Quantum Walk based Qubit Mapping

Smit Chaudhary

Supervised by : Sebastian Feld
Faculty of Applied Sciences
Delft University of Technology
Delft, The Netherlands

May 30, 2022

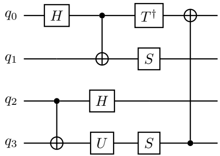
Outline

Motivation

We want to map qubits on a quantum circuit (**virtual qubits**) to the qubits on a quantum chip (**physical qubits**).

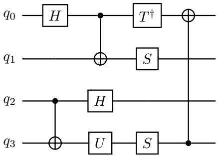
Motivation

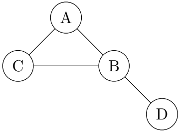
We want to map qubits on a quantum circuit (**virtual qubits**) to the qubits on a quantum chip (**physical qubits**).



Motivation

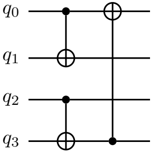
We want to map qubits on a quantum circuit (**virtual qubits**) to the qubits on a quantum chip (**physical qubits**).

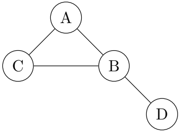




Motivation

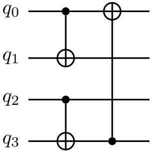
We want to map qubits on a quantum circuit (**virtual qubits**) to the qubits on a quantum chip (**physical qubits**).

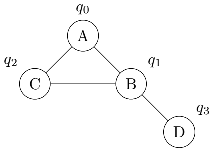




Motivation

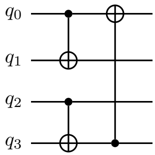
We want to map qubits on a quantum circuit (**virtual qubits**) to the qubits on a quantum chip (**physical qubits**).

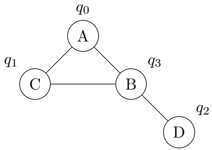




Motivation

We want to map qubits on a quantum circuit (**virtual qubits**) to the qubits on a quantum chip (**physical qubits**).



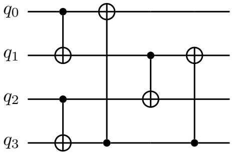


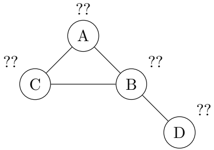
Motivation

Not always possible to find a mapping. Especially with a bigger circuit.

Motivation

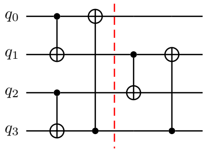
Not always possible to find a mapping. Especially with a bigger circuit.

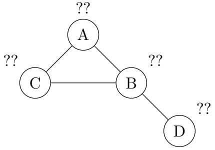




Motivation

Not always possible to find a mapping. Especially with a bigger circuit.

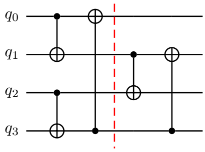


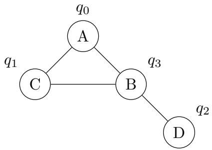


Map one part of the circuit.

Motivation

Not always possible to find a mapping. Especially with a bigger circuit.

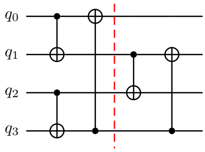


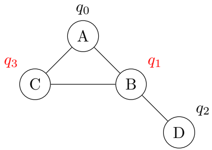


Map one part of the circuit.

Motivation

Not always possible to find a mapping. Especially with a bigger circuit.

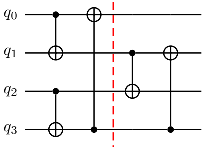


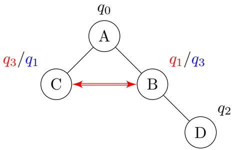


Map one part of the circuit. Map the next part of the circuit.

Motivation

Not always possible to find a mapping. Especially with a bigger circuit.





Map one part of the circuit. Map the next part of the circuit. Move the qubits to go from one mapping to the next (Routing).

Outline

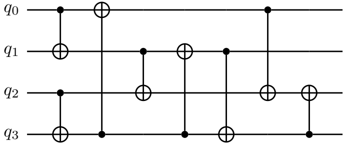
1. Motivation
2. Methods
3. Results
4. Conclusions

Problem Setup

Given: A quantum circuit to map.

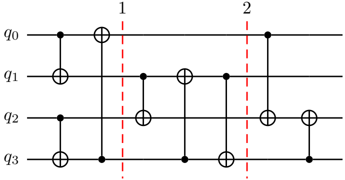
Problem Setup

Given: A quantum circuit to map.



Problem Setup

Given: A quantum circuit to map.

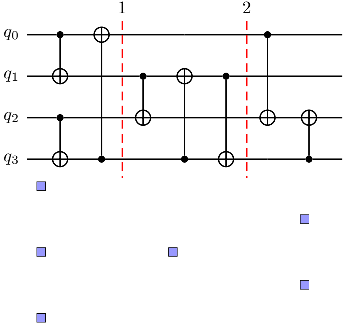


Idea:

1. Slice the circuit.

Problem Setup

Given: A quantum circuit to map.

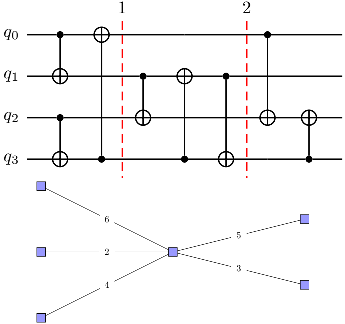


Idea:

1. Slice the circuit. Get (multiple) mappings for each slice.

Problem Setup

Given: A quantum circuit to map.

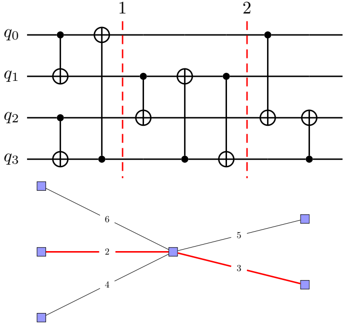


Idea:

1. Slice the circuit. Get (multiple) mappings for each slice.
2. Estimate cost of changes.

Problem Setup

Given: A quantum circuit to map.

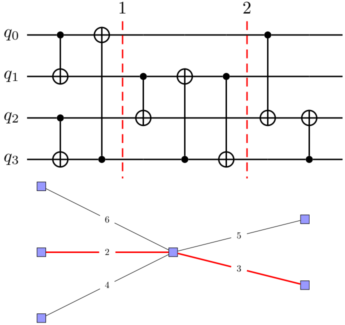


Idea:

1. Slice the circuit. Get (multiple) mappings for each slice.
2. Estimate cost of changes.
3. Optimize the cost.

Problem Setup

Given: A quantum circuit to map.



Idea:

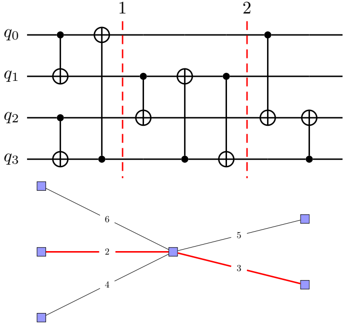
1. Slice the circuit. Get (multiple) mappings for each slice.
2. Estimate cost of changes.
3. Optimize the cost.

Two part problem:

Initial Mapping: Quantum Walk based search

Problem Setup

Given: A quantum circuit to map.



Idea:

1. Slice the circuit. Get (multiple) mappings for each slice.
2. Estimate cost of changes.
3. Optimize the cost.

Two part problem:

Initial Mapping:	Quantum Walk based search
Routing:	Classical Optimization

Solving Initial Mapping

Solving Initial Mapping

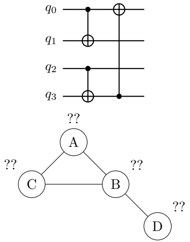
It is a constraint satisfaction problem (CSP).

Solving Initial Mapping

It is a constraint satisfaction problem (CSP). Classically, can do a backtracking search to find the solutions.

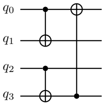
Solving Initial Mapping

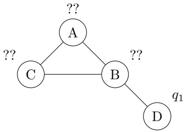
It is a constraint satisfaction problem (CSP). Classically, can do a backtracking search to find the solutions.

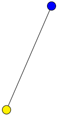


Solving Initial Mapping

It is a constraint satisfaction problem (CSP). Classically, can do a backtracking search to find the solutions.

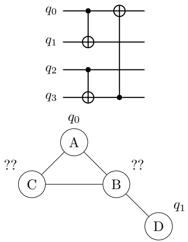


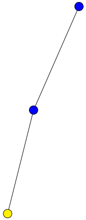




Solving Initial Mapping

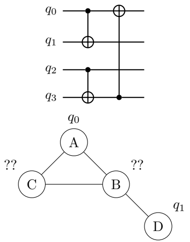
It is a constraint satisfaction problem (CSP). Classically, can do a backtracking search to find the solutions.

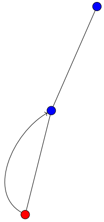




Solving Initial Mapping

It is a constraint satisfaction problem (CSP). Classically, can do a backtracking search to find the solutions.

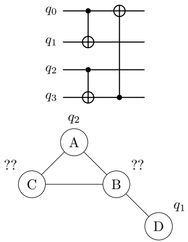


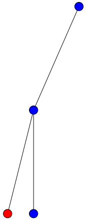


Backtrack!!

Solving Initial Mapping

It is a constraint satisfaction problem (CSP). Classically, can do a backtracking search to find the solutions.

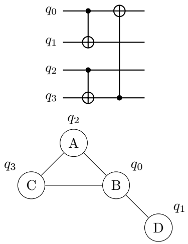


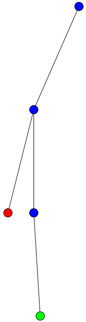


Backtrack!!

Solving Initial Mapping

It is a constraint satisfaction problem (CSP). Classically, can do a backtracking search to find the solutions.

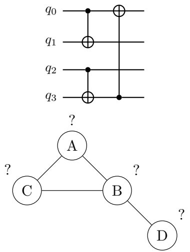


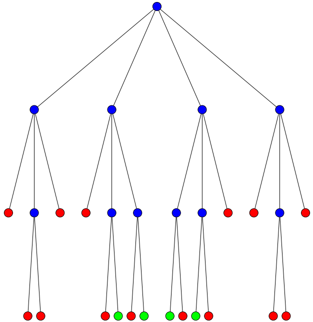


Backtrack!!

Solving Initial Mapping

It is a constraint satisfaction problem (CSP). Classically, can do a backtracking search to find the solutions.





Backtrack!!

Quantum walk speed up

Speed up this backtracking for the CSP.

Quantum walk speed up

Speed up this backtracking for the CSP.

Finding a solution (Montanaro, 2015)

Assume access to predicate $P(x) \rightarrow \{\text{true, false, indeterminate}\}$

Quantum walk speed up

Speed up this backtracking for the CSP.

Finding a solution (Montanaro, 2015)

Assume access to predicate $P(x) \rightarrow \{\text{true, false, indeterminate}\}$
And heuristic h to decide how to extend an assignment.

Quantum walk speed up

Speed up this backtracking for the CSP.

Finding a solution (Montanaro, 2015)

Assume access to predicate $P(x) \rightarrow \{\text{true, false, indeterminate}\}$
And heuristic h to decide how to extend an assignment.
A bounded error quantum algorithm evaluates P and h
$\mathcal{O}(\sqrt{Tn^{\frac{5}{2}}} log(n))$ times each and outputs solution.

Quantum walk speed up

Speed up this backtracking for the CSP.

Finding a solution (Montanaro, 2015)

Assume access to predicate $P(x) \rightarrow \{\text{true, false, indeterminate}\}$
And heuristic h to decide how to extend an assignment.
A bounded error quantum algorithm evaluates P and h $\mathcal{O}(\sqrt{Tn^{\frac{5}{2}}} log(n))$ times each and outputs solution.

- The backtracking tree is given implicitly using the predicate and heuristic.

Quantum walk speed up

Speed up this backtracking for the CSP.

Finding a solution (Montanaro, 2015)

Assume access to predicate $P(x) \rightarrow \{\text{true, false, indeterminate}\}$
And heuristic h to decide how to extend an assignment.
A bounded error quantum algorithm evaluates P and h $\mathcal{O}(\sqrt{Tn^{\frac{5}{2}}} log(n))$ times each and outputs solution.

- The backtracking tree is given implicitly using the predicate and heuristic.
- In the worst case, T grows as $n!$. We get a near quadratic separation between classical and quantum case.

Quantum walk speed up

Speed up this backtracking for the CSP.

Finding a solution (Montanaro, 2015)

Assume access to predicate $P(x) \rightarrow \{\text{true, false, indeterminate}\}$
And heuristic h to decide how to extend an assignment.
A bounded error quantum algorithm evaluates P and h
$\mathcal{O}(\sqrt{Tn^{\frac{5}{2}}} log(n))$ times each and outputs solution.

- The backtracking tree is given implicitly using the predicate and heuristic.
- In the worst case, T grows as $n!$. We get a near quadratic separation between classical and quantum case.
- Strike out previously found solutions to find all solutions.

Quantum walk in a tree

Quantum walk in a tree

The quantum walk operates on a T-dimensional Hilbert space spanned by $\{|r\rangle \cup |x\rangle \, x \in \{1, ..., T-1\}\}$ where r is root.

Quantum walk in a tree

The quantum walk operates on a T-dimensional Hilbert space spanned by $\{|r\rangle \cup |x\rangle \; x \in \{1, ..., T-1\}\}$ where r is root. Walk starts in state $|r\rangle$ and is based on diffusion operators D_x.

Quantum walk in a tree

The quantum walk operates on a T-dimensional Hilbert space spanned by $\{|r\rangle \cup |x\rangle\, x \in \{1, ..., T-1\}\}$ where r is root. Walk starts in state $|r\rangle$ and is based on diffusion operators D_x.

- If x is valid solution, then D_x is identity.

Quantum walk in a tree

The quantum walk operates on a T-dimensional Hilbert space spanned by $\{|r\rangle \cup |x\rangle \; x \in \{1, ..., T-1\}\}$ where r is root. Walk starts in state $|r\rangle$ and is based on diffusion operators D_x.

- If x is valid solution, then D_x is identity.
- If x is indeterminate, and $x \neq r$ then $D_x = \mathbb{I} - 2 |\psi_x\rangle \langle\psi_x|$

$$|\psi_x\rangle \propto \left(|x\rangle + \sum_{y, x \rightarrow y} |y\rangle \right)$$

Quantum walk in a tree

The quantum walk operates on a T-dimensional Hilbert space spanned by $\{|r\rangle \cup |x\rangle \; x \in \{1, ..., T-1\}\}$ where r is root. Walk starts in state $|r\rangle$ and is based on diffusion operators D_x.

- If x is valid solution, then D_x is identity.
- If x is indeterminate, and $x \neq r$ then $D_x = \mathbb{I} - 2\,|\psi_x\rangle\,\langle\psi_x|$

$$|\psi_x\rangle \propto \left(|x\rangle + \sum_{y, x \to y} |y\rangle\right)$$

- For the root, $D_r = \mathbb{I} - 2\,|\psi_r\rangle\,\langle\psi_r|$

$$|\psi_r\rangle \propto \left(|r\rangle + \sqrt{n} \sum_{y, r \to y} |y\rangle\right)$$

TU Delft

Quantum walk in a tree

Quantum walk in a tree

Let A and B be the set of vertices even and odd distance from root, respectively.

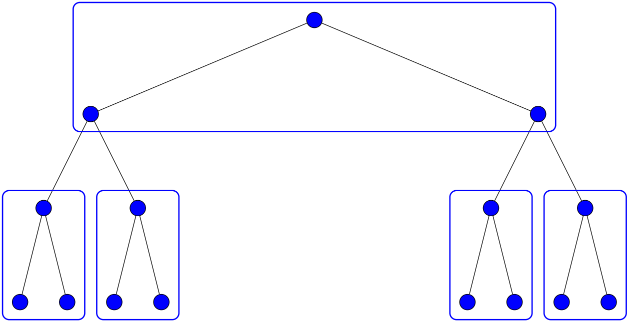
Quantum walk in a tree

Let A and B be the set of vertices even and odd distance from root, respectively.
A step of the walk is applying $R_B R_A$ where $R_A = \oplus_{x \in A} D_x$ and $R_B = |r\rangle\langle r| + \oplus_{x \in B} D_x$

Quantum walk in a tree

Let A and B be the set of vertices even and odd distance from root, respectively.

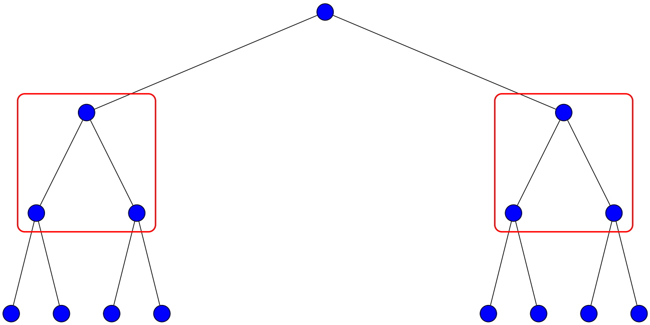
A step of the walk is applying $R_B R_A$ where $R_A = \oplus_{x \in A} D_x$ and $R_B = |r\rangle\langle r| + \oplus_{x \in B} D_x$



Quantum walk in a tree

Let A and B be the set of vertices even and odd distance from root, respectively.

A step of the walk is applying $R_B R_A$ where $R_A = \oplus_{x \in A} D_x$ and $R_B = |r\rangle\langle r| + \oplus_{x \in B} D_x$



Quantum Algorithm to detect solution

Quantum Algorithm to detect solution

The algorithm (Montanaro, 2015)

Apply phase estimation to $R_B R_A$ on state $|r\rangle$ with precision $\mathcal{O}\left(\frac{1}{\sqrt{Tn}}\right)$.

Quantum Algorithm to detect solution

The algorithm (Montanaro, 2015)

Apply phase estimation to $R_B R_A$ on state $|r\rangle$ with precision $\mathcal{O}\left(\frac{1}{\sqrt{Tn}}\right)$.

If there is a solution, $R_B R_A$ has an eigenvector with eigenvalue 1 and overlap $\geq \frac{1}{2}$ with $|r\rangle$.

Quantum Algorithm to detect solution

The algorithm (Montanaro, 2015)

Apply phase estimation to $R_B R_A$ on state $|r\rangle$ with precision $\mathcal{O}\left(\frac{1}{\sqrt{Tn}}\right)$.

If there is a solution, $R_B R_A$ has an eigenvector with eigenvalue 1 and overlap $\geq \frac{1}{2}$ with $|r\rangle$.

If there is no solution, $||P_\chi |r\rangle||^2 \leq \frac{1}{4}$ where P_χ is the projector onto the subspace spanned by eigenvectors of $R_B R_A$ with eigenvalue $e^{2i\theta}$ for $\theta \leq \frac{1}{2\sqrt{Tn}}$

Quantum Algorithm to detect solution

The algorithm (Montanaro, 2015)

Apply phase estimation to $R_B R_A$ on state $|r\rangle$ with precision $\mathcal{O}\left(\frac{1}{\sqrt{Tn}}\right)$.

If there is a solution, $R_B R_A$ has an eigenvector with eigenvalue 1 and overlap $\geq \frac{1}{2}$ with $|r\rangle$.

If there is no solution, $||P_\chi |r\rangle||^2 \leq \frac{1}{4}$ where P_χ is the projector onto the subspace spanned by eigenvectors of $R_B R_A$ with eigenvalue $e^{2i\theta}$ for $\theta \leq \frac{1}{2\sqrt{Tn}}$

Use the subroutine K times and if the measured eigenvalue is 1 at least $\frac{3K}{8}$ times, a solution exists.

Detection to search

Detection to search

- Can use this detection procedure as a subroutine to find the solutions in the tree via binary search.

TU Delft

Detection to search

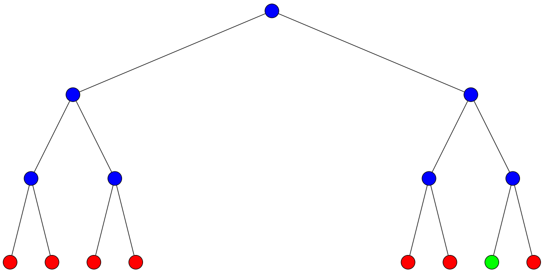
- Can use this detection procedure as a subroutine to find the solutions in the tree via binary search.
- Apply the procedure to the whole tree.

Detection to search

- Can use this detection procedure as a subroutine to find the solutions in the tree via binary search.
- Apply the procedure to the whole tree. If it detects a solution, apply it to the subtree rooted at each of the children, and repeat.

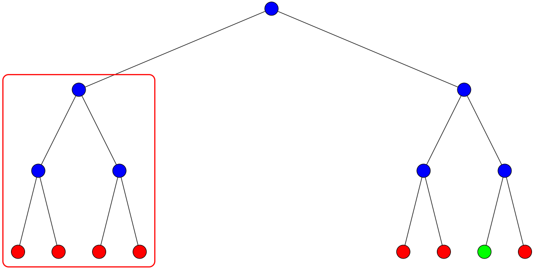
Detection to search

- Can use this detection procedure as a subroutine to find the solutions in the tree via binary search.
- Apply the procedure to the whole tree. If it detects a solution, apply it to the subtree rooted at each of the children, and repeat.



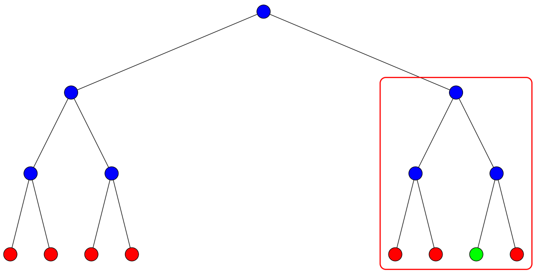
Detection to search

- Can use this detection procedure as a subroutine to find the solutions in the tree via binary search.
- Apply the procedure to the whole tree. If it detects a solution, apply it to the subtree rooted at each of the children, and repeat.



Detection to search

- Can use this detection procedure as a subroutine to find the solutions in the tree via binary search.
- Apply the procedure to the whole tree. If it detects a solution, apply it to the subtree rooted at each of the children, and repeat.



TUDelft

Optimizations

We propose a practical implementation of the method and further optimize both space and depth.

Optimizations

We propose a practical implementation of the method and further optimize both space and depth.

Depth optimization

Optimizations

We propose a practical implementation of the method and further optimize both space and depth.

Depth optimization

Eliminate dynamic heuristic h and fix the order of qubit assignment.

Optimizations

We propose a practical implementation of the method and further optimize both space and depth.

Depth optimization

Eliminate dynamic heuristic h and fix the order of qubit assignment.
Heuristic determines the shape of the tree.

Optimizations

We propose a practical implementation of the method and further optimize both space and depth.

Depth optimization

Eliminate dynamic heuristic h and fix the order of qubit assignment.
Heuristic determines the shape of the tree.
Potentially a more efficient implementation of predicate by checking fewer constraints.

Optimizations

We propose a practical implementation of the method and further optimize both space and depth.

Depth optimization

Eliminate dynamic heuristic h and fix the order of qubit assignment.
Heuristic determines the shape of the tree.
Potentially a more efficient implementation of predicate by checking fewer constraints.

More efficient encoding

Optimizations

We propose a practical implementation of the method and further optimize both space and depth.

Depth optimization

Eliminate dynamic heuristic h and fix the order of qubit assignment.
Heuristic determines the shape of the tree.
Potentially a more efficient implementation of predicate by checking fewer constraints.

More efficient encoding

We propose a more efficient encoding of the partial assignment

Optimizations

We propose a practical implementation of the method and further optimize both space and depth.

Depth optimization

Eliminate dynamic heuristic h and fix the order of qubit assignment.
Heuristic determines the shape of the tree.
Potentially a more efficient implementation of predicate by checking fewer constraints.

More efficient encoding

We propose a more efficient encoding of the partial assignment $x_1 = v_1, ..., x_l = v_l \rightarrow |l\rangle\,|v_1\rangle\,...\,|v_l\rangle\,|*\rangle\,...\,|*\rangle$

Optimizations

We propose a practical implementation of the method and further optimize both space and depth.

Depth optimization

Eliminate dynamic heuristic h and fix the order of qubit assignment.
Heuristic determines the shape of the tree.
Potentially a more efficient implementation of predicate by checking fewer constraints.

More efficient encoding

We propose a more efficient encoding of the partial assignment $x_1 = v_1, ..., x_l = v_l \rightarrow |l\rangle\,|v_1\rangle \ldots |v_l\rangle\,|*\rangle \ldots |*\rangle$
Saves $nlog(n)$ qubits.

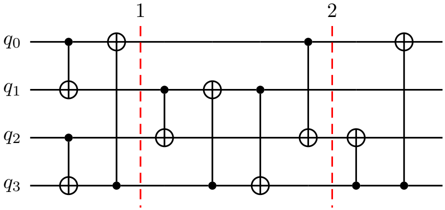
TUDelft

Slicing the circuit

What if the detection algorithm returns "no solution exists"?

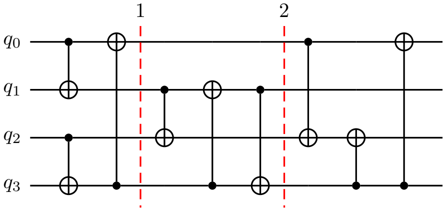
Slicing the circuit

What if the detection algorithm returns "no solution exists"?



Slicing the circuit

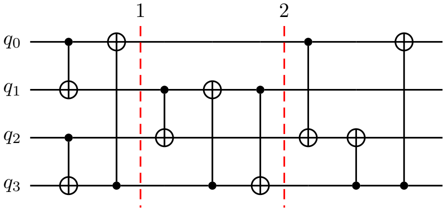
What if the detection algorithm returns "no solution exists"?



Need to make the "slice" smaller.

Slicing the circuit

What if the detection algorithm returns "no solution exists"?



Need to make the "slice" smaller.
Proposed guess for size of slice $= d_G \cdot |G_2|$
$d_G = \frac{\text{no. of edges}}{\text{no. of possible edges}}$, $|G_2| =$ no. of 2 qubit gates

Optimizing Routing

Optimizing Routing

What is the metric to be optimized?

Optimizing Routing

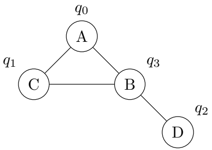
What is the metric to be optimized?
Distance measure between two mappings π_1 and π_2:

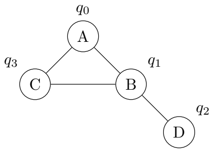
$$d_{\text{map}}(\pi_1, \pi_2) = \sum_{\alpha \in Z} d_{\text{chip}}(\pi_1^{-1}(\alpha), \pi_2^{-1}(\alpha))$$

Optimizing Routing

What is the metric to be optimized?
Distance measure between two mappings π_1 and π_2:

$$d_{\text{map}}(\pi_1, \pi_2) = \sum_{\alpha \in Z} d_{\text{chip}}(\pi_1^{-1}(\alpha), \pi_2^{-1}(\alpha))$$

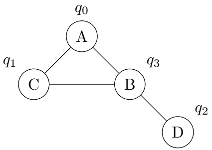


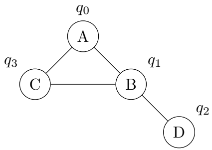


Optimizing Routing

What is the metric to be optimized?
Distance measure between two mappings π_1 and π_2:

$$d_{\text{map}}(\pi_1, \pi_2) = \sum_{\alpha \in Z} d_{\text{chip}}(\pi_1^{-1}(\alpha), \pi_2^{-1}(\alpha))$$



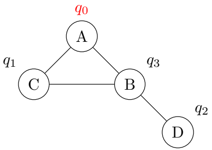


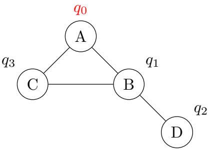
$$\begin{aligned} d_{\text{map}}(\pi_1, \pi_2) = & d_c(\pi_1^{-1}(q_0), \pi_2^{-1}(q_0)) \quad + d_c(\pi_1^{-1}(q_1), \pi_2^{-1}(q_1)) \\ & + d_c(\pi_1^{-1}(q_2), \pi_2^{-1}(q_2)) \quad + d_c(\pi_1^{-1}(q_3), \pi_2^{-1}(q_3)) \end{aligned}$$

Optimizing Routing

What is the metric to be optimized?
Distance measure between two mappings π_1 and π_2:

$$d_{\text{map}}(\pi_1, \pi_2) = \sum_{\alpha \in Z} d_{\text{chip}}(\pi_1^{-1}(\alpha), \pi_2^{-1}(\alpha))$$



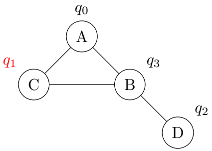


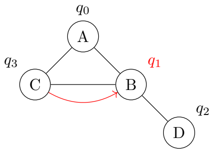
$$\begin{aligned} d_{\text{map}}(\pi_1, \pi_2) = \quad & 0 & & + d_c(\pi_1^{-1}(q_1), \pi_2^{-1}(q_1)) \\ & + d_c(\pi_1^{-1}(q_2), \pi_2^{-1}(q_2)) & & + d_c(\pi_1^{-1}(q_3), \pi_2^{-1}(q_3)) \end{aligned}$$

Optimizing Routing

What is the metric to be optimized?
Distance measure between two mappings π_1 and π_2:

$$d_{\text{map}}(\pi_1, \pi_2) = \sum_{\alpha \in Z} d_{\text{chip}}(\pi_1^{-1}(\alpha), \pi_2^{-1}(\alpha))$$





$$d_{\text{map}}(\pi_1, \pi_2) = 0 + 1$$

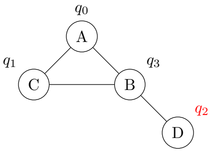
$$+d_c(\pi_1^{-1}(q_2), \pi_2^{-1}(q_2)) \quad +d_c(\pi_1^{-1}(q_3), \pi_2^{-1}(q_3))$$

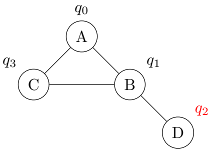
TU Delft

Optimizing Routing

What is the metric to be optimized?
Distance measure between two mappings π_1 and π_2:

$$d_{\mathsf{map}}(\pi_1, \pi_2) = \sum_{\alpha \in Z} d_{\mathsf{chip}}(\pi_1^{-1}(\alpha), \pi_2^{-1}(\alpha))$$





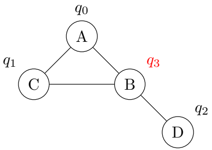
$$\begin{aligned} d_{\mathsf{map}}(\pi_1, \pi_2) = &\quad 0 \quad + \quad 1 \\ + &\quad 0 \quad + d_{\mathsf{c}}(\pi_1^{-1}(q_3), \pi_2^{-1}(q_3)) \end{aligned}$$

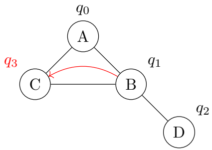
TU Delft

Optimizing Routing

What is the metric to be optimized?
Distance measure between two mappings π_1 and π_2:

$$d_{\text{map}}(\pi_1, \pi_2) = \sum_{\alpha \in Z} d_{\text{chip}}(\pi_1^{-1}(\alpha), \pi_2^{-1}(\alpha))$$





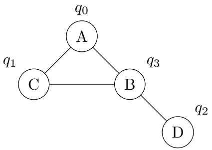
$$\begin{aligned} d_{\text{map}}(\pi_1, \pi_2) = & \quad 0 \quad + \quad 1 \\ + & \quad 0 \quad + \quad 1 \end{aligned}$$

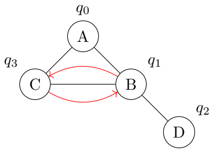
TU Delft

Optimizing Routing

What is the metric to be optimized?
Distance measure between two mappings π_1 and π_2:

$$d_{\text{map}}(\pi_1, \pi_2) = \sum_{\alpha \in Z} d_{\text{chip}}(\pi_1^{-1}(\alpha), \pi_2^{-1}(\alpha))$$





$$d_{\text{map}}(\pi_1, \pi_2) = 2$$

Optimizing Routing

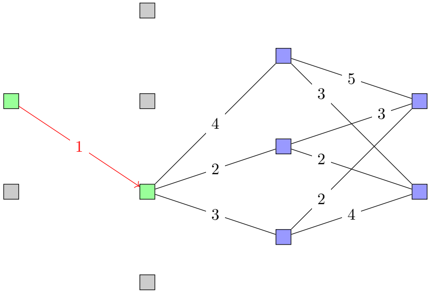
How to get the optimum "route"?

Optimizing Routing

How to get the optimum "route"?
Employ a simple greedy approach!

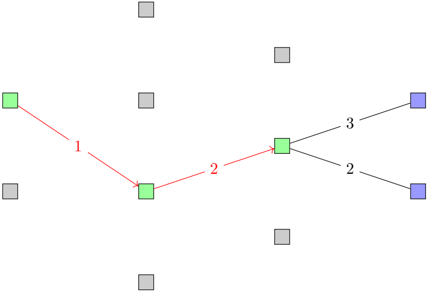
Optimizing Routing

How to get the optimum "route"?
Employ a simple greedy approach!



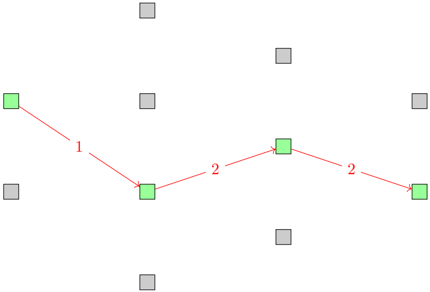
Optimizing Routing

How to get the optimum "route"?
Employ a simple greedy approach!



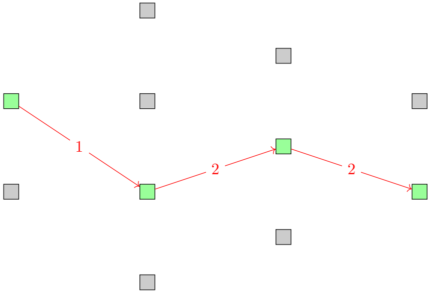
Optimizing Routing

How to get the optimum "route"?
Employ a simple greedy approach!



Optimizing Routing

How to get the optimum "route"?
Employ a simple greedy approach!



Potentially miss global optimum but we avoid combinatorial explosion.

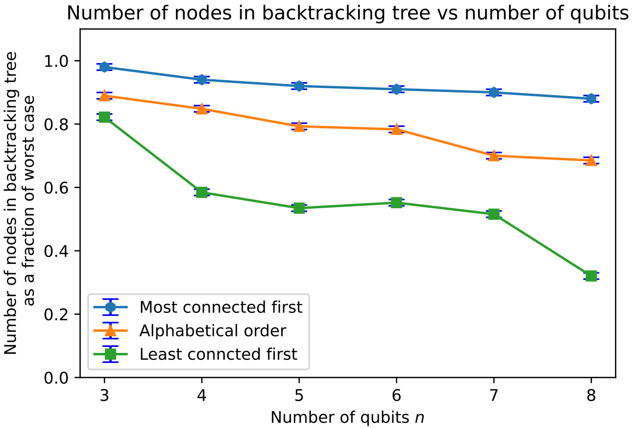
Outline

TU Delft

Results

Results

- Effect of choice of heuristic h:



Results

Benchmarking

Results

Benchmarking

Performance benchmarked against classical mappers for actual circuits (RevLib[a])

Results

Benchmarking

Performance benchmarked against classical mappers for actual circuits (RevLib[a])

Analysed the number of SWAPs needed.

Best performance matches QMAP[b] for small circuits (5 qubits, 30 CNOTs).

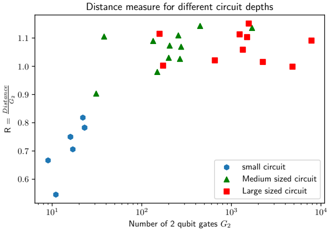
Results

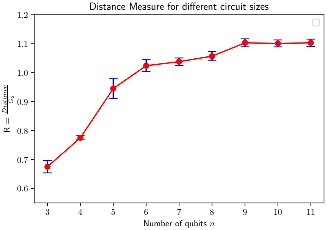
Benchmarking

Performance benchmarked against classical mappers for actual circuits (RevLib[a])
Analysed the number of SWAPs needed.
Best performance matches QMAP[b] for small circuits (5 qubits, 30 CNOTs).
Performance detoriates for bigger circuits, upto 80% more SWAPs needed.

[a] Wille, 2008
[b] Lao et. al., 2022

Results


Distance measure for different circuit depths
$R = \frac{Distance}{G_2}$
Number of 2 qubit gates G_2
small circuit
Medium sized circuit
Large sized circuit


Distance Measure for different circuit sizes
$R = \frac{Distance}{G_2}$
Number of qubits n

Outline

Conclusions

Conclusions

- Proposed a new model for qubit mapping based on CSP and combinatorial optimization.

Conclusions

- Proposed a new model for qubit mapping based on CSP and combinatorial optimization.
- Provided a practical implementation of a quantum algorithm for (initial) qubit mapping.

Conclusions

- Proposed a new model for qubit mapping based on CSP and combinatorial optimization.
- Provided a practical implementation of a quantum algorithm for (initial) qubit mapping.
- Further, optimized both the space and depth by eliminating dynamic heuristic and using a more efficient encoding.

Conclusions

- Proposed a new model for qubit mapping based on CSP and combinatorial optimization.
- Provided a practical implementation of a quantum algorithm for (initial) qubit mapping.
- Further, optimized both the space and depth by eliminating dynamic heuristic and using a more efficient encoding.
- Benchmarked and compared the depth overhead to other classical mapping and routing algorithms.

Thanks!