Solving Higher Order Binary Optimization Problems on NISQ Devices: Experiments and Limitations

Valentin Gilbert[1], Julien Rodriguez[1,2], Stéphane Louise[1], Renaud Sirdey[1]

[1] *Université Paris-Saclay, CEA-List* F-91120, Palaiseau, France

[2] *Université Bordeaux, INRIA-Tadaam* Bordeaux, France

{valentin.gilbert, julien.rodriguez, stephane.louise, renaud.sirdey}@cea.fr

Context and Approach

Context

Consequences of mapping higher order term on quantum computers:

- QA: requires a QUBO reduction, which adds extra variables. It leads to dense QUBOs and additional factors on couplers.
- QAOA: requires an efficient swapping strategy.

Higher order term reduction by substitution (Rosenberg I.G. [1]):

$$b_i b_j b_k \rightarrow b_a b_k + b_i b_j - 2b_i b_a - 2b_j b_a + 3b_a$$

What is the performance of QA and QAOA solving HOBO problems ?

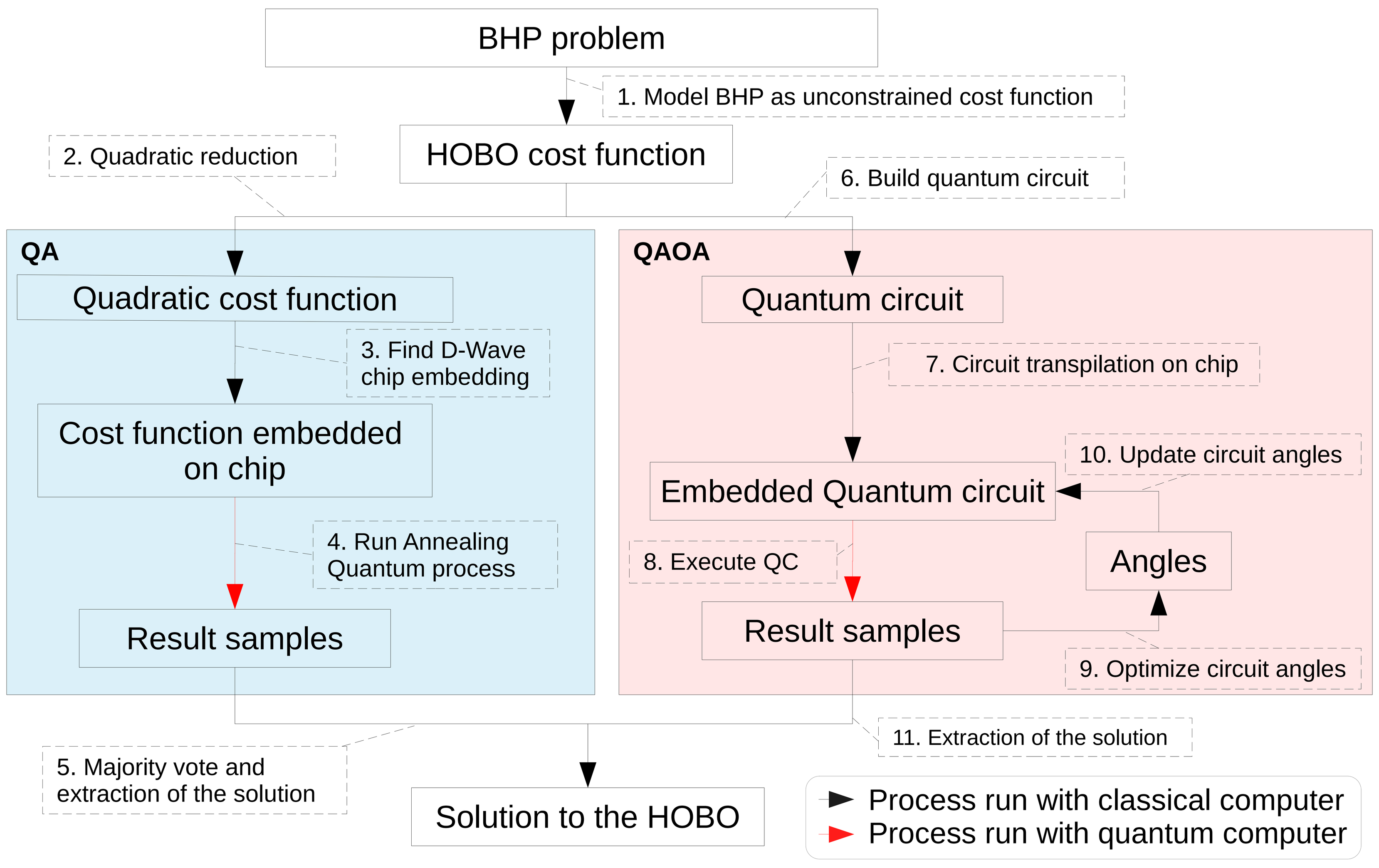

Fig 1. QA and QAOA workflow.

Related work

Formulations of the graph partitioning problem:

- Lucas A. [2]: Graph bi-partitioning
- Ushijima H. et al. [3]: Graph k-partitioning
- Rodriguez J. [4]: Hypergraph bi-partitioning

Recursive Hypergraph Partitioning

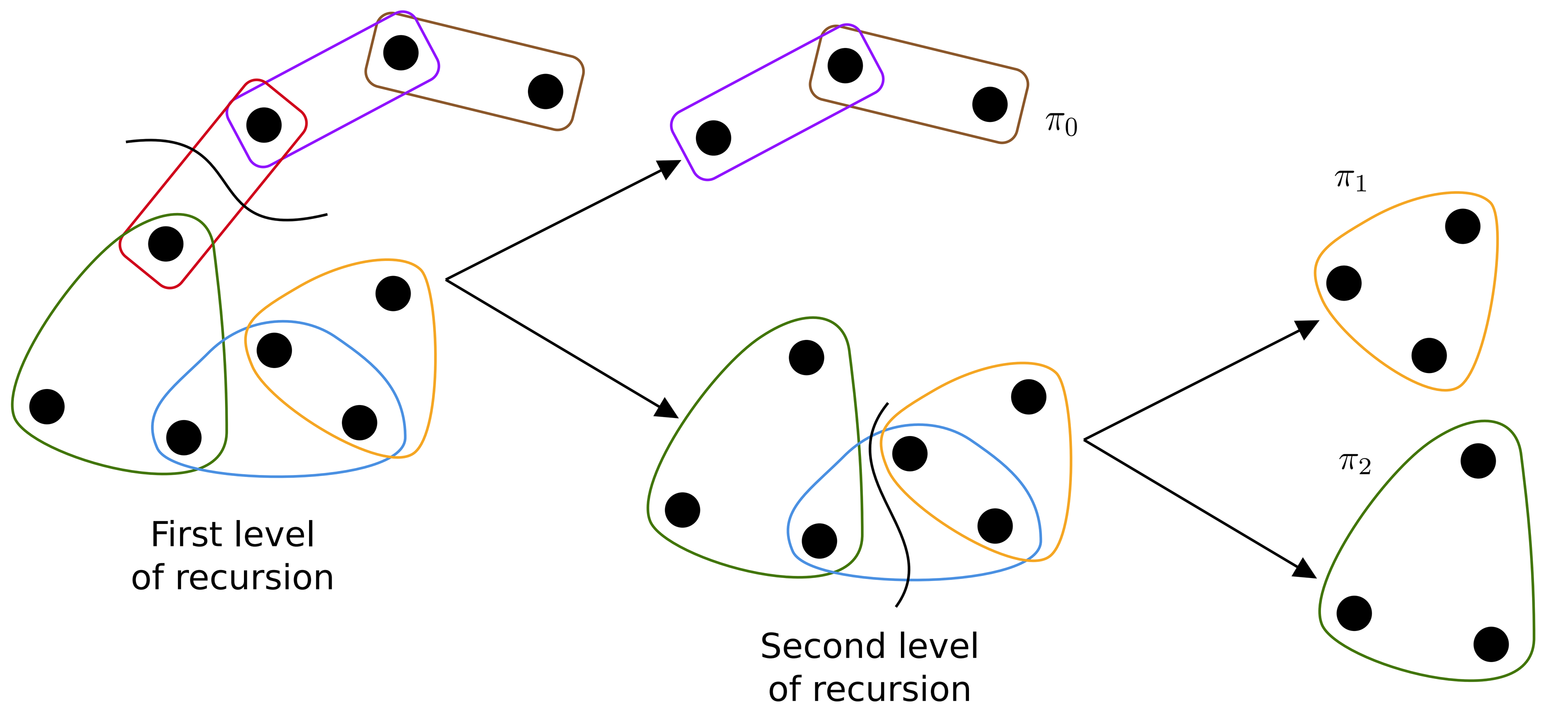

Fig 2. Recursive bi-partitioning example.

