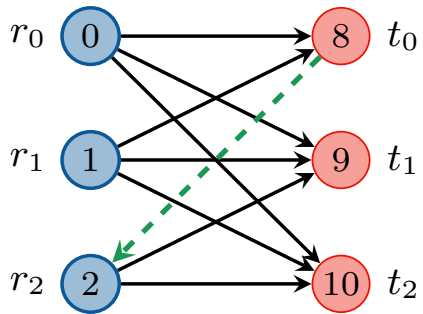
$$\sum_{i=0}^{n-1} F_{ij} \leq 1, \forall j \in \{0, 1, \dots, n-1\}$$

$$\begin{pmatrix} O_n & B \\ F & O_n \end{pmatrix} \text{ nilpotent}$$

Optimal Circuit Compilation for Maximizing Qubit Reuse

Optimal Quantum Circuit Compilation as a Binary Integer Programming Problem

Simplified DAG



Biadjacency Matrix

$$\begin{matrix} & t_0 & t_1 & t_2 \\ r_0 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_1 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_2 & \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

Candidate Matrix

$$\begin{matrix} & r_0 & r_1 & r_2 \\ t_0 & \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \\ t_1 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ t_2 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

Exact Characterization

$$\alpha := \max \sum_{i,j=0}^{n-1} F_{ij} \quad \text{Maximized added edges}$$

$$\text{s. t. } F_{ij} \leq \bar{B}_{ij}, \forall i, j \in \{0, 1, \dots, n-1\}$$

$$\sum_{j=0}^{n-1} F_{ij} \leq 1, \forall i \in \{0, 1, \dots, n-1\}$$

$$\sum_{i=0}^{n-1} F_{ij} \leq 1, \forall j \in \{0, 1, \dots, n-1\}$$

$$\begin{pmatrix} O_n & B \\ F & O_n \end{pmatrix} \quad \text{nilpotent}$$

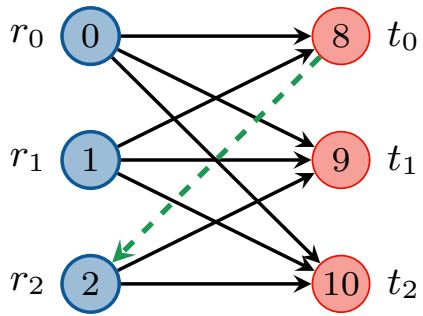
Graph Manipulation

- $\forall (t, r) \in E',$ it has $t \in T$ and $r \in R$;
- $\forall (t, r) \in E',$ it has $\text{outdegree}(t) = 1$ and $\text{indegree}(r) = 1$;
- $G' = (R \cup T, E \cup E')$ is an acyclic graph.

Optimal Circuit Compilation for Maximizing Qubit Reuse

Optimal Quantum Circuit Compilation as a Binary Integer Programming Problem

Simplified DAG



Biadjacency Matrix

$$\begin{matrix} & t_0 & t_1 & t_2 \\ r_0 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_1 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_2 & \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

Candidate Matrix

$$\begin{matrix} & r_0 & r_1 & r_2 \\ t_0 & \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \\ t_1 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ t_2 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

Exact Characterization

$$\alpha := \max \sum_{i,j=0}^{n-1} F_{ij} \quad \text{Maximized added edges}$$

$$\text{s. t. } F_{ij} \leq \bar{B}_{ij}, \forall i, j \in \{0, 1, \dots, n-1\}$$

$$\left\{ \begin{array}{l} \sum_{j=0}^{n-1} F_{ij} \leq 1, \forall i \in \{0, 1, \dots, n-1\} \\ \sum_{i=0}^{n-1} F_{ij} \leq 1, \forall j \in \{0, 1, \dots, n-1\} \end{array} \right.$$

$$\left\{ \begin{array}{l} \sum_{i=0}^{n-1} F_{ij} \leq 1, \forall j \in \{0, 1, \dots, n-1\} \end{array} \right.$$

$$\begin{pmatrix} O_n & B \\ F & O_n \end{pmatrix} \quad \text{nilpotent}$$

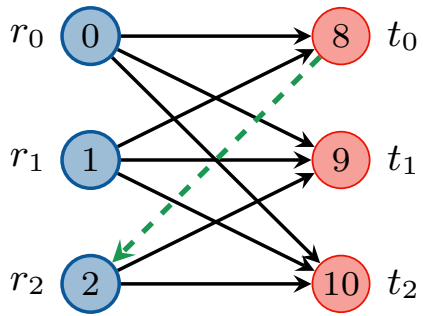
Graph Manipulation

- $\forall (t, r) \in E',$ it has $t \in T$ and $r \in R$;
- $\forall (t, r) \in E',$ it has $\text{outdegree}(t) = 1$ and $\text{indegree}(r) = 1$;
- $G' = (R \cup T, E \cup E')$ is an acyclic graph.

Optimal Circuit Compilation for Maximizing Qubit Reuse

Optimal Quantum Circuit Compilation as a Binary Integer Programming Problem

Simplified DAG



Biadjacency Matrix

$$\begin{matrix} & t_0 & t_1 & t_2 \\ r_0 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_1 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_2 & \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

Candidate Matrix

$$\begin{matrix} & r_0 & r_1 & r_2 \\ t_0 & \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \\ t_1 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ t_2 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

Exact Characterization

$$\alpha := \max \sum_{i,j=0}^{n-1} F_{ij} \quad \text{Maximized added edges}$$

$$\text{s. t. } F_{ij} \leq \bar{B}_{ij}, \forall i, j \in \{0, 1, \dots, n-1\}$$

$$\begin{cases} \sum_{j=0}^{n-1} F_{ij} \leq 1, \forall i \in \{0, 1, \dots, n-1\} \\ \sum_{i=0}^{n-1} F_{ij} \leq 1, \forall j \in \{0, 1, \dots, n-1\} \end{cases}$$

$$\begin{pmatrix} O_n & B \\ F & O_n \end{pmatrix} \quad \text{nilpotent}$$

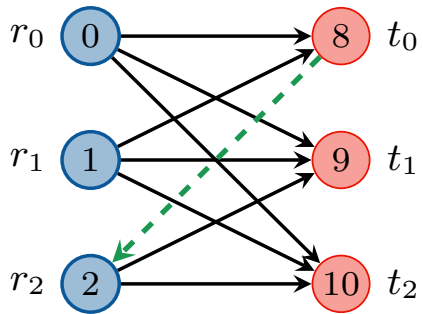
Graph Manipulation

- $\forall (t, r) \in E',$ it has $t \in T$ and $r \in R$;
- $\forall (t, r) \in E',$ it has $\text{outdegree}(t) = 1$ and $\text{indegree}(r) = 1$;
- $G' = (R \cup T, E \cup E')$ is an acyclic graph.

Optimal Circuit Compilation for Maximizing Qubit Reuse

Optimal Quantum Circuit Compilation as a Binary Integer Programming Problem

Simplified DAG



Biadjacency Matrix

$$\begin{matrix} & t_0 & t_1 & t_2 \\ r_0 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_1 & \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\ r_2 & \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

Candidate Matrix

$$\begin{matrix} & r_0 & r_1 & r_2 \\ t_0 & \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \\ t_1 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ t_2 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

Graph Manipulation

- $\forall (t, r) \in E', it has t \in T and r \in R;$
- $\forall (t, r) \in E', it has outdegree(t) = 1 and indegree(r) = 1;$
- $G' = (R \cup T, E \cup E')$ is an acyclic graph.

Exact Characterization

$$\alpha := \max \sum_{i,j=0}^{n-1} F_{ij} \quad \text{NP-hard [*]}$$

$$\text{s. t. } F_{ij} \leq \bar{B}_{ij}, \forall i, j \in \{0, 1, \dots, n-1\}$$

$$\begin{cases} \sum_{j=0}^{n-1} F_{ij} \leq 1, \forall i \in \{0, 1, \dots, n-1\} \\ \sum_{i=0}^{n-1} F_{ij} \leq 1, \forall j \in \{0, 1, \dots, n-1\} \end{cases}$$

$$\begin{pmatrix} O_n & B \\ F & O_n \end{pmatrix} \text{ nilpotent}$$

[1] Hanru Jiang. 2024. Qubit Recycling Revisited. Proc. ACM Program. Lang. 8, PLDI, Article 198 (June 2024), 24 pages.

[*] The author in [1] proved that an equivalent optimization problem is NP-hard.

Optimal Circuit Compilation for Maximizing Qubit Reuse

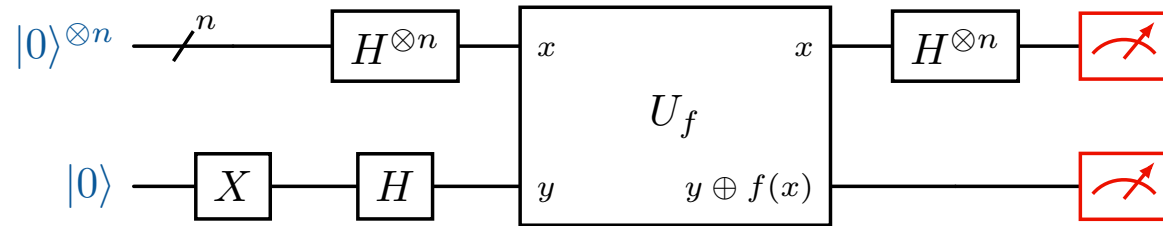
Summary of Explored Circuits

Quantum circuits	Original	Compiled
Deutsch-Jozsa algorithm	n	2 (1)
Bernstein-Vazirani algorithm	n	2 (1)
Simon's algorithm	$2n$	3
Quantum Fourier transform	n	n
Quantum phase estimation	n	n
Shor's algorithm	n	n
Grover's algorithm	n	n
Quantum counting algorithm	n	n
Linearly entangled circuit with l layers	n	$l + 1$
Circularly entangled circuit with l layers	n	3
Pairwisely entangled circuit with l layers	n	$2l + 1$
Fully entangled circuit	n	n
Diamond-structured quantum circuit	$2n$	$n + 1$
MBQC with cluster state of size (w, d)	wd	$w + 1$
MBQC with brickwork state of size (w, d)	wd	$w + 1$
Quantum ripple carry adder circuit	n (4)	4 (3)

Cases in red indicates the compiled width is **optimal** and cases in blue indicates **irreducible circuit**.

