Hardware-Optimal Quantum Algorithms

Krishnendu Chatterjee Thomas A. Henzinger
Stefanie Muroya

Workshop on Verification of Quantum Computing (VQC)
July 21st, 2025

Introduction

Motivation

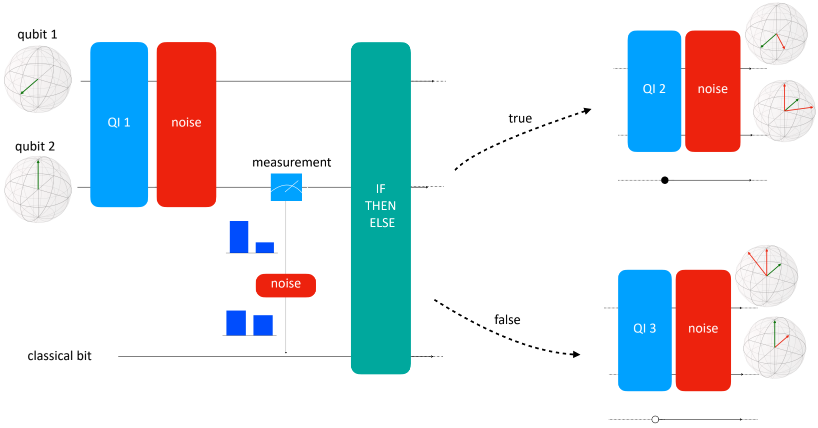
- quantum computers are very noisy
- algorithms do not behave as expected
- better performance is necessary for achieving fault-tolerant quantum computing

Motivation

- quantum computers are very noisy
- algorithms do not behave as expected
- better performance is necessary for achieving fault-tolerant quantum computing

"Automatically synthesize quantum algorithms considering a concrete hardware specification"

Quantum Algorithm



Hardware Specification and Semantics

Hardware Specification H

Unitary Instructions

$$H : \texttt{UnitaryIns} \to \underbrace{\texttt{LinearMap}}_{\text{quantum channel}}$$

Measurement Instructions

$$H : \texttt{MeasIns} \to \underbrace{(\texttt{Bit} \to \mathcal{D}(\texttt{Bit}))}_{\text{measurement channel}}$$

Hardware Specification and Semantics

Hardware Specification H

Unitary Instructions

$$H : \texttt{UnitaryIns} \to \underbrace{\texttt{LinearMap}}_{\text{quantum channel}}$$

Measurement Instructions

$$H : \texttt{MeasIns} \to \underbrace{(\texttt{Bit} \to \mathcal{D}(\texttt{Bit}))}_{\text{measurement channel}}$$

$\Downarrow$ hardware semantics

$$[\![H]\!] : \texttt{HybridState} \times \texttt{QuantumIns} \to \mathcal{D}(\texttt{HybridState})$$

Related Works

- Quantum synthesis approaches remain mostly noise agnostic [Xu et al., 2023, Kusyk et al., 2021, Guo and Wang, 2024, Kang and Oh, 2023].
- Some efforts towards noise aware synthesis:
  - pulse-level optimization [Lin et al., 2022, Voichick et al., 2025]
  - machine learning heuristics [Charrwi et al., 2024, Kikuchi et al., 2023]
  - noise-aware qubit mapping [Murali et al., 2019, Sun et al., 2025, Sivarajah et al., 2020]
- Stochastic models:
  - Quantum Markov Decision processes have been used to approximate one-qubit unitaries via Hilbert Space discretization [Alam et al., 2023] or to synthesize algorithms with a specific structure [Ming-Sheng et al., 2021, Ying and Ying, 2018].
  - Quantum Observable Markov Decision Processes (QOMDPs) [Barry et al., 2014] introduce stochastic transitions via Kraus operators.

