@agra_uni_bremen

Optimal SWAP Gate Insertion for Nearest Neighbor Quantum Circuits

Robert Wille[1,2], Aaron Lye[1], Rolf Drechsler[1,2]

[1] Institute of Computer Science
University of Bremen, Germany
[2] Cyber Physical Systems
DFKI GmbH Bremen, Germany

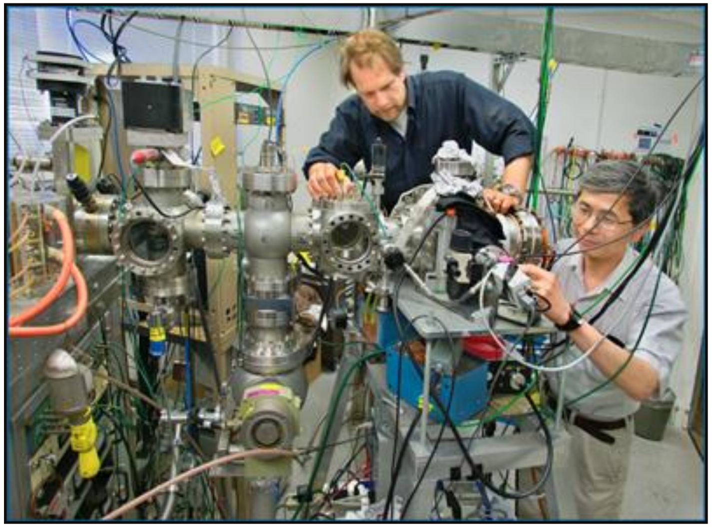
rwille@informatik.uni-bremen.de

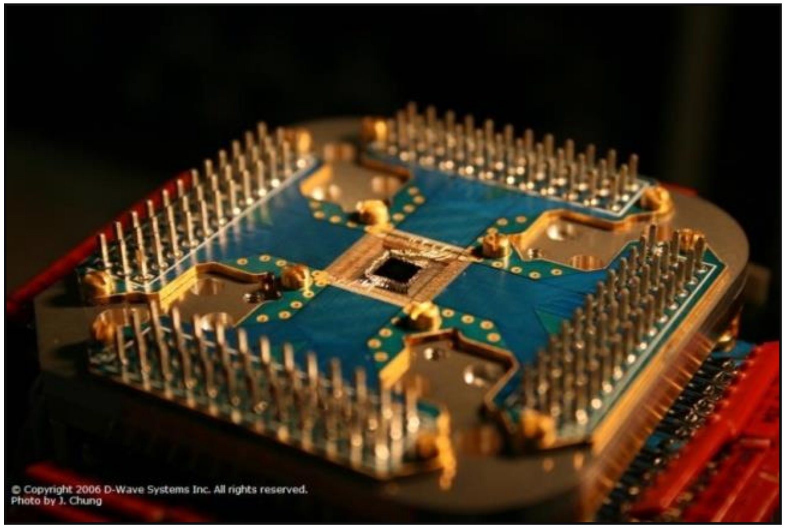
Outline

- (Motivation and Background)
- Nearest Neighbor Constraints
- Ensuring Nearest Neighbor Constraints through SWAP Gates
- Minimizing the Number of SWAP Gates
- Experimental Evaluation
- Conclusions

Quantum Circuits

- Computation not only with 0 and 1 but also superposition of both
- Enables significant speed-ups for certain problems (e.g. factorization, database search)