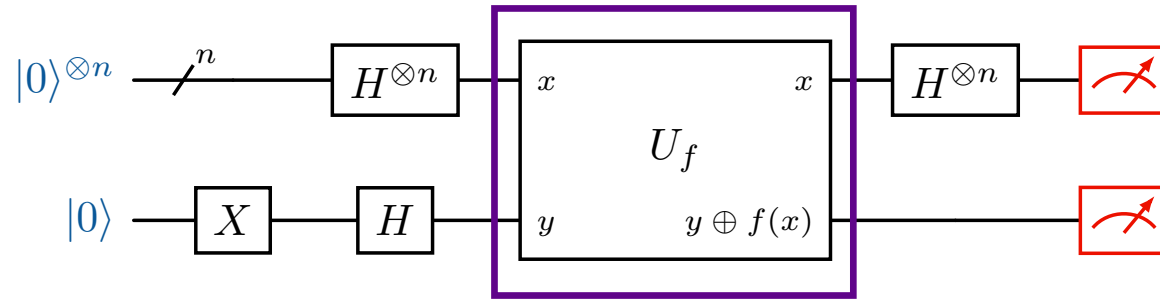
Optimal Circuit Compilation for Maximizing Qubit Reuse

e.g. Deutsch-Jozsa algorithm

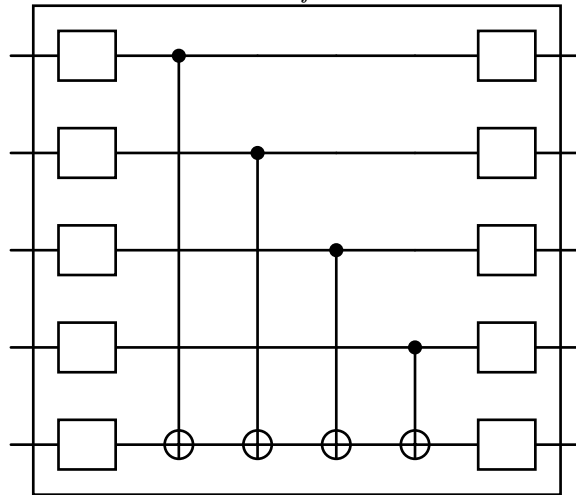


Optimal Circuit Compilation for Maximizing Qubit Reuse

e.g. Deutsch-Jozsa algorithm



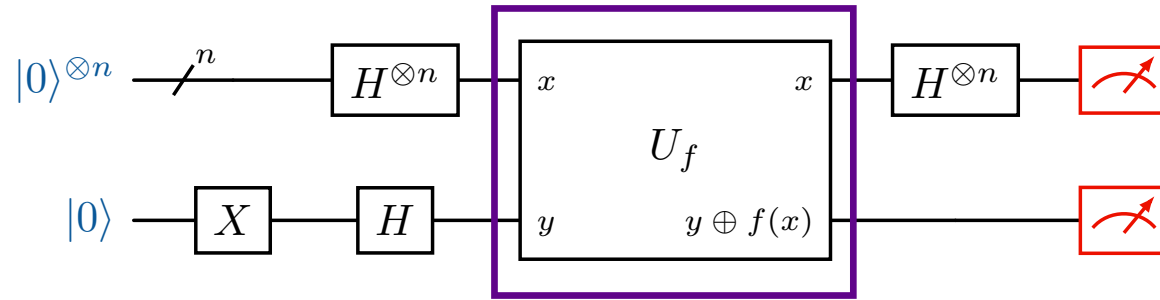
U_f



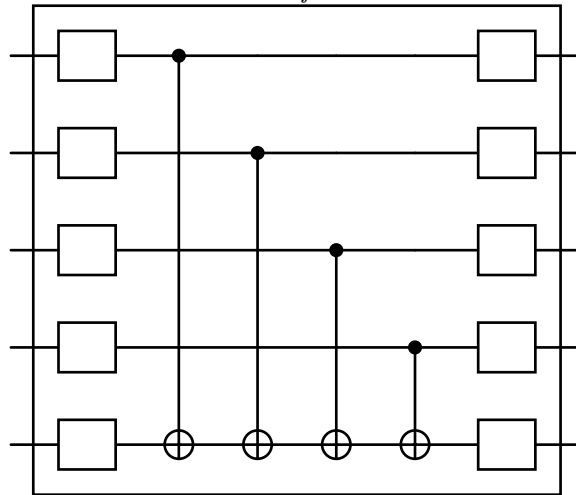
**Oracle
Implementation**

Optimal Circuit Compilation for Maximizing Qubit Reuse

e.g. Deutsch-Jozsa algorithm



U_f



**Oracle
Implementation**



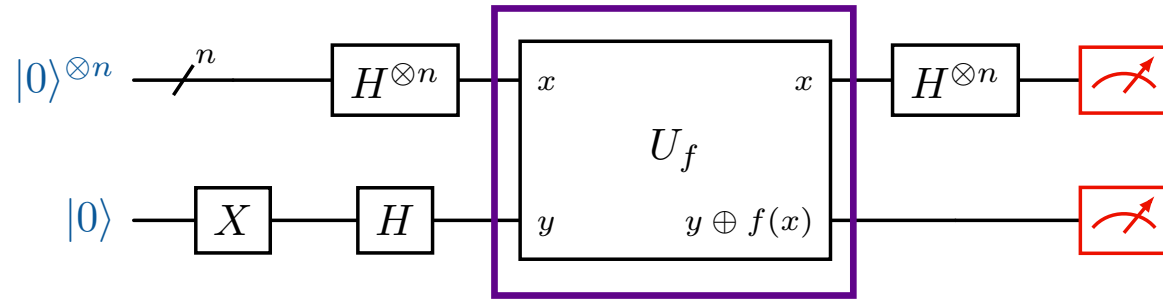
Candidate Matrix

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

$\begin{pmatrix} O_n & B \\ F & O_n \end{pmatrix}$ nilpotent

Optimal Circuit Compilation for Maximizing Qubit Reuse

e.g. Deutsch-Jozsa algorithm

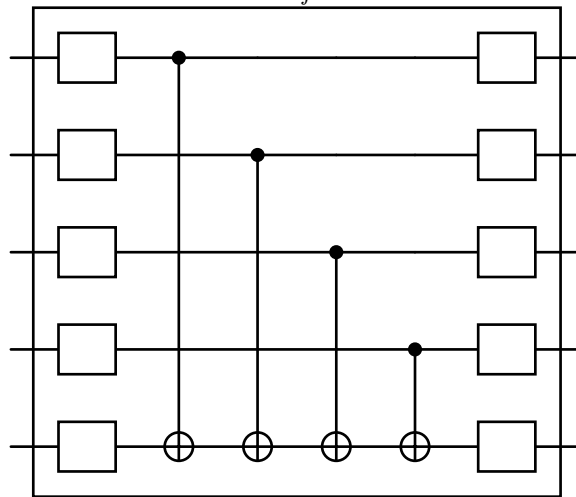


Optimal Solution

$$F = \sum_{i=0}^{n-2} E_{i,i+1}^{n+1} \quad n_{opt} = 2$$



Oracle Implementation



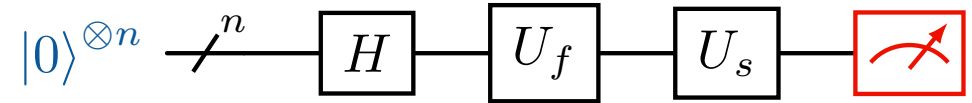
Candidate Matrix

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

$$\begin{pmatrix} O_n & B \\ F & O_n \end{pmatrix} \text{ nilpotent}$$

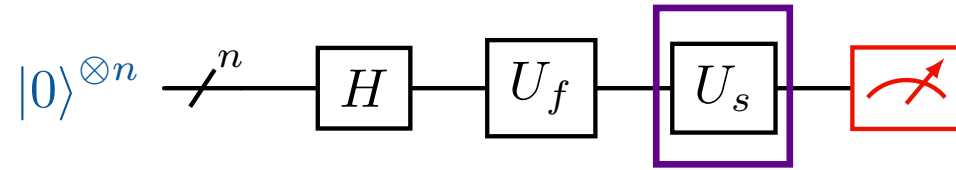
Optimal Circuit Compilation for Maximizing Qubit Reuse

e.g. Grover's algorithm

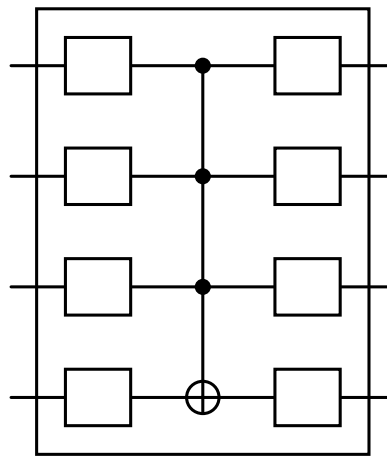


Optimal Circuit Compilation for Maximizing Qubit Reuse

e.g. Grover's algorithm



U_s



**Diffusion Operator
Implementation**

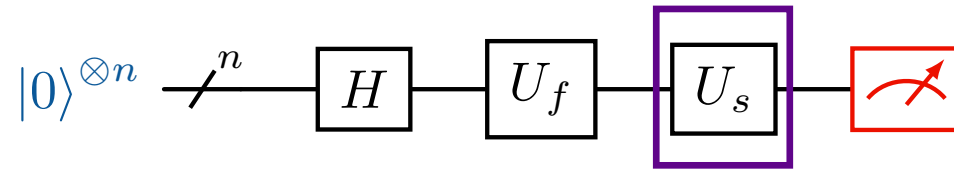
n-qubit gates !