Method

Quantitative Synthesis Problem for Quantum Algorithms

What is the algorithm for H

We are given:

- **hardware specification** H
- guard $G : \texttt{HybridState} \rightarrow 2^{\mathcal{I}} / \emptyset$
- set T of target states
- initial state s_*
- horizon k
- finite set $\mathcal{I}$ of instructions

Quantitative Synthesis Problem for Quantum Algorithms

We are given:

- **hardware specification** H
- guard $G : \texttt{HybridState} \rightarrow 2^{\mathcal{I}}/\emptyset$
- set T of target states
- initial state s_*
- horizon k
- finite set $\mathcal{I}$ of instructions

What is the algorithm for H

that with the
highest probability

Quantitative Synthesis Problem for Quantum Algorithms

We are given:

- hardware specification H
- **guard** G : $\texttt{HybridState} \to 2^{\mathcal{I}}/\emptyset$
- **set T of target states**
- initial state s_*
- horizon k
- finite set $\mathcal{I}$ of instructions

What is the algorithm for H

that with the
highest probability

satisfies G and reaches T

Quantitative Synthesis Problem for Quantum Algorithms

We are given:

- hardware specification H
- guard $G : \texttt{HybridState} \rightarrow 2^{\mathcal{I}} / \emptyset$
- set T of target states
- **initial state** s_*
- horizon k
- finite set $\mathcal{I}$ of instructions

What is the algorithm for H

that with the
highest probability

satisfies G and reaches T

starting from s_*

Quantitative Synthesis Problem for Quantum Algorithms

We are given:

- hardware specification H
- guard $G : \texttt{HybridState} \to 2^{\mathcal{I}} / \emptyset$
- set T of target states
- initial state s_*
- **horizon** k
- **finite set** $\mathcal{I}$ **of instructions**

What is the algorithm for H

that with the
highest probability

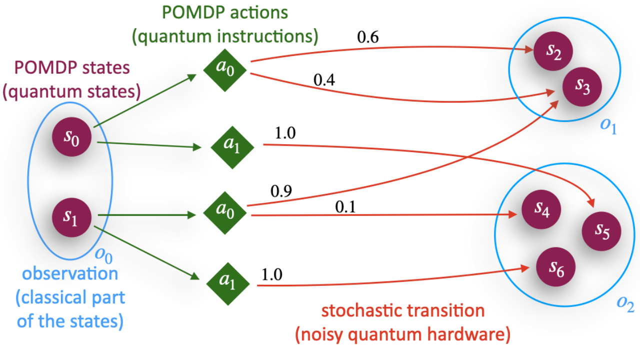
satisfies G and reaches T

starting from s_*

using at most k instructions from $\mathcal{I}$?

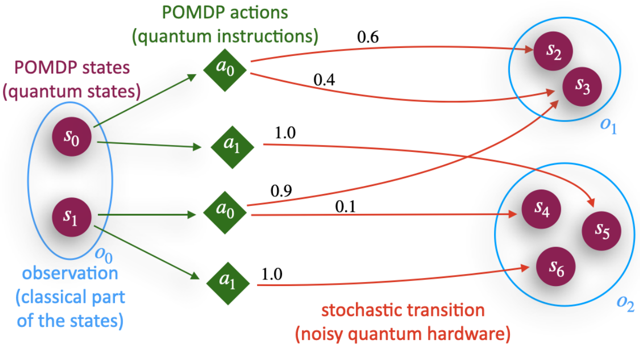
Partially Observable Markov Decision Process (POMDP)

$$P = \langle S, A, \mathcal{O}, \delta : S \times A \rightarrow \mathcal{D}(S), \gamma : S \rightarrow \mathcal{O} \rangle$$



Partially Observable Markov Decision Process (POMDP)

$$P = \langle S, A, \mathcal{O}, \delta : S \times A \to \mathcal{D}(S), \gamma : S \to \mathcal{O} \rangle$$



$$\tau^* : (\mathcal{O} \cdot A)^* \cdot \mathcal{O} \to \mathcal{D}(A)$$

Experiments

Parity-bitflip problem: setup

- **Hilbert space $\mathcal{H}_8$**

```
CX(0,2);
CX(1,2);
m = 0;
for _ in [0..meas_count]:
    m+= MEASURE(2);
if m >= ceil(meas_count/2):
    X(0)
```

Parity-bitflip problem: setup

- **Hilbert space $\mathcal{H}_8$**
- **Initial uniform distribution** over Bell states

```
CX(0,2);
CX(1,2);
m = 0;
for _ in [0..meas_count]:
    m+= MEASURE(2);
if m >= ceil(meas_count/2):
    X(0)
```

Parity-bitflip problem: setup

- **Hilbert space** $\mathcal{H}_8$
- **Initial uniform distribution** over Bell states
- **Target states** are hybrid states with an even parity Bell state.

```
CX(0,2);
CX(1,2);
m = 0;
for _ in [0..meas_count]:
    m+= MEASURE(2);
if m >= ceil(meas_count/2):
    X(0)
```

