Learning based hardware-centric quantum circuit generation

M.A. Schalkers,[@,*] M. Moller,[*] D. de Laat[*]

[@]m.a.schalkers@tudelft.nl, [*]Delft Institute of Applied Mathematics, TU Delft

Problem and motivation

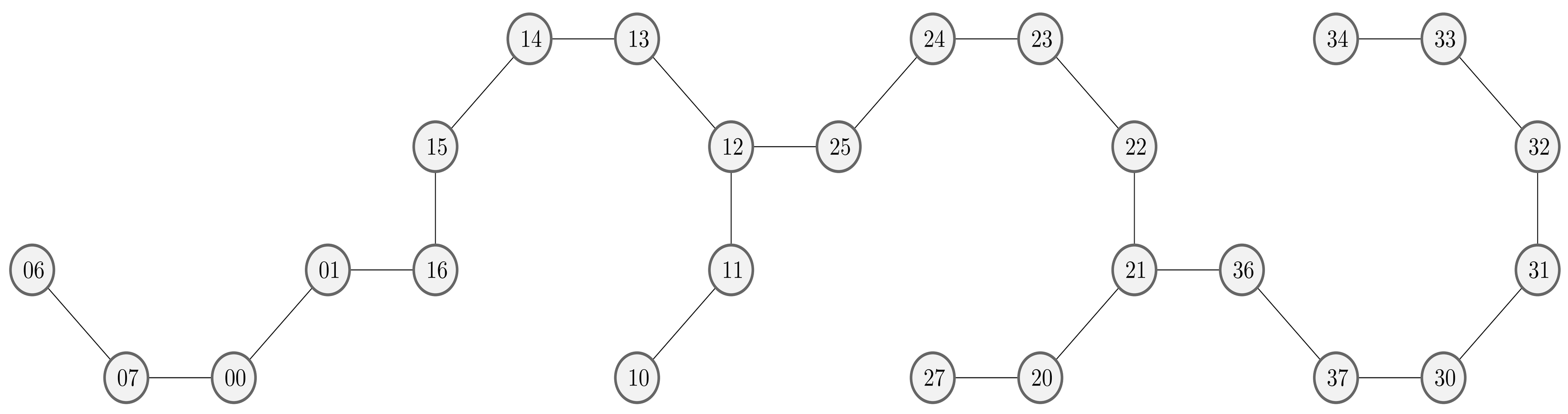
Noisy Intermediate Scale Quantum (**NISQ**) computers have properties that strongly influence their performance:

- Limited amount of qubits (~ 50)
- Short decoherence time
- Limited native gates
- Sparse connectivity

⇒ **Consequence**: Quantum algorithms of practical interest cannot be run on NISQ devices due to resource limitations.
⇒ **Idea**: Generate circuits which mimick a desired operation while respecting the limitations of NISQ devices.

Connectivity and native gate set

Only gates in the native gate set can be directly applied to the qubits. Furthermore we can only apply two-qubit gates between qubits that are connected on the hardware. Most QPUs have **sparse connectivity**, like the Rigetti Aspen 7 portrayed below.



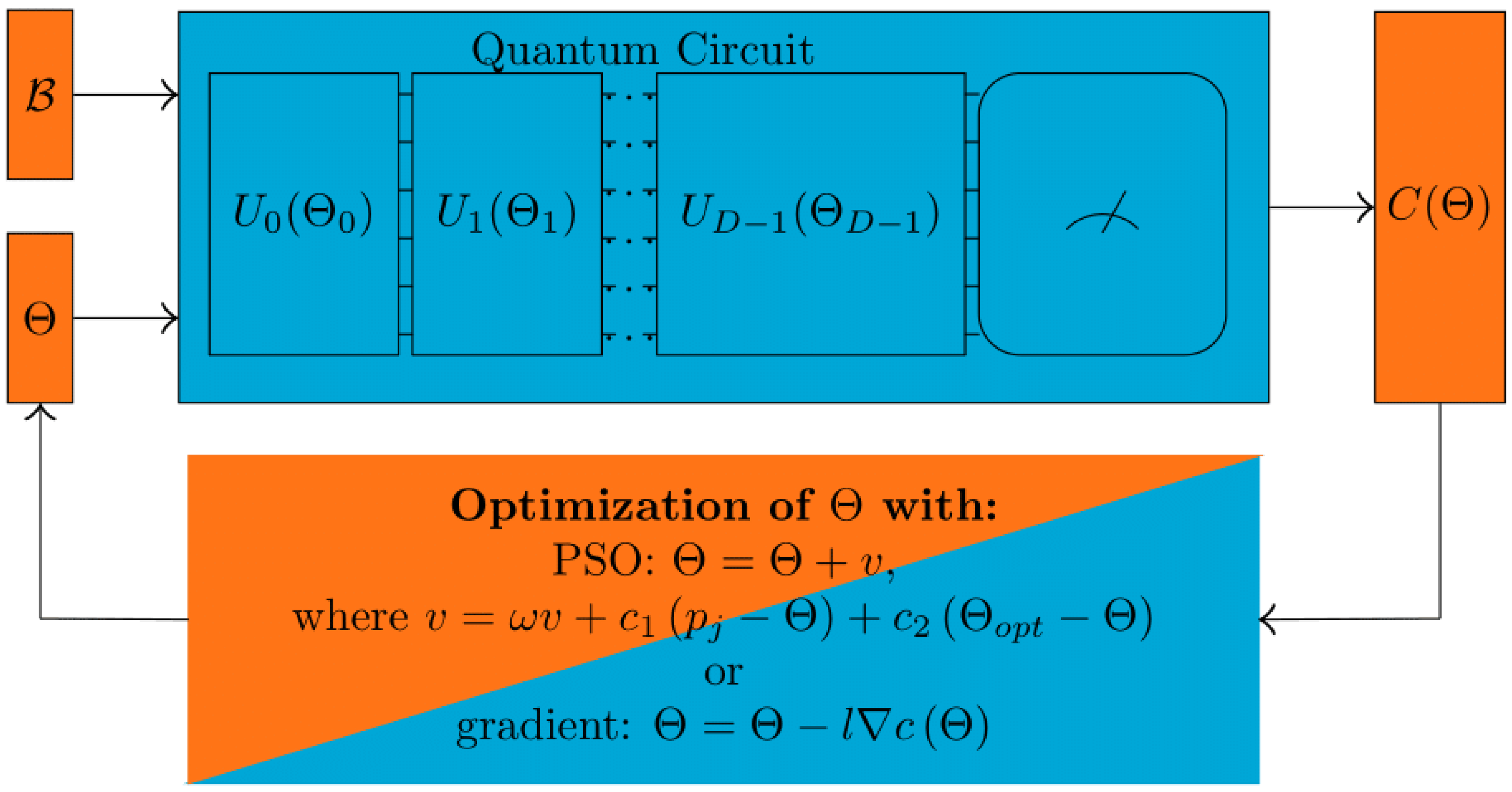
Goal and approach

Approach: Given a set $\mathcal{B}$ of quantum states $|\phi\rangle$ and associated probability output vectors $\mathbf{p}$, find a quantum circuit $V(\Theta)$ that creates a similar probability output vector for every input state.

- **Input:** $\mathcal{B} = \{(|\phi\rangle, \mathbf{p})\}$
- **Output:** $V(\Theta)$
- **Goal:** $(|\langle i \,|\, V(\Theta) \,|\, \phi\rangle|^2) \approx p_i$

Use the following **cost function** to optimize the continuous parameter Θ:

- $C(\Theta) = \frac{1}{|\mathcal{B}|}\sum_{(|\phi\rangle,\mathbf{p})\in\mathcal{B}} \|\mathbf{p} - \mathbf{f}(|\phi\rangle)\|_2$
- Where: $f(|\phi\rangle)_i = (|\langle i \,|\, V(\Theta) \,|\, \phi\rangle|^2)$



Optimization techniques:

We use the following methods to optimize Θ:

- **Gradient descent**: We can calculate the gradients of each quantum gate with respect to its input parameter θ_i directly on the quantum computer. Subsequently we use backpropagation to calculate the gradients of the cost function with respect to the θ_i on a classical computer.
- **Particle swarm optimization**: We use Particle Swarm Optimization to find a minimal point Θ, in the $|\Theta|$-dimensional space, representing the input parameter of the gates in the quantum circuit.

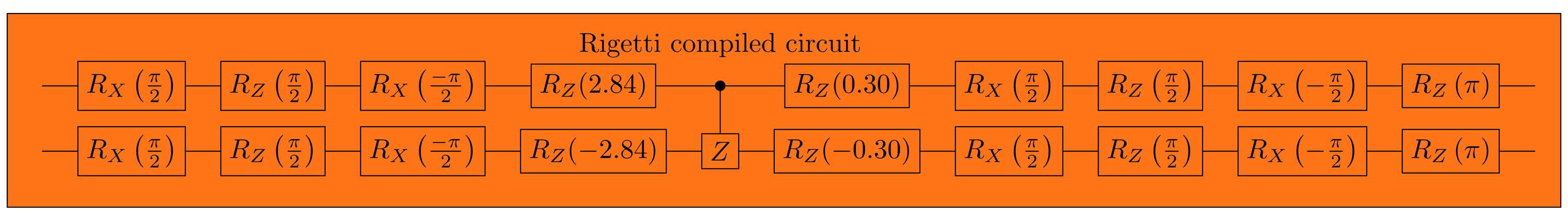
Use cases and example:

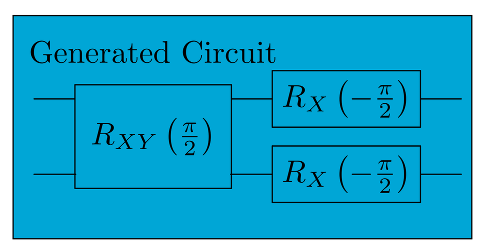
We can use our method to find quantum circuits which create a desired probability to measure each possible basis state. As such our method can be used for the following problem types:

- **Mimicking quantum circuits**: Our method can be used to mimick known quantum operations by finding a quantum circuit which creates, for all possible input states, the same probability of measuring each basis state.
- **Probabilistic operations**: Our method can be used for problems where, given an input state, we wish to return one of the indices with a desired property and/or probability.

Grover's Search for 2 qubits on Rigetti Aspen

⇒ **Example**: Our method can be used to mimick the amplitude amplification step of Grover's search. Compare the depth of our circuit with the classically compiled one.





TU Delft Delft University of Technology