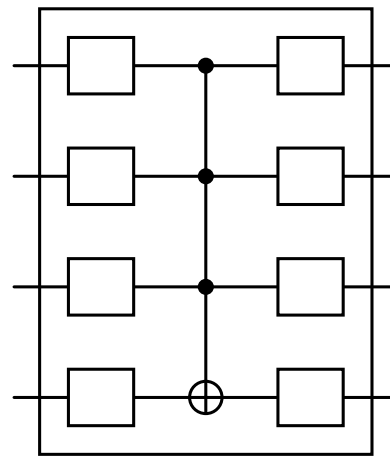
Irreducible Circuit

Optimal Circuit Compilation for Maximizing Qubit Reuse

e.g. Grover's algorithm



**Diffusion Operator
Implementation**



n-qubit gates !

Irreducible Circuit

- Note that the compilation depends on the specific circuit **implementation**.
- It is **possible** that there exist another circuit decomposition for Grover that is reducible.

Heuristic Algorithms

How to add as many edges as possible?

Heuristic Algorithm I: Minimum Remaining Values Algorithm

Algorithm 1: Minimum Remaining Values (MRV) Algorithm

MRV Heuristic: Designate the variable with **the fewest valid values** (i.e., the minimum remaining values) as the next one for value assignment

e.g. Sudoku Solver select the empty cell with the fewest potential values for the next value assignment

Heuristic Algorithm I: Minimum Remaining Values Algorithm

Algorithm 1: Minimum Remaining Values (MRV) Algorithm

MRV Heuristic: Designate the variable with **the fewest valid values** (i.e., the minimum remaining values) as the next one for value assignment

e.g. Sudoku Solver select the empty cell with the fewest potential values for the next value assignment

Procedure

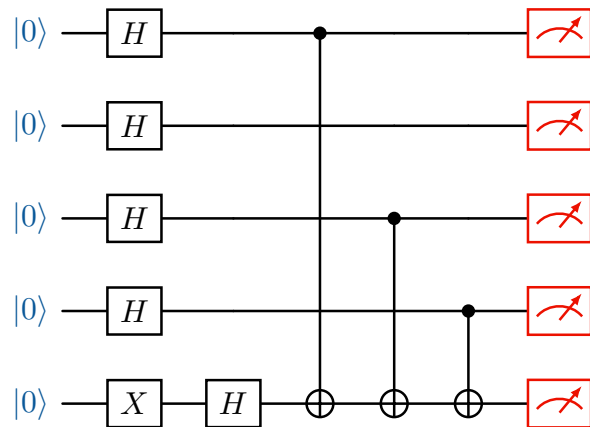
- Identify the terminal t_i with **the fewest candidate roots (smallest row sum)**
- Connect t_i with a candidate root r_j **with the least number of choices of terminals**
- Update candidate matrix and proceed to next iteration

Heuristic Algorithm I: Minimum Remaining Values Algorithm

Algorithm 1: Minimum Remaining Values (MRV) Algorithm

- Identify the terminal t_i with **the fewest candidate roots (smallest row sum)**
 - Connect t_i with a candidate root r_j with **the least number of choices of terminals (smallest column sum)**
 - Update candidate matrix and proceed to next iteration
-

e.g. 5-qubit Bernstein-Vazirani

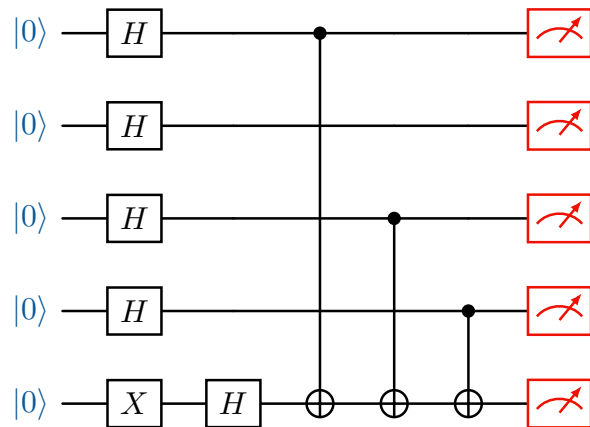


Heuristic Algorithm I: Minimum Remaining Values Algorithm

Algorithm 1: Minimum Remaining Values (MRV) Algorithm

- Identify the terminal t_i with the **fewest candidate roots (smallest row sum)**
 - Connect t_i with a candidate root r_j with the **least number of choices of terminals (smallest column sum)**
 - Update candidate matrix and proceed to next iteration
-

e.g. 5-qubit Bernstein-Vazirani



Candidate Matrix

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

Heuristic Algorithm I: Minimum Remaining Values Algorithm

Algorithm 1: Minimum Remaining Values (MRV) Algorithm

- Identify the terminal t_i with the fewest candidate roots (smallest row sum)

e.g. 5-qubit Bernstein-Vazirani

Identify an Edge

	r_0	r_1	r_2	r_3	r_4
t_0	0	1	1	1	0
t_1	1	0	1	1	1
t_2	0	1	0	1	0
t_3	0	1	0	0	0
t_4	0	1	0	0	0

Heuristic Algorithm I: Minimum Remaining Values Algorithm

Algorithm 1: Minimum Remaining Values (MRV) Algorithm

- Identify the terminal t_i with the fewest candidate roots (smallest row sum)
- Connect t_i with a candidate root r_j with the least number of choices of terminals (smallest column sum)

e.g. 5-qubit Bernstein-Vazirani

Identify an Edge

	r_0	r_1	r_2	r_3	r_4
t_0	0	1	1	1	0
t_1	1	0	1	1	1
t_2	0	1	0	1	0
t_3	0	1	0	0	0
t_4	0	1	0	0	0

	r_0	r_1	r_2	r_3	r_4
t_0	0	1	1	1	0
t_1	1	0	1	1	1
t_2	0	1	0	1	0
t_3	0	1	0	0	0
t_4	0	1	0	0	0

Heuristic Algorithm I: Minimum Remaining Values Algorithm

Algorithm 1: Minimum Remaining Values (MRV) Algorithm

- Identify the terminal t_i with the fewest candidate roots (smallest row sum)
- Connect t_i with a candidate root r_j with the least number of choices of terminals (smallest column sum)
- Update candidate matrix and proceed to next iteration

e.g. 5-qubit Bernstein-Vazirani

Identify an Edge

	r_0	r_1	r_2	r_3	r_4
t_0	0	1	1	1	0
t_1	1	0	1	1	1
t_2	0	1	0	1	0
t_3	0	1	0	0	0
t_4	0	1	0	0	0

	r_0	r_1	r_2	r_3	r_4
t_0	0	1	1	1	0
t_1	1	0	1	1	1
t_2	0	1	0	1	0
t_3	0	1	0	0	0
t_4	0	1	0	0	0

Update graph and biadjacency matrix

Update Candidate Matrix

0	1	1	1	0
0	0	0	0	0
0	1	0	1	0
0	1	0	0	0
0	1	0	0	0

0	0	1	1	0
0	0	0	0	0
0	0	0	1	0
0	0	0	0	0
0	0	0	0	0

Heuristic Algorithm I: Minimum Remaining Values Algorithm

Algorithm 1: Minimum Remaining Values (MRV) Algorithm

Worst-case Time Complexity $O(mn + n^3)$

e.g. 5-qubit Bernstein-Vazirani

Identify an Edge

	r_0	r_1	r_2	r_3	r_4
t_0	0	1	1	1	0
t_1	1	0	1	1	1
t_2	0	1	0	1	0
t_3	0	1	0	0	0
t_4	0	1	0	0	0

	r_0	r_1	r_2	r_3	r_4
t_0	0	1	1	1	0
t_1	1	0	1	1	1
t_2	0	1	0	1	0
t_3	0	1	0	0	0
t_4	0	1	0	0	0

Update graph and biadjacency matrix

Update Candidate Matrix

0	1	1	1	0
0	0	0	0	0
0	1	0	1	0
0	1	0	0	0
0	1	0	0	0

0	0	1	1	0
0	0	0	0	0
0	0	0	1	0
0	0	0	0	0
0	0	0	0	0

Heuristic Algorithm II: Greedy Algorithms*

Algorithm 2: Greedy Algorithm

Greedy Algorithm

Makes a **locally optimal** choice at each step

Heuristic Algorithm II: Greedy Algorithms*

Algorithm 2: Greedy Algorithm

Greedy Algorithm

Makes a **locally optimal** choice at each step



Maximize the possibility to add more edges in subsequent steps

Heuristic Algorithm II: Greedy Algorithms*

Algorithm 2: Greedy Algorithm

Greedy Algorithm

Makes a **locally optimal** choice at each step



Maximize the possibility to add more edges in subsequent steps

Procedure

- **Temporarily** integrates the edge into the simplified DAG and update the candidate matrix
- Score the candidate edge as the **summation** of all elements within the updated candidate matrix
- **Randomly** select a candidate edge with the **highest score**

Heuristic Algorithm II: Greedy Algorithms*

Algorithm 2: Greedy Algorithm

- **Temporarily** integrates the edge into the simplified DAG and update the candidate matrix

e.g. 5-qubit Bernstein-Vazirani

Edge under evaluation

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

Heuristic Algorithm II: Greedy Algorithms*

Algorithm 2: Greedy Algorithm

- **Temporarily** integrates the edge into the simplified DAG and update the candidate matrix

e.g. 5-qubit Bernstein-Vazirani

Edge under evaluation

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

Update candidate matrix

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

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

Heuristic Algorithm II: Greedy Algorithms*

Algorithm 2: Greedy Algorithm

- **Temporarily** integrates the edge into the simplified DAG and update the candidate matrix
- Score the candidate edge as the **summation** of all elements within the updated candidate matrix

e.g. 5-qubit Bernstein-Vazirani

Edge under evaluation

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

Update candidate matrix

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

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

Score matrix

$$\begin{pmatrix} 0 & 3 & 7 & 6 & 0 \\ 3 & 0 & 3 & 3 & 3 \\ 0 & 3 & 0 & 7 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \end{pmatrix}$$