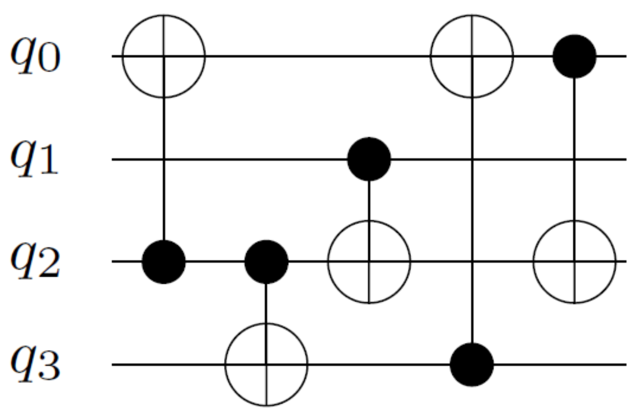
Quantum Circuits

- Cascade of quantum gates

Quantum Gates

- Cascade of quantum gates
- Realize unitary operations *U*
- If *c=0*: all states remain unchanged
- If *c=1*: *U* is applied to *t*

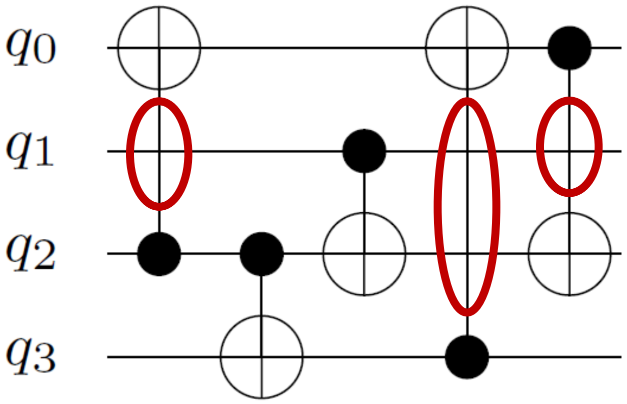
Control



Target line *t*

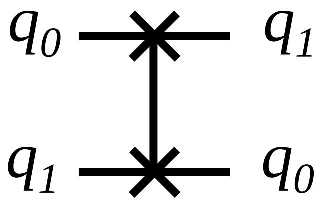
Nearest Neighbor Constraints

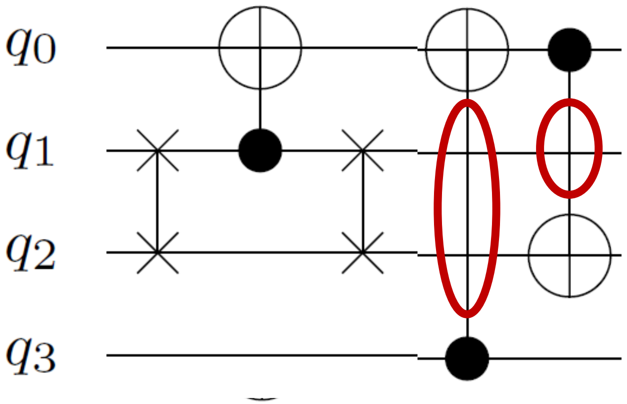
- Motivated by physical realizations
- Control and Target Lines need to be adjacent



Ensuring Nearest Neighbor

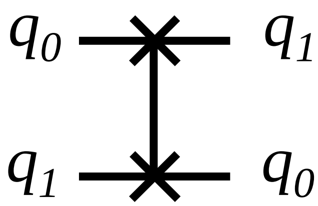
- Through SWAP gates

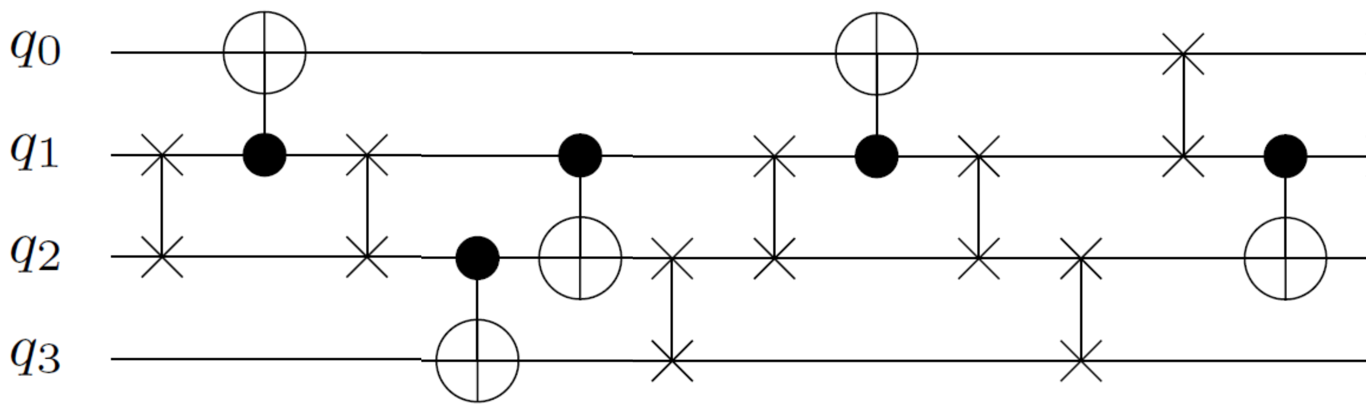




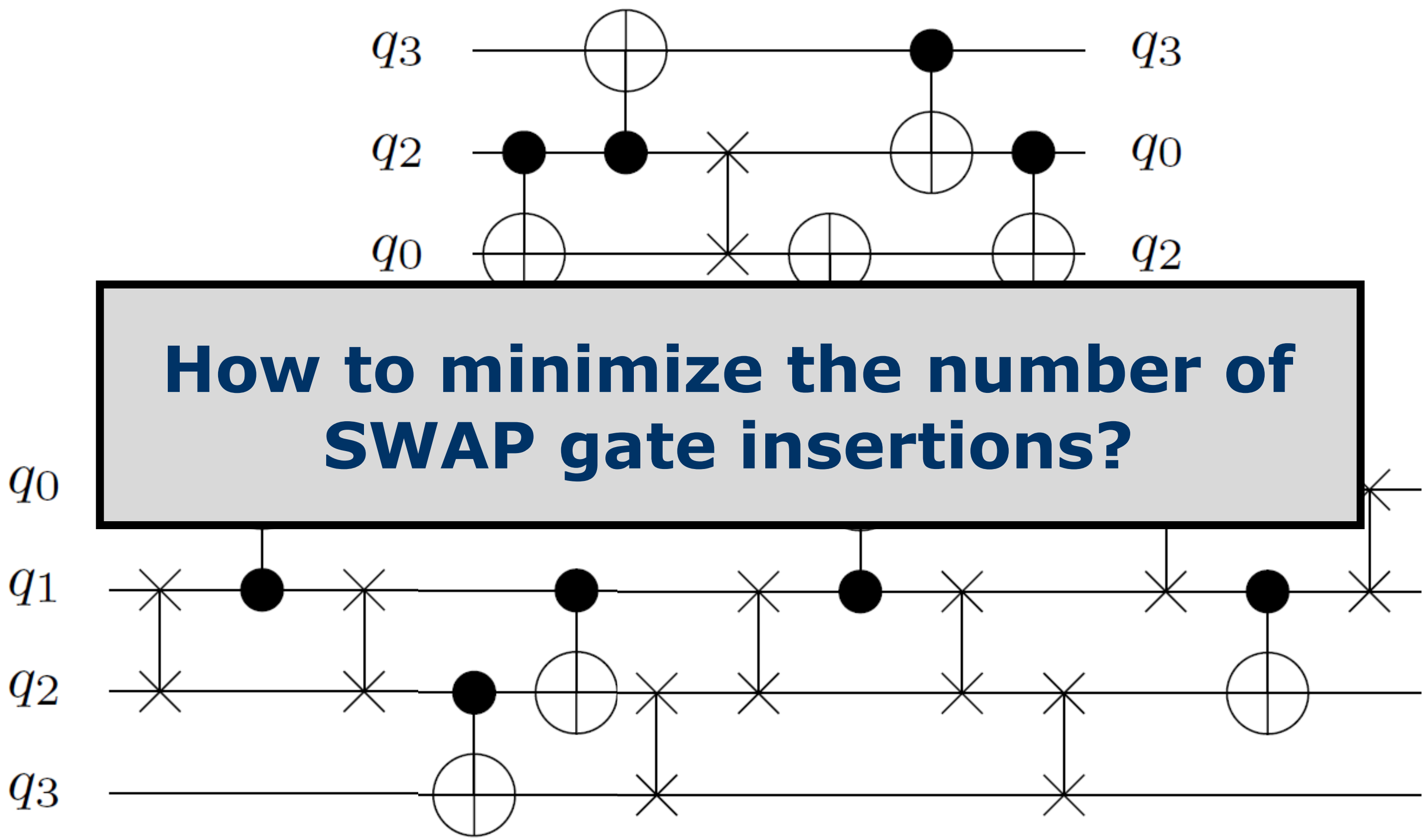
Ensuring Nearest Neighbor

- Through SWAP gates





Ensuring Nearest Neighbor



How to minimize the number of SWAP gate insertions?

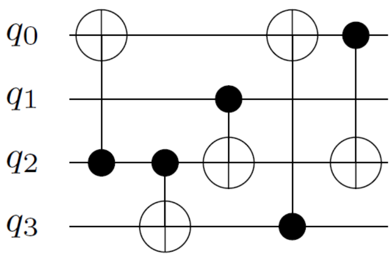
State-of-the-art

- Heuristic approaches
  - Re-ordering of circuit lines
  - Window-based schemes
  - Mapping the graph arrangement problem
  - …

  (Van Meter & Oskin, 2006; Mottonen & Vartiainen, 2006; Chakrabarti & Sur-Kolay, 2007; Khan, 2008; Saeedi, Wille & Drechsler, 2010; Shafaei, Saeedi & Pedram, 2013)

- Exact approaches
  - Enumerative
  - Through gate order changes

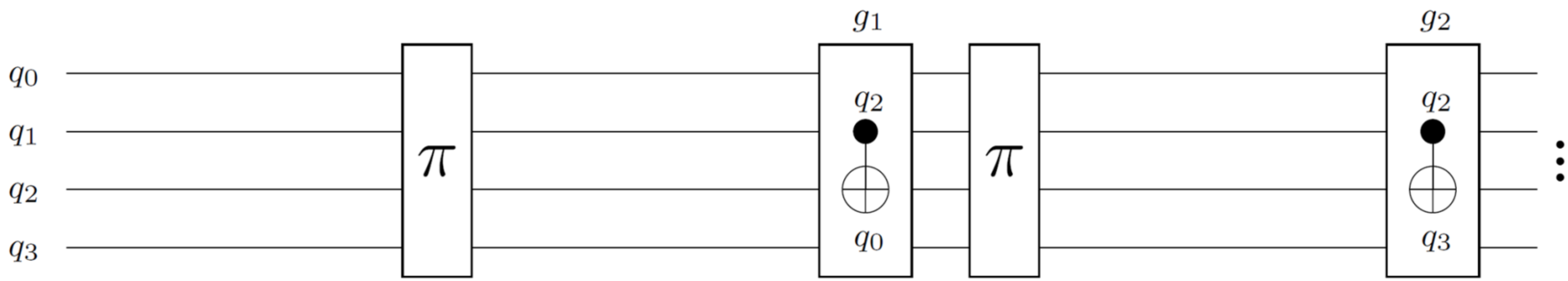
  (Hirata, Nakanishi, Yamashita & Nakashima, 2009; Matsuo & Yamashita, 2011)



General Idea

Consideration of

- all possible permutations before each gate

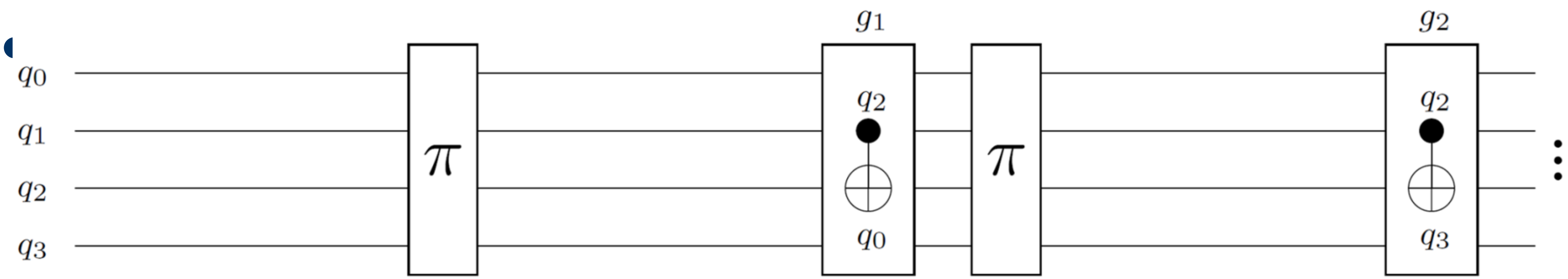


- the cost of implementing them (i.e. the number of SWAP gates)
  - Can be calculated using *inversion vectors*
  - Example: (0; 1; 2; 3) → (2; 3; 1; 0)
    - v = (3; 2; 0; 0)
    - 3 + 2 + 0 + 0 = 5 SWAP gates

Naïve Approach

1. Enumerately consider all possible permutations for all gates of the given circuit
2. For each permutation satisfying the nearest neighbor condition, calculate the number of SWAP gates required to realize the permutation
3. Afterwards, take the one with the smallest costs

➔ $n!^d$ possible combinations
(n…number of lines, d…number of gates)



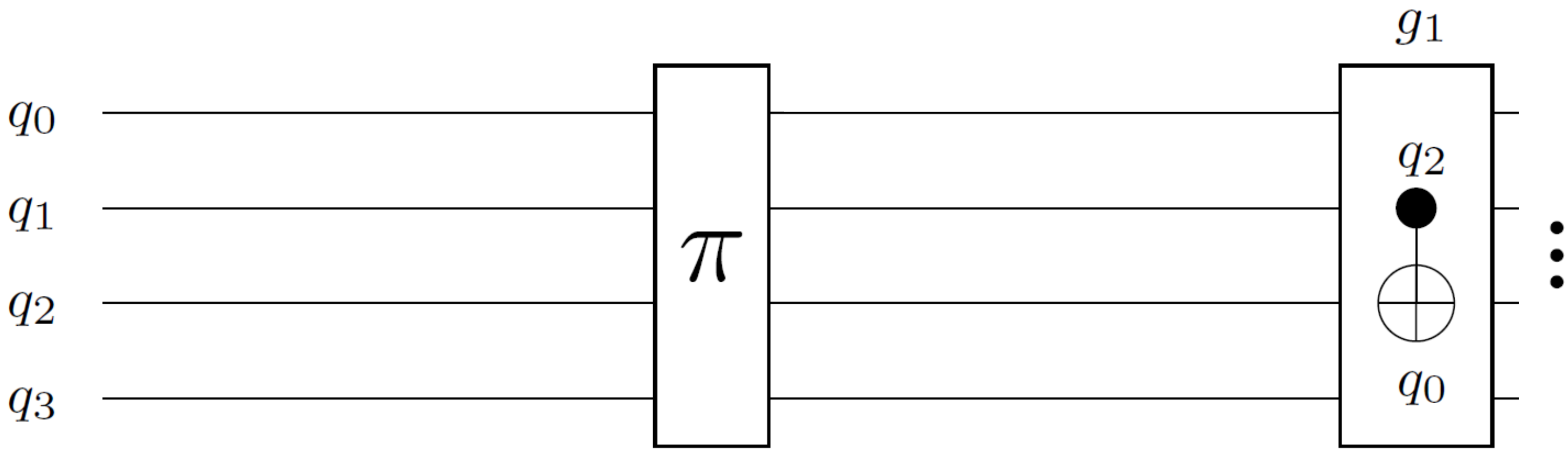
PBO Solver

- PBO solvers: An algorithm for solving the Pseudo Boolean Optimization problem
- Gets a Boolean function *f* and an optimization function *F* as input and determines
  - an assignment *a* such that $f(a)=1$ and *F* is minimized

  or

  - proofs that no such assignment exists
- Challenge:
  How to encode the PBO instance?

PBO Encoding



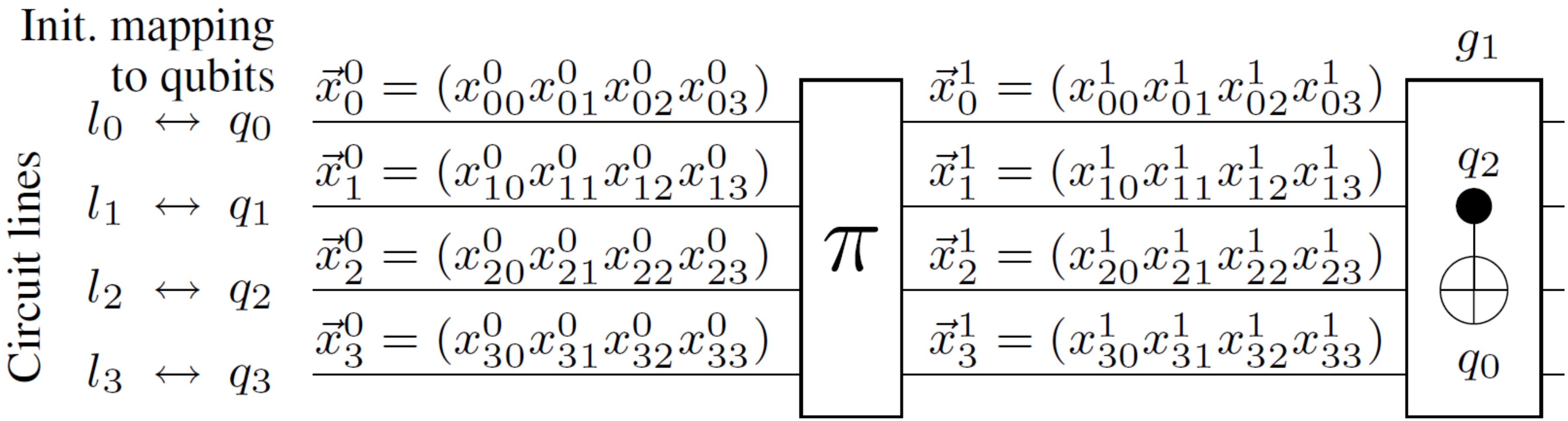
Consistency-constraints:

$$
\begin{aligned}
& x^0_{00} + x^0_{01} + x^0_{02} + x^0_{03} = 1 \\
\wedge & x^0_{10} + x^0_{11} + x^0_{12} + x^0_{13} = 1 \\
\wedge & x^0_{20} + x^0_{21} + x^0_{22} + x^0_{23} = 1 \\
\wedge & x^0_{30} + x^0_{31} + x^0_{32} + x^0_{33} = 1 \\
\wedge & x^0_{00} + x^0_{10} + x^0_{20} + x^0_{30} = 1 \\
\wedge & x^0_{01} + x^0_{11} + x^0_{21} + x^0_{31} = 1 \\
& \ldots
\end{aligned}
$$

Adjacency-constraints (for g_1 with q_0 and q_2):

$$
\begin{aligned}
& (x^1_{00} \wedge x^1_{12}) \\
\vee & (x^1_{10} \wedge x^1_{22}) \\
\vee & (x^1_{20} \wedge x^1_{32}) \\
\vee & (x^1_{02} \wedge x^1_{10}) \\
\vee & (x^1_{12} \wedge x^1_{20}) \\
\vee & (x^1_{22} \wedge x^1_{30})
\end{aligned}
$$

PBO Encoding



Permutation-constraint (for $\pi = (2310)$ and $k = 1$)

$$(\vec{x}_0^0 = \vec{x}_2^1 \;\wedge\; \vec{x}_1^0 = \vec{x}_3^1 \;\wedge\; \vec{x}_2^0 = \vec{x}_1^1 \;\wedge\; \vec{x}_3^0 = \vec{x}_0^1) \Leftrightarrow s_{2310}^1$$

Objective function:

$$\min((0\cdot s_{0123}^2 + 1\cdot s_{0132}^2 + 1\cdot s_{0213}^2 + 2\cdot s_{0231}^2 + 2\cdot s_{0312}^2 + 3\cdot s_{0321}^2 + 1\cdot s_{1023}^2 + 2\cdot s_{1032}^2 + 2\cdot s_{1203}^2 + 3\cdot s_{1230}^2 + 3\cdot s_{1302}^2 + 4\cdot s_{1320}^2 + 2\cdot s_{2013}^2 + 3\cdot s_{2031}^2 + 3\cdot s_{2103}^2 + 4\cdot s_{2130}^2 + 4\cdot s_{2301}^2 + 5\cdot s_{2310}^2 + 3\cdot s_{3012}^2 + 4\cdot s_{3021}^2 + 4\cdot s_{3102}^2 + 5\cdot s_{3120}^2 + 5\cdot s_{3201}^2 + 6\cdot s_{3210}^2) + \ldots)$$

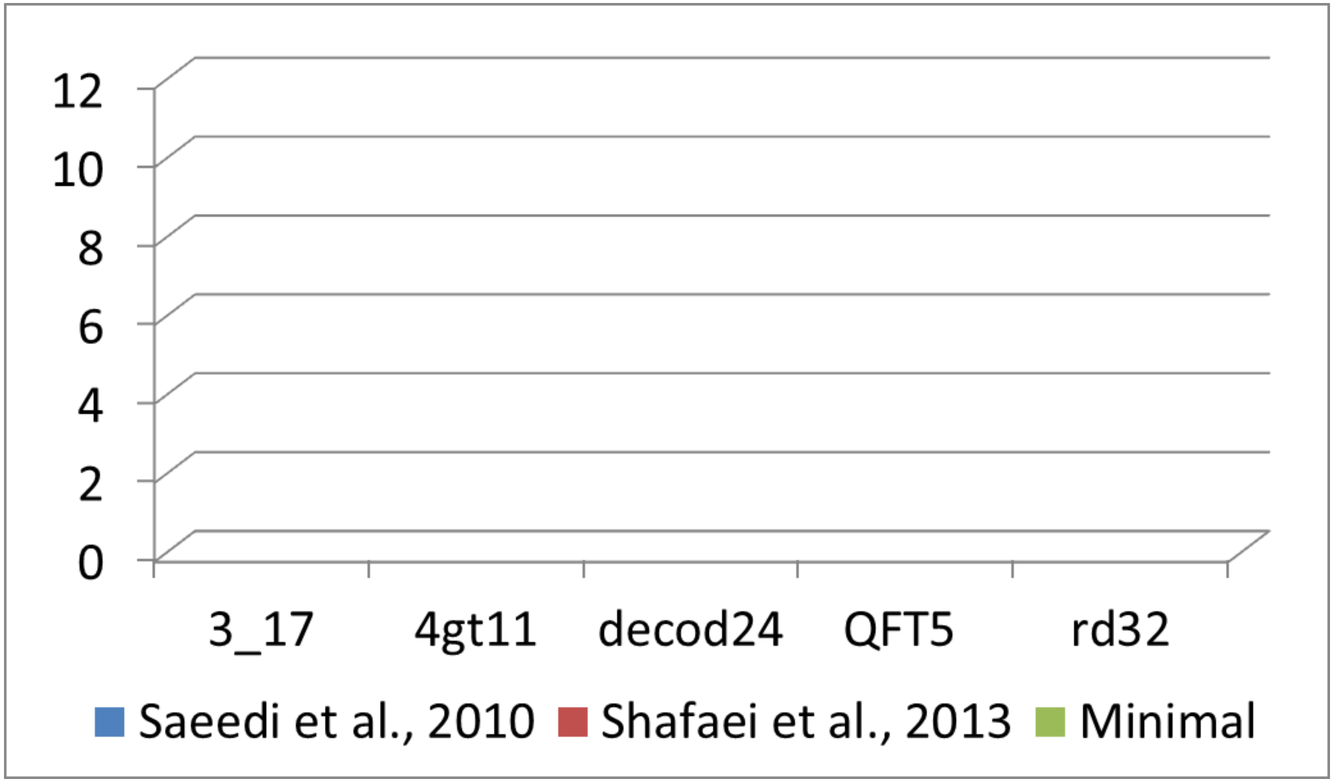
Experimental Evaluation

- Implemented on top of RevKit (www.revkit.org)
- *clasp* as PBO solver (www.cs.uni-potsdam.de/clasp/)
- Benchmarks from RevLib (www.revlib.org)

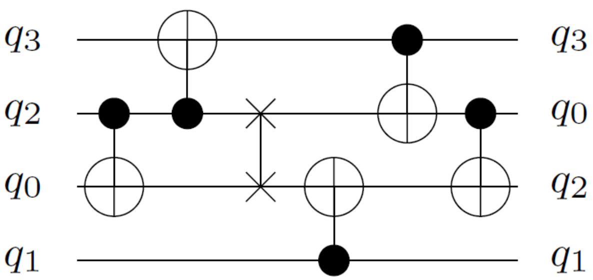
Benchmark	n	$\lvert G \rvert$	$n!^{\lvert G \rvert}$	Swaps	Time

Experimental Evaluation

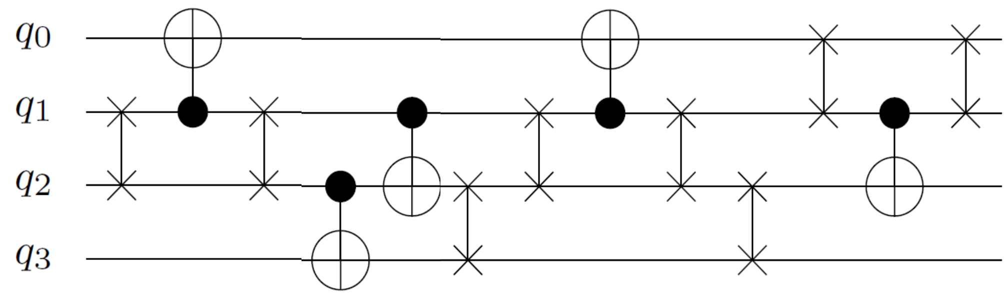
- Implemented on top of RevKit (www.revkit.org)
- *clasp* as PBO solver (www)
- Benchmarks from RevLib (www.revlib.org)



Conclusion



vs.



- Minimizing the number of SWAP gate insertions
- Exploiting the deductive power of PBO solvers

- Enabled to compare results obtained by heuristic methods to the actual optimum

- Future Work: Consideration of alternative architectures (e.g. nearest for 2D quantum architectures)

Optimal SWAP Gate Insertion for Nearest Neighbor Quantum Circuits

Robert Wille[1,2], Aaron Lye[1], Rolf Drechsler[1,2]

[1] Institute of Computer Science
University of Bremen, Germany
[2] Cyber Physical Systems
DFKI GmbH Bremen, Germany

rwille@informatik.uni-bremen.de