Parity-bitflip problem: setup

- **Hilbert space** $\mathcal{H}_8$
- **Initial uniform distribution** over Bell states
- **Target states** are hybrid states with an even parity Bell state.
- 44 **hardware specifications**

```
CX(0,2);
CX(1,2);
m = 0;
for _ in [0..meas_count]:
    m+= MEASURE(2);
if m >= ceil(meas_count/2):
    X(0)
```

Parity-bitflip problem: setup

- **Hilbert space** $\mathcal{H}_8$
- **Initial uniform distribution** over Bell states
- **Target states** are hybrid states with an even parity Bell state.
- 44 **hardware specifications**
- 225 **embeddings**

```
CX(0,2);
CX(1,2);
m = 0;
for _ in [0..meas_count]:
    m+= MEASURE(2);
if m >= ceil(meas_count/2):
    X(0)
```

Parity-bitflip problem: setup

- **Hilbert space** $\mathcal{H}_8$
- **Initial uniform distribution** over Bell states
- **Target states** are hybrid states with an even parity Bell state.
- 44 **hardware specifications**
- 225 **embeddings**
- **Horizons** 4 through 7

```
CX(0,2);
CX(1,2);
m = 0;
for _ in [0..meas_count]:
    m+= MEASURE(2);
if m >= ceil(meas_count/2):
    X(0)
```

Parity-bitflip problem: setup

- **Hilbert space** $\mathcal{H}_8$
- **Initial uniform distribution** over Bell states
- **Target states** are hybrid states with an even parity Bell state.
- 44 **hardware specifications**
- 225 **embeddings**
- **Horizons** 4 through 7
- **Guards** on measurements

```
CX(0,2);
CX(1,2);
m = 0;
for _ in [0..meas_count]:
    m+= MEASURE(2);
if m >= ceil(meas_count/2):
    X(0)
```

Parity-bitflip problem (results)

IPMA Instruction Set

{`CX(0,2)`, `CX(1,2)`, `X(0)`, `measure(2)`}

- Maximum improvement achieved: 9.8%
- Synthesized 27 different algorithms
- Repeating blocks of **CX** gates is optimal when there is a high measurement success probability.
- Performing more measurements is not always better.

CX+H Instruction Set

{`H(1)`, `H(2)`, `CX(2,1)`, `CX(0,1)`, `measure(2)`}

- Maximum improvement achieved: 34.07% (from 61.13% to 95.2%)
- Synthesized 4 different algorithms.
- Only found algorithms at horizon 6.
- Optimal algorithms using up to 4 `CX` gates.

Parity-bitflip problem (samples)

```
CX(0,2);
CX(1,2);
if MEASURE(2) == 0:
    CX(0,2);
    CX(1,2);
    if MEASURE(2) == 1:
        if MEASURE(2) == 1:
            X(0)
    else:
        CX(0,2);
        CX(1,2);
        if MEASURE(2) == 0:
            X(0)
else:
    CX(0,2);
    CX(1,2);
    if MEASURE(2) == 0:
        if MEASURE(2) == 0:
            X(0)
        else:
            if MEASURE(2) == 0:
            X(0)
    else:
        CX(0,2);
        CX(1,2);
        if MEASURE(2) == 0:
            X(0)
```

```
H(2);
CX(2,1);
H(1);
CX(0,1);
H(2);
CX(2,1);
MEASURE(2);
```

Qubit-reset problem: setup

- **Hilbert space $\mathcal{H}_2$**

```
m = 0;
for _ in [0..meas_count]:
    m+= MEASURE(0);
if m >= ceil(meas_count/2):
    X(0)
```

Qubit-reset problem: setup

- **Hilbert space $\mathcal{H}_2$**
- **Initial uniform distribution** over $|0\rangle$ and $|1\rangle$

```
m = 0;
for _ in [0..meas_count]:
    m+= MEASURE(0);
if m >= ceil(meas_count/2):
    X(0)
```

Qubit-reset problem: setup

- **Hilbert space** $\mathcal{H}_2$
- **Initial uniform distribution** over $|0\rangle$ and $|1\rangle$
- **Target states** is any hybrid state with quantum state $|0\rangle$.

```
m = 0;
for _ in [0..meas_count]:
    m+= MEASURE(0);
if m >= ceil(meas_count/2):
    X(0)
```

Qubit-reset problem: setup

- **Hilbert space** $\mathcal{H}_2$
- **Initial uniform distribution** over $|0\rangle$ and $|1\rangle$
- **Target states** is any hybrid state with quantum state $|0\rangle$.
- 44 **hardware specifications**

```
m = 0;
for _ in [0..meas_count]:
    m+= MEASURE(0);
if m >= ceil(meas_count/2):
    X(0)
```

Qubit-reset problem: setup

- **Hilbert space** $\mathcal{H}_2$
- **Initial uniform distribution** over $|0\rangle$ and $|1\rangle$
- **Target states** is any hybrid state with quantum state $|0\rangle$.
- 44 **hardware specifications**
- 156 **embeddings**

```
m = 0;
for _ in [0..meas_count]:
    m+= MEASURE(0);
if m >= ceil(meas_count/2):
    X(0)
```

Qubit-reset problem: setup

- **Hilbert space** $\mathcal{H}_2$
- **Initial uniform distribution** over $|0\rangle$ and $|1\rangle$
- **Target states** is any hybrid state with quantum state $|0\rangle$.
- 44 **hardware specifications**
- 156 **embeddings**
- **Horizons** 2 through 7

```
m = 0;
for _ in [0..meas_count]:
    m+= MEASURE(0);
if m >= ceil(meas_count/2):
    X(0)
```

Qubit-reset problem: setup

- **Hilbert space** $\mathcal{H}_2$
- **Initial uniform distribution** over $|0\rangle$ and $|1\rangle$
- **Target states** is any hybrid state with quantum state $|0\rangle$.
- 44 **hardware specifications**
- 156 **embeddings**
- **Horizons** 2 through 7
- No **guards**

```
m = 0;
for _ in [0..meas_count]:
    m+= MEASURE(0);
if m >= ceil(meas_count/2):
    X(0)
```

Qubit-reset problem: results

$\{\texttt{X(0)}, \texttt{measure(0)}\}$

- Maximum improvement achieved: 13.5% (from 61.9% to 75.4%)
- Synthesized 42 different algorithms.
- Several optimal algorithms that exploit the higher accuracy of measurements for one of the basis states.
- For some hardware specifications, we provide guarantees above 99.999%.

```
if MEASURE(0) == 0:
    if MEASURE(0) == 0:
        if MEASURE(0) == 0:
            if MEASURE(0) == 1:
                X(0);
        else:
            X(0);
    else:
        X(0);
        if MEASURE(0) == 1:
            X(0);
else:
    if MEASURE(0) == 0:
        X(0);
        if MEASURE(0) == 1:
            X(0);
    else:
        X(0);
```

GHZ state preparation problem: setup

- **Hilbert space** $\mathcal{H}_8$

```
H(0);
CX(0,1);
CX(1,2);
```

GHZ state preparation problem: setup

- **Hilbert space** $\mathcal{H}_8$
- **Initial state** is $|000\rangle$.

```
H(0);
CX(0,1);
CX(1,2);
```

GHZ state preparation problem: setup

- **Hilbert space** $\mathcal{H}_8$
- **Initial state** is $|000\rangle$.
- **Target states** is any hybrid state with quantum state $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$.

```
H(0);
CX(0,1);
CX(1,2);
```

GHZ state preparation problem: setup

- **Hilbert space** $\mathcal{H}_8$
- **Initial state** is $|000\rangle$.
- **Target states** is any hybrid state with quantum state $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$.
- 44 **hardware specifications**

```
H(0);
CX(0,1);
CX(1,2);
```

GHZ state preparation problem: setup

- **Hilbert space** $\mathcal{H}_8$
- **Initial state** is $|000\rangle$.
- **Target states** is any hybrid state with quantum state $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$.
- 44 **hardware specifications**
- 1275 **embeddings**

```
H(0);
CX(0,1);
CX(1,2);
```

GHZ state preparation problem: setup

- **Hilbert space** $\mathcal{H}_8$
- **Initial state** is $|000\rangle$.
- **Target states** is any hybrid state with quantum state $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$.
- 44 **hardware specifications**
- 1275 **embeddings**
- A single instruction set per embedding

```
H(0);
CX(0,1);
CX(1,2);
```

GHZ state preparation problem: setup

- **Hilbert space** $\mathcal{H}_8$
- **Initial state** is $|000\rangle$.
- **Target states** is any hybrid state with quantum state $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$.
- 44 **hardware specifications**
- 1275 **embeddings**
- A single instruction set per embedding
- **Horizon** 3

```
H(0);
CX(0,1);
CX(1,2);
```

GHZ state preparation problem: setup

- **Hilbert space** $\mathcal{H}_8$
- **Initial state** is $|000\rangle$.
- **Target states** is any hybrid state with quantum state $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$.
- 44 **hardware specifications**
- 1275 **embeddings**
- A single instruction set per embedding
- **Horizon** 3
- No **guards**

```
H(0);
CX(0,1);
CX(1,2);
```

GHZ state preparation problem: results

- Goal: compare against Qiskit transpiler
- use optimization level 2 and 3
- Average improvement is of 5.72% and 7.48% respectively.
- Highest improvement: 92% (6.1% vs. 98.1%)
- In 12 quantum hardware the improvement is at least 2.5%

```
H(1);
CX(1,0);
CX(1,2);
```

Conclusion

Conclusion

- We presented a framework that incorporates hardware specifications into the synthesis process.
- We target the synthesis process of practical algorithms while providing provable guarantees about their performance.
- Our results highlight the need for hardware-specific algorithms.

References

Alam, M. S., Berthusen, N. F., and Orth, P. P. (2023).
Quantum logic gate synthesis as a markov decision process.

Barry, J., Barry, D. T., and Aaronson, S. (2014).
Quantum partially observable markov decision processes.
Phys. Rev. A, 90:032311.

Charrwi, M. W., Ioannou, G., Younis, E., de Jong, W. A., and Saeed, S. M. (2024).
Quantum circuit partitioning for scalable noise-aware quantum circuit re-synthesis.
In *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 359–364.
10.1109/QCE60285.2024.10306.

Guo, Z.-H. and Wang, T.-C. (2024).
Smt-based layout synthesis approaches for quantum circuits.
In *Proceedings of the 2024 International Symposium on Physical Design*, ISPD '24, page 235–243, New York, NY, USA. Association for Computing Machinery.

Kang, C. and Oh, H. (2023).
Modular component-based quantum circuit synthesis.
Proceedings of the ACM on Programming Languages, 7:348–375.

Kikuchi, Y., Mc Keever, C., Coopmans, L., Lubasch, M., and Benedetti, M. (2023).
Realization of quantum signal processing on a noisy quantum computer.
npj Quantum Information, 9:93.
10.1038/s41534-023-00762-0.

Kusyk, J., Saeed, S. M., and Uyar, M. U. (2021).
Survey on quantum circuit compilation for noisy intermediate-scale quantum computers: Artificial intelligence to heuristics.
IEEE Transactions on Quantum Engineering, 2:1–16.

Lin, S. F., Sussman, S., Duckering, C., Mundada, P. S., Baker, J. M., Kumar, R. S., Houck, A. A., and Chong, F. T. (2022).
Let each quantum bit choose its basis gates.
In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 1042–1058.
10.1109/MICRO56248.2022.00075.

Ming-Sheng, Y., Yuan, F., and Sheng-Gang, Y. (2021).
Optimal policies for quantum markov decision processes.
In *International Journal of Automation and Computing*, volume 18, page 410.

Murali, P., Baker, J. M., Javadi-Abhari, A., Chong, F. T., and Martonosi, M. (2019).
Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers.
In *Proceedings of the twenty-fourth international conference on architectural support for programming languages and operating systems*, pages 1015–1029.
10.1145/3297858.3304075.

Sivarajah, S., Dilkes, S., Cowtan, A., Simmons, W., Edgington, A., and Duncan, R. (2020).
t—ket⟩: a retargetable compiler for nisq devices.
Quantum Science and Technology, 6:014003.
10.1088/2058-9565/ab8e92.

Sun, W., Li, X., Yu, L., Wang, Z., Chen, G., Zheng, D., and Yang, G. (2025).
Haqa: A hardware-guided and fidelity-aware strategy for efficient qubit mapping optimization.
arXiv:2504.16468.

Voichick, F., Lampropoulos, L., and Rand, R. (2025).
Cognac: Circuit optimization via gradients and noise-aware compilation.
arXiv:2311.02769.

Xu, A., Molavi, A., Pick, L., Tannu, S., and Albarghouthi, A. (2023).
Synthesizing quantum-circuit optimizers.
Proceedings of the ACM on Programming Languages, 7(PLDI):835–859.

Ying, S. and Ying, M. (2018).
Reachability analysis of quantum markov decision processes.
Information and Computation, 263:31–51.