Heuristic Algorithm II: Greedy Algorithms*

Algorithm 2: Greedy Algorithm

- **Temporarily** integrates the edge into the simplified DAG and update the candidate matrix
- Score the candidate edge as the **summation** of all elements within the updated candidate matrix
- **Randomly** select a candidate edge with the **highest score**

(randomness helps to avoid the dependence of the circuit relabeling, better performance in practice)

e.g. 5-qubit Bernstein-Vazirani

Edge under evaluation

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

Update candidate matrix

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

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

Score matrix

$$\begin{pmatrix} 0 & 3 & 7 & 6 & 0 \\ 3 & 0 & 3 & 3 & 3 \\ 0 & 3 & 0 & 7 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \end{pmatrix}$$

Heuristic Algorithm II: Greedy Algorithms*

Algorithm 2: Greedy Algorithm

Worst-case Time Complexity $O(mn + n^5)$

e.g. 5-qubit Bernstein-Vazirani

Edge under evaluation

0	1	1	1	0
1	0	1	1	1
0	1	0	1	0
0	1	0	0	0
0	1	0	0	0

Update candidate matrix

0	1	1	1	0
0	0	0	0	0
0	1	0	1	0
0	1	0	0	0
0	1	0	0	0

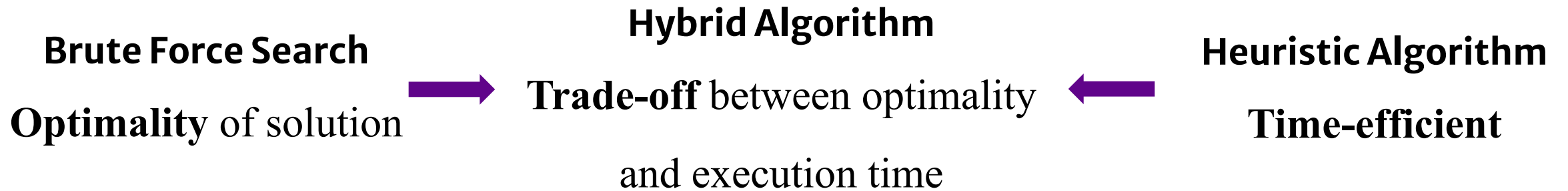
0	0	1	1	0
0	0	0	0	0
0	0	0	1	0
0	0	0	0	0
0	0	0	0	0

Score matrix

0	3	7	6	0
3	0	3	3	3
0	3	0	7	0
0	3	0	0	0
0	3	0	0	0

Heuristic Algorithm III: Hybrid Algorithm

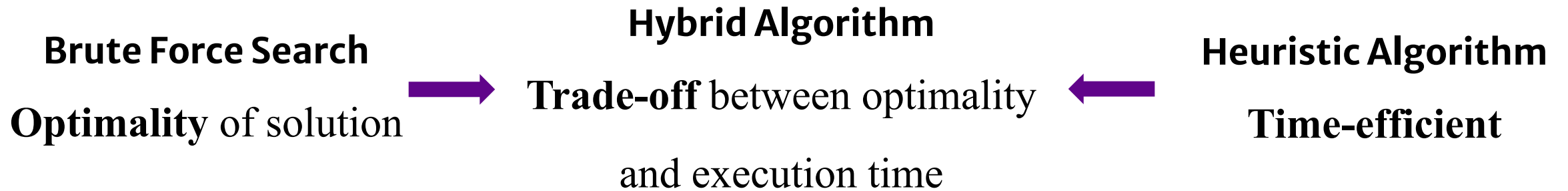
Algorithm 3: Hybrid Algorithm



- **Brute force search** on a designated subset of terminals $T_E \subseteq T$, which exhaustively enumerates all feasible edge additions. **Hierarchy level: $L = |T_E|$**
- For each feasible addition, employ **MRV heuristic (or other heuristics)** to identify edges can be added to the remaining terminals in $T \setminus T_E$.
- Select **the best solution** among all enumeration.

Heuristic Algorithm III: Hybrid Algorithm

Algorithm 3: Hybrid Algorithm

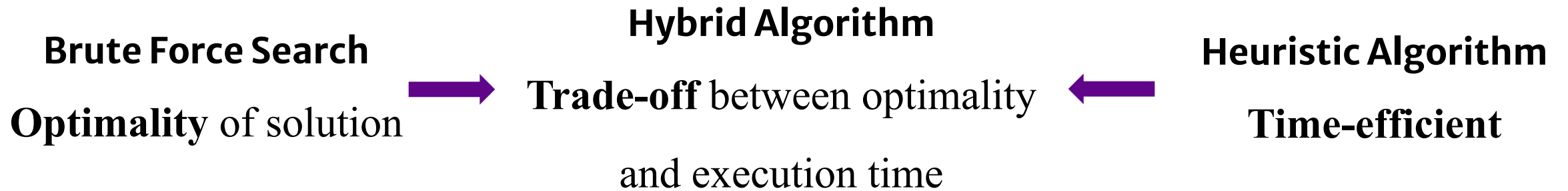


Hierarchy level: $L = |T_E|$



Heuristic Algorithm III: Hybrid Algorithm

Algorithm 3: Hybrid Algorithm



Hierarchy level: $L = |T_E|$



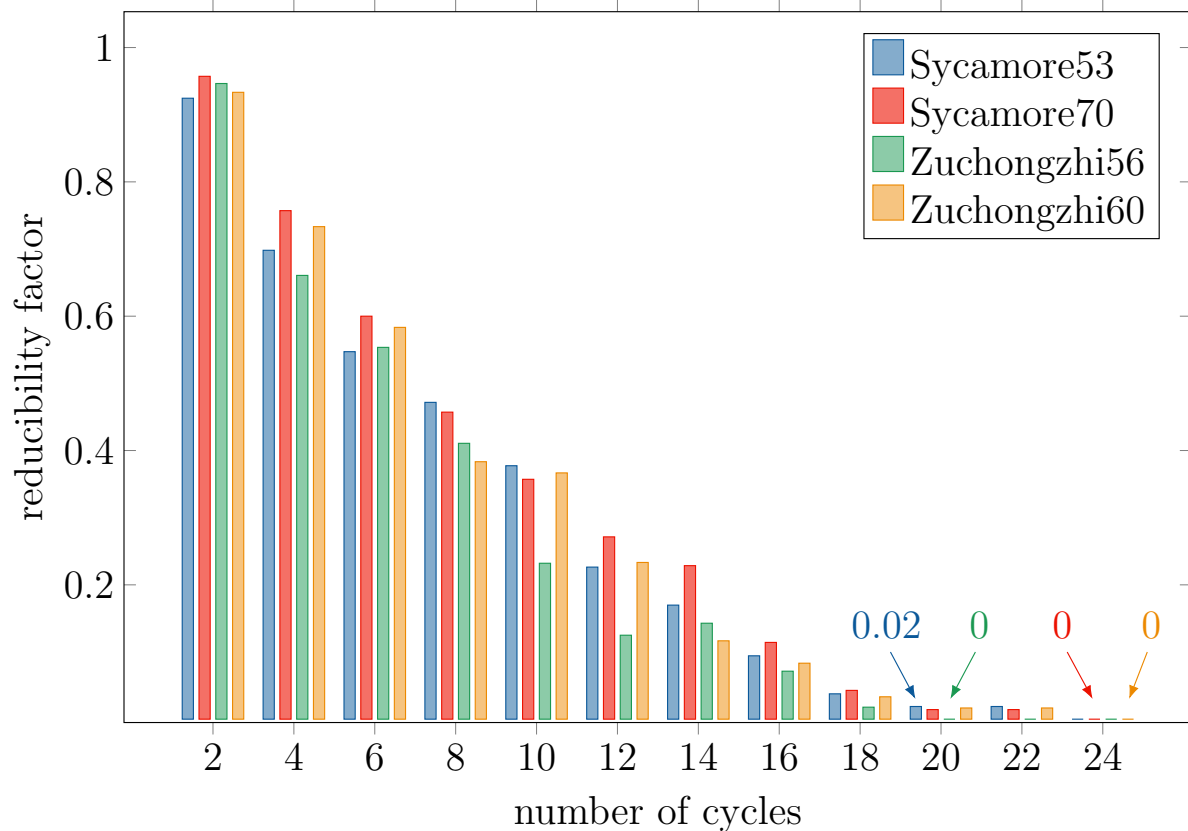
Worst-case Time Complexity $O(n^L m^2 (n - L)^2)$

Numerical Evaluation

Performance of the heuristic algorithms?

Numerical Evaluation: Quantum Supremacy Circuits

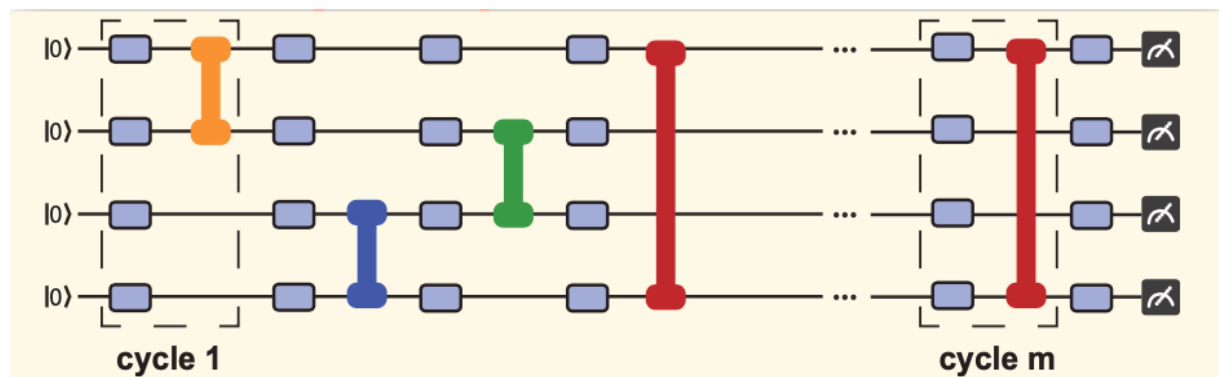
*Demonstrating Quantum Supremacy



$$\text{Reducibility Factor } r = 1 - \frac{\text{compiled width}}{\text{original width}} \in [0, 1)$$

The larger, the better

Random Circuit Generation



Alternating layers of single-qubit and two-qubit gates in random quantum circuits.

