PURDUE UNIVERSITY

Quantum-Classical Hybrid Methods for Optimization*

David E. Bernal Neira

Principal Investigator of SECQUOIA

Assistant Professor - Davidson School of Chemical Engineering, Purdue University

May 28th, 2024

Master Class - Quantum Computing for CP, AI, and OR, and vice-versa

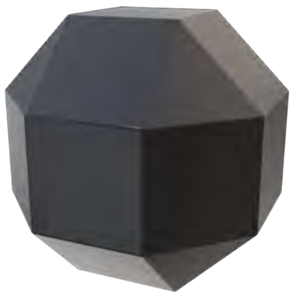
CPAIOR 2024, Uppsala, Sweden

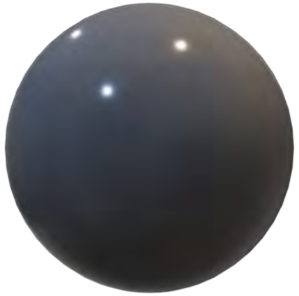
Discrete Nonlinear Optimization – What is it and how do we solve it?

Optimization
Modeling the real world

The real-world is abrupt, unexpected, discontinuous, non-smooth, and our physical models are **nonlinear**!

Our mathematical programs need to reflect these conditions





$$\min_{\mathbf{x}=\{x_1,\dots,x_n\}} Obj(x_1,\dots,x_n)$$
s.t. $Const(x_1,\dots,x_n) \le 0$
Some or all x_i are integer

Although we can model non-convexities with discrete variables, and some quantities are continuous, we can represent more information via nonlinear constraints and objectives

- Assumption that

$$Obj(x_1,\dots,x_n) = f(\mathbf{x}) = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$
$$Const_1(x_1,\dots,x_n) = g_1(\mathbf{x}): a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \le b_1$$
$$\vdots$$
$$Const_m(x_1,\dots,x_n) = g_m(\mathbf{x}): a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \le b_m$$

does not hold in general

Resulting Discrete Nonlinear Problem is computationally hard!

Discrete Nonlinear Optimization

What can you formulate with MINLP?

Formulation

$$\min_{\mathbf{x},\mathbf{y}} \quad f(\mathbf{x},\mathbf{y})$$
$$\text{s.t.} \quad g_j(\mathbf{x},\mathbf{y}) \leq 0$$
$$\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} \leq \mathbf{b},$$
$$\mathbf{x} \in \mathbb{R}^n, \ \mathbf{y} \in \mathbb{Z}^m.$$

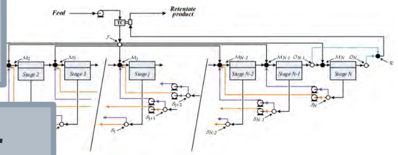
Objective function

Nonlinear Constraints

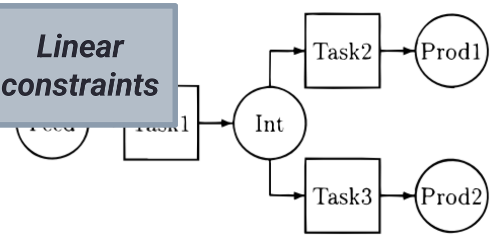
Linear constraints

Continuous variables

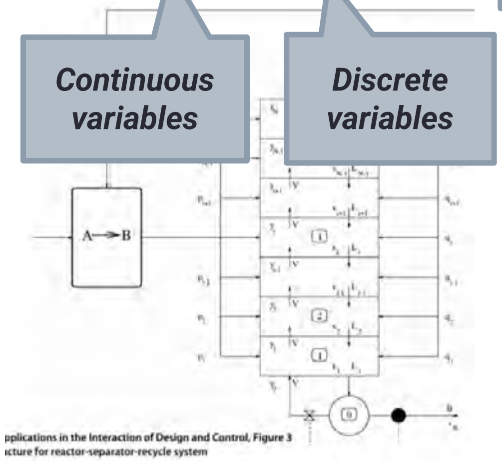
Discrete variables



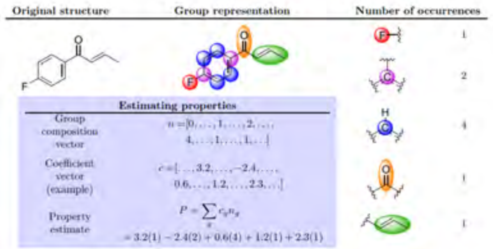
Flowsheet optimization[2]



Process planning and scheduling[3]

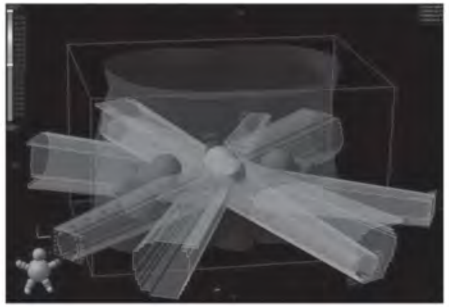


Simultaneous process control and design[4]

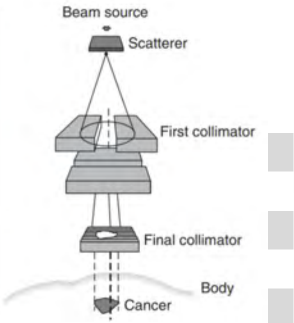


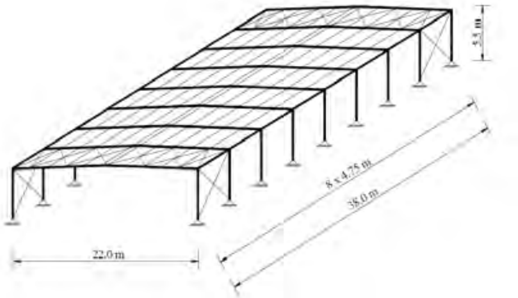
Molecular structure encoding for MINLP modeling[5]

Applications in PSE[1]



Gamma knife cancer treatment[6]





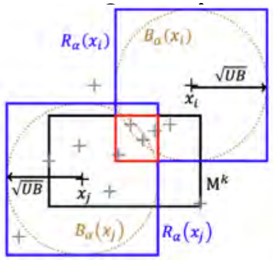
Topology of a single-story building[7]

ng

e optim

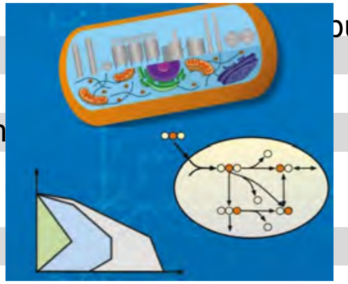
trol

PC

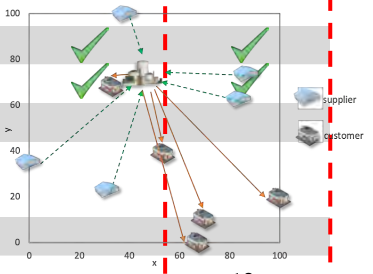


K-centroid clustering in ML[8]

- Nonlinear MPC
- Hybrid



Computational Biology[9]



Network design[10]

Molecular design

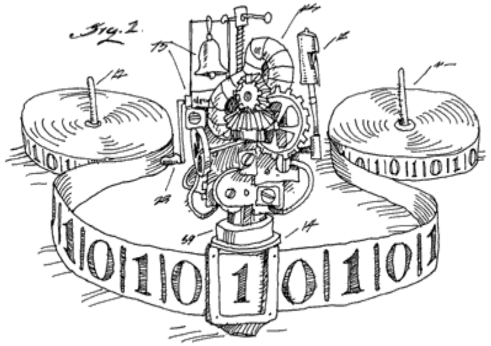
1. Modified from Biegler, L. T.; Grossmann, I. E. "Retrospective on optimization.". 2004
2. Chavez, J.A., Gooty, R., Tawarmalani, M. Agrawal, R.. "Optimal Design of Membrane Cascades for Gaseous and Liquid Mixtures via MINLPs." 2021
3. L. Mockus and G. V. Reklaitis "Continuous Time Representation Approach to Batch and Continuous Process Scheduling. 1. MINLP Formulation." 1999
4. Schweiger, C., Floudas, C. "MINLP: Applications in the Interaction of Design and Control" 2008
5. Austin, N., Sahinidis, N., Trahan, D. "Computer-aided molecular design: An introduction and review of tools, applications, and solution techniques" 2017
6. Cao, W. ,Lim, G. J. "Optimization models for cancer treatment planning" 2011
7. Silih, S., Zula, T., Kravanja, S. "The MINLP Optimization in civil engineering" 2007
8. Shi, Mingfei, et al. "Global optimization of k-center clustering." ICML. PMLR, 2022.
9. Maranas, C.D., Zomorrodi, A.R. "Optimization Methods in Metabolic Networks" 2016
10. Lara, C., Bernal D.E., Li, C., Grossmann, I.E. "Global Optimization Algorithm for Multi-period Design and Planning of Centralized and Distributed Manufacturing Networks" 2019

Discrete Nonlinear Optimization

What can you formulate with MINLP?

Formulation

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{y}} \quad & f(\mathbf{x},\mathbf{y}) \\
\text{s.t.} \quad & g_j(\mathbf{x},\mathbf{y}) \leq 0 \\
& \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} \leq \mathbf{b}, \\
& \mathbf{x} \in \mathbb{R}^n, \ \mathbf{y} \in \mathbb{Z}^m.
\end{aligned}
$$



Representation of Turing Machine[1]

"***Most industrial processes** can be formulated as MINLP. Its expressive power is remarkable: it can encode any Turing Machine, including universal ones, such as Minsky's Recording Machine, which means that **every problem** can be formulated as a MINLP*.*"

- L. Liberti, Mathematical Programming. 2017

MINLP is **NP-Hard** since:

$$\text{SAT} \rightarrow \text{BIP} \subset \text{ILP} \subset \text{MILP} \subset \text{MINLP}$$

Although one can prove they can be undecidable!

- Theorem 3 in Liberti, L. and Martinelli, F. "Mathematical programming: Turing completeness and applications to software analysis". 2014
- Problem 2, main Theorem in Karp, R.M. "Reducibility Among Combinatorial Problems". 1972

1. COSC 545 - Theory of Computation, Georgetown University. Retrieved from http://people.cs.georgetown.edu/~cnewport/teaching/cosc545-spring14/ on 02/17/2019.

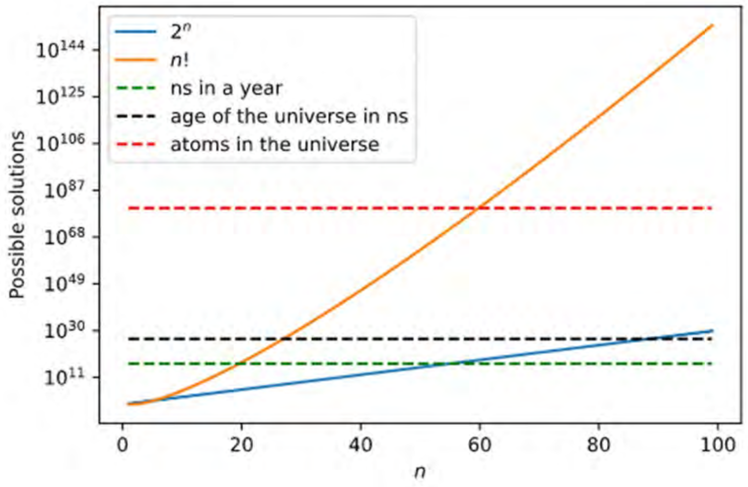
Discrete Nonlinear Optimization

How hard can solving these problems be?

Enumerating solutions

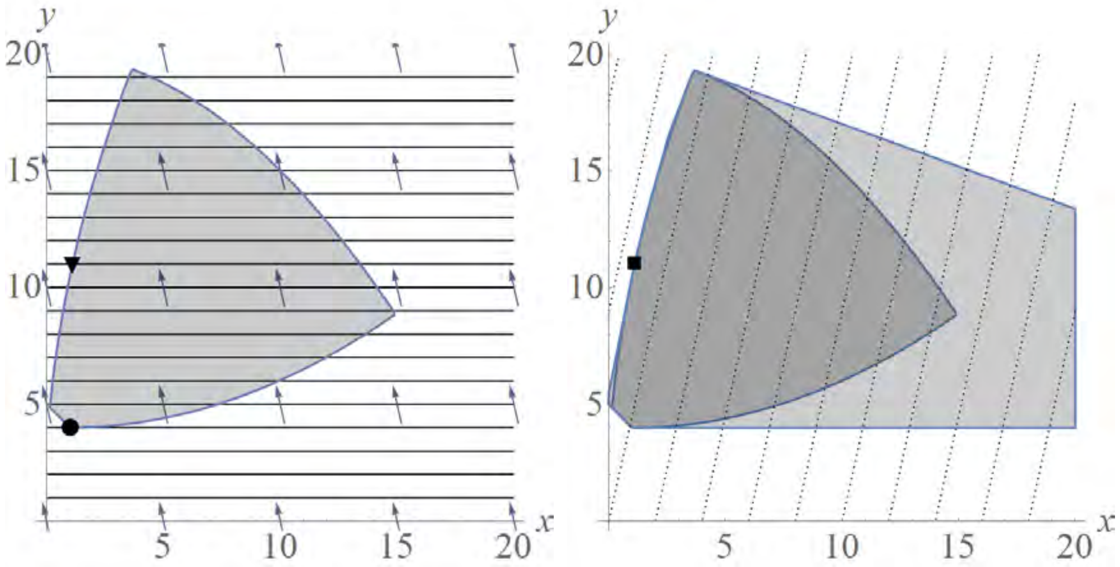
Look at **all the possible values**, check if they **satisfy the constraints** and **compare their objective**?