Mathematical Model

$x_v \in \{0,1\}$: vertex part

$\omega_v \in \mathbb{R}$: vertex weight

$\omega_e \in \mathbb{R}$: edge weight

k: number of parts

$\Omega(\mathcal{V}) = \sum_{v\in\mathcal{V}} \omega_v$: vertices weight

Balancing constraint, for each level of recursion:

$$H_A = \left(\sum_{v\in\mathcal{V}} \omega_v x_v - \left[\left\lfloor \frac{k}{2} \right\rfloor / k \times \Omega(\mathcal{V})\right]\right)^2 \tag{1}$$

This expression weights the number of nodes that should appear in π_0 and $\pi_1 \cup \pi_2$ according to their total weights (see Fig 2).

Cut cost:

$$H_B = \sum_{e\in\mathcal{E}} \left(\omega_e \times \left(1 - \prod_{v\in e} x_v - \prod_{v\in e}(1 - x_v)\right)\right) \tag{2}$$

For example, the red hyperedge is cut with a penalty equal to:

$$1 - (x_u x_v) - ((1-x_u)(1-x_v)) = 1 \tag{3}$$

Final objective function to minimize:

$$C(x) = AH_A + BH_B \tag{4}$$

Experiment and Results

Experimental setup and instances

QA settings (D-Wave):

- Quadratization: Rosenberg I.G. [1]
- Embedding: Cai J. et al. [5]
- Tested annealers: DW_2000Q, Adv4.1, Adv6.1, Adv2 prototype
- Post-processing method: Majority vote
- Chain strength: set experimentally

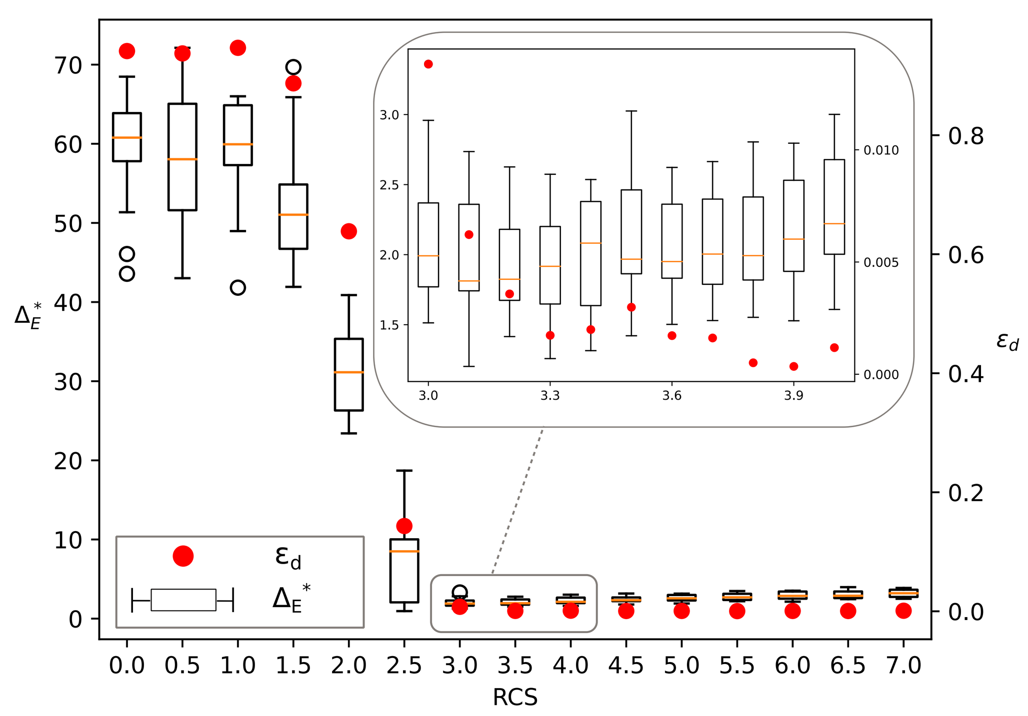

Fig 3. Relative Chain Strength setting. Δ_E^* is the minimum energy gap and ϵ_d is the duplication error rate.