[1] Frank Arute et al. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 505-510 (2019).

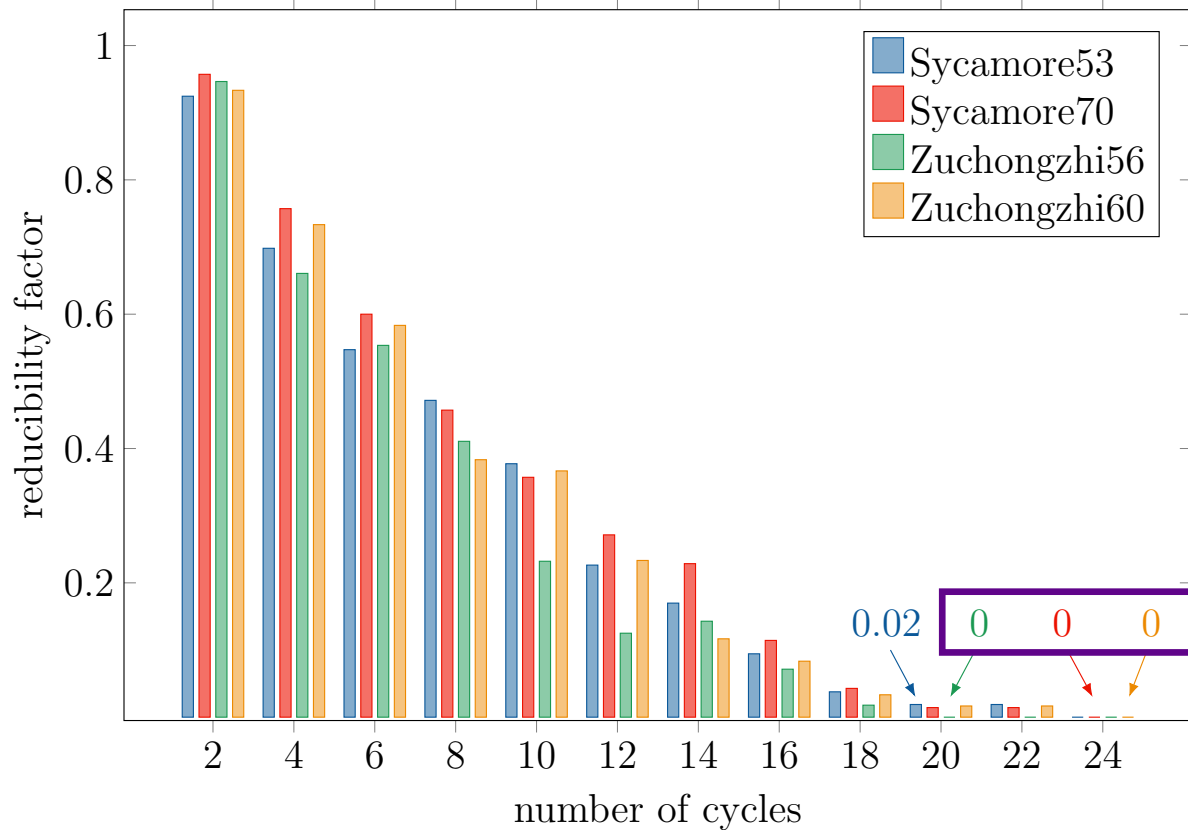
[2] Alexis Morvan et al. Phase transition in Random Circuit Sampling. arXiv:2304.11119 (2023).

[3] Yulin Wu et al. Strong quantum computational advantage using a superconducting quantum processor. *Phys. Rev. Lett.* 127, 180501 (2021).

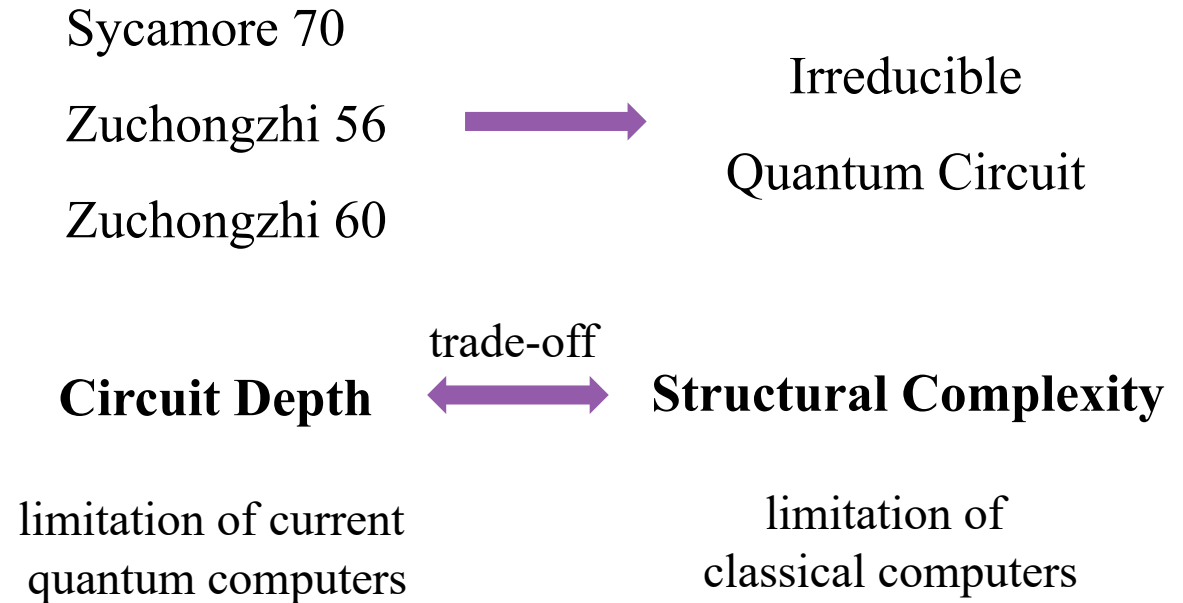
[4] Qingling Zhu et al. Quantum computational advantage via 60-qubit 24-cycle random circuit sampling. *Science Bulletin* 67, 240-245 (2022).

Numerical Evaluation: Quantum Supremacy Circuits

*Demonstrating Quantum Supremacy



Interesting Observation I



[1] Frank Arute et al. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 505-510 (2019).

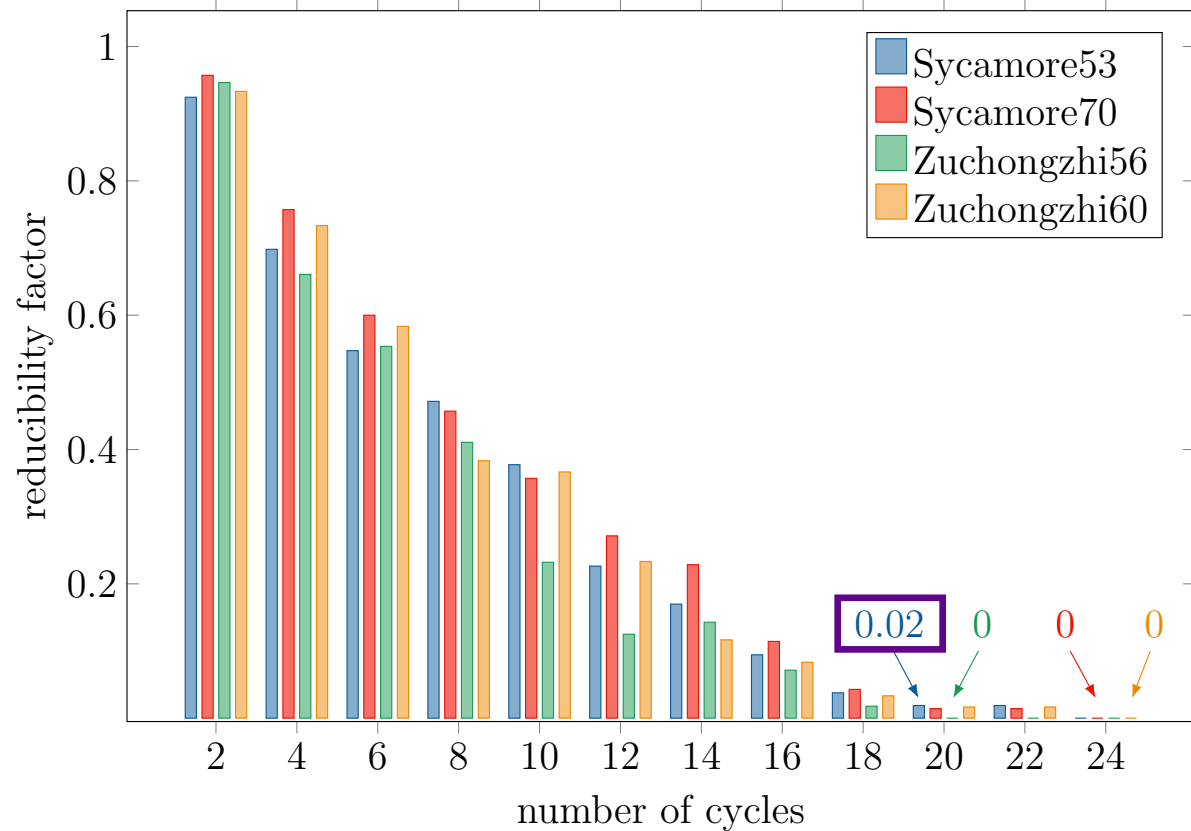
[2] Alexis Morvan et al. Phase transition in Random Circuit Sampling. arXiv:2304.11119 (2023).

[3] Yulin Wu et al. Strong quantum computational advantage using a superconducting quantum processor. *Phys. Rev. Lett.* 127, 180501 (2021).

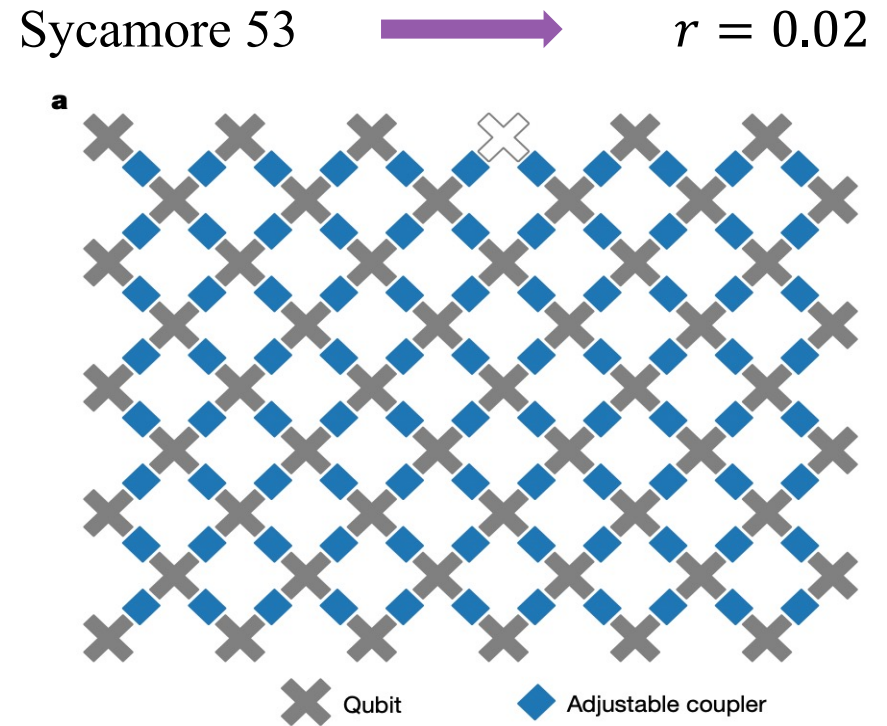
[4] Qingling Zhu et al. Quantum computational advantage via 60-qubit 24-cycle random circuit sampling. *Science Bulletin* 67, 240-245 (2022).

Numerical Evaluation: Quantum Supremacy Circuits

*Demonstrating Quantum Supremacy



Interesting Observation II



[1] Frank Arute et al. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 505-510 (2019).

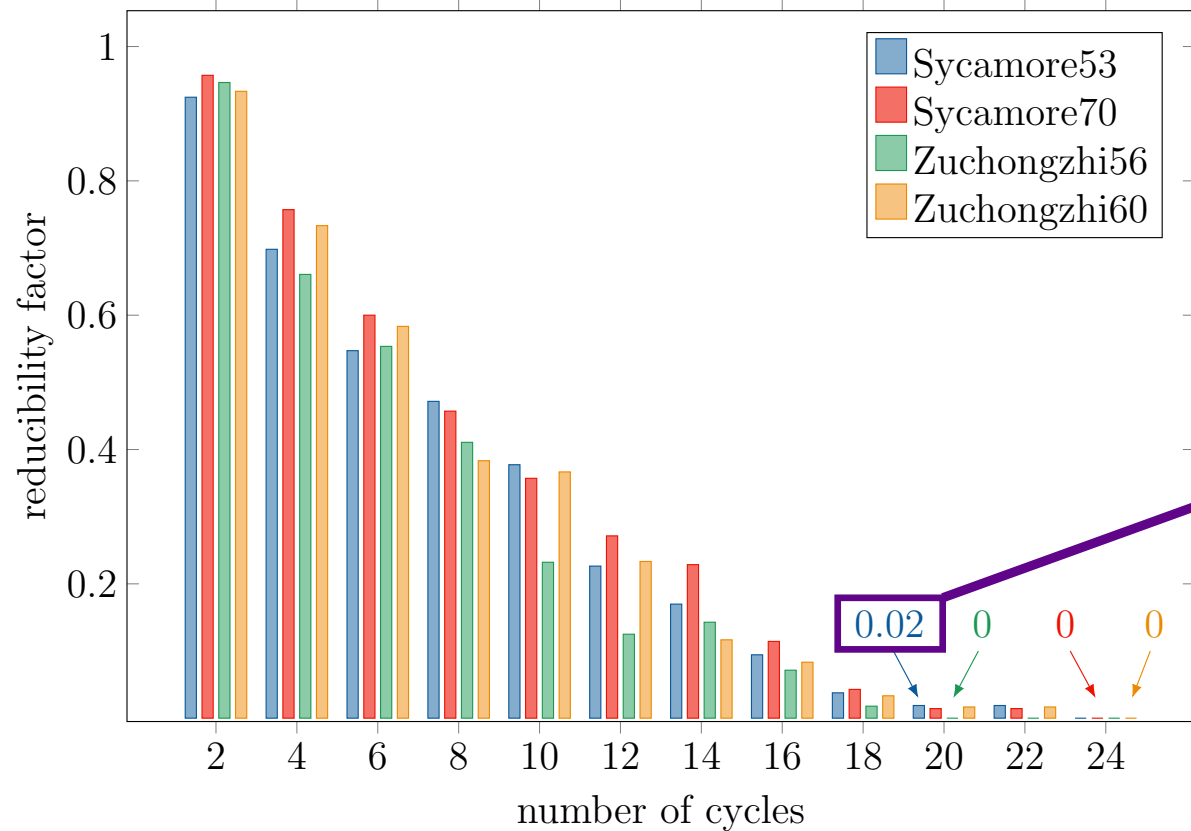
[2] Alexis Morvan et al. Phase transition in Random Circuit Sampling. arXiv:2304.11119 (2023).

[3] Yulin Wu et al. Strong quantum computational advantage using a superconducting quantum processor. *Phys. Rev. Lett.* 127, 180501 (2021).

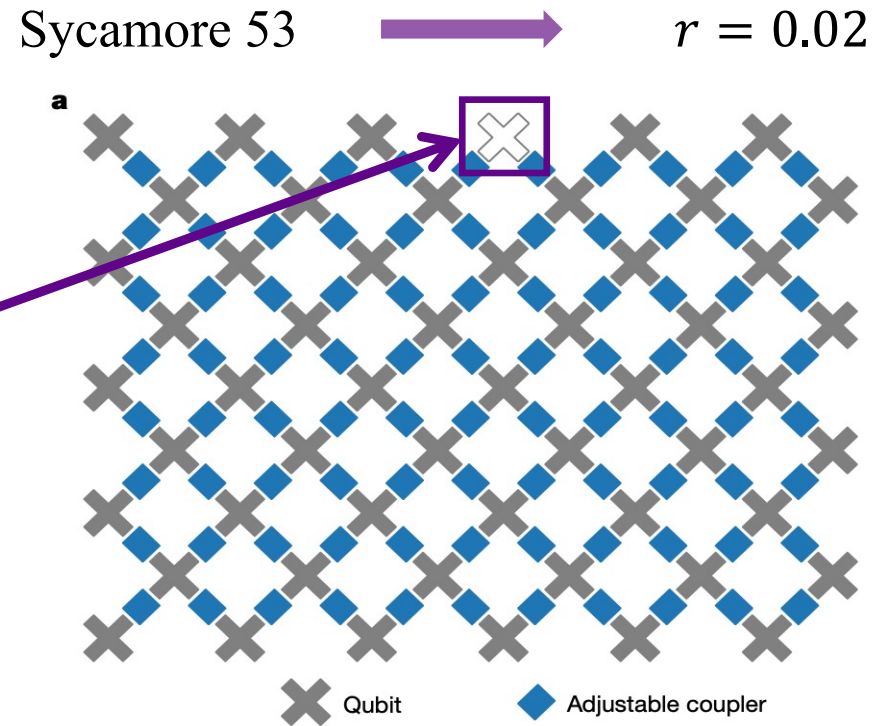
[4] Qingling Zhu et al. Quantum computational advantage via 60-qubit 24-cycle random circuit sampling. *Science Bulletin* 67, 240-245 (2022).

Numerical Evaluation: Quantum Supremacy Circuits

*Demonstrating Quantum Supremacy



Interesting Observation II



[1] Frank Arute et al. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 505-510 (2019).

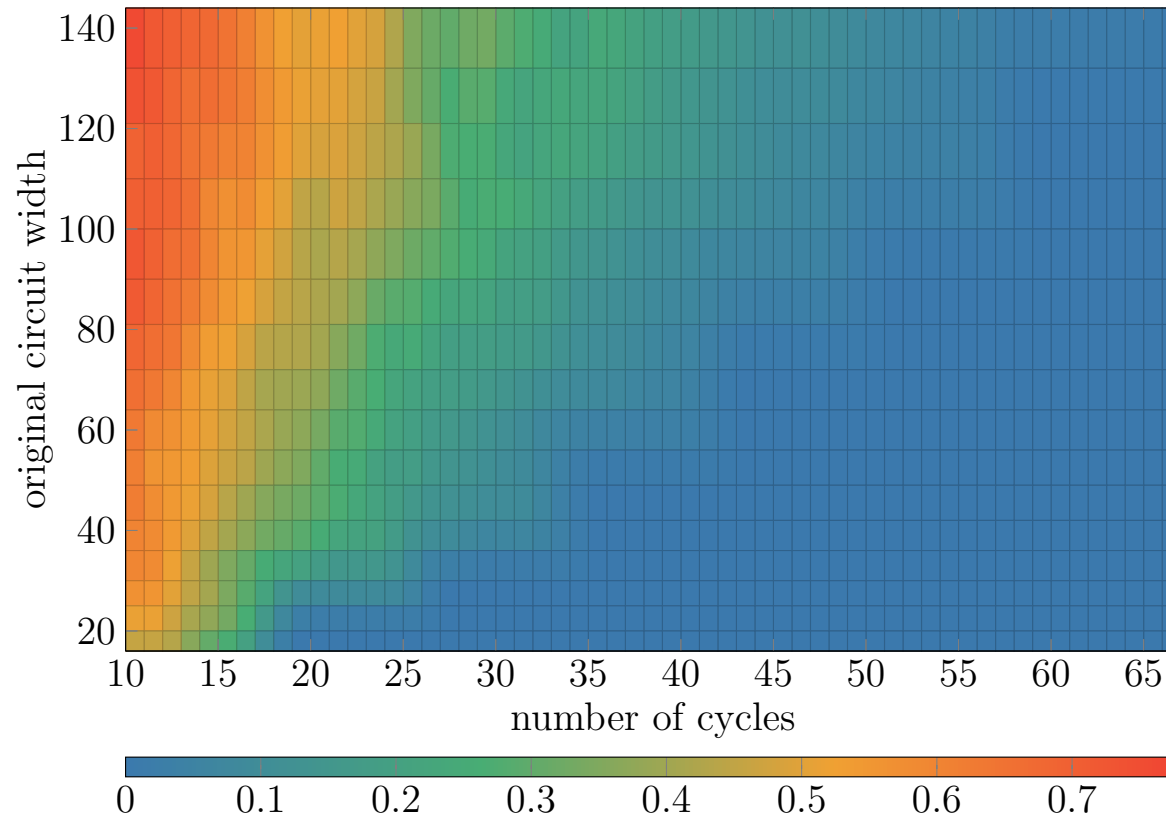
[2] Alexis Morvan et al. Phase transition in Random Circuit Sampling. arXiv:2304.11119 (2023).

[3] Yulin Wu et al. Strong quantum computational advantage using a superconducting quantum processor. *Phys. Rev. Lett.* 127, 180501 (2021).

[4] Qingling Zhu et al. Quantum computational advantage via 60-qubit 24-cycle random circuit sampling. *Science Bulletin* 67, 240-245 (2022).

Numerical Evaluation: Quantum Supremacy Circuits

Google Random Circuit Sampling (GRCS)



The larger the number of cycles
The less reducible the circuit

Reducibility Factor

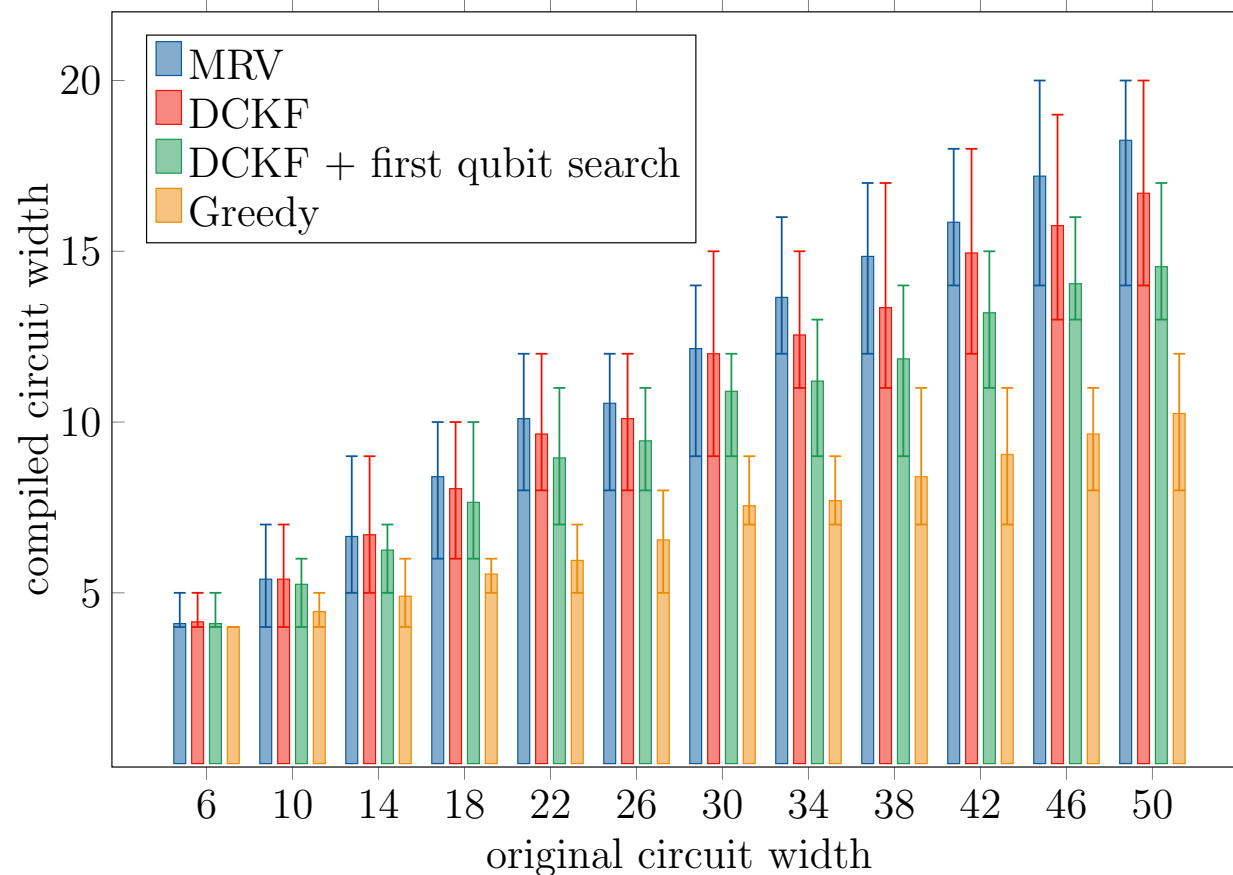
[1] Sergio Boixo et al. Characterizing quantum supremacy in near-term devices. *Nature Phys* **14**, 595–600 (2018).

[2] GRCS. <https://github.com/sboixo/GRCS>

Numerical Evaluation: QAOA circuits

QAOA circuits for max-cut problem on random 3-regular graph

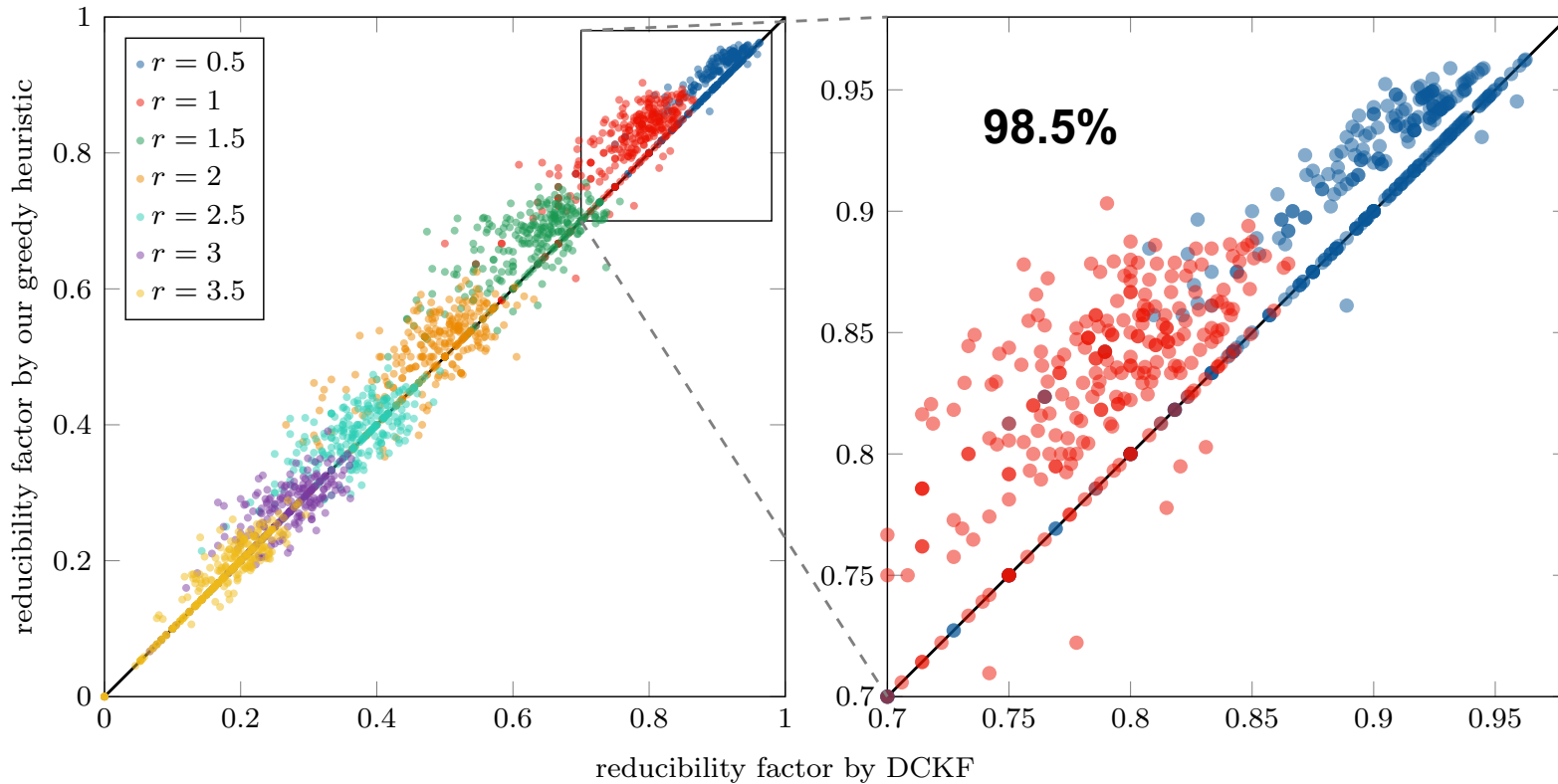
Commutable ZZ gates



Our methods account for **commutable gates**, which leads to better performance.

Numerical Evaluation: Random Quantum Circuits

Normal Random Circuit

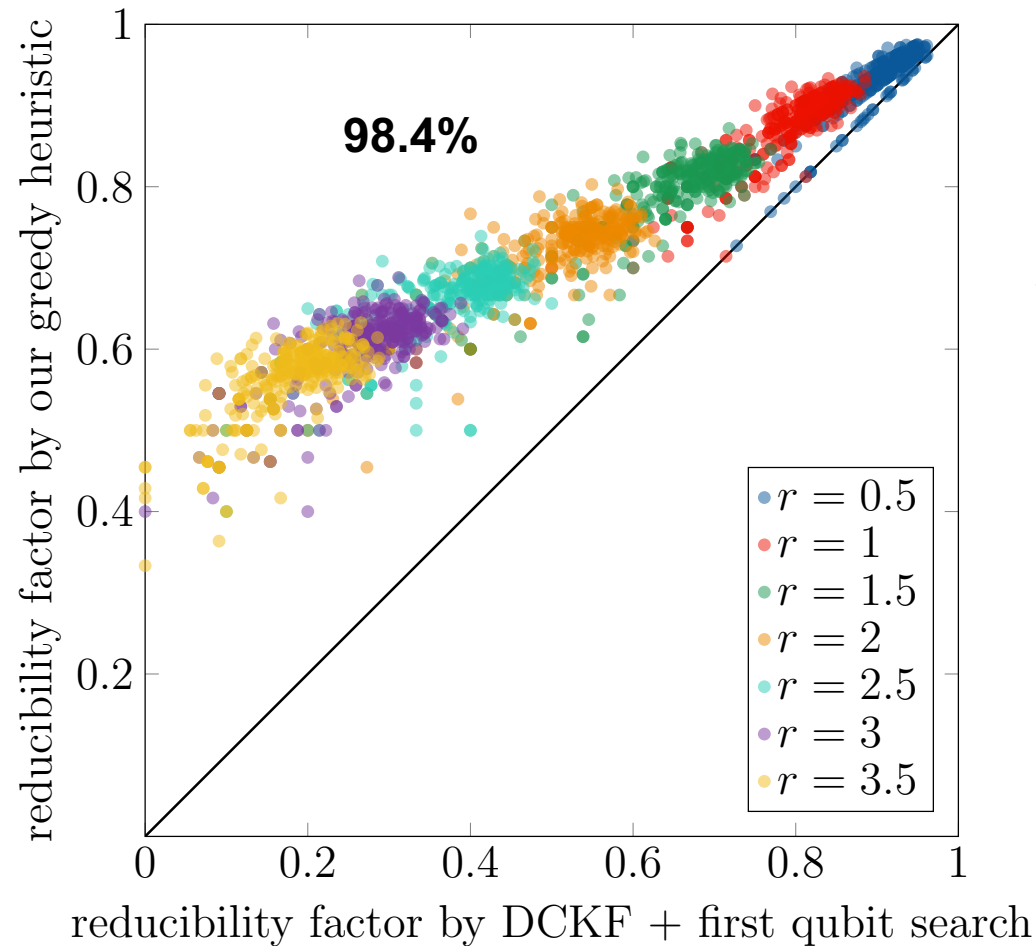


$$r = \frac{\# \text{ of two - qubit gates}}{\# \text{ of qubits}}$$

Outperforms in approximately **87.6%**
of cases, with a **strict advantage** over
49.6% of random circuits

Numerical Evaluation: Random Quantum Circuits

Random IQP Circuit



A lot of commutable gates

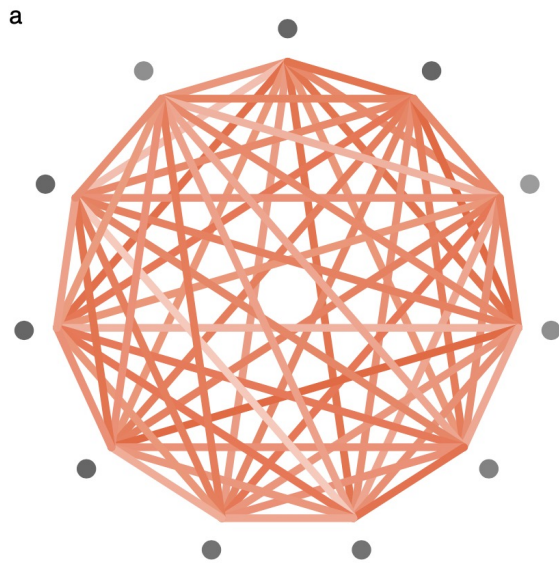
**Outperforms in nearly
100% of instances**

Numerical Evaluation: Noisy Simulation

Noisy Simulation of a 11-qubit Bernstein-Vazirani Algorithm

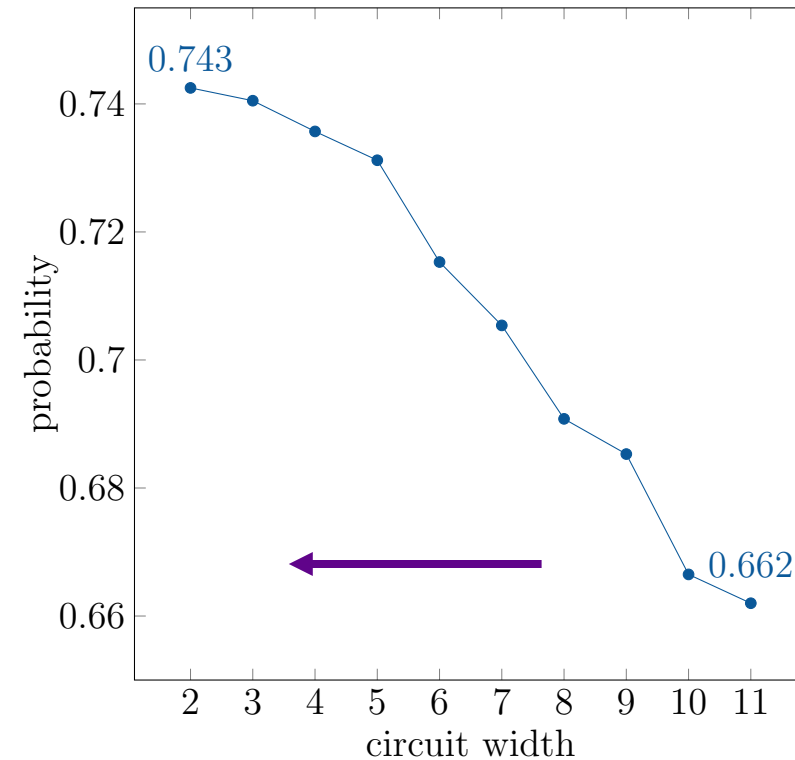
Compilation reduces the number of qubits required, allowing us to choose physical qubits with better performance.

11-qubit Trapped-Ion Quantum Computer^[1]



All-to-all connectivity

Experiment Results



Qubit reduce up to 82% and probability improve 9%

[1] K. Wright et al. Benchmarking an 11-qubit quantum computer. *Nature Communications* 10, 5464 (2019).

Summary

Large-Scale
Quantum Algorithm



Quantum Computer
with Limited Resource

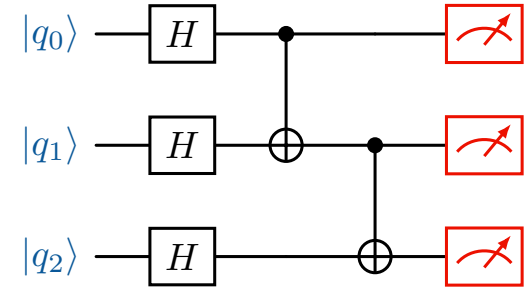
Dynamic Quantum



Circuit Compilation

Challenges

- Hardware Support for Dynamic Circuit
- Reducibility of Quantum Advantageous Applications



- Qubit Saving

Advantages

- Swap Reduction
- Improved Fidelity

Thanks for your attention!

See [arXiv:2310.11021](https://arxiv.org/abs/2310.11021)
for more details.

Frequently Asked Questions

- **Q: Dynamic circuit compilation also increases the circuit depth, why do we mainly focus on minimizing the circuit width?**
- **A:** Our work aims to explore the **fundamental limits** of qubit reuse compilation. The circuit depth may be reduced if mapping is considered. Besides, our framework is also **adaptable to other scenarios**, such as optimizing tradeoffs among circuit width, depth, and related factors.
- **Q: It seems that many important quantum algorithms (e.g. Shor) are irreducible...**
- **A:** Dynamic circuit compilation depends on the **specific circuit implementation**. It's possible that certain decompositions could be reducible. The compilation at the algorithmic level is left for future work.
- **Q: Real machine evaluation ?**
- **A:** We do hope to have the opportunity to test our methods on real quantum device.

Appendix

Numerical Evaluation: Noisy simulation

Noisy Simulation of a 11-qubit Bernstein–Vazirani Algorithm

Qubit Mapping and Probability of getting correct outcome

	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	probability
BV_11	Q_3	Q_0	Q_2	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}	Q_1	66.2%
BV_10	Q_3	Q_0	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}	Q_1	-	66.7%
BV_9	Q_3	Q_0	Q_4	Q_6	Q_7	Q_8	Q_9	Q_{10}	Q_1	-	-	68.5%
BV_8	Q_3	Q_0	Q_4	Q_6	Q_7	Q_8	Q_9	Q_1	-	-	-	69.1%
BV_7	Q_3	Q_0	Q_4	Q_7	Q_8	Q_9	Q_1	-	-	-	-	70.5%
BV_6	Q_3	Q_0	Q_4	Q_7	Q_9	Q_1	-	-	-	-	-	71.5%
BV_5	Q_3	Q_0	Q_7	Q_9	Q_1	-	-	-	-	-	-	73.1%
BV_4	Q_3	Q_0	Q_7	Q_1	-	-	-	-	-	-	-	73.6%
BV_3	Q_3	Q_0	Q_1	-	-	-	-	-	-	-	-	74.1%
BV_2	Q_3	Q_1	-	-	-	-	-	-	-	-	-	74.3%