- Only binary variables, solutions are 2^n
- Permutations, solutions grow as $n!$



Decomposition methods

Split problems into **easily solvable** parts (**subproblems**) and **design algorithms** that put those parts together to find the solution to the original problem.



Decomposition algorithms for MIP

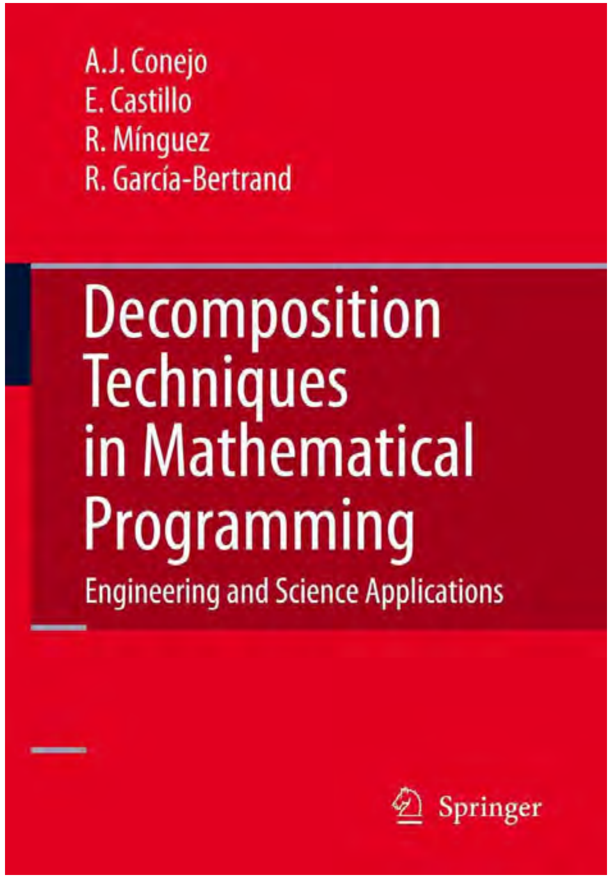
Which are these subproblems? Easier problems

- Continuous subproblems are usually solved (fixing discrete variables makes problem way easier)
- Surprisingly, in many cases, the subproblems can also be MIP!
  - Speedup between CPLEX 1.2 (1991) and CPLEX 11 (2007): **29,000 times**
    - Gurobi 1.0 (2009) comparable to CPLEX 11
  - Speedup between Gurobi 1.0 and Gurobi 5.5 (2013): **20 times**
  - Speedup between Gurobi 5.5 and Gurobi 11 (2024): **6 times**
- Total algorithmic speedup **1991-2024**: **3'450,000 times**
  - Together with computer improvements authors are not afraid of claiming **200 billion speedups**!
- Where is this improvement coming from?
  - CPAIOR!

Feature	Speedup factor
Cuts	54
Preprocessing	11
Branching variable selection	3
Heuristics	1.5

1. Bixby. A brief history of linear and mixed-integer programming computation. Documenta Mathematica, Extra Volume: Optimization Stories, pages 107–121, 2012.
2. Nemhauser. Integer programming: the global impact. Presented at EURO, INFORMS, Rome, Italy, 2013.
3. https://www.gurobi.com/solutions/gurobi-optimizer/prior-version-enhancements/
4. Bertsimas. "Statistics and machine learning via a modern optimization lens." INFORMS Annual Meeting. 2014.
5. Conforti, Cornuéjols, Zambelli. Integer programming models. Springer International Publishing, 2014.

Decomposition Techniques in Mathematical Programming



Preface

Optimization plainly dominates the design, planning, operation, and control of engineering systems. This is a book on optimization that considers particular cases of optimization problems, those with a decomposable structure that can be advantageously exploited. Those decomposable optimization problems are ubiquitous in engineering and science applications. The book considers problems with both complicating constraints and complicating variables, and analyzes linear and nonlinear problems, with and without integer variables. The decomposition techniques analyzed include Dantzig-Wolfe, Benders, Lagrangian relaxation, Augmented Lagrangian decomposition, and others. Heuristic techniques are also considered.

Additionally, a comprehensive sensitivity analysis for characterizing the solution of optimization problems is carried out. This material is particularly novel and of high practical interest.

This book is built based on many clarifying, illustrative, and computational examples, which facilitate the learning procedure. For the sake of clarity, theoretical concepts and computational algorithms are assembled based on these examples. The results are simplicity, clarity, and easy-learning.

We feel that this book is needed by the engineering community that has to tackle complex optimization problems, particularly by practitioners and researchers in Engineering, Operations Research, and Applied Economics. The descriptions of most decomposition techniques are available only in complex and specialized mathematical journals, difficult to understand by engineers.

1. Conejo, Antonio J., et al. Decomposition techniques in mathematical programming: engineering and science applications. Springer Science & Business Media, 2006.

Classical methods for MIP and MINLP

What have we learned?

- Classical methods are extraordinary!
  - Even when facing worst-case NP-Hard problems
- Decomposition methods are useful for taming complexity
  - Understanding the structure of the problem is key to success
  - Using the right tool for the right task pays off
  - Handwavy argument: If problem is NP-Hard, maybe it pays off to solve problems of half the size many times!

- Keep doing CPAIOR! We need it! (for classical and quantum)

1. Bixby. A brief history of linear and mixed-integer programming computation. Documenta Mathematica, Extra Volume: Optimization Stories, pages 107–121, 2012.
2. Nemhauser. Integer programming: the global impact. Presented at EURO, INFORMS, Rome, Italy, 2013.
3. https://www.gurobi.com/solutions/gurobi-optimizer/prior-version-enhancements/
4. Bertsimas. "Statistics and machine learning via a modern optimization lens." INFORMS Annual Meeting. 2014.
5. Conforti, Cornuéjols, Zambelli. Integer programming models. Springer International Publishing, 2014.

Quantum Methods for Discrete Optimization

Solving an ISING/QUBO problem

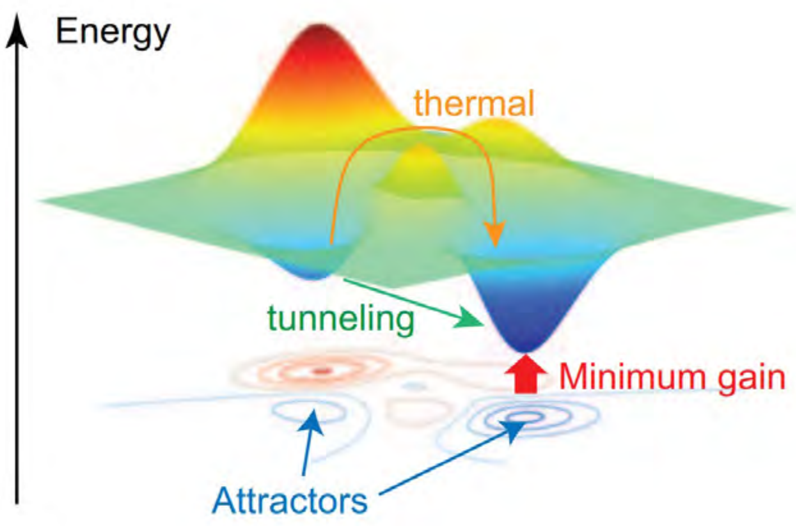
What is it? How?

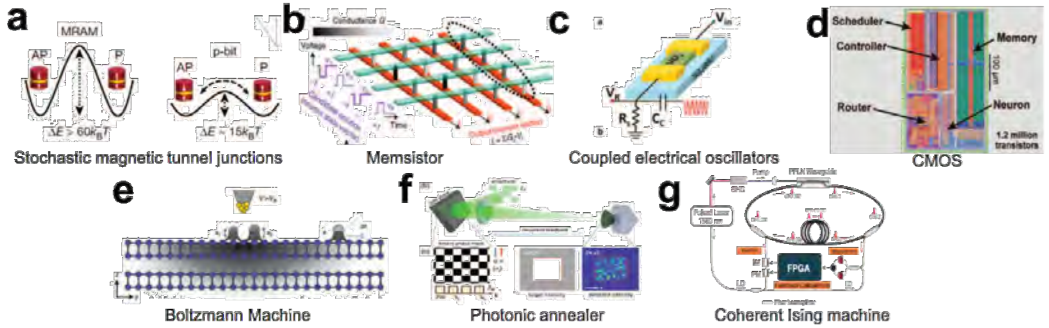
"Solving the Model"

- **Physicists:** identifying thermodynamics expectation values (find the distribution of solutions/energies)
- **Optimizers:** find the actual variable values that minimizes the objective function (find the ground state)

How to solve the Ising model? With an Ising solver!

- **Ising solver?** End-to-end solution method including software/algorithms and hardware/machines
  - **Classical Thermal Annealing**
  - Other classical algorithms
    - MIP or Physics-inspired classical algorithms
  - Hardware solution-based methods
    - **Coherent Ising Machines,** GPU, CMOS, …
  - Quantum Approaches
    - **Quantum Annealing**
    - **Hybrid Quantum-Classical Algorithms**
      - **Variational Quantum Algorithms – QAOA, VQE**

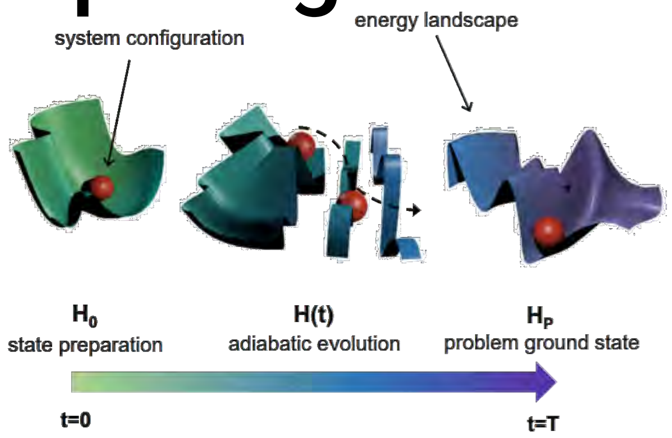




1. Mohseni, Naeimeh, Peter L. McMahon, and Tim Byrnes. "Ising machines as hardware solvers of combinatorial optimization problems." Nature Reviews Physics 4.6 (2022): 363-379.

Solving ISING/QUBOs using Quantum Computing

How?



Quantum Adiabatic Algorithm (QAA)

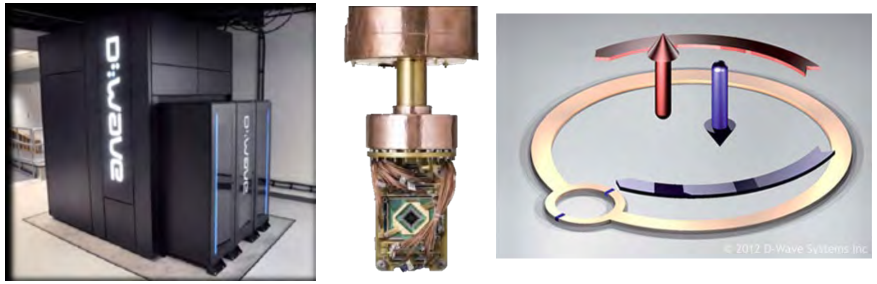
- Adiabatic evolution preserves energy ranking of states
- Start evolution from system at ground-state
- Evolve system adiabatically to one that encodes optimization problem

Quantum Annealing

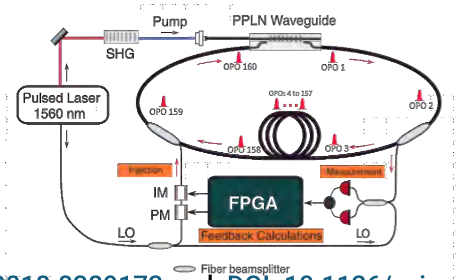
- They run a single quantum algorithm
- Physical implementation of idealized QAA
- Analog computation
- D-Wave quantum annealer is the best-known example

Myriad of physical or Physics-inspired methods

- Coherent Ising Machines, Simulated Bifurcation Machines, Digital annealers, …



Retrieved from www.dwavesys.com/



Retrieved from 10.1109/MSPEC.2018.8389173 and DOI: 10.1126/science.aah5178

1. Pierangeli, Davide, Giulia Marcucci, and Claudio Conti. "Adiabatic evolution on a spatial-photonic Ising machine." Optica 7.11 (2020): 1535-1543.
2. Salloum, Hadi, et al. "Integration of machine learning with quantum annealing." International Conference on Advanced Information Networking and Applications. Cham: Springer Nature Switzerland, 2024.

Solving ISING/QUBOs using Quantum Computing

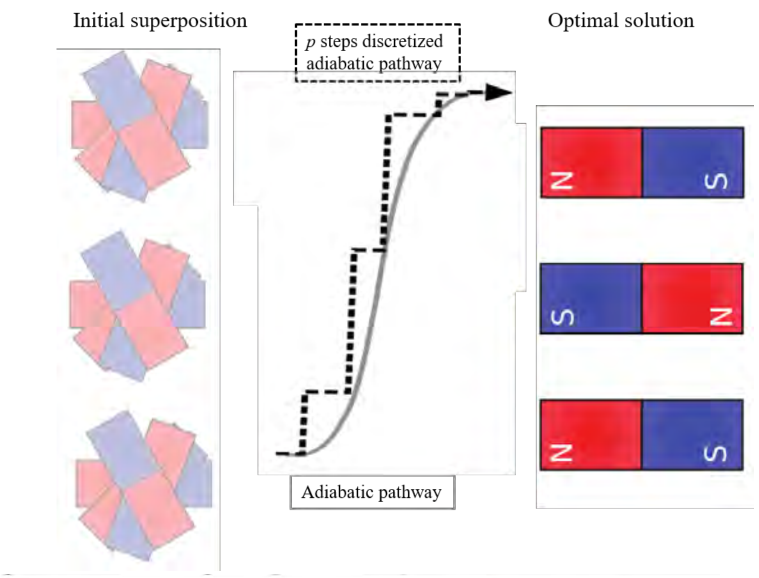
How?

Gate-based NISQ computers

- We can implement algorithms via noisy quantum operators (gates)
- For solving QUBOs, we can use variational algorithms like:
  - Quantum Approximate Optimization Ansatz (QAOA)
- For optimization, algorithms can be understood as discretized adiabatic computation

Fault tolerant Quantum Computers

- Amplitude Amplification
- Variational algorithms
- Hamiltonian Walks
- Grover's method



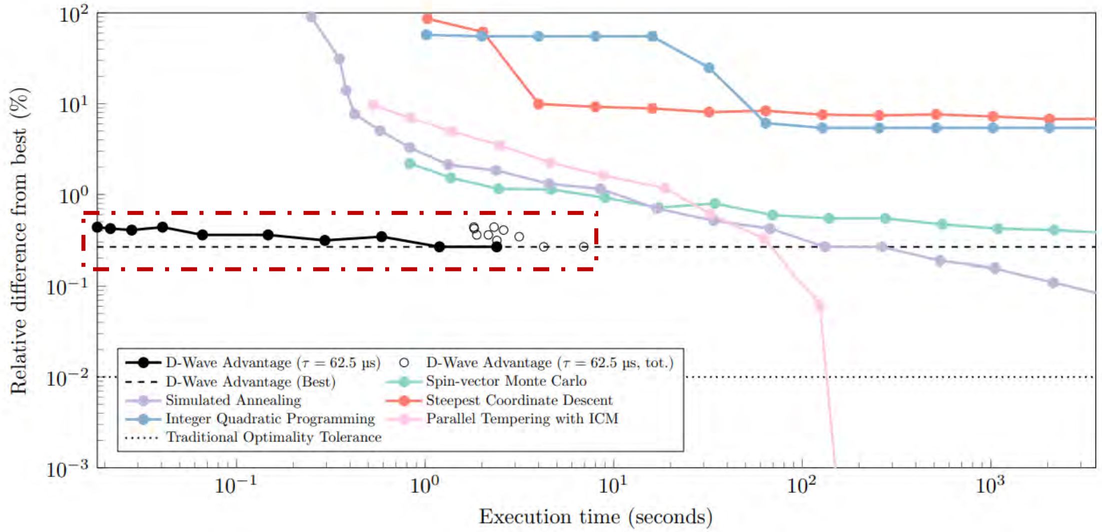
Problem	Algorithm primitive
L-term spin	Amplitude amplification (Sec. III A) QAOA/first-order Trotter (Sec. III B) Hamiltonian walk (Sec. III C) QSA/qubitized (Sec. III E) QSA/gap amplification (Sec. III F)
QUBO	Amplitude amplification (Sec. III A) QAOA/first-order Trotter (Sec. III B) Hamiltonian walk (Sec. III C) QSA/qubitized (Sec. III E) QSA/gap amplification (Sec. III F)

1. Sanders, Yuval R., et al. "Compilation of fault-tolerant quantum heuristics for combinatorial optimization." PRX Quantum 1.2 (2020): 020312.

Solving ISING/QUBOs using Quantum Computing

How good?

Quantum Annealing



Empirical performance advantages [1]

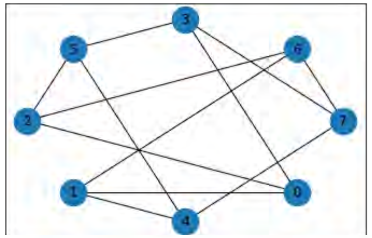
Limited by size and connectivity

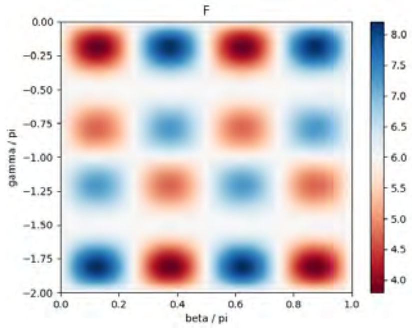
Understood as Heuristic methods

Gate-based computing (QAOA)

For quite general problems (MAXCUT 3-reg-degree graphs) provides provably better approximation ratio than random guessing [2].

For a particular problem (SAT E3Lin2), QAOA provided the best approximation ratio [3].





Interesting theoretical guarantees

Even more limited by size and noise

Approximation algorithms

1. Tasseff, Byron, et al. "On the emerging potential of quantum annealing hardware for combinatorial optimization." arXiv preprint arXiv:2210.04291 (2022).
2. Farhi, Edward, Jeffrey Goldstone, and Sam Gutmann. "A quantum approximate optimization algorithm." arXiv preprint arXiv:1411.4028 (2014).
3. Hadfield, Stuart, et al. "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz." Algorithms 12.2 (2019): 34.

Where do these ISING/QUBO come from?
Where are the nonlinear discrete problems in this talk?

Formulations of many discrete optimization problems in Ising/QUBO

- Mappings of Karp's 21 NP-Complete problems to Ising and others

"Reformulation" of discrete nonlinear optimization to Ising/QUBO

Some mappings are inefficient

- Continuous variables – Discretize
- Integer variables – Binarize
- Nonlinear Equations – Quadratize
- Constraints – Penalize with ρ

Problem size growth

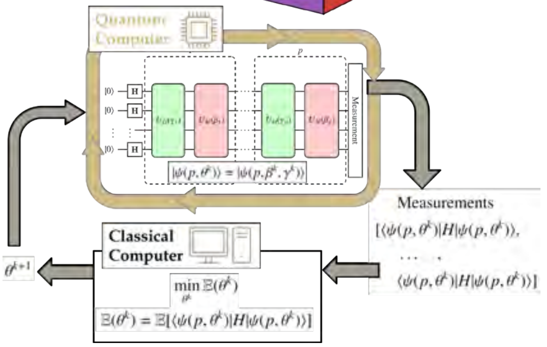
Extra variables and large coefficients

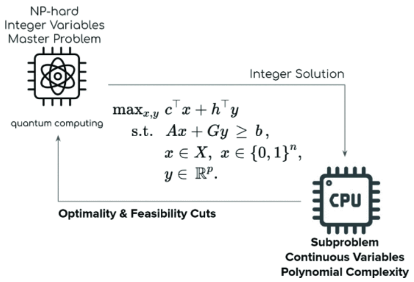
Extra variables

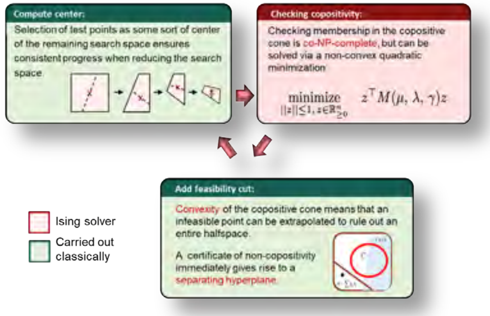
"Softening" of constraints

Decomposition Algorithms!

- Next slides!







1. Lucas, Andrew. "Ising formulations of many NP problems." Frontiers in physics 2 (2014): 5.
2. Xavier, Pedro Maciel, et al. "Qubo. jl: A julia ecosystem for quadratic unconstrained binary optimization." arXiv preprint arXiv:2307.02577 (2023).
3. Bernal, David E., et al. "Perspectives of quantum computing for chemical engineering." AIChE Journal 68.6 (2022): e17651.
4. Zhao, Zhongqi, Lei Fan, and Zhu Han. "Hybrid quantum benders' decomposition for mixed-integer linear programming." 2022 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2022.
5. Brown, Robin, et al. "A copositive framework for analysis of hybrid ising-classical algorithms." SIAM Journal on Optimization 34.2 (2024): 1455-1489.

QUBO.jl

A Julia Ecosystem for Ising and QUBO optimization

				
Pedro Maciel Xavier	Pedro Ripper	Tiago Andrade	Joaquim Dias Garcia	Nelson Maculan

Pedro Maciel Xavier[1,2], Pedro Ripper[1], Tiago Andrade[1], Joaquim Dias Garcia[1], Nelson Maculan[2], DEBN

[1]Energy Consulting and Analytics, PSR Inc.

[2]Department of Systems Engineering and Computer Science, UF Rio de Janeiro

PSR

Universidade Federal do Rio de Janeiro

USRA

NASA

https://github.com/psrenergy/QUBO.jl

1. Xavier, Pedro Maciel, et al. "Qubo. jl: A julia ecosystem for quadratic unconstrained binary optimization." arXiv preprint arXiv:2307.02577 (2023).

QUBO.jl

What do our Julia packages do?

Table 2 List of supported variable encodings and constraint penalization methods across QUBO modelling platforms

	ToQUBO.jl	PyQUBO	qubovert	Qiskit	OpenQAOA	Amplify
Automatic variable encoding methods implemented						
Binary [1]	■ ⋆	■	■			■ !
Unary [1]	■ ⋆	■	■	■	■	■ !
One-Hot [1]	■ ⋆	■				
Domain-Wall [2]	■ ⋆	■				
Bounded-Coefficient [3]	■ ⋆	■				
Arithmetic Progression	■ ⋆					■ !
Supported constraints						
$\mathbf{a}'\mathbf{x} \leq b$	■		■	■	■	■
$\mathbf{a}'\mathbf{x} = b$	■		■	■	■	■
$\mathbf{x}'Q\mathbf{x} + \mathbf{a}'\mathbf{x} \leq b$	■					■
$\mathbf{x}'Q\mathbf{x} + \mathbf{a}'\mathbf{x} = b$	■					■
SOS1	■					
$\bigwedge_i x_i$ †			■			■
$\bigvee_i x_i$ †			■			■
$\bigoplus_i x_i$ †						■

[1] (Tamura et al. 2021); [2] (Chancellor 2019); [3] (Karimi and Ronagh 2019);

† For logical constraints, it is assumed that $x_i \in \mathbb{B}$.

⋆ ToQUBO.jl is the only platform in the board whose automatic encoding routines are also applicable to continuous variables. Furthermore, its *Bounded-Coefficient* implementation allows the technique to be used not just with the *Binary* encoding, as proposed in the original paper, but also with the *Unary* and *Arithmetic Progression* modes.

! Amplify only supports these encoding methods when introducing integer slack variables for mapping inequality constraints. They are not applicable to regular variables, as their models only accept binary or spin sites so far.

Table 3 External samplers coverage

Sampler	Hardware	QUBODrivers.jl	PyQUBO	qubovert	Qiskit	OpenQAOA	Amplify
Simulated Annealing	CPU/ GPU	■ [3]	■	■			■
Quantum Annealing	QA[1]	■ [4]	■		■ [9]		■
Simulated Quantum Annealing	CPU/ GPU	■ [5]					■
VQE	QGC [2]	■ [6]			■	■	
QAOA	QGC [2]	■ [6]			■	■	■
MQLib Heuristics Library	CPU	■ [7]					
Parallel Tempering	CPU	■ [8]					
Simulated Bifurcation Machine	GPU/FPGA						■
Digital Annealing	ASIC						■
CMOS Annealing	CMOS						■

[1] Quantum Annealer; [2] Quantum Gate Circuit; [3] DWave.jl (Xavier and Ripper 2022a); [4] DWaveNeal.jl (Xavier and Ripper 2023a); [5] QuantumAnnealingInterface.jl (Morrell and Coffrin 2022, Xavier and Ripper 2022b); [6] QiskitOpt.jl (Xavier and Ripper 2023c); [7] MQLib.jl (Dunning et al. 2018, Xavier and Ripper 2023b); [8] PySA.jl (Xavier 2023); [9] Dwave-Qiskit-plugin (D-Wave 2020);

ToQUBO.jl

QUBO Drivers.jl

Qiskit
Elements for building a quantum future

MQ LIB

D-Wave

Ocean

QUBO Lib

1. Xavier, Pedro Maciel, et al. "Qubo. jl: A julia ecosystem for quadratic unconstrained binary optimization." arXiv preprint arXiv:2307.02577 (2023).

QiskitOpt.jl

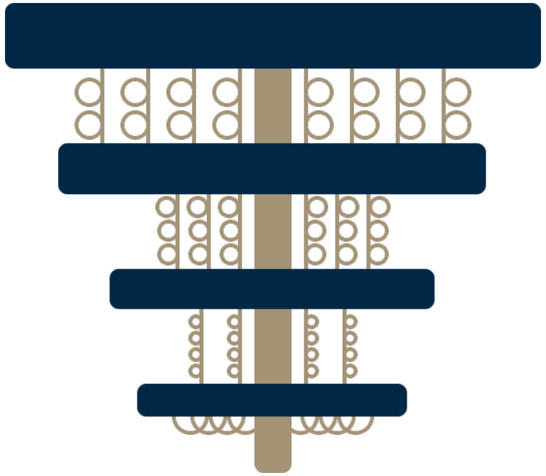
QUBODriver example

Leverages **Qiskit** to **interface** with the **IBMQ** platform:

Simulator + Quantum Hardware

$$
\begin{aligned}
\min \quad & x'Qx + \ell'x + c \\
\text{s.t.} \quad & x \in \{0,1\}^n
\end{aligned}
$$

```julia
using JuMP
using QiskitOpt

# If using QAOA
model = Model(QiskitOpt.QAOA.Optimizer)

# If using VQE
model = Model(QiskitOpt.VQE.Optimizer)

Q = [
   -1  2  2
    2 -1  2
    2  2 -1
]

@variable(model, x[1:3], Bin)
@objective(model, Min, x' * Q * x)

optimize!(model)
```

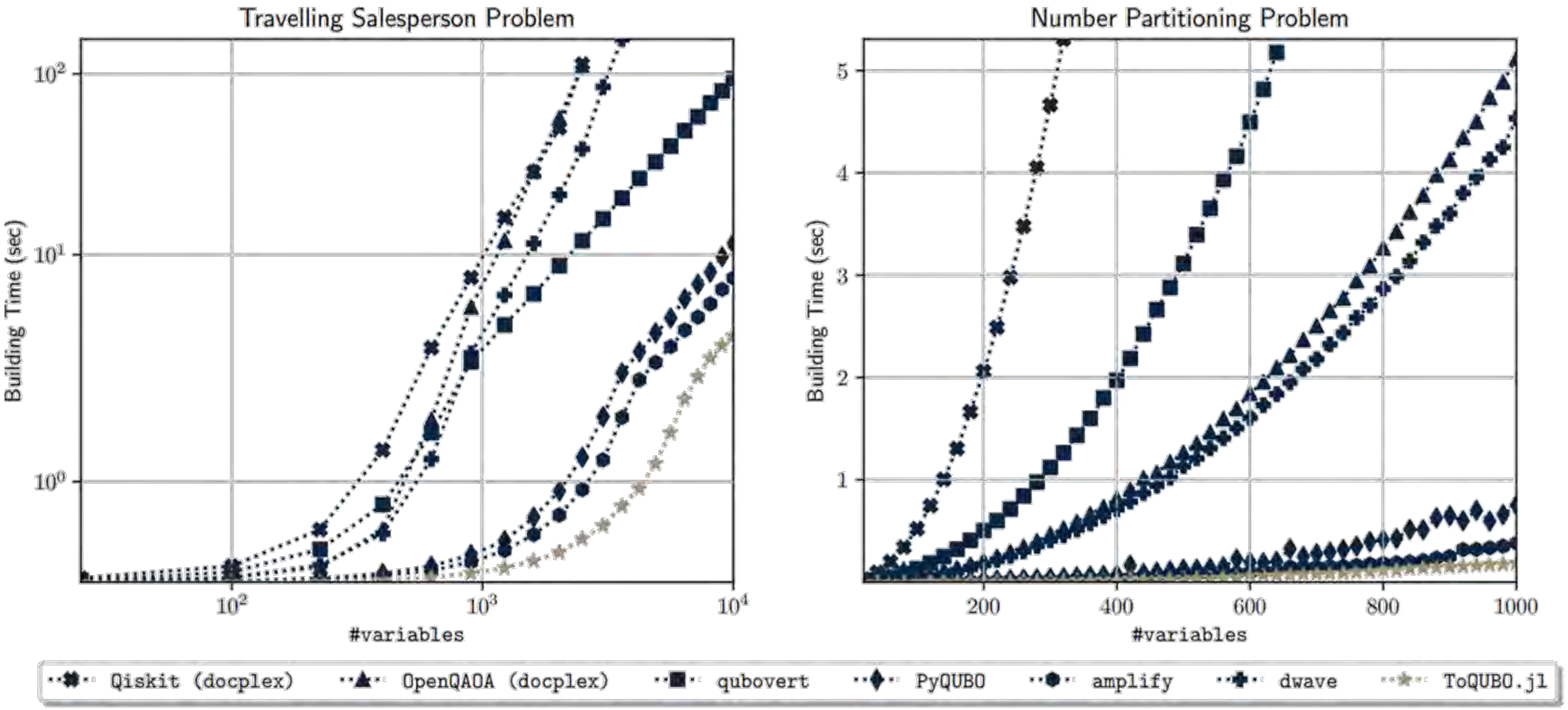


Smooth modeling experience for Quantum Optimization!

QUBO.jl

How do our Julia packages perform?

Outcome:
We provide the **fastest and most complete reformulator** of traditional optimization problems **to Ising/QUBO** as **open-source**



1. Xavier, Pedro Maciel, et al. "Qubo. jl: A julia ecosystem for quadratic unconstrained binary optimization." arXiv preprint arXiv:2307.02577 (2023).

Quantum Methods for Discrete Optimization – Decomposition methods

Decomposition methods for quantum optimization

Tackling the original QUBO problems

- One family of decomposition algorithms aims to address directly the QUBOs that monolithic Quantum algorithms try to solve
  - Usual methods need one qubit per variable in the QUBO
- From the lessons learned of classical decomposition:
  - Leverage problem structure
  - Use the right tool for the right task

… we can design decomposition or multi-level methods that will use for empirical advantage in optimization with quantum subroutines

- Key question: Is the empirical advantage relying on the quantum subroutine or because of the decomposition method?

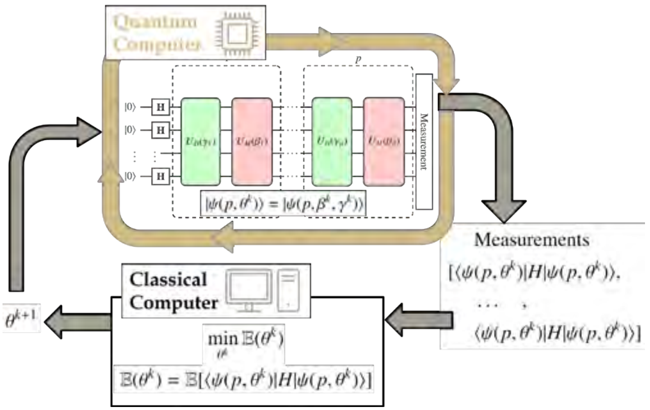
Variational Quantum Algorithms

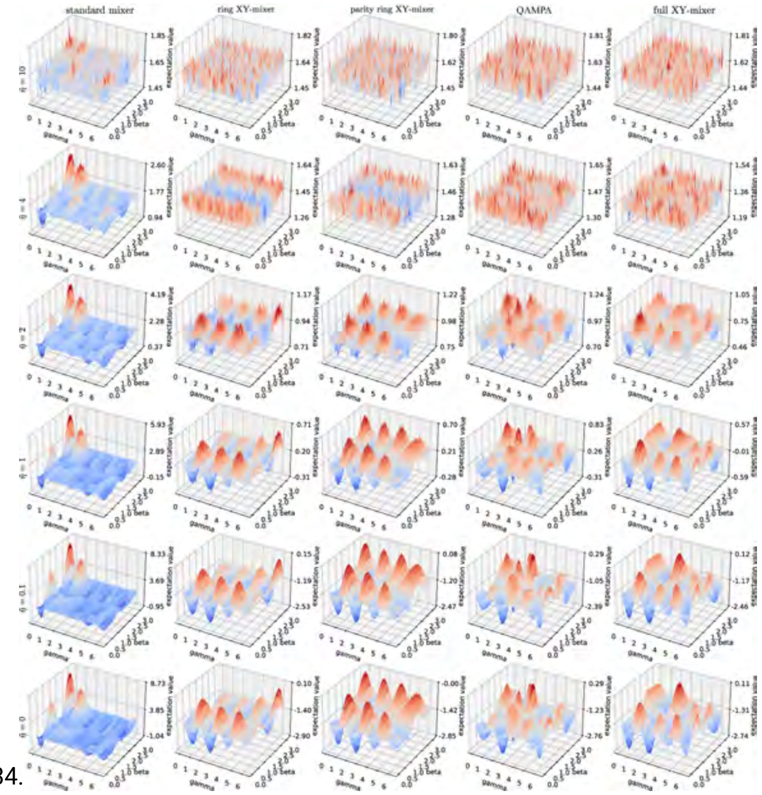
Well known but non-trivial example

In variational algorithms, combinatorial optimization is decomposed

- Sampling of the quantum state is performed in a gate-based QC
    - Quantum device is able to represent a complex probability distribution
- Parameter setting problem is solved in the classical computer
    - Classical computer can aim to perform continuous non-convex optimization

Well designed mixers can implement constraints!





1. Hadfield, Stuart, et al. "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz." Algorithms 12.2 (2019): 34.
2. Fuchs, Franz Georg, et al. "Constraint preserving mixers for the quantum approximate optimization algorithm." Algorithms 15.6 (2022): 202.
3. Brandhofer, Sebastian, et al. "Benchmarking the performance of portfolio optimization with QAOA." Quantum Information Processing 22.1 (2022): 25.

Graph-decomposition methods

Rely on equivalence that QUBOs can be represented as graphs

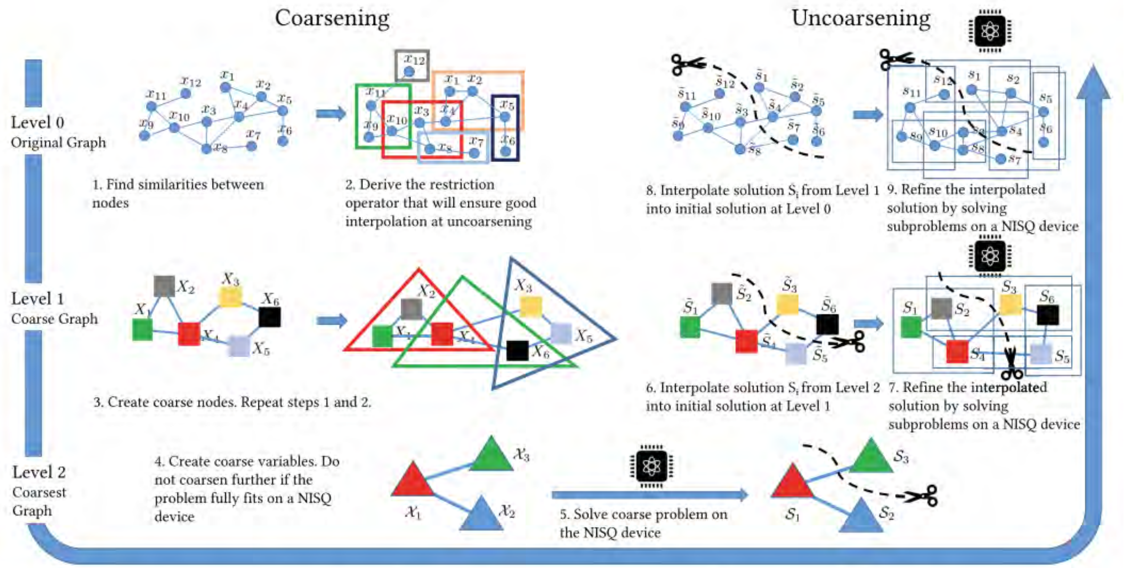
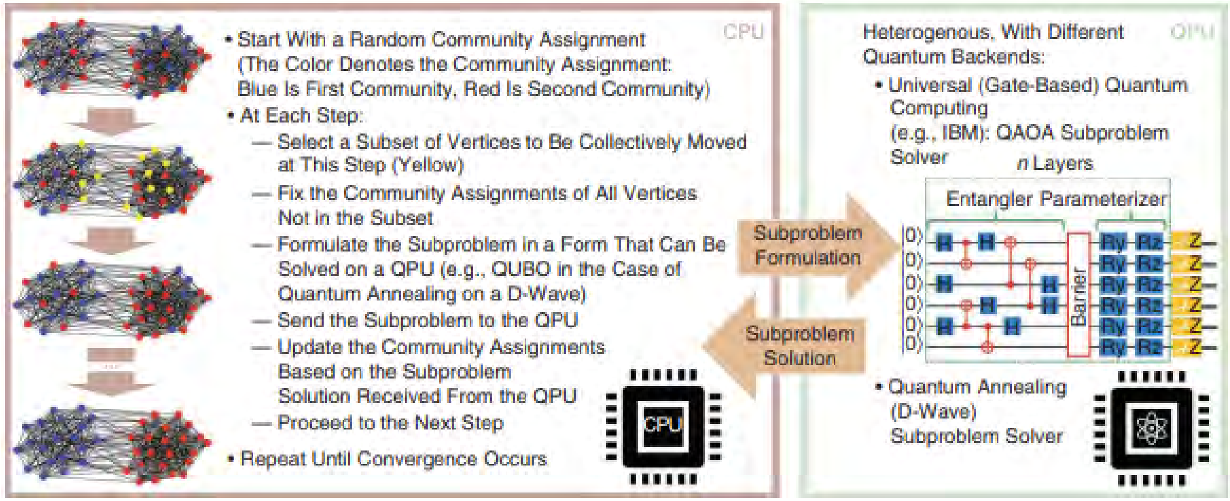
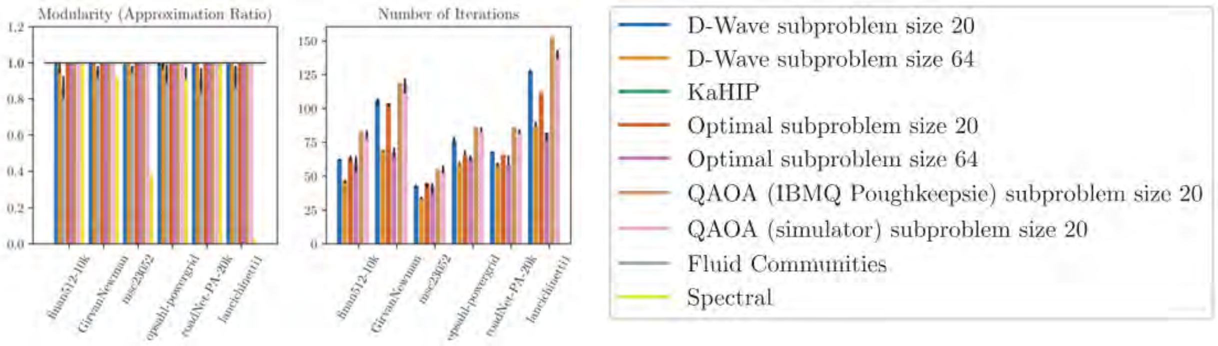


Fig. 1. V-cycle for a graph problem. First, the problem is iteratively coarsened (left). Second, the coarse problem is solved using a NISQ optimization solver (bottom). Finally, the problem is iteratively uncoarsened and the solution is refined using a NISQ solver (right).





1. Shaydulin, Ruslan, et al. "A hybrid approach for solving optimization problems on small quantum computers." Computer 52.6 (2019): 18-26.
2. Ushijima-Mwesigwa, Hayato, et al. "Multilevel combinatorial optimization across quantum architectures." ACM Transactions on Quantum Computing 2.1 (2021): 1-29.
3. Angone, Anthony, et al. "Hybrid quantum-classical multilevel approach for maximum cuts on graphs." 2023 IEEE High Performance Extreme Computing Conference (HPEC). IEEE, 2023.

Graph-decomposition methods

Rely on equivalence that QUBOs can be represented as graphs

FIG. 3: Left: Graph G. Middle: G is split into two components from the removal of K. Right: $V_2 \cup K$

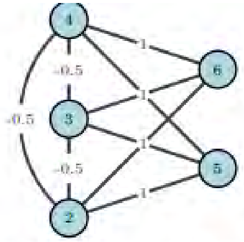


FIG. 4: Graph $\mathcal{G}$ for 3-Cut Example

- After ~80 iterations in 100 nodes graphs solving subproblems of size 16

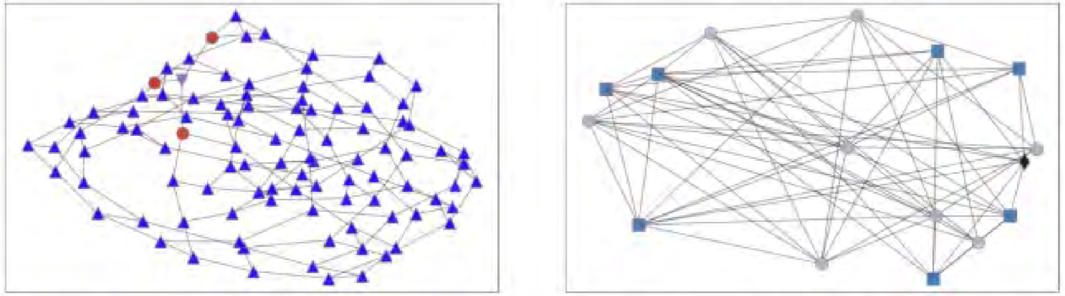


FIG. 5: An example of one of the 100-vertex 3-regular graph solved using QAOA and then used in the decomposition algorithm (left). Marked in the red (circle): $K^{(0)}$, in blue (upward-facing triangle): $V_1^{(0)}$, and in purple (down-facing triangle): $V_2^{(0)}$. The final iteration of the decomposition before the termination criteria is met (right). Marked in the light blue (square): $K^{(84)}$, in gray (circle): $V_1^{(84)}$, and in black (rhombus): $V_2^{(84)}$.

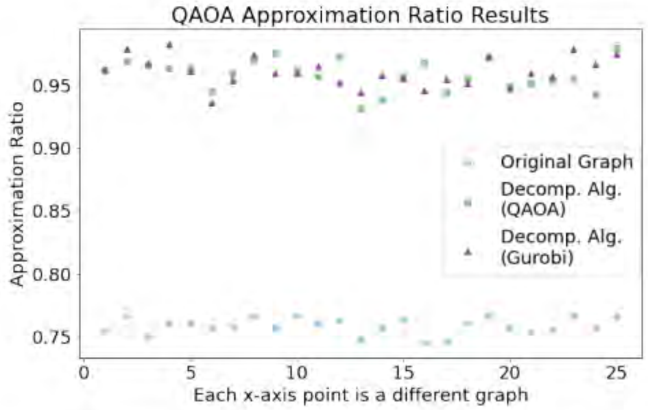


FIG. 6: Comparing approximation ratios for the twenty-five different 100-vertex 3-regular random graphs.

1. Ponce, Moises, et al. "Graph decomposition techniques for solving combinatorial optimization problems with variational quantum algorithms." arXiv preprint arXiv:2306.00494 (2023).

Quantum Optimization for the Maximum Cut Problem on a Superconducting Quantum Computer

Maxime Dupont

Bhuvanesh Sundar

Bram Evert

Zedong Peng

Mark Hodson

Maxime Dupont,[1] Bhuvanesh Sundar,[1] Bram Evert,[1] DEBN,[2, 3, 4]
Zedong Peng,[2] Stephen Jeffrey,[1] and Mark J. Hodson[1]

[1]Rigetti Computing

[2]Purdue University

[2]Quantum Artificial Intelligence Laboratory, NASA Ames Research Center

[3]Research Institute of Advanced Computer Science, USRA

arxiv.org/abs/2404.17579

Quantum Optimization for the Maximum Cut Problem on a Superconducting Quantum Computer

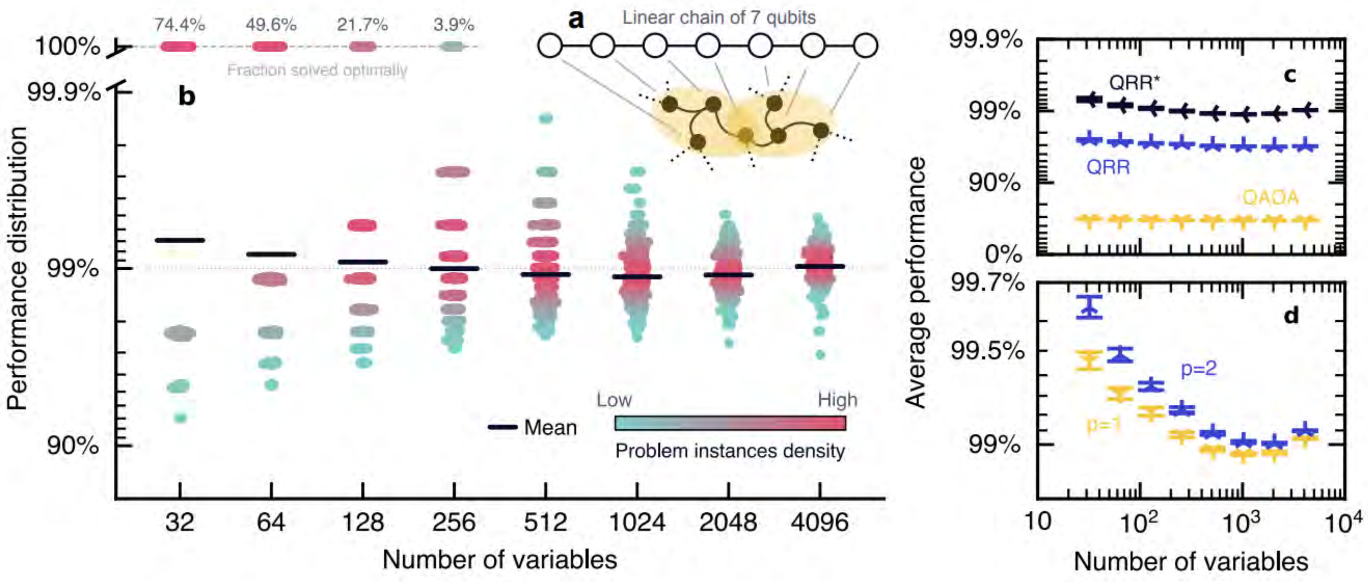


FIG. 1. (a) Embedding of a light-cone-induced 7-vertex subgraph by the QAOA at $p = 1$ for a 3-regular graph onto a linear chain of 7 qubits. (b) Experimental performance distribution of the greedy-enhanced quantum relax-and-round (QRR*) solver with $p = 1$ over 1,000 randomly generated problem instances for various number of variables. (c) Average experimental performance as a function of the problem size for the quantum approximate optimization algorithm (QAOA), the quantum relax-and-round (QRR), and the QRR* solvers with $p = 1$. (d) Simulated average performance of the QRR* solver as a function of the number of variables for $p = 1$ and $p = 2$ layers. Error bars indicate one standard deviation.

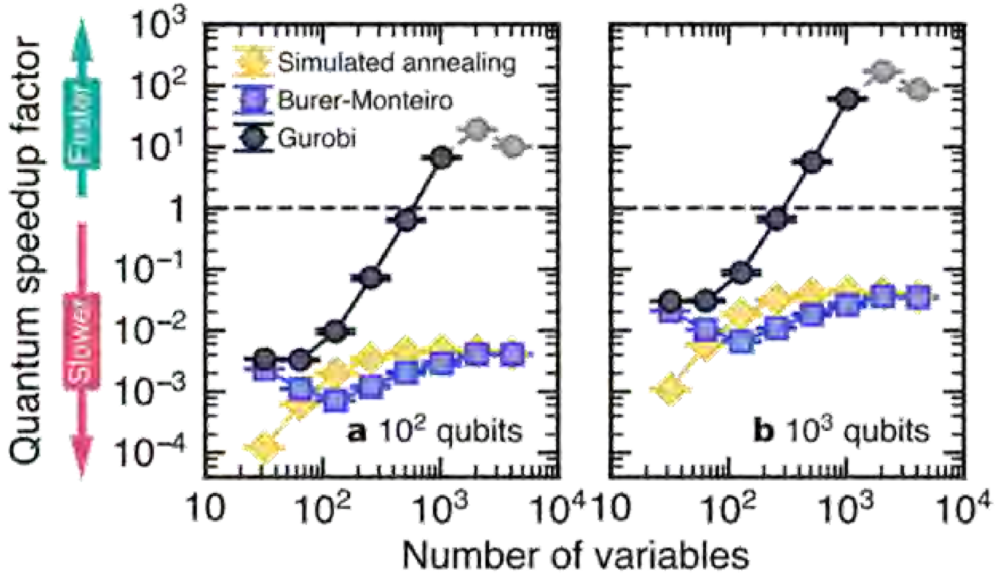


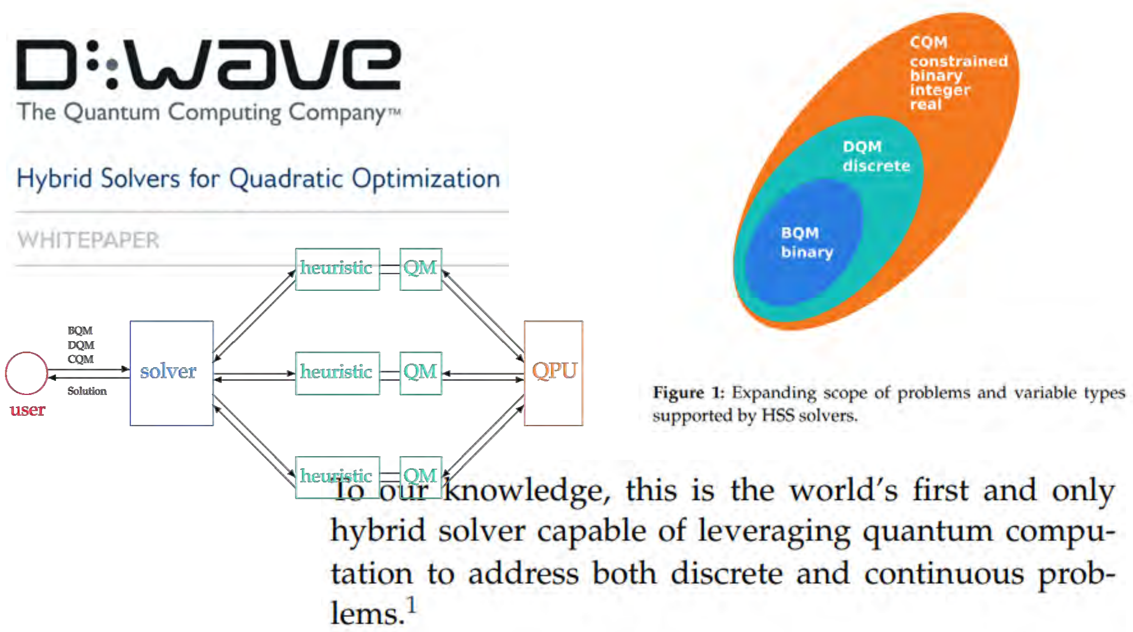
FIG. 2. Optimal runtime for various classical solvers to match the performance of the greedy-enhanced quantum relax-and-round (QRR*) solver with $p = 1$, normalized by the runtime of QRR* [30], as a function of the number of variables. The Gurobi data points for the two largest sizes are lower bounds where no solutions matching that of QRR* were found in 600 seconds for most of the problem instances [30]. The quantum runtime component of QRR* is based on $t_{\mathrm{Rx}} = 40$ ns, $t_{\mathrm{ISWAP}} = 122$ ns, $t_{\mathrm{m+r}} = 6$ μs, and $n = 5 \times 10^4$ shots [30]. (a) Utilizing a quantum computer with 10^2 qubits. (b) Utilizing a quantum computer with 10^3 qubits. Each data point is averaged over 1,000 randomly generated problem instances for each number of variables. Error bars indicate one standard deviation.

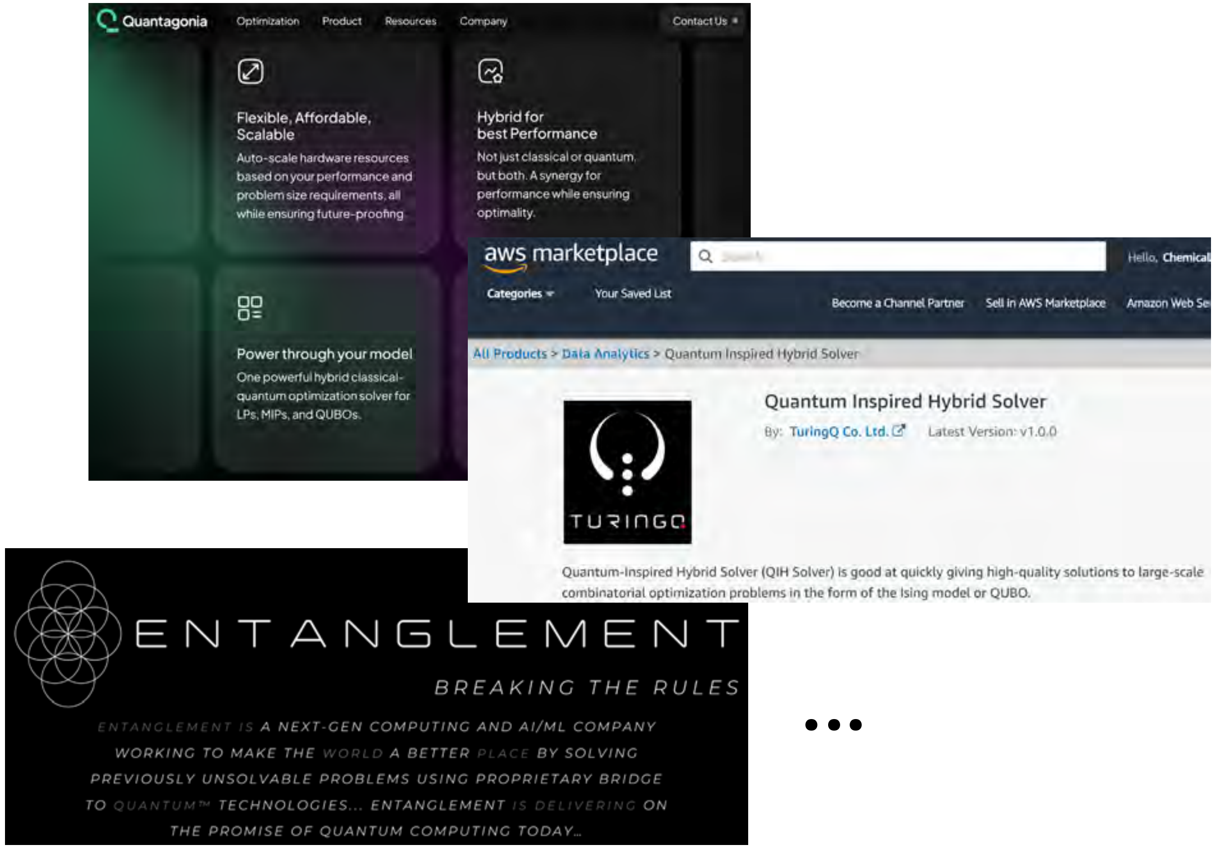
1. Dupont, Maxime, et al. "Quantum Optimization for the Maximum Cut Problem on a Superconducting Quantum Computer." arXiv preprint arXiv:2404.17579 (2024).

More general problems than QUBO – Constraints

Using all the techniques from mathematical programming

Companies are being built around this idea (this could drive you into it)





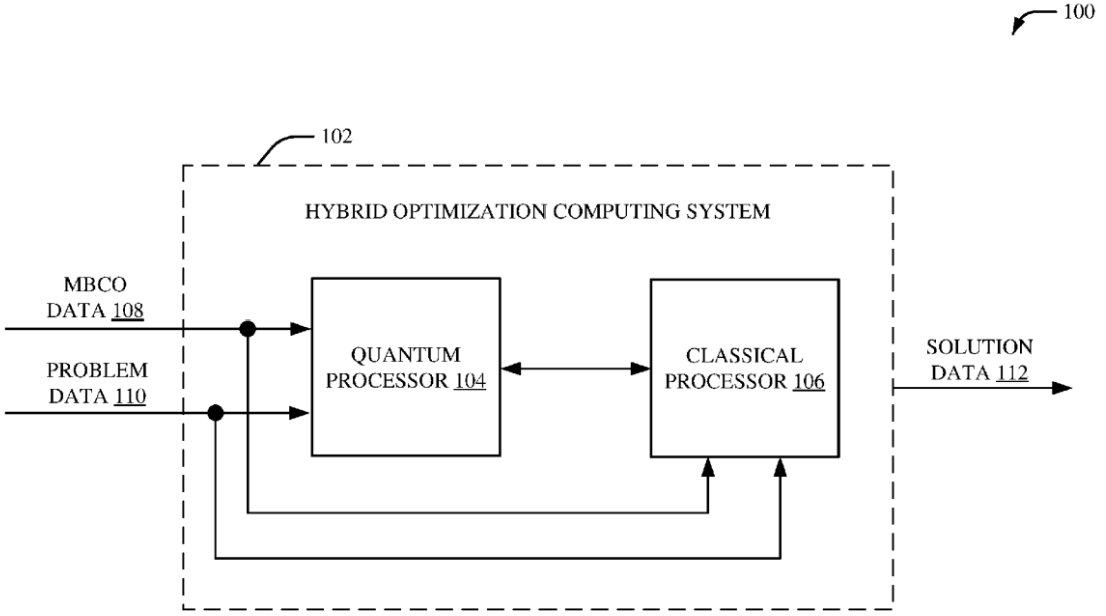
…

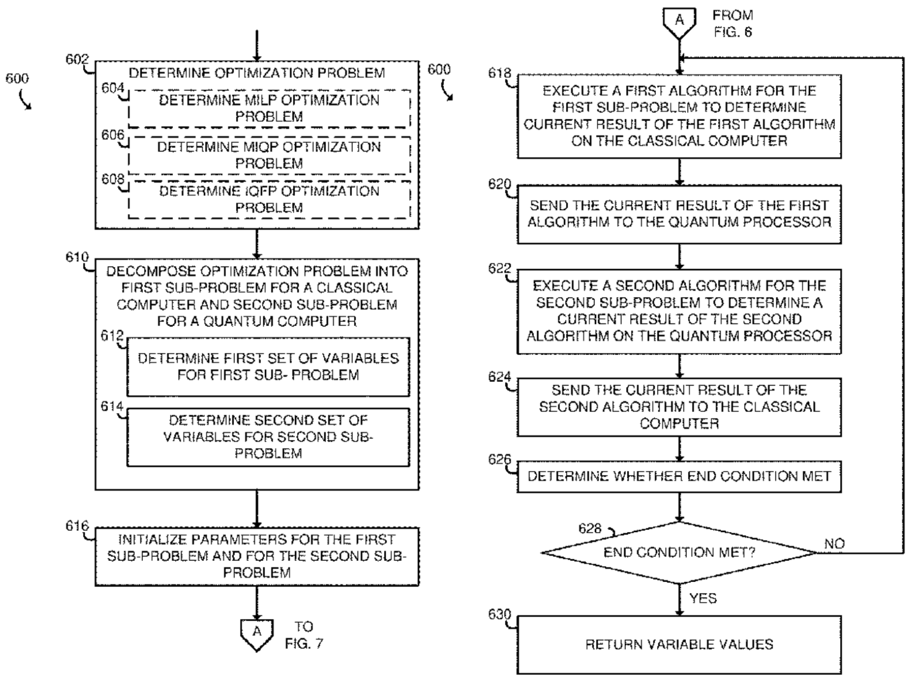
1. https://www.dwavesys.com/media/soxph512/hybrid-solvers-for-quadratic-optimization.pdf
2. https://www.quantagonia.com/hybridsolver
3. https://www.turingq.com/
4. https://www.entanglement.com/

More general problems than QUBO – MIP

Using all the techniques from mathematical programming

Patents out there (which should not steer you away from this)





1. Gambella, Claudio, et al. "Mixed-binary constrained optimization on quantum computers." U.S. Patent Application No. 16/395,560.
2. You, Fengqi, and Akshay Ajagekar. "Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems." U.S. Patent No. 11,769,070. 26 Sep. 2023.

Mixed-Binary Optimization

ADMM Optimization

MBO →

$$\begin{aligned} \underset{x\in\mathcal{X},u\in\mathcal{U}\subseteq\mathbb{R}^l}{\text{minimize}} \quad & q(x)+\varphi(u) \\ \text{subject to :} \quad & Gx=b, \quad g(x)\le 0 \\ & \ell(x,u)\le 0, \end{aligned}$$

M-ADMM-H Decomposition

iterative approach:

outer iterations

1. **QUBO** solution →
2. **Convex problems** solution [CPLEX]
3. Dual update
4. Evaluate merit metric
5. Go to 1.

QUBO Solution

Either:

(i) Classical [CPLEX]

(ii) Quantum [VQE/QAOA] →

VQE

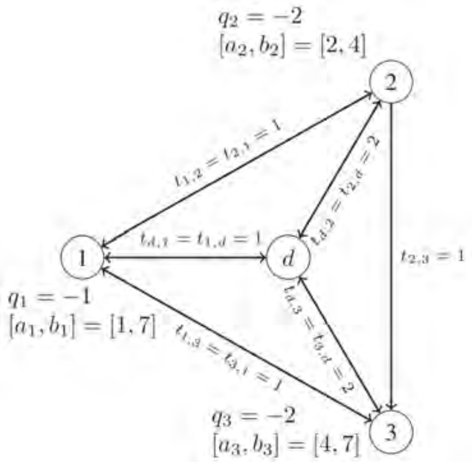
Solves QUBO as

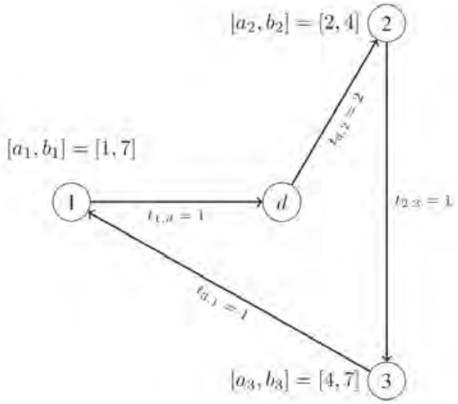
$$\min_\theta \langle\psi(\theta)|H|\psi(\theta)\rangle$$

by iterating as

inner iterations

1. Choose θ with a classical optimizer:
   Either: (i) SPSA or (ii) COBYLA
2. Evaluate $\langle\psi(\theta)|H|\psi(\theta)\rangle$
   and its gradient on the quantum device
3. Go to 1





1. Gambella, Claudio, and Andrea Simonetto. "Multiblock ADMM heuristics for mixed-binary optimization on classical and quantum computers." IEEE Transactions on Quantum Engineering 1 (2020): 1-22.
2. Harwood, Stuart, et al. "Formulating and solving routing problems on quantum computers." IEEE Transactions on Quantum Engineering 2 (2021): 1-17.

Mixed-Binary Optimization

ADMM Optimization

$$\begin{aligned} \min_{x\in\mathcal{X}, u\in\mathcal{U}\subseteq\mathbb{R}^l} \quad & q(x)+\varphi(u) \\ \text{subject to:} \quad & Gx=b, \quad g(x)\le 0 \\ & \ell(x,u)\le 0, \end{aligned}$$

$$\begin{aligned} \min_{x\in\mathcal{X}, z\in\mathbb{R}^n, u\in\mathcal{U}\subseteq\mathbb{R}^l} \quad & q(x)+\frac{c}{2}\|Gx-b\|_2^2+\varphi(u) \\ \text{subject to:} \quad & g(z)\le 0, \quad \ell(z,u)\le 0, \\ & x=z, \end{aligned}$$

$$\begin{aligned} \min_{x\in\mathcal{X}, \bar{x}\in\mathbb{R}^m} \quad & f_0(x)+f_1(\bar{x}) \\ \text{subject to:} \quad & A_0x+A_1\bar{x}=0, \end{aligned}$$

$$\begin{aligned} \min_{x\in\mathcal{X}, \bar{x}\in\mathbb{R}^m, y\in\mathbb{R}^n} \quad & f_0(x)+f_1(\bar{x})+\frac{\beta}{2}\|y\|_2^2 \\ \text{subject to:} \quad & A_0x+A_1\bar{x}=y, \end{aligned}$$

Algorithm 1 2-ADMM-H mixed-binary heuristic

Require: Initial choice of x_0, y_0, λ_0. Choice of $\varrho, c, \mu > 0$, tolerance $\epsilon > 0$, and maximum number of iterations $K_{\max}$.

while $k < K_{\max}$ **and** $\|A_0x^k + A_1\bar{x}^k\| > \epsilon$, **do**

First block update (QUBO):

$$x_k = \arg\min_{x\in\{0,1\}^n} q(x)+\frac{c}{2}\|Gx-b\|_2^2+\lambda_{k-1}^\intercal A_0x + \frac{\varrho}{2}\|A_0x+A_1\bar{x}_{k-1}\|^2$$

Second block update (Convex):

$$\bar{x}_k = \arg\min_{\bar{x}\in\mathbb{R}^m} f_1(\bar{x})+\lambda_{k-1}^\intercal A_1\bar{x}+\frac{\varrho}{2}\|A_0x_k+A_1\bar{x}\|^2$$

Dual variable update:

$$\lambda_k = \lambda_{k-1}+\varrho(A_0x_k+A_1\bar{x}_k)$$

Compute merit value:

$$\eta_k = q(x_k)+\phi(\bar{x}_k) + \mu(\max(g(x_k),0)+\max(l(x_k,\bar{x}_k),0))$$

end while

return $x_{k^*}, \bar{x}_{k^*}, y_{k^*}$, with $k^* = \min_k \eta_k$.

Algorithm 2 3-ADMM-H mixed-binary heuristic

Require: Initial choice of $x_0, \bar{x}_0, y_0, \lambda_0$. Choice of $\varrho, \beta, c >$ 0, tolerance $\epsilon > 0$, and maximum number of iterations $K_{\max}$.

while $k < K_{\max}$ **and** $\|A_0x^k + A_1\bar{x}^k - y_k\| > \epsilon$, **do**

First block update (QUBO):

$$x_k = \arg\min_{x\in\{0,1\}^n} q(x)+\frac{c}{2}\|Gx-b\|_2^2 + \lambda_{k-1}^\intercal A_0x+\frac{\varrho}{2}\|A_0x+A_1\bar{x}_{k-1}-y_{k-1}\|^2 \qquad (17)$$

Second block update (Convex):

$$\bar{x}_k = \arg\min_{\bar{x}\in\mathbb{R}^m} f_1(\bar{x})+\lambda_{k-1}^\intercal A_1\bar{x}+ \frac{\varrho}{2}\|A_0x_k+A_1\bar{x}-y_{k-1}\|^2 \qquad (18)$$

Third block update (Convex+quadratic):

$$y_k = \arg\min_{y\in\mathbb{R}^n} \frac{\beta}{2}\|y\|_2^2-\lambda_{k-1}^\intercal y+\frac{\varrho}{2}\|A_0x_k+A_1\bar{x}_k-y\|^2$$

Dual variable update:

$$\lambda_k = \lambda_{k-1}+\varrho(A_0x_k+A_1\bar{x}_k-y_k)$$

Compute merit value:

$$\eta_k = q(x_k)+\phi(\bar{x}_k)+ + \mu(\max(g(x_k),0)+\max(l(x_k,\bar{x}_k),0)) \qquad (19)$$

end while

return $x_{k^*}, \bar{x}_{k^*}, y_{k^*}$, with $k^* = \min_k \eta_k$.

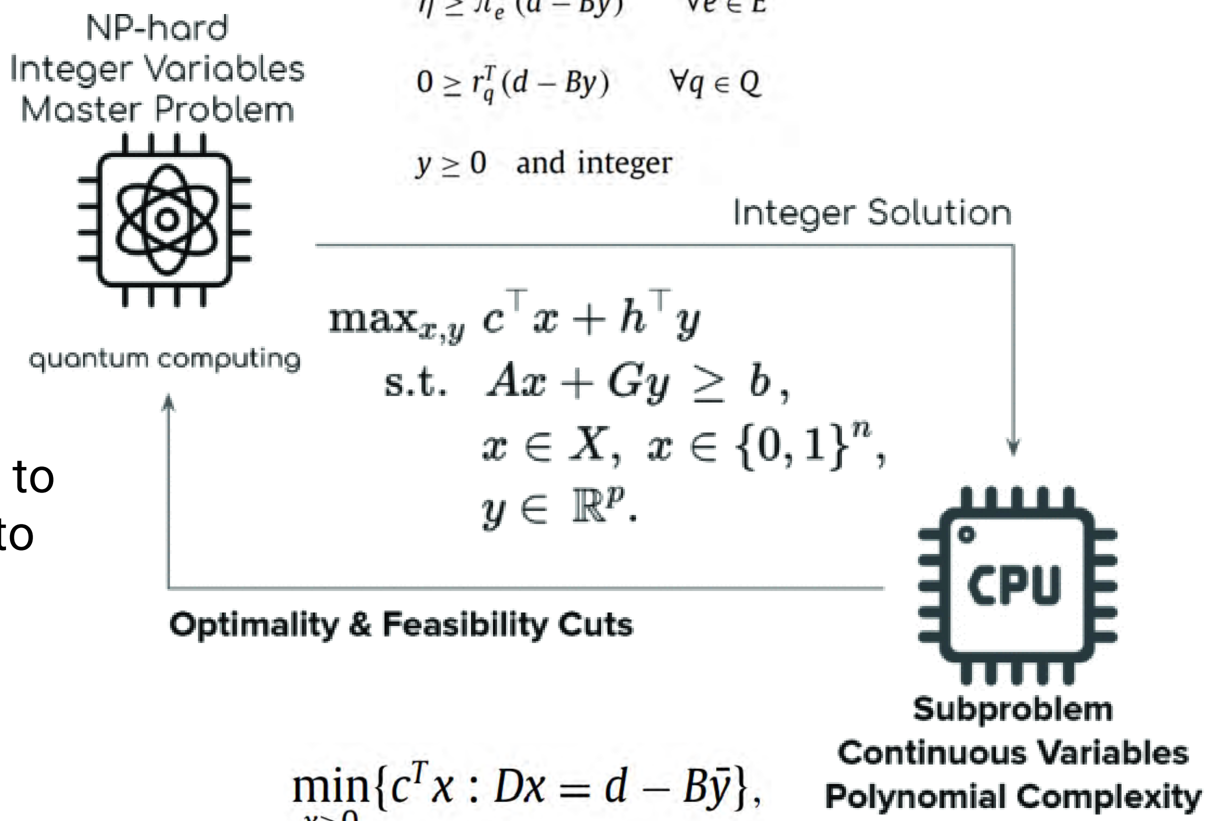
1. Gambella, Claudio, and Andrea Simonetto. "Multiblock ADMM heuristics for mixed-binary optimization on classical and quantum computers." IEEE Transactions on Quantum Engineering 1 (2020): 1-22.

Mixed-Binary Optimization
Benders Decomposition

$$
\begin{aligned}
\text{Minimize} \quad & f^T y + c^T x \\
\text{subject to} \quad & Ay = b \\
& By + Dx = d \\
& x \geq 0 \\
& y \geq 0 \quad \text{and integer},
\end{aligned}
$$

Or even use QUBO to select which cuts to add [6]

$$
\begin{aligned}
\min_{y,\eta} \quad & f^T y + \eta \\
\text{subject to} \quad & Ay = b \\
& \eta \geq \pi_e^T (d - By) \qquad \forall e \in E \\
& 0 \geq r_q^T (d - By) \qquad \forall q \in Q \\
& y \geq 0 \quad \text{and integer}
\end{aligned}
$$



$$
\begin{aligned}
\max_{x,y} \ & c^\top x + h^\top y \\
\text{s.t.} \ & Ax + Gy \geq b, \\
& x \in X, \ x \in \{0,1\}^n, \\
& y \in \mathbb{R}^p.
\end{aligned}
$$

$$\min_{x \geq 0} \{c^T x : Dx = d - B\bar{y}\},$$

1. Rahmaniani, Ragheb, et al. "The Benders decomposition algorithm: A literature review." European Journal of Operational Research 259.3 (2017): 801-817.
2. Ajagekar, Akshay, et al. "Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems." Computers & Chemical Engineering 132 (2020): 106630.
3. Chang, Chin-Yao, et al. "On hybrid quantum and classical computing algorithms for mixed-integer programming." arXiv preprint arXiv:2010.07852 (2020).
4. Gao, Fang, et al. "Hybrid quantum-classical general benders decomposition algorithm for unit commitment with multiple networked microgrids." arXiv preprint arXiv:2210.06678 (2022).
5. Zhao, Zhongqi, Lei Fan, and Zhu Han. "Hybrid quantum benders' decomposition for mixed-integer linear programming." 2022 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2022.
6. Paterakis, Nikolaos G. "Hybrid quantum-classical multi-cut benders approach with a power system application." Computers & Chemical Engineering 172 (2023): 108161.

More general problems than QUBO – Constraints

Reformulate into continuous problems

Mixed-integer programs can be reformulated into high-dimensional continuous programs

- LPs through Dantzig-Wolfe [1]
- SDPs using moment/sum-of-squares [2,3]
- Or copositive programs! [4]

1. Dantzig, George B., and Philip Wolfe. "Decomposition principle for linear programs." Operations research 8.1 (1960): 101-111.
2. Parrilo, Pablo A. Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. California Institute of Technology, 2000.
3. Lasserre, Jean B. "Global optimization with polynomials and the problem of moments." SIAM Journal on optimization 11.3 (2001): 796-817.
4. Burer, Samuel. "On the copositive representation of binary and continuous nonconvex quadratic programs." Mathematical Programming 120.2 (2009): 479-495.
5. Abbas, Amira, et al. "Quantum optimization: Potential, challenges, and the path forward." arXiv preprint arXiv:2312.02279 (2023).

Copositive Optimization for mixed-binary quadratic optimization via Ising solvers

Robin Brown

Davide Venturelli

Marco Pavone

arxiv > math > arXiv:2207.13630

Search... All fields Search

Help | Advanced Search

Mathematics > Optimization and Control

[Submitted on 27 Jul 2022]

Copositive programming for mixed-binary quadratic optimization via Ising solvers

Robin Brown, David E. Bernal Neira, Davide Venturelli, Marco Pavone

Recent years have seen significant advances in quantum/quantum-inspired technologies capable of approximately searching for the ground state of Ising spin Hamiltonians. The promise of leveraging such technologies to accelerate the solution of difficult optimization problems has spurred an increased interest in exploring methods to integrate Ising problems as part of their solution process, with existing approaches ranging from direct transcription to hybrid quantum-classical approaches rooted in existing optimization algorithms. Due

Download:
- PDF
- Other formats

(license)

Current browse context: math.OC

< prev | next >

new | recent | 2207

Change to browse by:

cs

cs.DM

math

quant-ph

References & Citations
- INSPIRE HEP
- NASA ADS

Robin Brown[1], DEBN, Davide Venturelli[2,3], Marco Pavone[1]

[1]Institute of Computational and Mathematical Engineering, Stanford University

[2]Quantum Artificial Intelligence Laboratory, NASA Ames Research Center

[3]Research Institute of Advanced Computer Science, USRA

ICME INSTITUTE for COMPUTATIONAL & MATHEMATICAL ENGINEERING at STANFORD UNIVERSITY

USRA

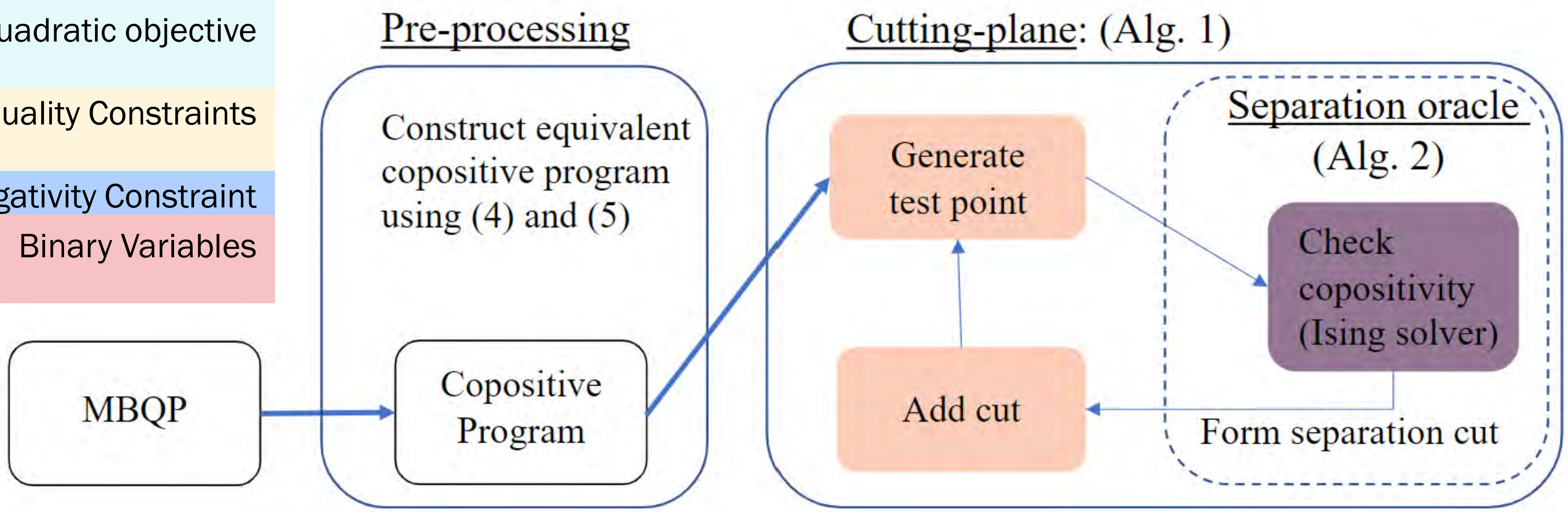
NASA

arxiv.org/abs/2207.13630

Copositive decomposition for MBQP using QUBO

Solution process in a nutshell

$$
\begin{aligned}
\min\quad & x^\top Q x + 2c^\top x && \text{Nonconvex quadratic objective}\\
& a_i^\top x = b_i,\ i \in 1, \dots, N && \text{Linear Equality Constraints}\\
& x \geq 0 && \text{Non-negativity Constraint}\\
& x_j \in \{0, 1\},\ j \in B && \text{Binary Variables}
\end{aligned}
$$



$$
\begin{aligned}
\max\quad & \gamma + \sum_i \mu_i^{(\mathrm{lin})} b_i + \mu_i^{(\mathrm{quad})} b_i^2\\
& M(\mu, \lambda, \gamma) \in \mathcal{C}
\end{aligned}
$$

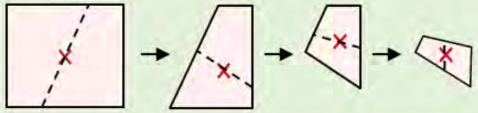
$$
M(\mu,\lambda,\gamma) := \begin{pmatrix} Q & c \\ c^\top & \cdot \end{pmatrix} - \sum_i \mu_i^{(\mathrm{lin})} \begin{pmatrix} \cdot & \frac{1}{2}a_i \\ \frac{1}{2}a_i^\top & \cdot \end{pmatrix} - \sum_i \mu_i^{(\mathrm{quad})} \begin{pmatrix} a_i a_i^\top & \cdot \\ \cdot & \cdot \end{pmatrix} - \sum_{j \in B} \lambda_j \begin{pmatrix} -e_j e_j^\top & \frac{1}{2}e_j \\ \frac{1}{2}e_j^\top & \cdot \end{pmatrix} - \gamma \begin{pmatrix} \cdot & \cdot \\ \cdot & 1 \end{pmatrix}
$$

1. Brown, Robin, et al. "A copositive framework for analysis of hybrid ising-classical algorithms." SIAM Journal on Optimization 34.2 (2024): 1455-1489.

Copositive decomposition for MBQP using QUBO

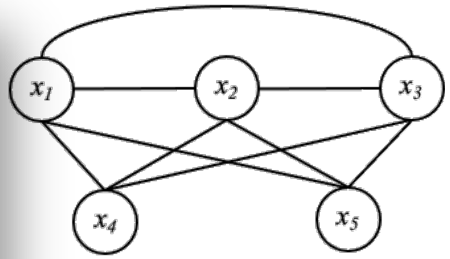
How to solve MBQPs to optimality?

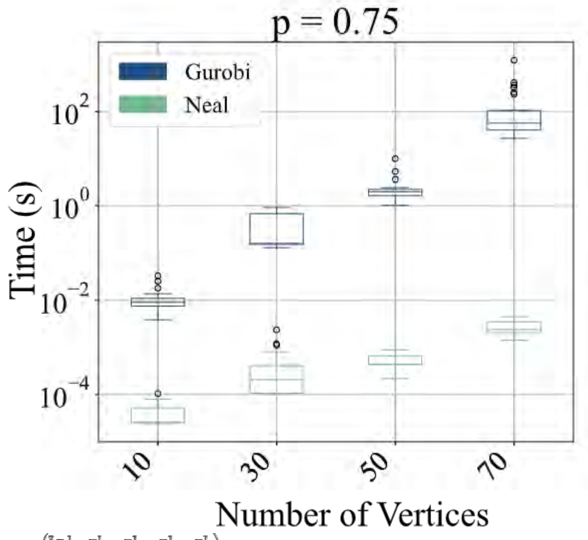
Compute center:
Selection of test points as some sort of center of the remaining search space ensures consistent progress when reducing the search space.

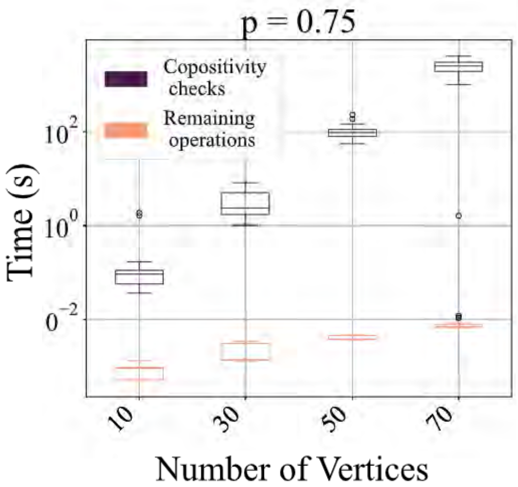


Checking copositivity:
Checking membership in the copositive cone is co-NP-complete, but can be solved via a non-convex quadratic minimization:

MAXCLIQUE

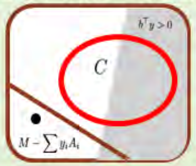


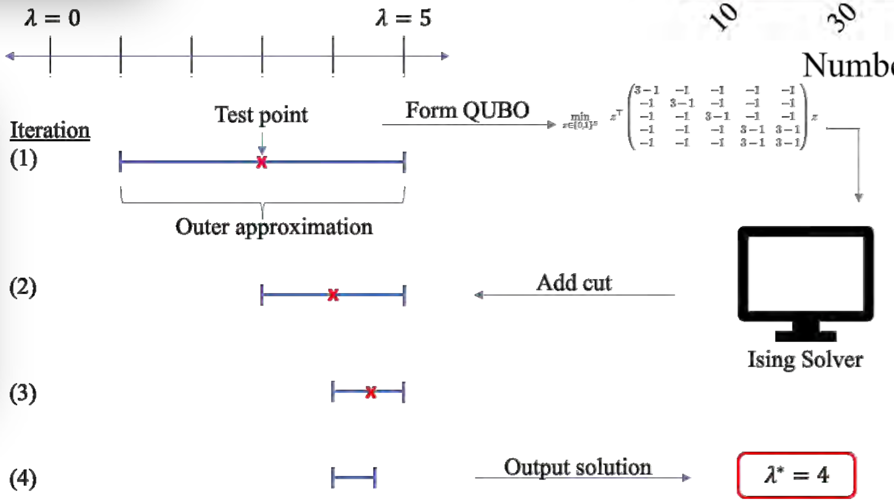




Add feasibility cut:
Convexity of the copositive cone means that an infeasible point can be extrapolated to rule out an entire halfspace.

A certificate of non-copositivity immediately gives rise to a separating hyperplane.

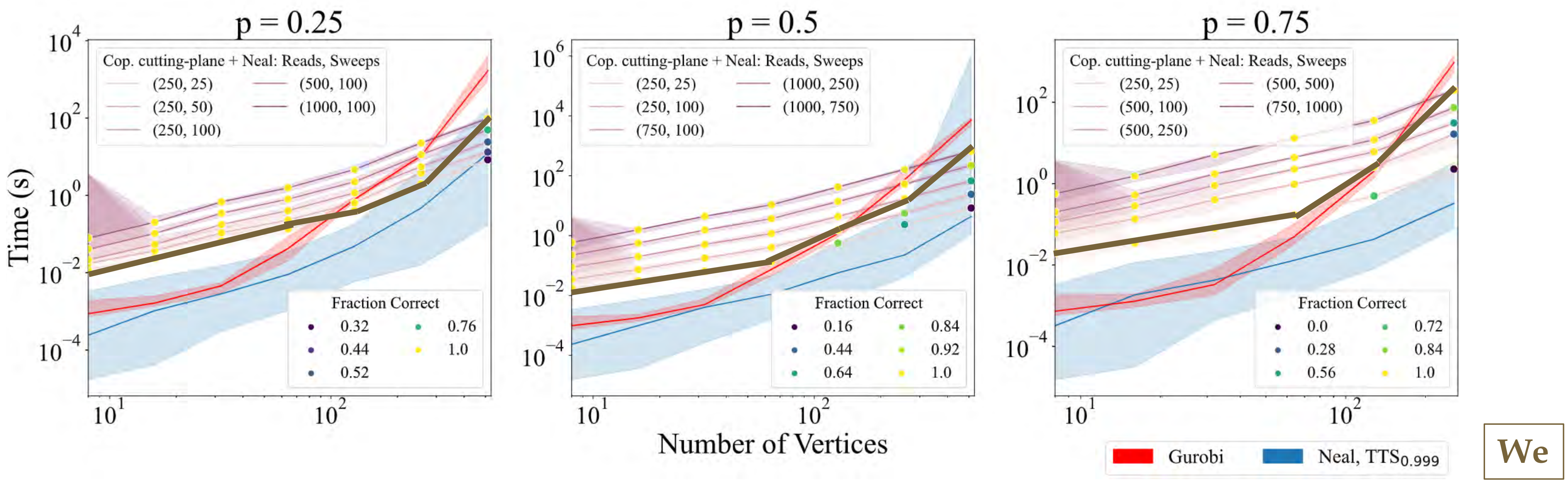




1. Brown, Robin, et al. "A copositive framework for analysis of hybrid ising-classical algorithms." SIAM Journal on Optimization 34.2 (2024): 1455-1489.

Copositive decomposition using QUBO
How does it perform?

MAXCLIQUE comparison against MIP formulation and Ising heuristic



Solution time for the copositive cutting-plane algorithm with Simulated Annealing as the Ising solver, the solution time when solving a mixed-integer programming (MIP) formulation of maximum-clique directly with Gurobi, and the corresponding best penalty version of Simulated Annealing TTT to 99.9% confidence

1. Brown, Robin, et al. "A copositive framework for analysis of hybrid ising-classical algorithms." SIAM Journal on Optimization 34.2 (2024): 1455-1489.

Time is up!

What did we learn?

Quantum algorithms have been designed for discrete optimization problems are limited in terms of

- Problem class: mainly unconstrained for annealing and NISQ
- Problem sizes
- Convergence guarantees

Think of very cool and potentially efficient heuristics!

- Fertile ground for decomposition algorithms where classical tackles each of these weaknesses!
  - Look for inspiration in classical literature but open your mind to new ideas!

BONUS: Decomposition in Quantum Chemistry

communications physics

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › communications physics › articles › article

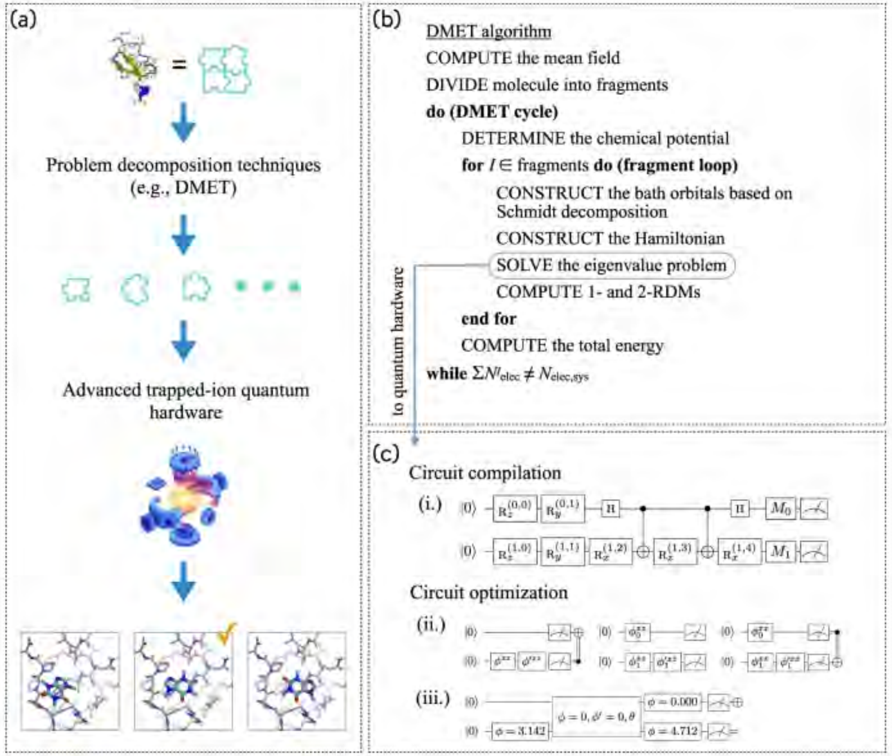
Article | Open access | Published: 18 November 2021

Optimizing electronic structure simulations on a trapped-ion quantum computer using problem decomposition

Yukio Kawashima, Erika Lloyd, Marc P. Coons, Yunseong Nam, Shunji Matsuura, Alejandro J. Garza, Sonika Johri, Lee Huntington, Valentin Senicourt, Andrii O. Maksymov, Jason H. V. Nguyen, