

# A Review on Quantum Circuit Optimization using ZX-Calculus

**Tobias Fischbach**, Pierre Talbot, Pascal Bouvry

02.02.2026

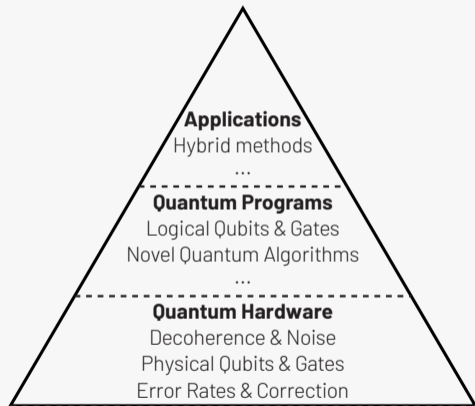


UNIVERSITY OF LUXEMBOURG  
Institute for Advanced Studies

# Quantum computing

Current challenges in quantum computing

- **Resource restrictions (e.g., Superconducting Architectures):**
  - 127 logical qubits [1]
  - Up to  $\approx 5000$  logical gates
  - Short coherence time ( $80[\mu\text{s}]$  to  $1[\text{ms}]$ ) [2]
- **Error correction:**
  - Noise drives gate error rate [3]
  - Limits the number of usable gates
  - Overhead vary by order of magnitude for different gates [4]
- **Quantum computing limited to artificial problems**

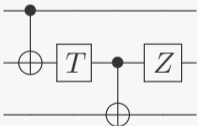


# Quantum Computing

Current Challenges in Quantum Computing

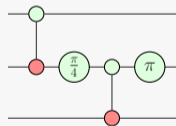
## QC Optimization

- Infinite universal gate sets
- Infinite gate commutation rules
- Equivalence verification computational expensive



## ZX Calculus [5, 6]

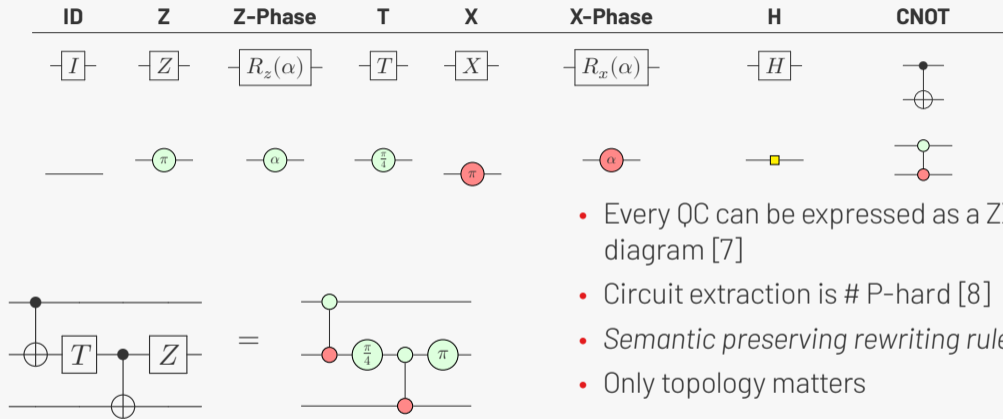
- 8 generators
- 9 well defined rewriting rules
- Rewriting rules preserve semantics



⇒ **State-space for combinatorial optimization is (infinitely) large**

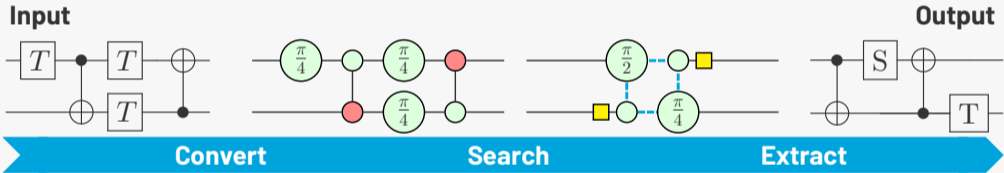
# ZX calculus

From Quantum Circuits to ZX-Diagrams

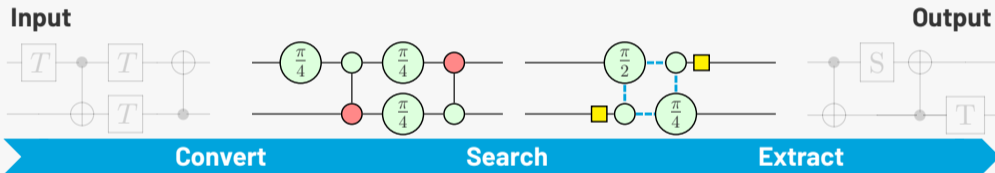


- Every QC can be expressed as a ZX diagram [7]
- Circuit extraction is # P-hard [8]
- *Semantic preserving rewriting rules*
- Only topology matters

# ZX-based QC Optimization Pipeline



# ZX-Diagram Optimization



# Definition of ZX-Diagram Optimization

Quantum Circuit Optimization

- Input & output quantum circuit
  - Let **QC** be the set of quantum circuits
- *opt* is a function that maps a quantum circuit to a metric
  - $opt : \mathbf{QC} \rightarrow \mathbb{Z}$

Define quantum circuits

# Definition of ZX-Diagram Optimization

## Quantum Circuit Optimization

- Input & output quantum circuit
  - Let **QC** be the set of quantum circuits
- *opt* is a function that maps a quantum circuit to a metric
  - $opt : \mathbf{QC} \rightarrow \mathbb{Z}$

Define quantum circuits

- Linear map of qubits
  - Let **LM** be the set of the linear map of qubits
- $\gamma$  is a function that maps a quantum circuit to its linear map
  - $\gamma : \mathbf{QC} \rightarrow \mathbf{LM}$

Quantum circuits  
implement a  
linear map of qubits

# Definition of ZX-Diagram Optimization

## Quantum Circuit Optimization

- Input & output quantum circuit
  - Let **QC** be the set of quantum circuits
- $opt$  is a function that maps a quantum circuit to a metric
  - $opt : \mathbf{QC} \rightarrow \mathbb{Z}$

Define quantum circuits

- Linear map of qubits
  - Let **LM** be the set of the linear map of qubits
- $\gamma$  is a function that maps a quantum circuit to its linear map
  - $\gamma : \mathbf{QC} \rightarrow \mathbf{LM}$

Quantum circuits implement a linear map of qubits

- $f$  is a function to transform quantum circuits
  - $f : \mathbf{QC} \rightarrow \mathbf{QC}$
  - $f$  is semantic-preserving  $\forall q \in \mathbf{QC}, \gamma(q) = \gamma(f(q))$
  - $f$  minimizes the objective function  $\min opt$

Functions that transform quantum circuits need to **keep the same** linear map of qubits

$\Rightarrow f$  is a quantum circuit optimization algorithm

## Definition of ZX-Diagram Optimization

ZX-based Quantum Circuit Optimization

- ZX diagrams
  - Let **ZX** be the infinite set of all finite ZX diagrams
- $\alpha$  is a function that converts quantum circuits to ZX diagrams
  - $\alpha : \mathbf{QC} \rightarrow \mathbf{ZX}$

**Every**  
quantum circuit  
can be converted  
to a ZX diagram

# Definition of ZX-Diagram Optimization

ZX-based Quantum Circuit Optimization

- ZX diagrams
  - Let **ZX** be the infinite set of all finite ZX diagrams
- $\alpha$  is a function that converts quantum circuits to ZX diagrams
  - $\alpha : \mathbf{QC} \rightarrow \mathbf{ZX}$
- *extract* is a function that converts ZX diagrams to quantum circuits
  - $extract : \mathbf{ZX} \rightarrow \mathbf{QC} \cup \{\perp\}$
  - A ZX diagram is extractable if  $zx \in \mathbf{ZX}, extract(zx) \neq \perp$

**Every**  
quantum circuit  
can be converted  
to a ZX diagram

**Not every**  
ZX diagram can  
be extracted

# Definition of ZX-Diagram Optimization

## ZX-based Quantum Circuit Optimization

- ZX diagrams
  - Let **ZX** be the infinite set of all finite ZX diagrams
- $\alpha$  is a function that converts quantum circuits to ZX diagrams
  - $\alpha : \mathbf{QC} \rightarrow \mathbf{ZX}$

- *extract* is a function that converts ZX diagrams to quantum circuits
  - $extract : \mathbf{ZX} \rightarrow \mathbf{QC} \cup \{\perp\}$
  - A ZX diagram is extractable if  $zx \in \mathbf{ZX}, extract(zx) \neq \perp$

- Rewriting rules
  - Let  $\mathbf{R} = \{i1, i2, f, h, b, c, \pi, hd, lc\}$  be the set of rewriting rules
- $r \in R$  is a function that transforms a ZX diagram
  - $r \in R : \mathbf{ZX} \rightarrow \mathbf{ZX}$

**Every**

quantum circuit  
can be converted  
to a ZX diagram

**Not every**

ZX diagram can  
be extracted

Rules are functions  
that transform  
ZX diagrams

⇒ A ZX based QC algorithm searches for extractable ZX diagrams that improves QC metric(s)

## Definition of ZX-Diagram Optimization

ZX-based Quantum Circuit Optimization

State-space formed by finite sequence  
of rules on converted quantum circuit

- $W$  is the ZX state-space
  - $q \in QC, W \subseteq \mathbf{ZX}$  s.t.  $w \in W$  exists for a finite sequence of rewriting rules
  - $w = (r_n \circ \dots \circ r_1 \circ \alpha)(q)$
- $w$  is a function that converts a QC to a ZX diagram and applies a finite sequence of rewriting rules
  - $w = (r_n \circ \dots \circ r_1 \circ \alpha)(q)$

## Definition of ZX-Diagram Optimization

ZX-based Quantum Circuit Optimization

State-space formed by finite sequence of rules on converted quantum circuit

- $W$  is the ZX state-space
  - $q \in QC, W \subseteq \mathbf{ZX}$  s.t.  $w \in W$  exists for a finite sequence of rewriting rules
  - $w = (r_n \circ \dots \circ r_1 \circ \alpha)(q)$
- $w$  is a function that converts a QC to a ZX diagram and applies a finite sequence of rewriting rules
  - $w = (r_n \circ \dots \circ r_1 \circ \alpha)(q)$
- Solutions are all extractable ZX diagrams in the state-space  $W$ :
  - $S \subseteq W$
  - $\forall w \in W, \text{extract}(w) \neq \perp \Leftrightarrow w \in S$

Extractable ZX diagrams of the state-space are solutions

## Definition of ZX-Diagram Optimization

ZX-based Quantum Circuit Optimization

State-space formed by finite sequence of rules on converted quantum circuit

- $W$  is the ZX state-space
  - $q \in QC, W \subseteq \mathbf{ZX}$  s.t.  $w \in W$  exists for a finite sequence of rewriting rules
  - $w = (r_n \circ \dots \circ r_1 \circ \alpha)(q)$
- $w$  is a function that converts a QC to a ZX diagram and applies a finite sequence of rewriting rules
  - $w = (r_n \circ \dots \circ r_1 \circ \alpha)(q)$

- Solutions are all extractable ZX diagrams in the state-space  $W$ :
  - $S \subseteq W$
  - $\forall w \in W, \text{extract}(w) \neq \perp \Leftrightarrow w \in S$

Extractable ZX diagrams of the state-space are solutions

- **Optimal solutions** is the largest subset of solutions such that a metric is minimized
  - $O \subseteq S$
  - $\forall o \in O, s \in S, \text{opt}(\text{extract}(o)) \leq \text{opt}(\text{extract}(s))$

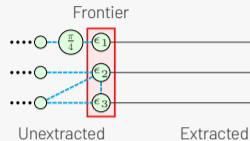
All solutions ZX diagrams that minimize a metric are optimal

# Circuit Extraction

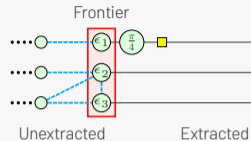


## Circuit Extraction

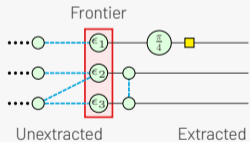
- Assertion of causality
- Circuit extraction is #P-hard [8]
- Necessitates the existence general [9] or causal [10] flow
- Polynomial-time algorithm exist for ZX-diagrams in the MBQC-form [11]



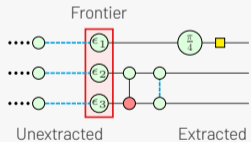
**(a)** Unextracted ZX-diagram.



**(b)** Spiders with a single input and output can be extracted directly.



**(c)** Connections between frontier spiders are extracted by CZ gates.



**(d)** CNOT gates extract wires between frontier and unextracted spiders.

# Methodology

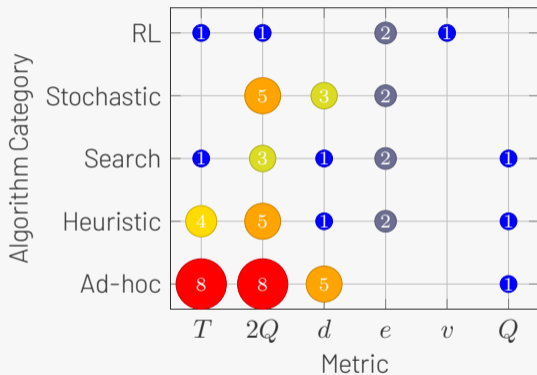
- 26 surveyed works until Sept. 2025
- Selection criteria:
  - work uses ZX-diagram rewriting for circuit optimization
  - work improves circuit extraction
- Categorized by:
  1. Architecture-independent (14) vs. architecture-aware (12)
  2. Optimization technique
  3. Target metrics
- Full survey available on arXiv:



Author(s)	Algorithm	T	2Q	Depth	Qubits	Edges	Vertices
Fagan et al. [12]	Ad-hoc		y	y			
Kissinger et al. [13]	Ad-hoc		y				
Beaudrap et al. [14]	Ad-hoc+Heuristic	y					
Beaudrap et al. [15]	Ad-hoc+Heuristic	y					
Duncan et al. [16]	Ad-hoc	y					
Kissinger et al. [17]	Ad-hoc	y					
Cowtan et al. [18]	Ad-hoc+Greedy		y				
Cowtan et al. [19]	Ad-hoc		y				
Munson et al. [20]	Ad-hoc+Heuristic	y					
Zilk et al. [21]	Ad-hoc	y	y	y			
Gogioso et al. [22]	SA		y				
Staudacher et al. [23]	Rand.+Heuristic	y				y	
Winderl et al. [24]	Ad-hoc+Heuristic	y					
Riu et al. [25]	RL	y				y	
Griend et al. [26]	Ad-hoc+Heuristic			y			
Vandaele [27]	Ad-hoc+DVFS+Greedy				y		
Holker [28]	Rand.+Heuristic	y				y	
Nägele et al. [29]	RL						y
Staudacher et al. [30]	Ad-hoc			y			
Ewen et al. [31]	GA		y	y			
Huang et al. [32]	Ad-hoc+Heuristic	y	y				
Mattick et al. [33]	RL+Tree search		y			y	
Fischbach et al. [34]	Tree search	y				y	
Liu et al. [35]	TM		y	y			
Chen et al. [36]	SA+LA		y	y			
Villoria et al. [37]	ILP+Peephole		y				



# Taxonomy



## Categories

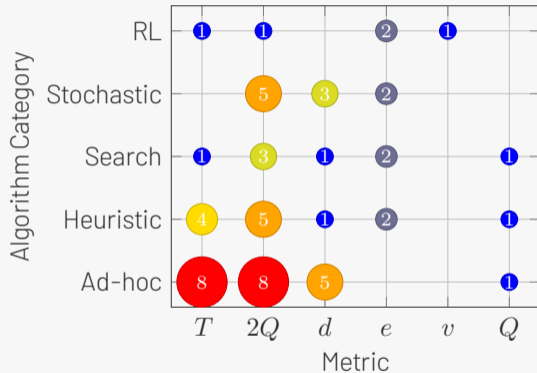
- *Ad-hoc*: {Ad-hoc, Template Matching}
- *Heuristic*: {Heuristic, Greedy}
- *Search*: {Integer Linear Programming, Tree Search, Look-ahead Search, Directed Vertex Feedback Solver}
- *Stochastic*: {Simulated Annealing, Genetic Algorithm, Random Algorithm}
- *RL*: {Reinforcement Learning}

## Notable Developments

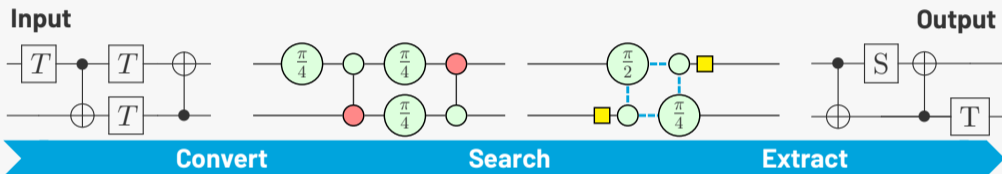
- From ad-hoc towards stochastic and search algorithms (e.g., genetic [31], simulated annealing [22], and tree search [33])
- ZX-Calculus as an intermediate representation for architecture-aware optimization (e.g., photonic [21], neutral atom [30], trapped ion [37], and superconducting [13] architectures)
- Edge reduction heuristic to optimize two-qubit gate count (first introduced by Staudacher et al. [23], improved by Holker [28], and used by Mattick et al. [33])
- Architecture-aware optimization during circuit extraction based on the biadjacency matrix (e.g., [13], [37])

# Challenges

- Scalability
- Multi-objective optimization
- Circuit extraction
- Architecture-awareness



# What is the performance of Algorithm XY compared to Algorithm YZ?



# ZX Benchmarking Framework

## Goals

- Declarative experiments (decouple optimizer and extractor)
- Representative benchmark (support of multiple data sets)
- CLI and PyZX & Qiskit integration
- Public website with up to date results
- *No reimplementations of existing algorithms*
- *Maintainable*

## GitLab Repository



[gitlab.com/NetForceExplorer/zx-benchmark](https://gitlab.com/NetForceExplorer/zx-benchmark)

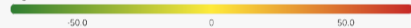
## Current Features

- Configuration files for declarative experiments
- Separation of optimizer and extractor
- HTML and LaTeX analysis output
- Works on standard data sets (e.g., QASMBench) and random circuits
- Implemented optimizers (joint work with Fabio Perreira):
  - Holker [28]
  - Staudacher et al. [23]
  - Ewen et al. [31]
  - Kissinger et al. [17]
  - Fischbach et al. [34]

### Pruning Condition: Cflow

Algorithm	DFS		FR		IDDFS		LDS		LE	
	2Q	T	2Q	T	2Q	T	2Q	T	2Q	T
<b>Small</b>	0.00	0.00	45.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
<b>Medium</b>	0.00	-28.57	10.71	-33.33	0.00	-28.57	0.00	-28.57	0.00	-28.57
<b>Large</b>	70.26	-42.86	50.26	-42.86	64.10	-42.86	67.69	-42.86	70.26	-42.86
<b>full</b>	39.19	-25.60	38.60	-27.38	50.00	-27.38	48.66	-27.38	39.19	-25.60
<b>Median</b>	19.59	-27.08	41.80	-30.36	25.00	-27.98	24.33	-27.98	19.59	-27.08

Legend



Algorithm

LDS := Limited-discrepancy search  
LE := Local-elimination  
IDDFS := Iterative-deepening depth-first search  
DFS := Depth-first search  
FR := Full reduce

Metric

T := T-gate  
2Q := Two-qubit



**Figure:** Example HTML analysis.



# Configuration

```
1 python config.py -c config.json
2 # or
3 uv run -c config.json
4 # or
5 docker run -c config.json
6 # or
7 experiment -c config.json
```

- Global and instance parameters configuration (e.g., timeouts)
- Instances are constructed as the combination of all optimizers, extractors, and data sets
- Filenames and instance identifier from optional metadata
- Optimizer and extractor parameters are passed `**kwargs`

```
1 {
2   "experiment":{
3     "data_dir": "./data/example",
4     "b_write_result": true,
5     "quiet": true,
6     "instance_timeout": 3600,
7     "global_timeout": 170000
8   },
9   "setup":{
10    "solvers":{
11      "dfs-enumerate-ilps-no-constraints":{
12        "optimizer": "zx_bench.optimizers.dfs_optimizer"
13      },
14      "extractor": "zx_bench.extractors.enumerate_ilps_extractor",
15      "optimizer_name": "dfs",
16      "extractor_name": "enumerate-ilps-extractor",
17      "extractor_params": {"quiet": true, "max_sol": 2},
18      "optimizer_params": {"quiet": true, "metric": ["MetricEdge()"], "pruning": ["PruneNonCircuits()"]}
19    },
20    "qcs":{
21      "qasm_bench": {"category": "test-set"}
22    }
23  }
24 }
```

## Optimizer Integration

```
1 $ ls src/zx_bench/optimizers/holker_2024
2 graph
3 heuristics.py
4 __init__.py
5 rules.py
6 simplify.py
7 utils.py
```

- `__init__.py` contains the wrapping function that maps `BaseGraph` → `BaseGraph`
- `**kwargs` upstream optimizer arguments
- Translation between up to date and incompatible legacy PyZX if required

```
1 # __init__.py
2 def holker_2024(g: BaseGraph, **kwargs)
3     -> BaseGraph:
4     g_conv = pyzx_graph_to_holker_graph(g
5         )
6     g_opt = g_conv.copy()
7     simplify.flow_2Q_simp(g_opt, **kwargs
8         )
9     g_conv = holker_graph_to_pyzx_graph(
10         g_opt.copy())
11     return g_conv
```



## Planned Features

- Automatization with GitLab runners that:
  - execute benchmark for newly added optimizers
  - update public website
  - Check if upstream PyZX changes break implementation
- Integrate more optimizers from Table 11
- Public testing

## Discussion Points

- Would you create a pull request with your optimizer?
- Do you wish for more advanced metrics or analysis?
- In what advanced experiments are you interested? E.g.:
  - computational efficiency
  - circuit size dependencies
  - two-qubit and T-gate density
- How can we test architecture-aware optimizers and extractors?
- Are we missing something fundamental that would prevent you from using the framework?



# Looking for Postdoc Opportunities

## Contact

- Email: [fischbach.tobias-michael@outlook.com](mailto:fischbach.tobias-michael@outlook.com)
- CV: [tobias-fischbach.de/cv](https://tobias-fischbach.de/cv)



- Website: [tobias-fischbach.de](https://tobias-fischbach.de)



## Ressources

- Survey [38]:



- ZX Benchmarking Framework:



# Looking for Postdoc Opportunities

## Contact

- Email: [fischbach.tobias-michael@outlook.com](mailto:fischbach.tobias-michael@outlook.com)
- CV: [tobias-fischbach.de/cv](https://tobias-fischbach.de/cv)



## Ressources

- Survey [38]:



**Thanks for your Attention!**

- Website: [tobias-fischbach.de](https://tobias-fischbach.de)



## Discussion Points

- Would you create a pull request with your optimizer?
- Do you wish for more advanced metrics or analysis?
- In what advanced experiments are you interested? E.g.:
  - computational efficiency
  - circuit size dependencies
  - two-qubit and T-gate density
- How can we test architecture-aware optimizers and extractors?
- Are we missing something fundamental that would prevent you from using the framework?



## Bibliography I

- [1] **Eagle's Quantum Performance Progress | IBM Quantum Computing Blog.**  
<https://www.ibm.com/quantum/blog/eagle-quantum-processor-performance>.
- [2] A. Somoroff, Q. Ficheux, R. A. Mencia, H. Xiong, R. Kuzmin, and V. E. Manucharyan. **"Millisecond Coherence in a Superconducting Qubit"**. In: *Physical Review Letters* 130.26 (June 2023), p. 267001. DOI: 10.1103/PhysRevLett.130.267001.
- [3] A. M. Steane. **"Overhead and Noise Threshold of Fault-Tolerant Quantum Error Correction"**. In: *Physical Review A* 68.4 (Oct. 2003), p. 042322. DOI: 10.1103/PhysRevA.68.042322.
- [4] J. Roffe. **Quantum Error Correction: An Introductory Guide.** July 2019. DOI: 10.48550/arXiv.1907.11157.

## Bibliography II

- [5] B. Coecke and R. Duncan. **“Interacting Quantum Observables”**. In: *Automata, Languages and Programming*. Ed. by L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfssdóttir, and I. Walukiewicz. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008, pp. 298–310. DOI: 10.1007/978-3-540-70583-3\_25.
- [6] B. Coecke and R. Duncan. **“Interacting Quantum Observables: Categorical Algebra and Diagrammatics”**. In: *New Journal of Physics* 13.4 (Apr. 2011), p. 043016. DOI: 10.1088/1367-2630/13/4/043016.
- [7] J. van de Wetering. **ZX-calculus for the Working Quantum Computer Scientist**. Dec. 2020. DOI: 10.48550/arXiv.2012.13966.

## Bibliography III

- [8] N. de Beaudrap, A. Kissinger, and J. van de Wetering. **“Circuit Extraction for ZX-Diagrams Can Be #P-Hard”**. In: *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*. Ed. by M. Bojańczyk, E. Merelli, and D. P. Woodruff. Vol. 229. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 119:1–119:19. DOI: 10.4230/LIPIcs.ICALP.2022.119.
- [9] D. E. Browne, E. Kashefi, M. Mhalla, and S. Perdrix. **“Generalized flow and determinism in measurement-based quantum computation”**. In: *New Journal of Physics* 9.8 (Aug. 2007), p. 250. DOI: 10.1088/1367-2630/9/8/250.
- [10] V. Danos and E. Kashefi. **“Determinism in the one-way model”**. In: *Phys. Rev. A* 74 (5 Nov. 2006), p. 052310. DOI: 10.1103/PhysRevA.74.052310.
- [11] M. Backens, H. Miller-Bakewell, G. de Felice, L. Lobski, and J. van de Wetering. **“There and back again: A circuit extraction tale”**. In: *Quantum* 5 (Mar. 2021), p. 421. DOI: 10.22331/q-2021-03-25-421.

## Bibliography IV

- [12] A. Fagan and R. Duncan. **“Optimising Clifford Circuits with Qantomatic”**. In: *Electronic Proceedings in Theoretical Computer Science* 287 (Jan. 2019), pp. 85–105. DOI: 10.4204/EPTCS.287.5.
- [13] A. Kissinger and A. M.-v. de Griend. **CNOT circuit extraction for topologically-constrained quantum memories**. 2019.
- [14] N. de Beaudrap, X. Bian, and Q. Wang. **“Techniques to Reduce  $\pi/4$ -Parity-Phase Circuits, Motivated by the ZX Calculus”**. In: *Electronic Proceedings in Theoretical Computer Science* 318 (May 2020), pp. 131–149. DOI: 10.4204/eptcs.318.9.
- [15] N. de Beaudrap, X. Bian, and Q. Wang. **“Fast and Effective Techniques for T-Count Reduction via Spider Nest Identities”**. In: *15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020)*. Ed. by S. T. Flammia. Vol. 158. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020, 11:1–11:23. DOI: 10.4230/LIPIcs.TQC.2020.11.

## Bibliography V

- [16] R. Duncan, A. Kissinger, S. Perdrix, and J. van de Wetering. **“Graph-Theoretic Simplification of Quantum Circuits with the ZX-calculus”**. In: *Quantum* 4 (June 2020), p. 279. DOI: 10.22331/q-2020-06-04-279.
- [17] A. Kissinger and J. van de Wetering. **“Reducing the number of non-Clifford gates in quantum circuits”**. In: *Phys. Rev. A* 102 (2 Aug. 2020), p. 022406. DOI: 10.1103/PhysRevA.102.022406.
- [18] A. Cowtan, S. Dilkes, R. Duncan, W. Simmons, and S. Sivarajah. **“Phase Gadget Synthesis for Shallow Circuits”**. In: *Electronic Proceedings in Theoretical Computer Science* 318 (May 2020), pp. 213–228. DOI: 10.4204/EPTCS.318.13.
- [19] A. Cowtan, W. Simmons, and R. Duncan. **A Generic Compilation Strategy for the Unitary Coupled Cluster Ansatz**. 2020.
- [20] A. Munson, B. Coecke, and Q. Wang. **“AND-gates in ZX-calculus: Spider Nest Identities and QBC-completeness”**. In: *Electronic Proceedings in Theoretical Computer Science* 340 (Sept. 2021), pp. 230–255. DOI: 10.4204/eptcs.340.12.

## Bibliography VI

- [21] F. Zilk, K. Staudacher, T. Guggemos, K. Furlinger, D. Kranzlmüller, and P. Walther. **“A compiler for universal photonic quantum computers”**. In: *2022 IEEE/ACM Third International Workshop on Quantum Computing Software (QCS)*. 2022, pp. 57–67. DOI: 10.1109/QCS56647.2022.00012.
- [22] S. Gogioso and R. Yeung. **“Annealing Optimisation of Mixed ZX Phase Circuits”**. In: *Proceedings 19th International Conference on Quantum Physics and Logic, Wolfson College, Oxford, UK, 27 June - 1 July 2022*. Ed. by S. Gogioso and M. Hoban. Vol. 394. *Electronic Proceedings in Theoretical Computer Science*. Open Publishing Association, 2023, pp. 415–431. DOI: 10.4204/EPTCS.394.20.
- [23] K. Staudacher, T. Guggemos, S. Grundner-Culemann, and W. Gehrke. **“Reducing 2-Qubit Gate Count for ZX-Calculus Based Quantum Circuit Optimization”**. In: *Electronic Proceedings in Theoretical Computer Science*. Vol. 394. Nov. 2023, pp. 29–45. DOI: 10.4204/EPTCS.394.3.

## Bibliography VII

- [24] D. Winderl, Q. Huang, and C. B. Mendl. **“A Recursively Partitioned Approach to Architecture-Aware ZX Polynomial Synthesis and Optimization”**. In: *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, Sept. 2023, pp. 837–847. DOI: 10.1109/qce57702.2023.00098.
- [25] J. Riu, J. Nogué, G. Vilaplana, A. Garcia-Saez, and M. P. Estarellas. **“Reinforcement Learning Based Quantum Circuit Optimization via ZX-Calculus”**. In: *Quantum* 9 (May 2025), p. 1758. DOI: 10.22331/q-2025-05-28-1758.
- [26] A. van de Griend and R. Duncan. **“Architecture-Aware Synthesis of Phase Polynomials for NISQ Devices”**. In: *Electronic Proceedings in Theoretical Computer Science* 394 (Nov. 2023), pp. 116–140. DOI: 10.4204/eptcs.394.8.
- [27] V. Vandaele. ***Qubit-count optimization using ZX-calculus***. 2024.
- [28] C. Holker. ***Causal Flow Preserving Optimisation of Quantum Circuits in the ZX-calculus***. Jan. 2024. DOI: 10.48550/arXiv.2312.02793.

## Bibliography VIII

- [29] M. Nägele and F. Marquardt. **“Optimizing ZX-diagrams with deep reinforcement learning”**. In: *Machine Learning: Science and Technology* 5.3 (Sept. 2024), p. 035077. DOI: 10.1088/2632-2153/ad76f7.
- [30] K. Staudacher, L. Schmid, J. Zeiher, R. Wille, and D. Krausz Müller. **“Multi-controlled Phase Gate Synthesis with ZX-calculus applied to Neutral Atom Hardware”**. In: *Electronic Proceedings in Theoretical Computer Science* 406 (Aug. 2024), pp. 96–116. DOI: 10.4204/eptcs.406.5.
- [31] T. Ewen, I. Turkalj, P. Holzer, and M.-O. Wolf. **“Application of ZX-calculus to quantum architecture search”**. In: *Quantum Machine Intelligence* 7.1 (Mar. 2025). DOI: 10.1007/s42484-025-00264-6.
- [32] Q. Huang, D. Winderl, A. Meijer-van de Griend, and R. Yeung. **“Redefining Lexicographical Ordering: Optimizing Pauli String Decompositions for Quantum Compiling”**. In: *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Vol. 01. 2024, pp. 885–896. DOI: 10.1109/QCE60285.2024.00108.

## Bibliography IX

- [33] A. Mattick, M. Periyasamy, C. Ufrecht, A. Y. Dubey, C. Mutschler, A. Plinge, and D. D. Scherer. **Optimizing Quantum Circuits via ZX Diagrams using Reinforcement Learning and Graph Neural Networks**. 2025.
- [34] T. M. Fischbach, P. Talbot, and P. Bouvry. **“Exhaustive Search for Quantum Circuit Optimization Using ZX Calculus”**. In: *Optimization and Learning*. Ed. by B. Dorronsoro, E.-G. Talbi, D. Weraikat, and S. Bouamama. Cham: Springer Nature Switzerland, 2026, pp. 239–253.
- [35] Y. Liu, Z. Zhang, Y. Hu, F. Meng, T. Luan, X. Zhang, and X. Yu. **“Practical circuit optimization algorithm for quantum simulation based on template matching”**. en. In: *Quantum Inf. Process.* 23.2 (Feb. 2024). DOI: <https://doi.org/10.1007/s11128-023-04252-2>.
- [36] K. Chen, W. Liu, G. Xu, Y. Li, M. Li, and S. He. **Quantum Circuit Optimization Based on Dynamic Grouping and ZX-Calculus for Reducing 2-Qubit Gate Count**. 2025.

## Bibliography X

- [37] A. Villoria, H. Basold, and A. Laarman. **“Optimisation and synthesis of quantum circuits with global gates”**. In: *Quantum Science and Technology* 11.1 (Jan. 2026), p. 015040. DOI: 10.1088/2058-9565/ae3029.
- [38] T. M. Fischbach, P. Talbot, and P. Bouvry. **“A Review on Quantum Circuit Optimization using ZX-Calculus”**. In: *arXiv* (Sept. 2025). DOI: <https://doi.org/10.48550/arXiv.2509.20663>.