

QAOA Tutorial Outline

- Quantum Approximate Optimization Algorithm: review and status
- The «Quantum Alternating Operator Ansatz»
 - Mixing Operators
 - Examples
- Compiling and Executing
 - The gate synthesis problem
 - Review of compilation methods
 - Compiling framework in nearest-neighbor architectures

Davide Venturelli, Ph.D.
USRA Research Institute for Advanced Computer Science
Quantum Artificial Intelligence Laboratory
NASA Ames Research Center
davide.venturelli@nasa.gov

READING LIST

- **Optimizing Variational Quantum Algorithms Using Pontryagin's Minimum Principle** Z. Yang, A. Rahmani, A. Shabani, H. Neven, and C. Chamon *Phys. Rev. X* **7**, 021027 (2017)
- **Quantum Approximate Optimization with Hard and Soft Constraints**. Hadfield, S., Wang, Z., Rieffel, E. G., O'Gorman, B., Venturelli, D., & Biswas, R. (2017, November). In *Proceedings of the Second International Workshop on Post Moores Era Supercomputing* (pp. 15-21). ACM.
- **From the quantum approximate optimization algorithm to a quantum alternating operator Ansatz** Hadfield, S, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas. *arXiv preprint arXiv:1709.03489* (2017).
- **Compiling quantum circuits to realistic hardware architectures using temporal planners** Venturelli, D., Do, M., Rieffel, E. and Frank, J., 2018. *Quantum Science and Technology*, 3(2), p.025004.

1. Review on QAOA Research

QAOA Review

1. Design a binary optimization classical Hamiltonian (**“phase separation”**)
2. Design a unitary operator that can connect and allow jumps between different states (**“mixing”**)
3. Prepare a QAOA state for some parameters

$$|\beta, \gamma\rangle = Q_p(\beta, \gamma) |s\rangle$$

$$Q_p(\beta, \gamma) = U_M(\beta_p)U_P(\gamma_p) \cdots U_M(\beta_1)U_P(\gamma_1).$$

4. Measure the state in the computational value and compute the exp. value of $C(z)$

$$F_p(\gamma, \beta) = \langle \gamma, \beta | C | \gamma, \beta \rangle \quad M_p \geq M_{p-1}$$

$$M_p = \max_{\gamma, \beta} F_p(\gamma, \beta) \quad \lim_{p \rightarrow \infty} M_p = \max_z C(z)$$

5. Change the parameters if they are not proven optimal and repeat 3-4

QAOA Status of Research



MIT-CTP/4610

A Quantum Approximate Optimization Algorithm

Edward Farhi and Jeffrey Goldstone
Center for Theoretical Physics
Massachusetts Institute of Technology
Cambridge, MA 02139

Sam Gutmann

Abstract

We introduce a quantum algorithm that produces approximate solutions for combinatorial optimization problems. The algorithm depends on an integer $p \geq 1$ and the quality of the approximation improves as p is increased. The quantum circuit that implements the algorithm consists of unitary gates whose locality is at most the locality of the objective function whose optimum is sought. The depth of the circuit grows linearly with p times (at worst) the number of constraints. If p is fixed, that is, independent of the input size, the algorithm makes use of efficient classical preprocessing. If p grows with the input size a different strategy is proposed. We study the algorithm as applied to MaxCut on regular graphs and analyze its performance on 2-regular and 3-regular graphs for fixed p . For $p = 1$, on 3-regular graphs the quantum algorithm always finds a cut that is at least 0.6924 times the size of the optimal cut.

1. UNDERSTANDING

- a. **Power/Supremacy:** Fahri, Arrow (2016), Jiang (2017)
- b. **Parameter Setting:** Yang (2017), Wang (2017)

2. PRACTICAL ASPECTS

- a. **Compilation:** Venturelli (2017), Guerreschi (2017), Oddi (2018)
- b. **Hybridization:** Guerreschi (2016)

3. GENERALIZATIONS/APPLICATIONS

- a. **Combinatorial Optimization:** Hadfield (2016)
- b. **Factoring:** Anschuetz (2018)
- c. **Deep Learning:** Verdon (2018)

4. EXPERIMENTS

- a. **Superconducting:** Otterbach (2017)
- b. **Neutral Atoms:** Pichler (2018)

QAOA Status of Research

MIT-CTP/4610

A Quantum Approximate Optimization Algorithm

Edward Farhi and Jeffrey Goldstone
*Center for Theoretical Physics
Massachusetts Institute of Technology
Cambridge, MA 02139*

Sam Gutmann
Abstract

We introduce a quantum algorithm that produces approximate solutions for combinatorial optimization problems. The algorithm depends on an integer $p \geq 1$ and the quality of the approximation improves as p is increased. The quantum circuit that implements the algorithm consists of unitary gates whose locality is at most the locality of the objective function whose optimum is sought. The depth of the circuit grows linearly with p times (at worst) the number of constraints. If p is fixed, that is, independent of the input size, the algorithm makes use of efficient classical preprocessing. If p grows with the input size a different strategy is proposed. We study the algorithm as applied to MaxCut on regular graphs and analyze its performance on 2-regular and 3-regular graphs for fixed p . For $p = 1$, on 3-regular graphs the quantum algorithm always finds a cut that is at least 0.6924 times the size of the optimal cut.

1. UNDERSTANDING

- a. **Power/Supremacy:** Fahri, Arrow (2016), Jiang (2017)
- b. **Parameter Setting:** Yang (2017), Wang (2017)

2. PRACTICAL ASPECTS

- a. **Compilation:** Venturelli (2017), Guerreschi (2017), Oddi (2018)
- b. **Hybrid:** **Optimizing Variational Quantum Algorithms Using Pontryagin's Minimum Principle**

Optimizing Variational Quantum Algorithms Using Pontryagin's Minimum Principle

Zhi-Cheng Yang, Armin Rahmani, Alireza Shabani, Hartmut Neven, and Claudio Chamon Phys. Rev. X 7, 021027 – Published 18 May 2017

3. GENERALIZATION

- a. **Combinatorial Optimization:** Hadfield (2016)
- b. **Factoring:** Aïmeur (2017)
- c. **Deep Learning:**
 - QAOA is a Bang-Bang quantum control protocol
 - Same «ansatz» (not params) is robust to noise/decoherence
 - It seems that certain aspects of parameter settings are universal (to be verified)

4. EXPERIMENTS

- a. **Superconducting:** Otterbach (2017)
- b. **Neutral Atoms:** Pichler (2018)

QAOA Status of Research

MIT-CTP/4610

A Quantum Approximate Optimization Algorithm

Edward Farhi and Jeffrey Goldstone
*Center for Theoretical Physics
Massachusetts Institute of Technology
Cambridge, MA 02139*

Sam Gutmann

Abstract

We introduce a quantum algorithm that produces approximate solutions for combinatorial optimization problems. The algorithm depends on an integer $p \geq 1$ and the quality of the approximation improves as p is increased. The quantum circuit that implements the algorithm consists of unitary gates whose locality is at most the locality of the objective function whose optimum is sought. The depth of the circuit grows linearly with p times (at worst) the number of constraints. If p is fixed, that is, independent of the input size, the algorithm makes use of efficient classical preprocessing. If p grows with the input size a different strategy is proposed. We study the algorithm as applied to MaxCut on regular graphs and analyze its performance on 2-regular and 3-regular graphs for fixed p . For $p = 1$, on 3-regular graphs the quantum algorithm always finds a cut that is at least 0.6924 times the size of the optimal cut.

1. UNDERSTANDING

- a. **Power/Supremacy:** Fahri, Arrow (2016), Jiang (2017)
- b. **Parameter Setting:** Yang (2017), Wang (2017)

2. PRACTICAL ASPECTS

- a. **Compilation:** Venturelli (2017), Guerreschi (2017), Oddi (2018)
- b. **Hybridization:** Guerreschi (2016)

3. GENERALIZATIONS/APPLICATIONS

- a. **Combinatorial Optimization:** Hadfield (2016)
- b. **Factoring:** Anschuetz (2018)
- c. **Deep Learning:** Verdon (2018)

This tutorial !

4. EXPERIMENTS

- a. **Superconducting:** Otterbach (2017)
- b. **Neutral Atoms:** Pichler (2018)

2. QAOA for Constrained Optimization Problems

QAOA Applications

- Maximum Cut
- Max-SAT, Min-SAT, NAE-SAT
- Set Splitting
- MaxE3LIN2
- Max-ColorableSubgraph
- Graph Partitioning
- Maximum Bisection
- Max Vertex k-Cover
- MaxIndependentSet
- MaxClique
- MinVertexCover
- MaxSetPacking
- MinSetCover
- TSP
- SMS with various metrics and constraints
- ...

Objective Function: Soft Constraints

Feasible States: Hard Constraints

Quantum Approximate Optimization with Hard and Soft Constraints

Stuart Hadfield*, Zihui Wang^{+,**}, Eleanor G. Rieffel⁺,

Bryan O’Gorman^{+,†}, Davide Venturelli^{+,**}, Rupak Biswas⁺

^{*} Department of Computer Science, Columbia University, New York, NY

⁺ Quantum Artificial Intelligence Lab., NASA Ames Research Center, Moffett Field, CA

^{**} Universities Space Research Association, Mountain View, CA

[†]Stinger Ghaffarian Technologies, Inc., Greenbelt, MD

ABSTRACT

Challenging computational problems arising in the practical world are frequently tackled by heuristic algorithms. Small universal quantum computers will emerge in the next year or two, enabling a substantial broadening of the types of quantum heuristics that can be investigated beyond quantum annealing. The immediate question is: what experiments should we prioritize that will give us insight into quantum heuristics? One leading candidate is the quantum approximate optimization algorithm (QAOA) metaheuristic. In this work, we provide a framework for designing QAOA circuits for a variety of combinatorial optimization problems with both hard constraints that must be met and soft constraints whose violation we wish to minimize. We work through a number of examples, and discuss design principles.

CCS CONCEPTS

• Mathematics of computing → Approximation algorithms;
• Hardware → Emerging technologies; Quantum computation;
• Theory of computation → *Quantum computation theory*;
Mathematical optimization;

advantage, and if so, how to design quantum algorithms that realize such advantages. Today, challenging computational problems arising in the practical world are frequently tackled by heuristic algorithms, which by definition have not been analytically proven to be the best approach, or even proven analytically to outperform the best approach of the previous year. Rather, these algorithms are empirically shown to be effective, by running them on characteristic sets of problems, or demonstrating their effectiveness in practical applications. As prototype quantum hardware emerges, this approach to algorithm design becomes available for the evaluation of quantum heuristic algorithms.

For several years now, special-purpose quantum hardware has been used to explore one quantum heuristic algorithm, quantum annealing. Emerging gate-model processors, which are universal in that, once scaled up, they can run any quantum algorithm, will enable investigation of a much broader array of quantum heuristics beyond quantum annealing. Within the last year, IBM has made available publicly through the cloud a 5-qubit gate-model chip [13], and announced recently an upgrade to a 17-qubit chip. Likewise, Google [3] and Rigetti Computing [22], anticipate providing processors with 40–100 qubits within a year or two [18]. Many academic

Alternating Operator Ansatz

$$Q_p(\beta, \gamma) = U_M(\beta_p)U_P(\gamma_p) \cdots U_M(\beta_1)U_P(\gamma_1).$$

$$|\beta, \gamma\rangle = Q_p(\beta, \gamma) |s\rangle$$

Some initial state respecting:

- It is a superposition of several solutions in the feasible subspace
- It can be prepared efficiently

$$e^{-i\beta H_M}$$

Some unitary respecting:

- Preserve the feasible subspace
- Provide all-to-all nonzero transitions between all feasible states
- Non-necessarily time evolution of a local Hamiltonian

$$e^{-i\gamma H_P}$$

Some unitary respecting:

- Is diagonal in the computational basis
- The spectrum of H_P encodes the objective function

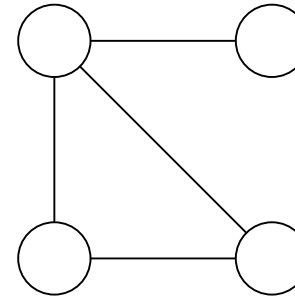
$$H_f |\mathbf{x}\rangle = f(\mathbf{x}) |\mathbf{x}\rangle$$

Alternating Operator Ansatz

Node u is colored by c

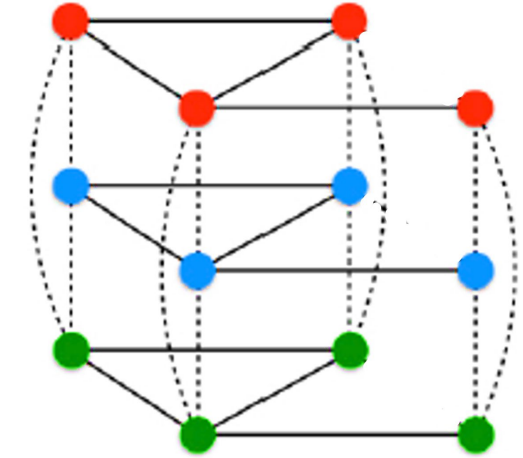
$$X_{u,c} = 1$$

$$m - \sum_{\{u,v\} \in E} \sum_{a=1}^k x_{u,a} x_{v,a}$$



Phase Separator (QUBO objective function)

$$H'_P = \frac{4 - \kappa}{4} mI + \frac{1}{4} \sum_{\{u,v\} \in E} \sum_{a=1}^{\kappa} (Z_{u,a} + Z_{v,a} - Z_{u,a} Z_{v,a})$$



Initial state:

$$|W\rangle_v = \frac{1}{\sqrt{k}} (|100 \dots 0\rangle + |010 \dots 0\rangle + |0 \dots 01\rangle)$$

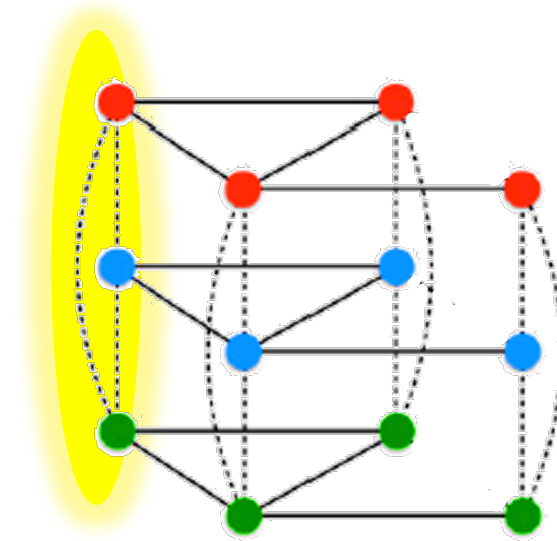
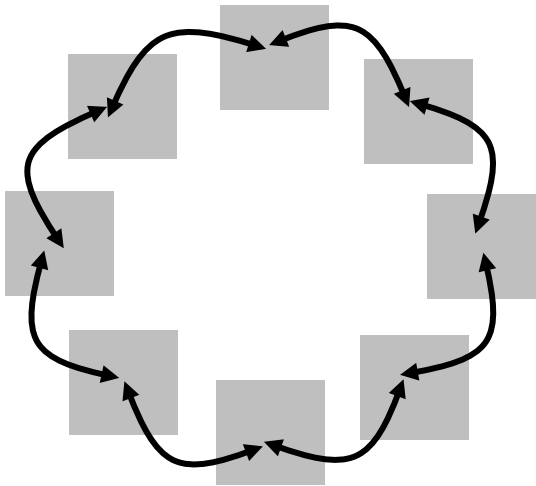
- Babbush (2017)
- Verstraete (2009)
- Wang (2009)
- Childs (2002)
- ...

Engineering Mixing Operators

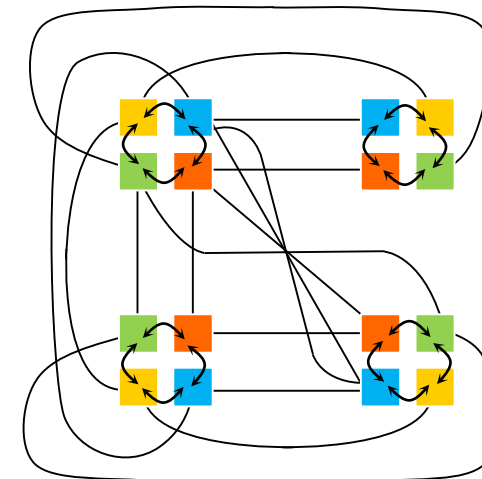
$$H_{\text{ring}}^{(\text{enc})} = \sum_a^d (X_a X_{a+1} + Y_a Y_{a+1})$$

Respects the Hamming Weight constraint

$\text{Exp}(iH_{\text{ring}})$ is difficult to implement



3-coloring



4-coloring

Design Freedom and Tradeoffs

$$H_{\text{ring}}^{(\text{enc})} = \sum_a^d (X_a X_{a+1} + Y_a Y_{a+1})$$

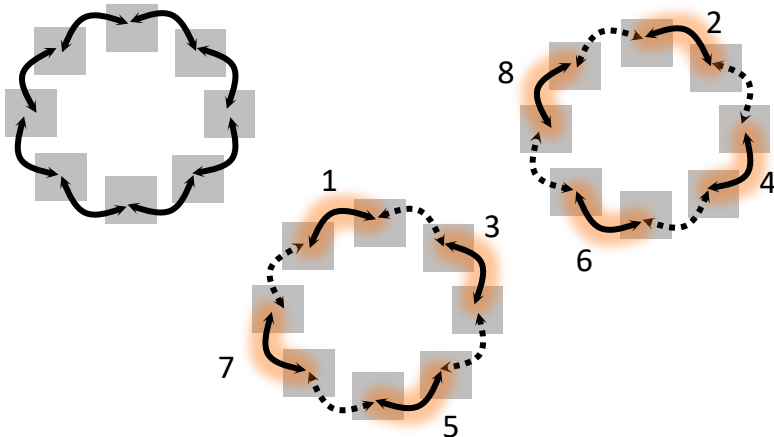
Respects the Hamming Weight constraint

$\text{Exp}(iH_{\text{ring}})$ is difficult to implement

$$U_M = \prod_{v=1}^n U_{v,\text{parity}}^{(\text{enc})} \quad \prod_{a \in \text{parity}} \text{Exp}(iX_a X_{a+1} + Y_a Y_{a+1})$$

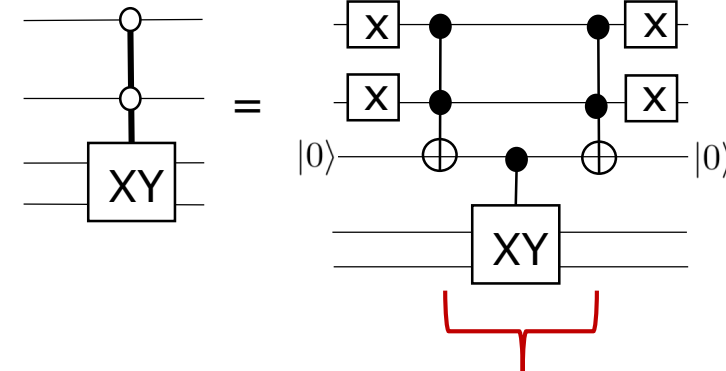
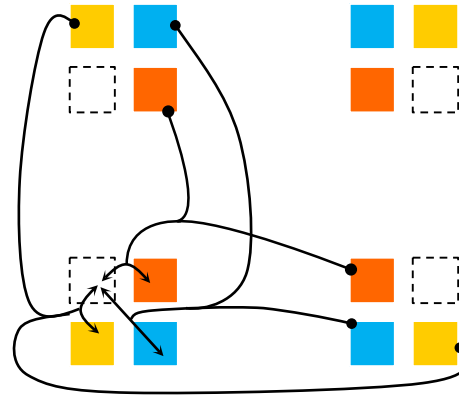
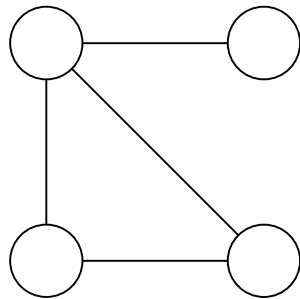
$$U_M = \underbrace{[U_1 U_3 U_5 U_7] [U_2 U_4 U_6 U_8] [U_1 U_3 U_5 U_7] [U_2 U_4 U_6 U_8] \dots}_{\text{This couples only distance 2; has to be repeated } k/2 \text{ times}}$$

All these 2-qubit $k^2/2$ gates need to be scheduled



Other Mixers (controlled XY)

Finding the largest induced subgraph colorable by k colors



Still needs to be compiled to 2 qubit gates

All these gates need to be scheduled

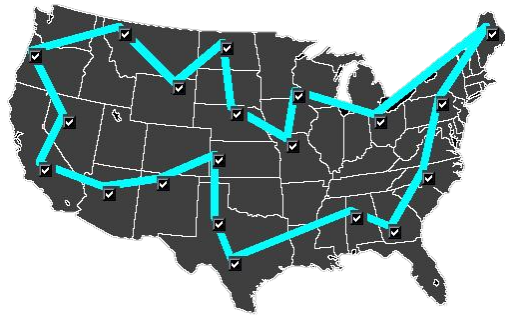
Node u is colored by c or uncolored (c=0)

$$X_{u,c} = 1$$

$$C = m - \sum_v x_{v,0} \rightarrow H_C = \frac{1}{2} \sum_v Z_{v,0}$$

Mixers Navigation & Scheduling

In **traveling salesman** encoding



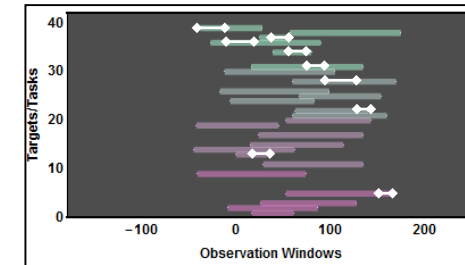
$X_{vj}=1$ if city v is visited as j^{th}

$$\sum_{\{u,v\} \in E} d_{u,v} \sum_{j=1}^n (x_{u,j} x_{v,j+1} + x_{v,j} x_{u,j+1})$$

$$H_{\text{PS},\{i,j\},\{u,v\}}^{(\text{enc})} = S_{u,i}^+ S_{v,j}^+ S_{u,j}^- S_{v,i}^- + S_{u,i}^- S_{v,j}^- S_{u,j}^+ S_{v,i}^+$$

(partitioned using edge coloring and parity
 $\approx (n-1)n^2/4$ mixers)
 (needs to be repeated $n(n-1)/2$ times for all-to-all)

In **single machine scheduling**



$X_{jt}=1$ if job j starts at time t

$$C = \sum_j w_j \sum_{(d_j - p_j) < t < h} x_{j,t} (t + p_j - d_j)$$

$$H_{\text{TS},t,\{i,j\}}^{(\text{enc})} = S_{i,t+p_j}^+ S_{j,t}^+ S_{i,t}^- S_{j,t+p_i}^- + S_{i,t}^+ S_{j,t+p_i}^+ S_{i,t+p_j}^- S_{j,t}^-$$

(But if we add release dates then we need controls on the no-overlap constraint)

Zoology of Ansätze

Bitflip mixers

- Maximum Cut
- Max-SAT, Min-SAT, NAE-SAT
- Set Splitting
- MaxE3LIN2

...

Controlled Bitflip mixers

- MaxIndependentSet
- MaxClique
- MinVertexCover
- MaxSetPacking
- MinSetCover

...

XY mixers

- Max-ColorableSubgraph
- Graph Partitioning
- Maximum Bisection
- Max Vertex k-Cover

...

Controlled XY mixers

- Max-k-ColorableInducedSubgraph
- MinGraphColoring
- MinCliqueCover

...

Permutation mixers

- TSP
- SMS with various metrics and constraints

...

From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz

Stuart Hadfield*, Zhihui Wang^{+,**}, Bryan O’Gorman^{+,†,‡}, Eleanor G. Rieffel⁺, Davide Venturelli^{+,**}, Rupak Biswas⁺

* Department of Computer Science, Columbia University, New York, NY

⁺ Quantum Artificial Intelligence Lab., NASA Ames Research Center, Moffett Field, CA

^{**} USRA Research Institute for Advanced Computer Science (RIACS), Mountain View, CA

[†] Stinger Ghaffarian Technologies, Inc., Greenbelt, MD

[‡] Berkeley Quantum Information and Computation Center and Departments of Chemistry and Computer Science, University of California, Berkeley, CA

September 12, 2017

The next few years will be exciting as prototype universal quantum processors emerge, enabling implementation of a wider variety of algorithms. Of particular interest are quantum heuristics, which require experimentation on quantum hardware for their evaluation, and which have the potential to significantly expand the breadth of applications for which quantum computers have an established advantage. A leading candidate is Farhi et al.’s Quantum Approximate Optimization Algorithm, which alternates between applying a cost-function-based Hamiltonian and a mixing Hamiltonian. Here, we extend this framework to allow alternation between more general families of operators. The essence of this extension, the Quantum Alternating Operator Ansatz, is the consideration of general parameterized families of unitaries rather than only those corresponding to the time-evolution under a fixed local Hamiltonian for a time specified by the parameter. This ansatz supports the representation of a larger, and potentially more useful, set of states than the original formulation, with potential long-term impact on a broad array of application areas.

3. QAOA in the "Real World"

The Gate Synthesis Problem

Quantum Circuits can be composed by single and two-qubit gates of universal set*

CNOT, $R_y(\theta)$ and $R_z(\alpha)$

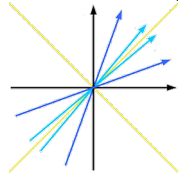
Each single qubit gate can be decomposed by single qubit rotations.

$$U_1 = R_z(\alpha) R_y(\beta) R_z(\gamma) e^{i\phi}$$

Each two qubit gate is reversible and it is representable by a Unitary Matrix.

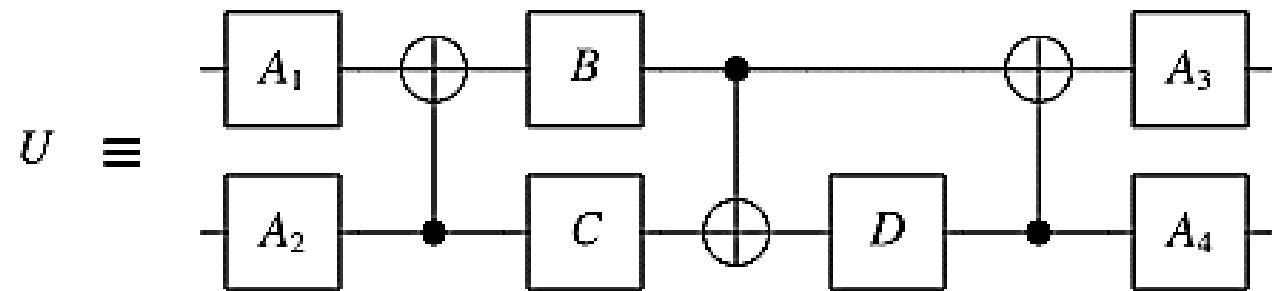
R_z gates can be «virtually» compiled.

(McKay 2017 and refs)



* active research to natively support multi-qubit gates

Barenco et al. (1995)
Kraus, Cirac (2001)
Vatan, Williams (2003)



Maximum number of elementary 1-qubit gates: **15**

Maximum number of CNOTs: **3**

Maximum depth assuming R_y , R_z and simplifications: **11**

SWAP-Compilation (review)



Performance of algorithms in NISQ will depend on aspects such as gate fidelities, parallelization, idle time, crosstalks..

Different Metrics to optimize correlate to final performance:

- Total Quantum Factor
- Quantum Volume
- Number of Two-Qubit Gates
- **Makespan**

Guerreschi and Park (2018). Two-step approach to scheduling quantum circuits. *arXiv preprint arXiv:1708.00023*.

Khatri, Sumeet, et al. "Quantum assisted quantum compiling." *arXiv preprint arXiv:1807.00800* (2018).

Li, G., Ding, Y., & Xie, Y. (2018). Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices. *arXiv preprint arXiv:1809.02573* (2018).

Oddi, Angelo, and Riccardo Rasconi. "Greedy Randomized Search for Scalable Compilation of Quantum Circuits." *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Springer, Cham, (2018).

...

Example: MaxCut

$$U = \frac{1}{2} \sum_{(i,j) \in E} (1 - s_i s_j)$$

$$s_i = \pm 1$$

Defines the cut

Counts the edges in the cut

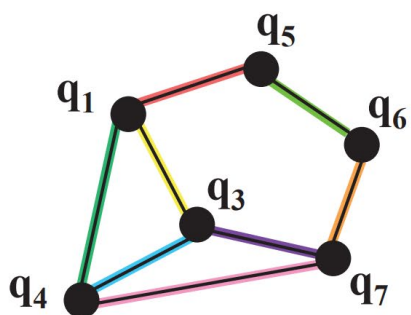
$$U_{PS} = \prod_{\langle jk \rangle} \text{Exp}(ibZ_j Z_k)$$

$$\sum_i X_i$$

Mixes the two partitions

$$U_M = \prod_j \text{Exp}(igX_j)$$

Interaction graph obtained from quadratic objective function (MAXCUT)



PS1

MX

PS2

P-S(q_1, q_4)
P-S(q_1, q_3)
P-S(q_3, q_4)
P-S(q_3, q_7)
P-S(q_4, q_7)
P-S(q_6, q_7)
P-S(q_5, q_6)
P-S(q_1, q_5)

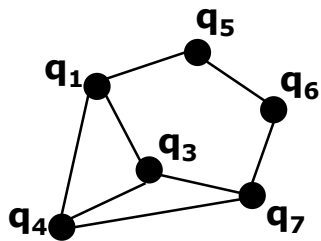
MIX(q_1)
MIX(q_3)
MIX(q_4)
MIX(q_5)
MIX(q_6)
MIX(q_7)

P-S(q_1, q_4)
P-S(q_1, q_3)
P-S(q_3, q_4)
P-S(q_3, q_7)
P-S(q_4, q_7)
P-S(q_6, q_7)
P-S(q_5, q_6)
P-S(q_1, q_5)

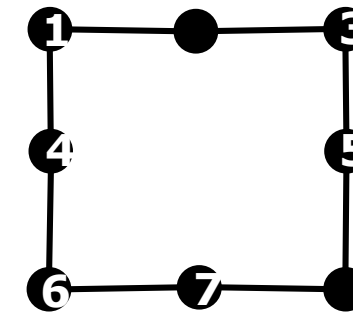
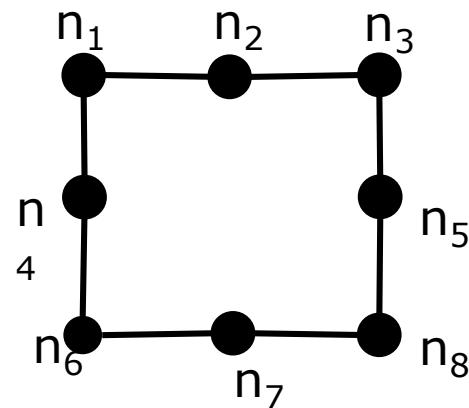
- Every edge is a gate that needs to be executed (in arbitrary order)
- The same graph has to be executed multiple times (*p rounds*).
- Every qubit has to complete all the gates of round *p* before being involved in *p+1*

Circuit Execution Schedule

Interaction graph
obtained from
quadratic objective
function (MAXCUT)

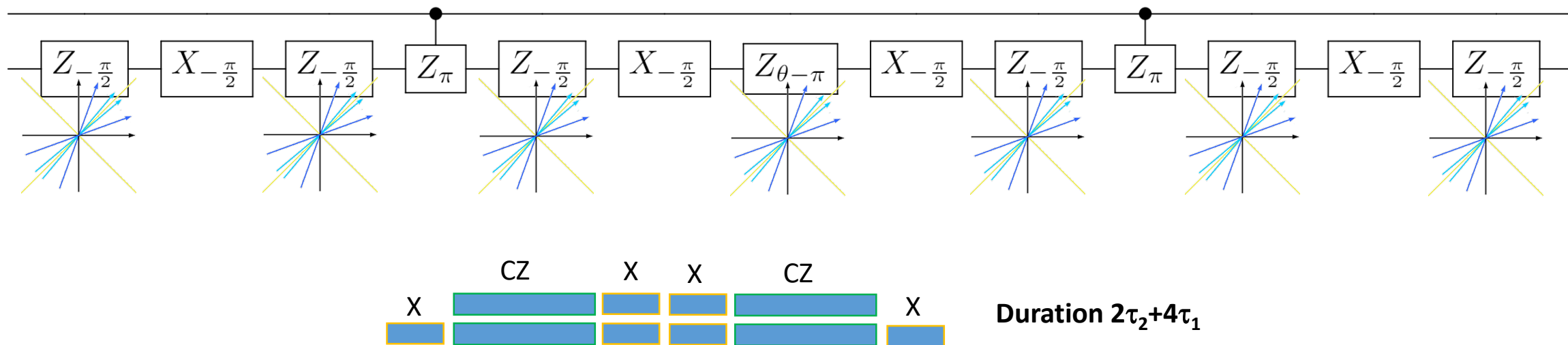
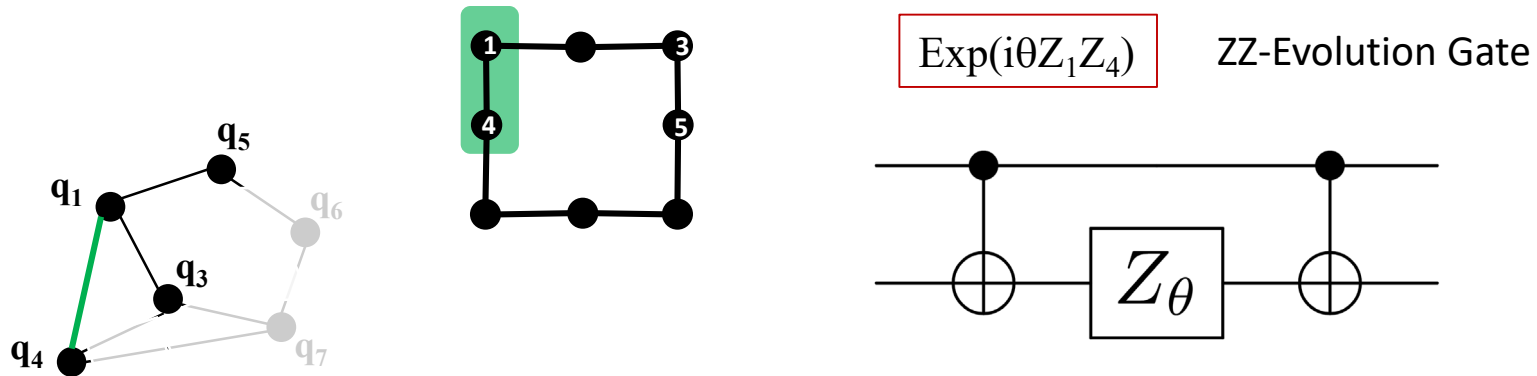


Initial assignment
 $q_i \rightarrow n_i$

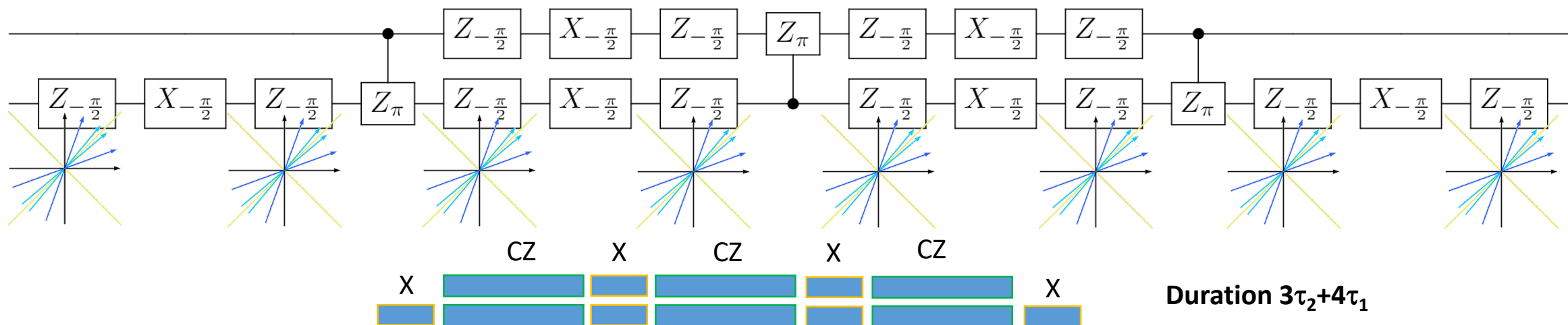
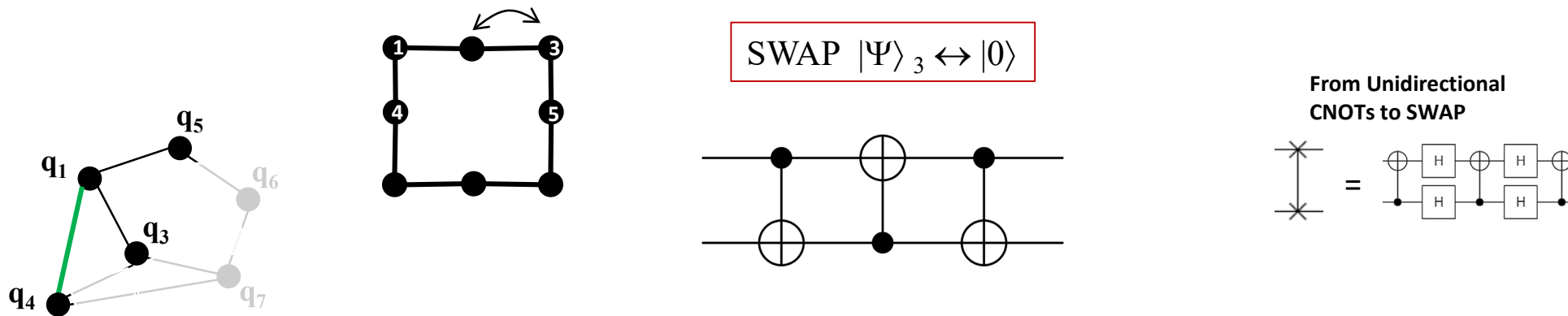


- Every edge is a gate that needs to be executed (in arbitrary order)
- The same graph has to be executed multiple times (p rounds).
- Every qubit has to complete all the gates of round p before being involved in $p+1$

Circuit Execution Schedule



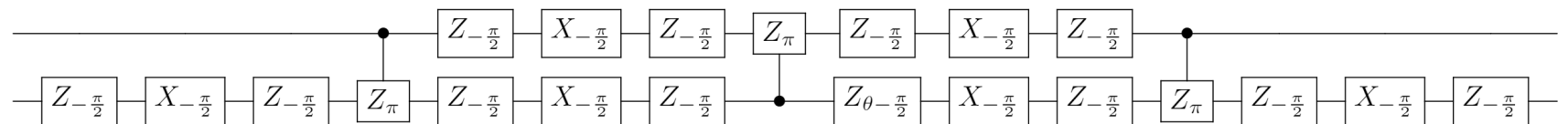
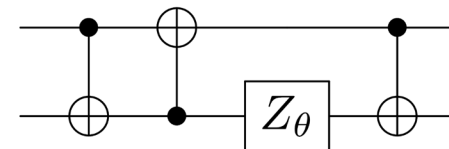
Circuit Execution Schedule



Circuit Execution Schedule

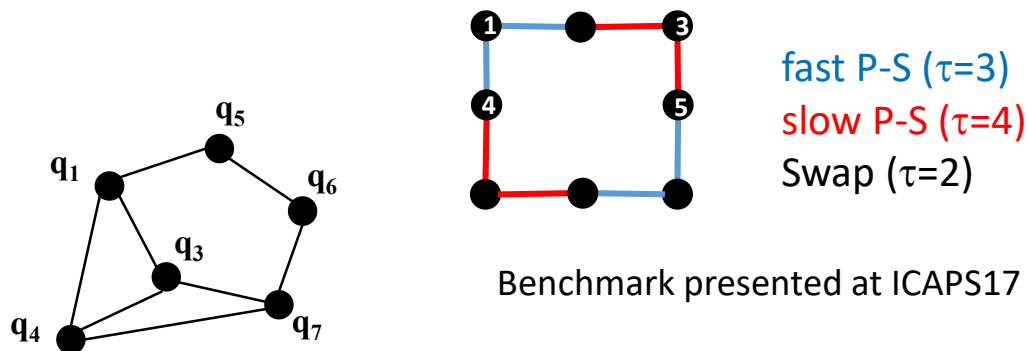
SWAPS can also be inserted as part of the UZZ interaction without the need to be sequential.

SWAP+ZZ-Evolution Gate

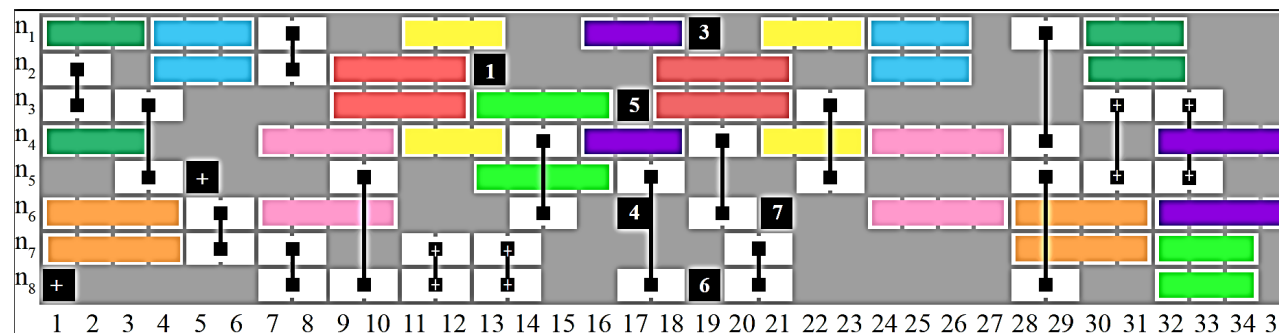


Duration $3\tau_2+4\tau_1$ (same as SWAP)

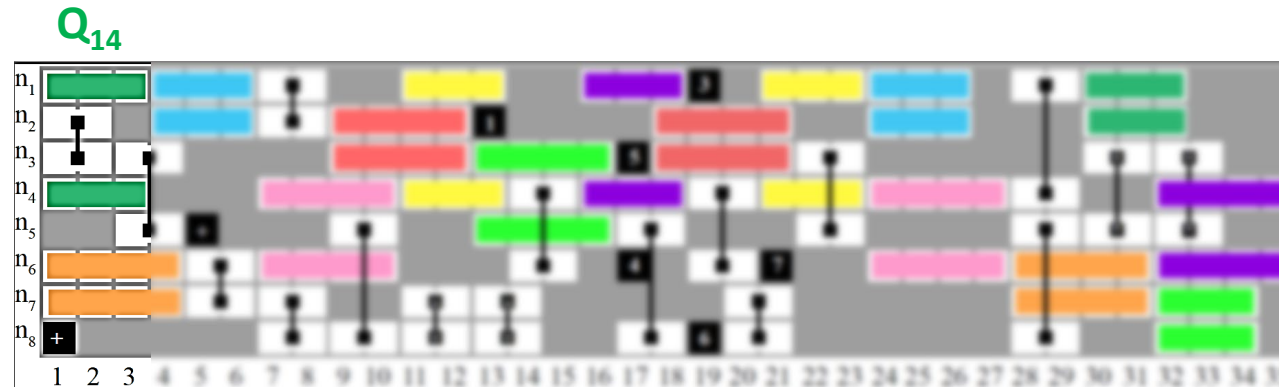
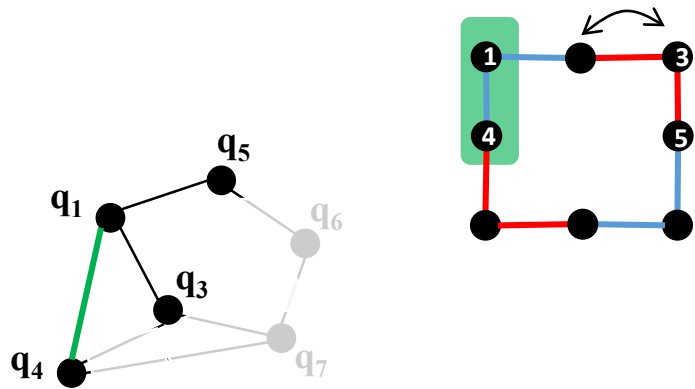
Circuit Execution Schedule



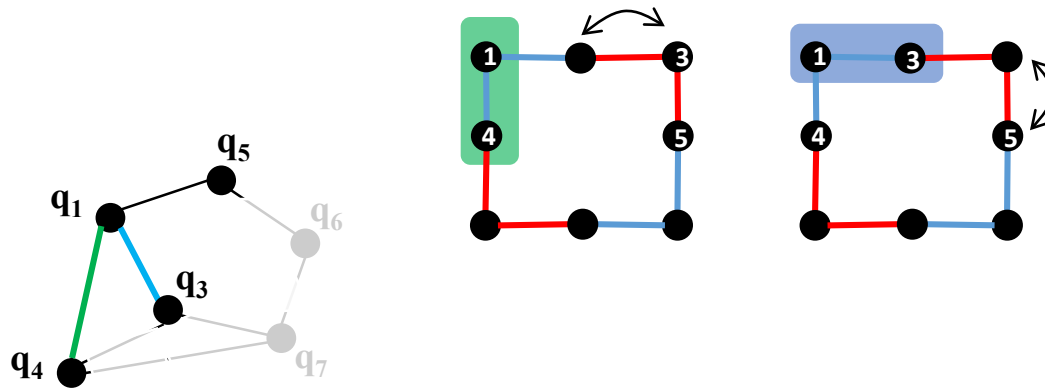
Objective: finding the makespan-minimizing Gantt Schedule for $p=1$, $p=2$, $N=8$, $N=21$



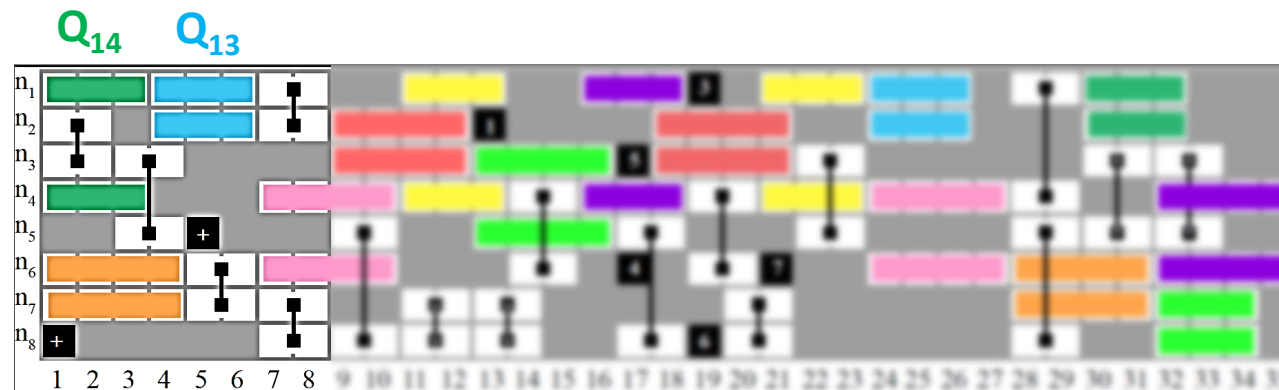
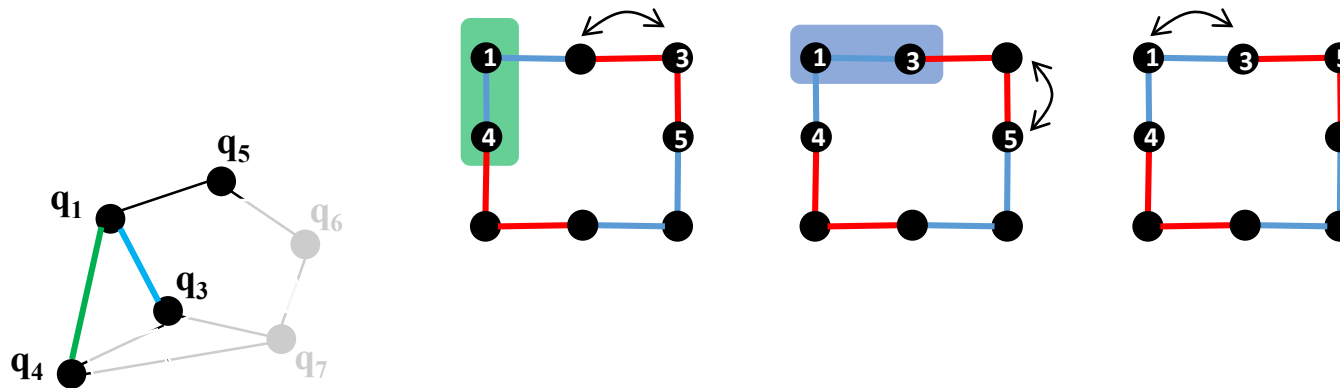
Circuit Execution Schedule



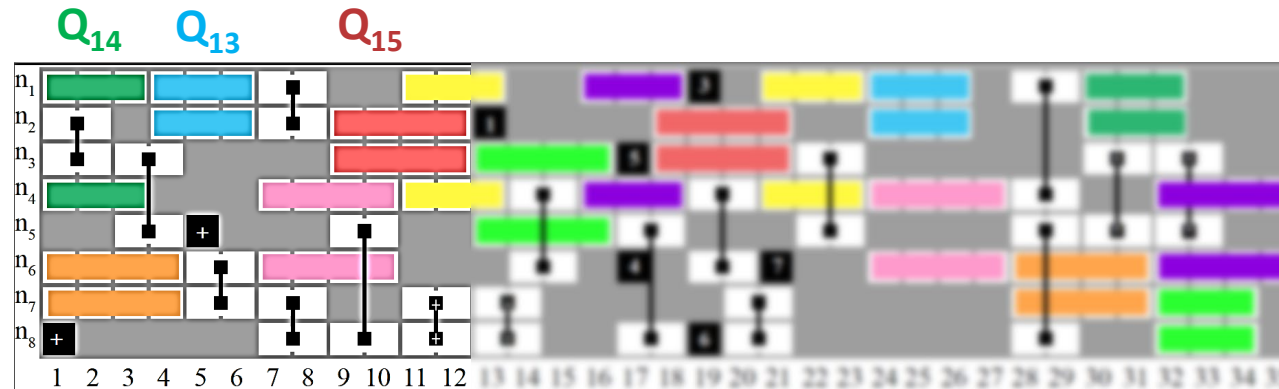
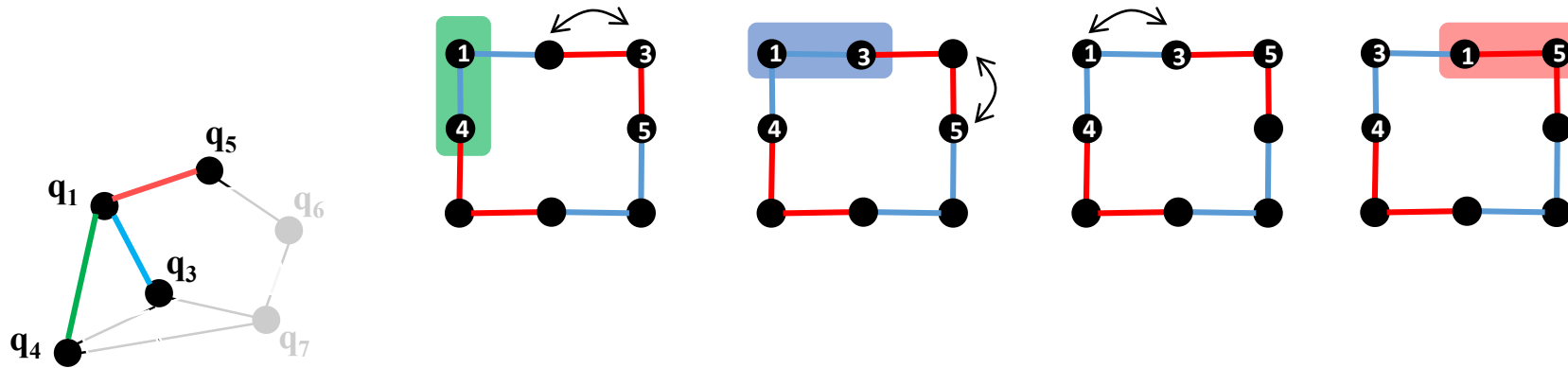
Circuit Execution Schedule



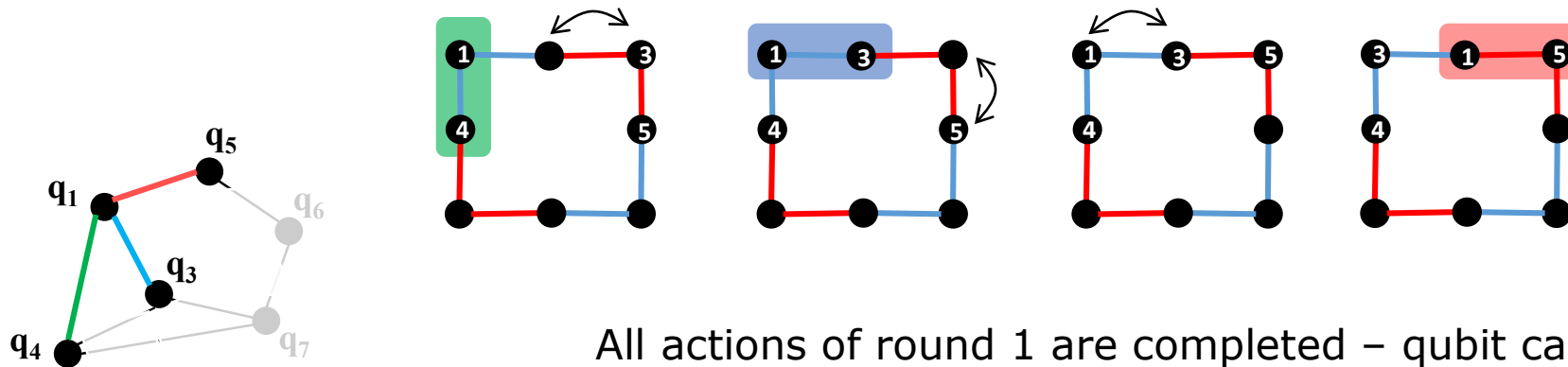
Circuit Execution Schedule



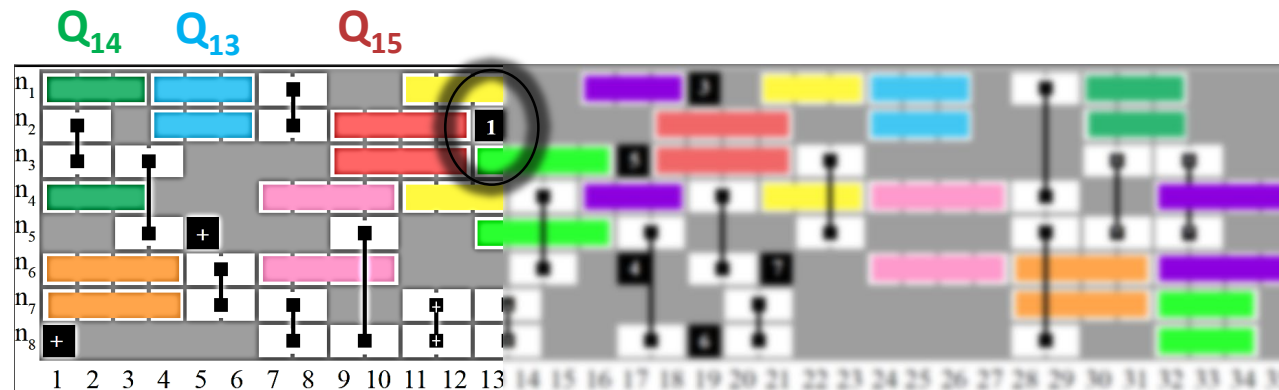
Circuit Execution Schedule



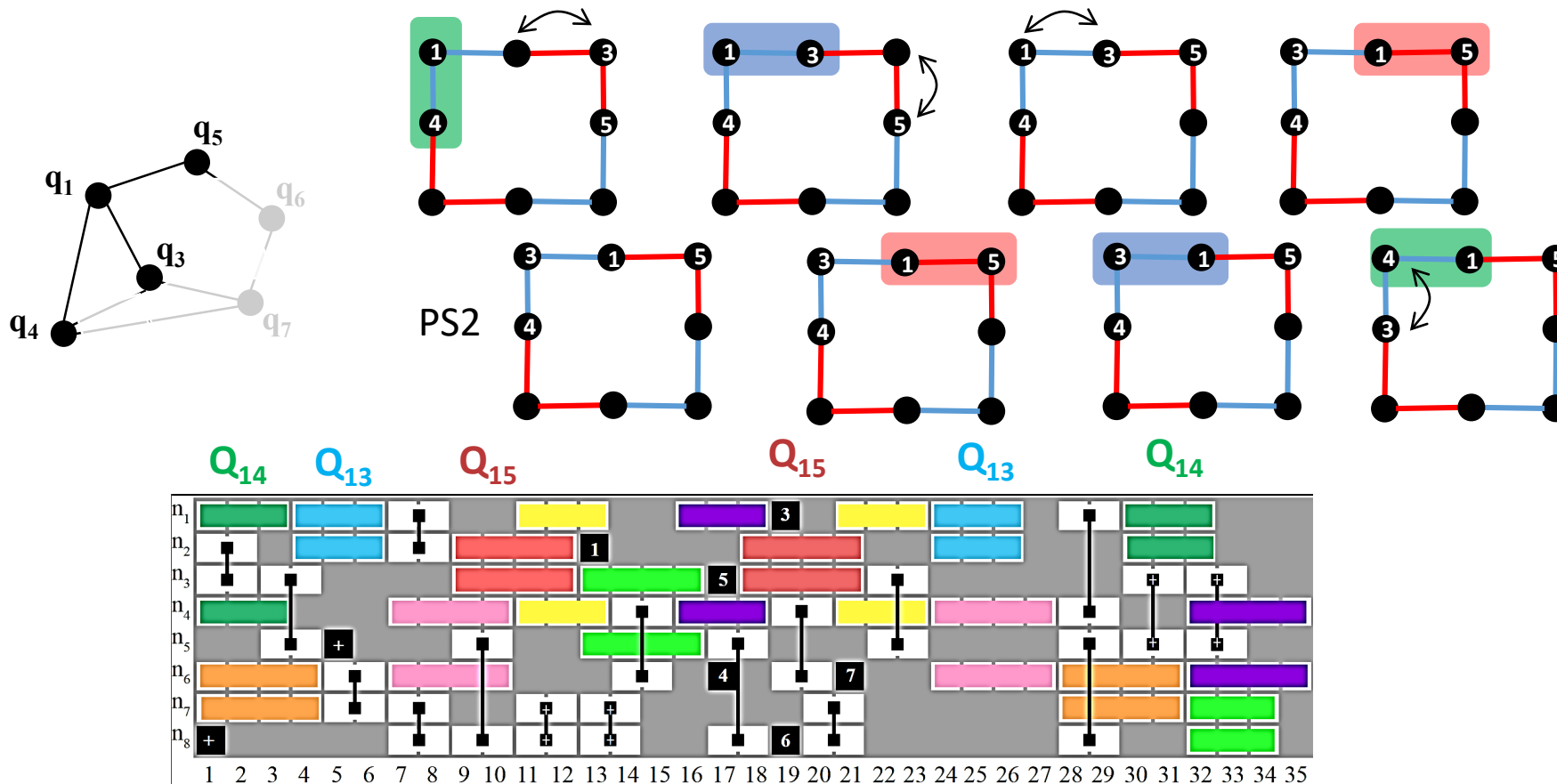
Circuit Execution Schedule



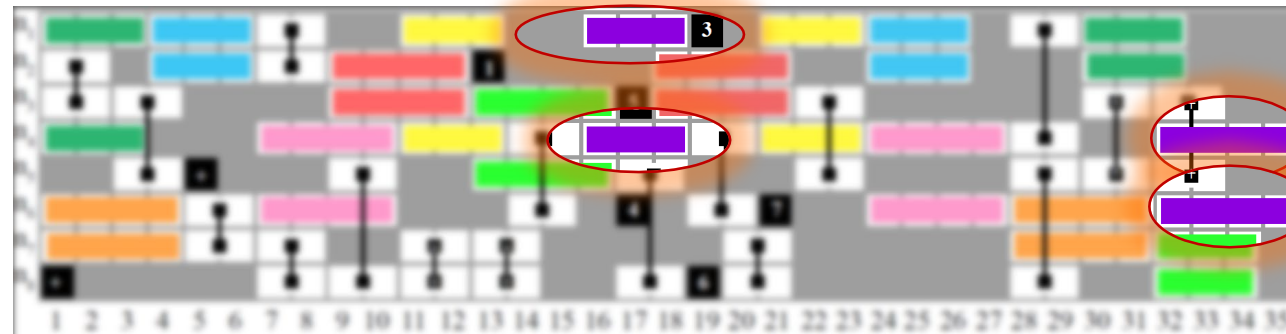
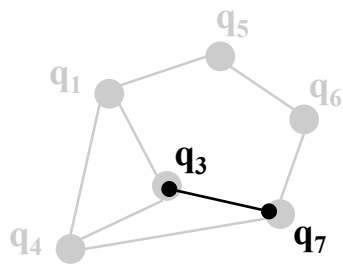
All actions of round 1 are completed – qubit can be mixed.
Qubit 1 can start participating to round 2.



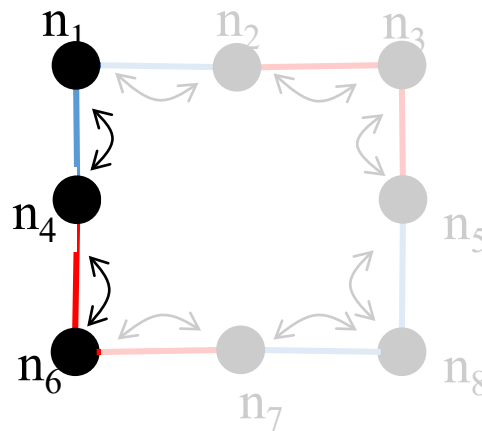
Circuit Execution Schedule



Circuit Execution Schedule

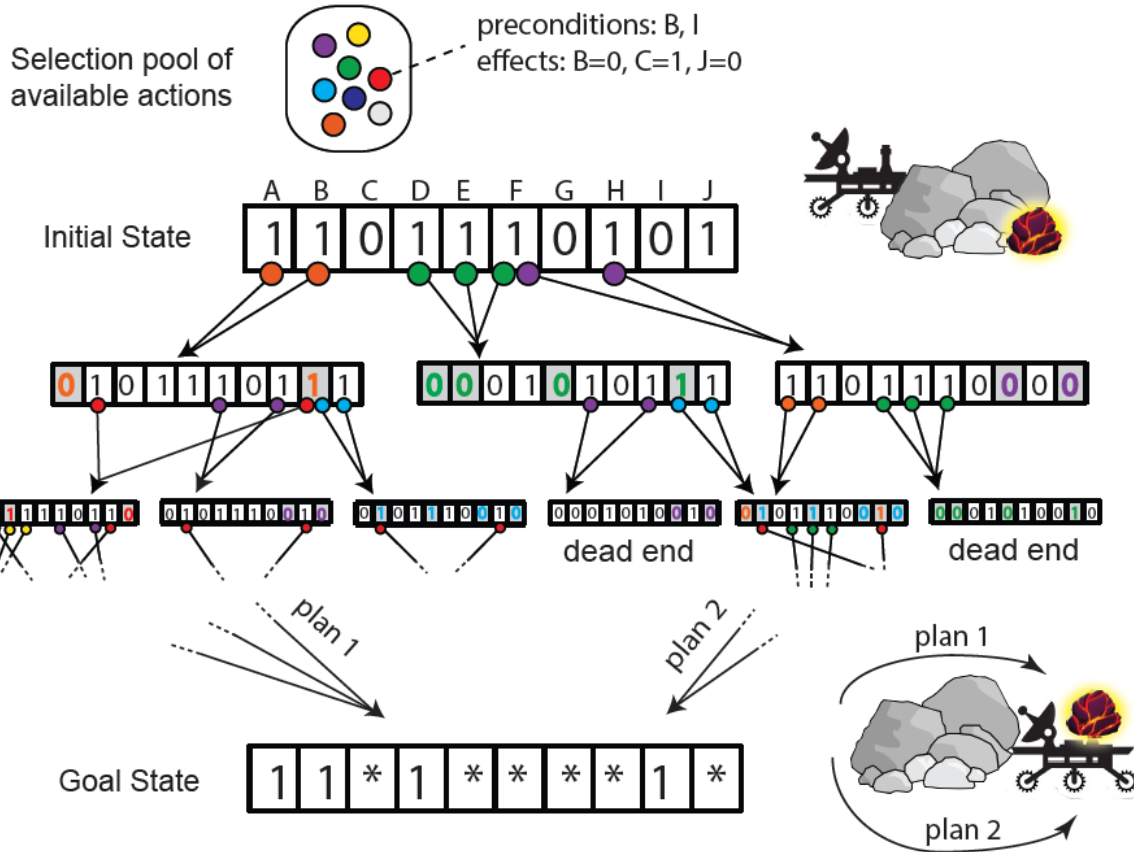


P-S(3,7) is **fast** on n_1, n_4
 P-S(3,7) is **slow** on n_4, n_6



How to obtain these schedules efficiently?

Temporal Planning

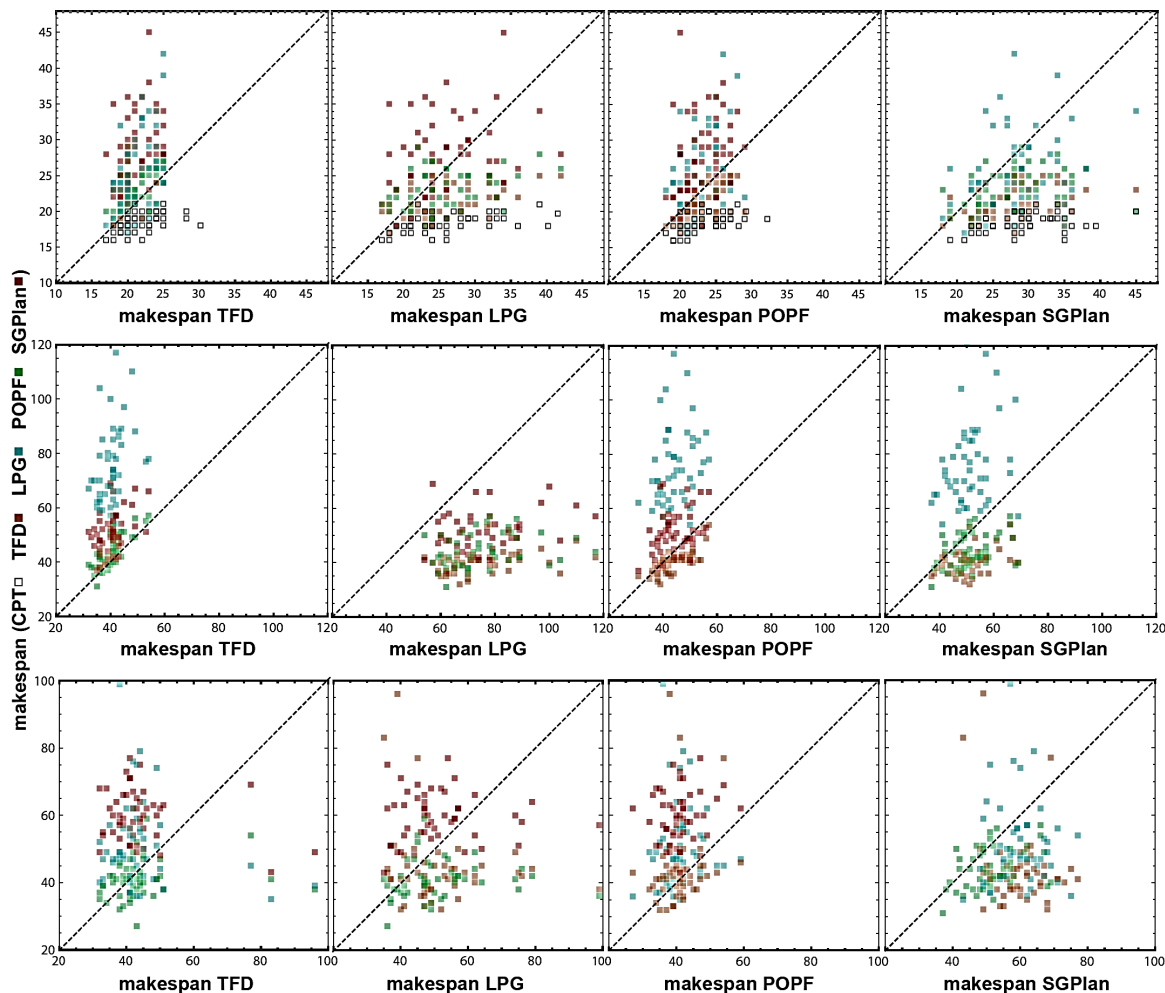


Planning Domain Specification Language (PDDL)

```
(:durative-action swap_1_2
  :parameters (?q1 - qstate ?q2 - qstate)
  :duration (= ?duration 2)
  :condition (and (at start (located_at_1 ?q1))
                  (at start (located_at_2 ?q2)))
  :effect (and (at start (not (located_at_1 ?q1)))
               (at start (not (located_at_2 ?q2)))
               (at end (located_at_1 ?q2))
               (at end (located_at_2 ?q1))))
```



Planning & Compilation



LPG (Gerevini 2003): *Local Search*

SGPlan (Wah 2004): *Subproblem planning decomposition*

CPT (Vidal 2006): *Partial-Order-Causal Link (POCL)*

TFD (Eyerich 2009): *Fast-Forward Search (FFS) + postprocessing*

POPF (Coles 2010): *POCL+FFS+Simple Temporal Network*

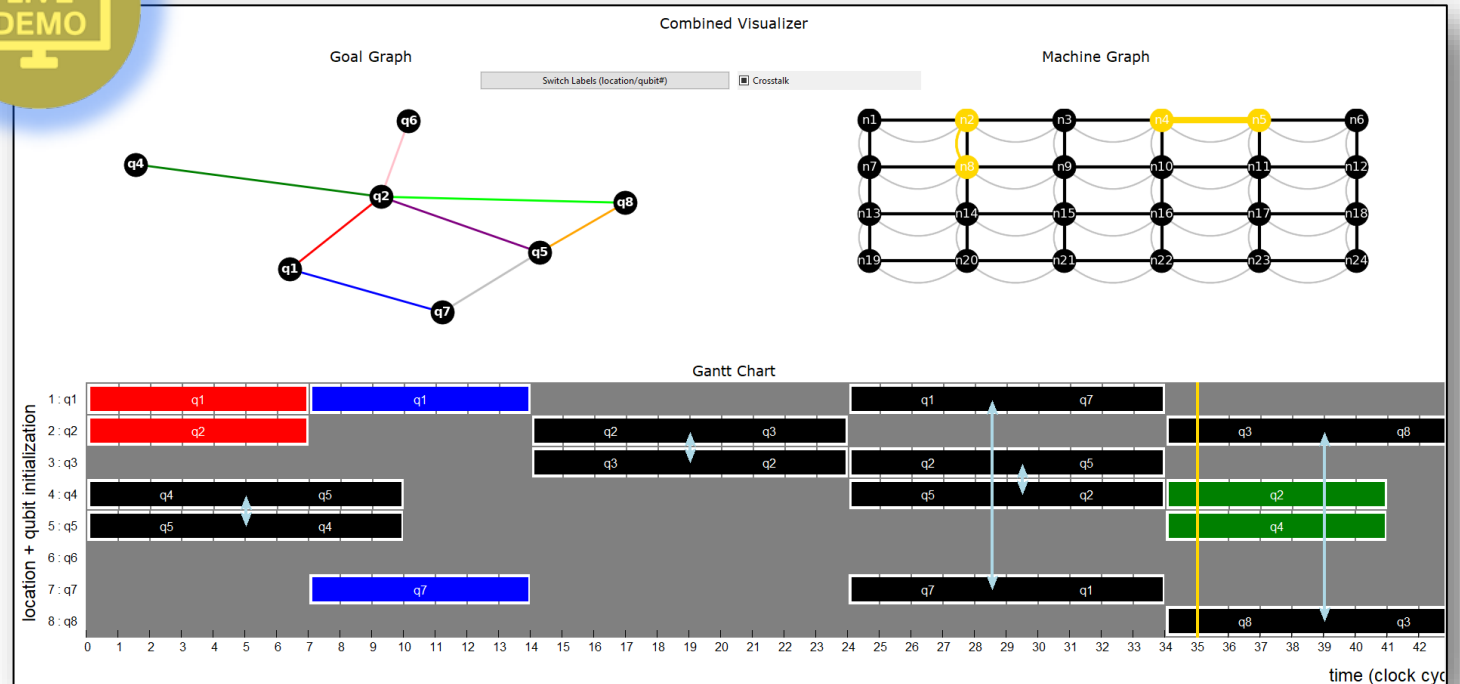
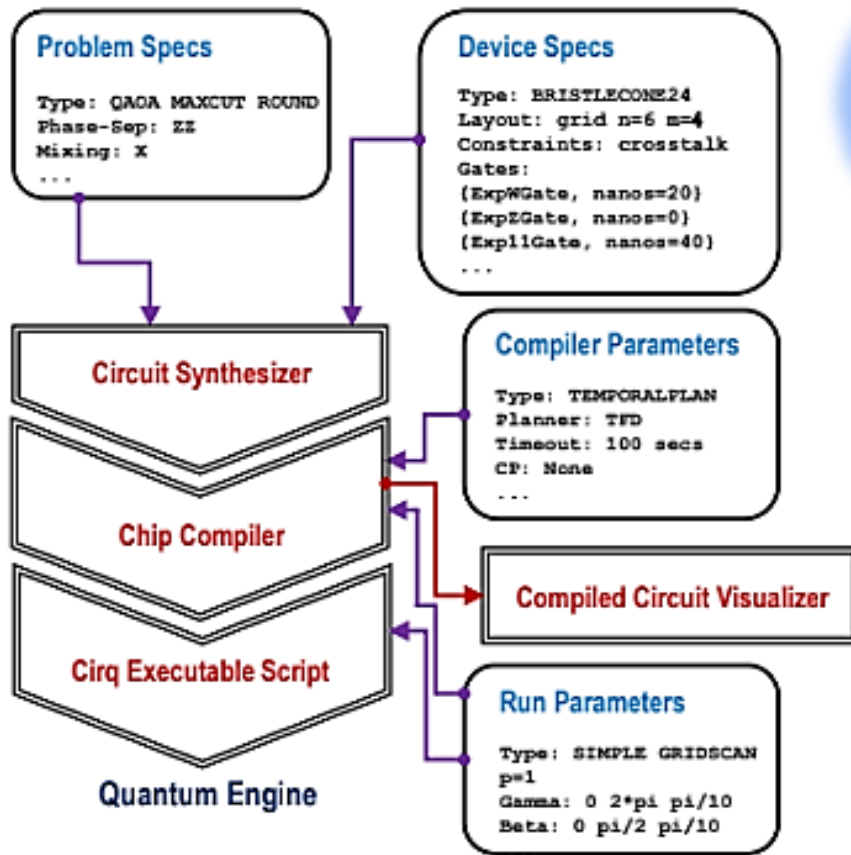
Plan quality relative comparison

	p=1, N8		p=1, N21		p=2, N8	
Utilization u	0.9	1.0	0.9	1.0	0.9	1.0
LPG	0.88	0.89	0.91	0.87	0.50	0.52
TFD	0.87	0.87	0.95	0.90	0.99	0.99
SGPlan	0.67	0.68	0.64	0.67	0.75	0.79
POPF	0.81	0.81	0.90	0.94	0.87	0.91

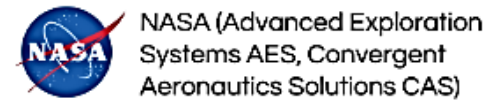
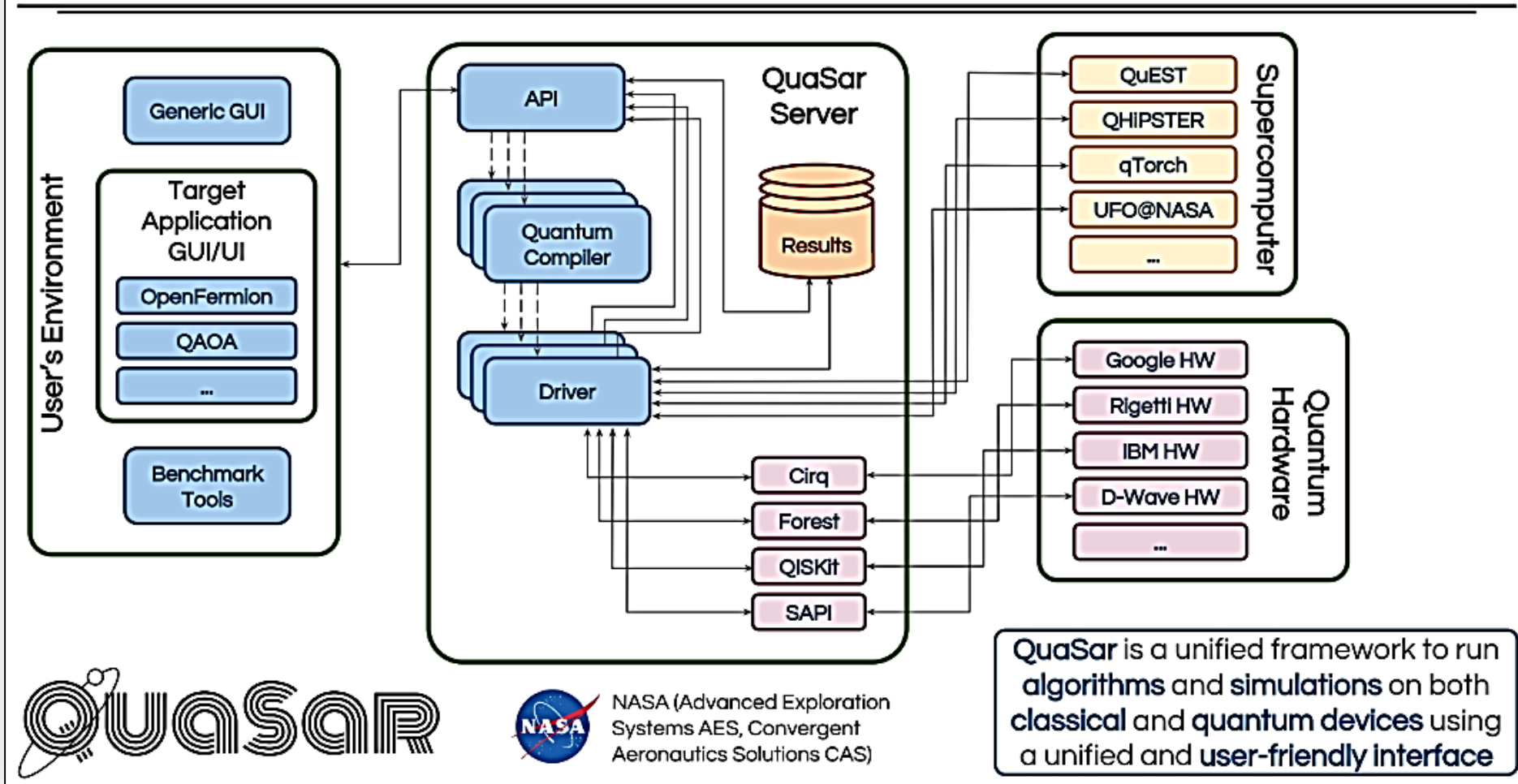
(CPT is provably optimal but timing scales badly)

CIRQ Toolchain

<https://ai.googleblog.com/2018/07/announcing-cirq-open-source-framework.html>



Quantum user-assisting Software for applied research (QuaSar)



Contacts



Davide Venturelli, Ph.D.
USRA Research Institute for Advanced Computer Science
Quantum Artificial Intelligence Laboratory
NASA Ames Research Center
davide.venturelli@nasa.gov