QAOA settings:

- Global optimization method: FOURIER$[\infty, 10]$ [6]
- Local optimization method: Nelder-Mead [7]
- Simulation type (Qiskit *Aer*):
  - Perfect simulation
  - Simulation with Pauli noise
- Transpilation: Qiskit (optim level=1)

Instances set: 3 groups of uniform hypergraphs with $|\mathcal{V}| = 10$ and $|\mathcal{E}| = 15$. Each group is composed of 15 instances that are connected and randomly generated.

Instance	QA qubit overhead (Adv2 prototype)	QAOA depth overhead	
		cycle	grid
2-uniform	175%	173%	178%
3-uniform	175%	159%	166%
4-uniform	581%	155%	135%

Table 1. Overheads induced by the compilation.

Results

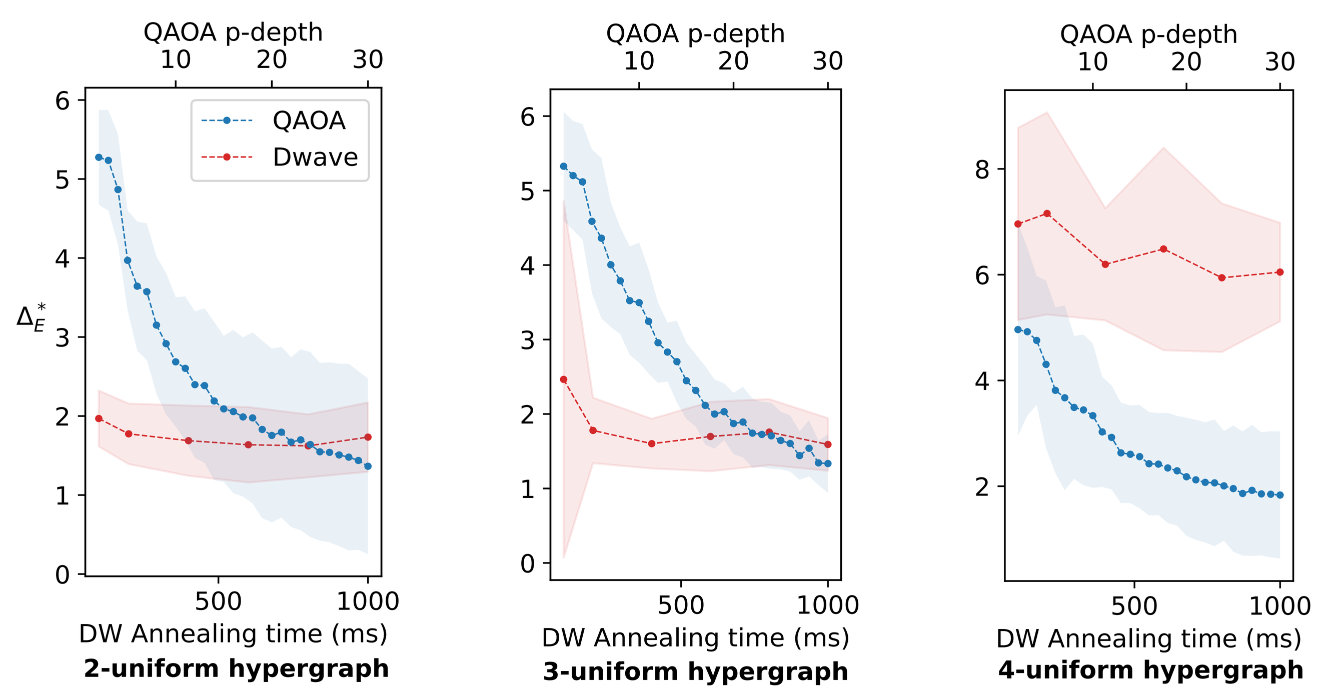

Fig 4. QA and QAOA on uniform graphs

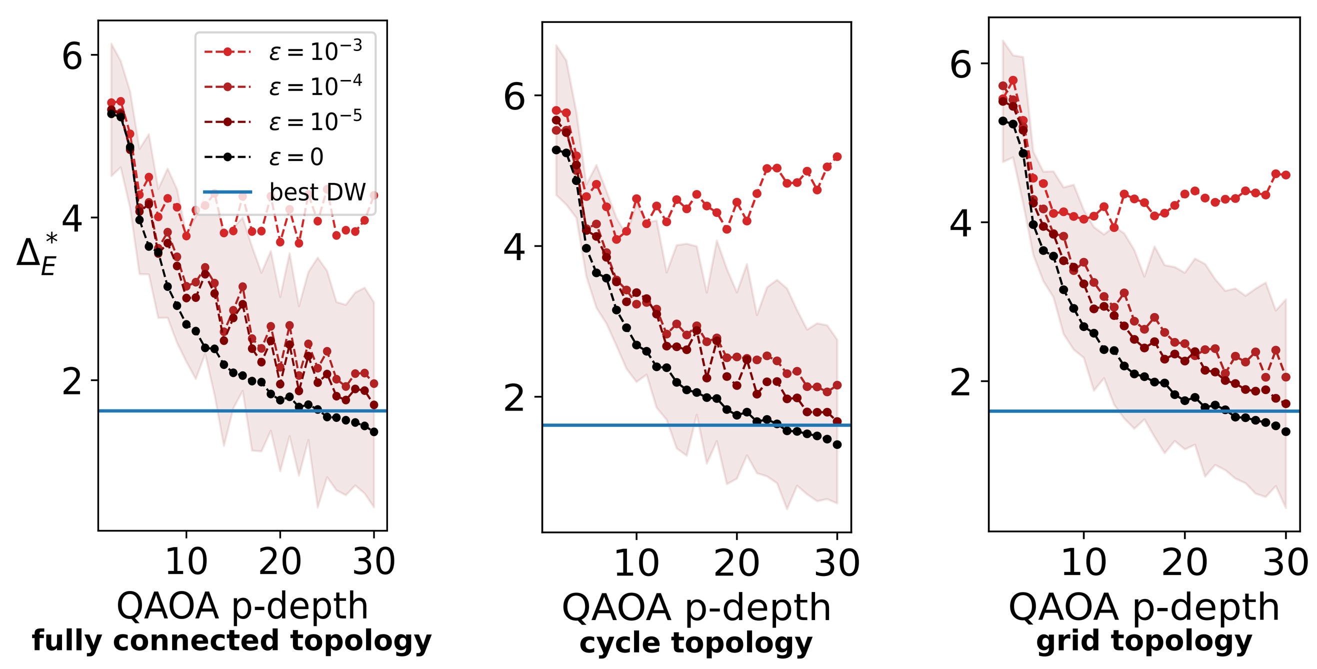

Fig 5. Noisy simulations of QAOA on 2-uniform graphs

Bibliography

[1] Rosenberg, I.G.: Reduction of bivalent maximization to the quadratic case. Cahiers du Centre d'Etudes de Recherche Operationnelle 17, 71–74 (1975)
[2] Lucas, A.: Ising formulations of many np problems. Frontiers in physics p. 5 (2014)
[3] Ushijima-Mwesigwa, H., Negre, C.F., Mniszewski, S.M.: Graph partitioning using quantum annealing on the d-wave system. In: Proceedings of the PMES. pp. 22–29 (2017)
[4] Rodriguez, J.: Quantum algorithms for hypergraph bi-partitioning. 23rd ROADEF . INSA Lyon, Villeurbanne - Lyon, France (Feb 2022), https://hal.archives-ouvertes.fr/hal-03595234
[5] Cai, J., Macready, W.G., Roy, A.: A practical heuristic for finding graph minors. arXiv preprint arXiv:1406.2741 (2014)
[6] Zhou, L., Wang, S.T., Choi, S., Pichler, H., Lukin, M.D.: Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. Physical Review X 10(2), 021067 (2020)
[7] Gao, F., Han, L.: Implementing the nelder-mead simplex algorithm with adaptive parameters. Computational Optimization and Applications 51(1), 259–277 (2012)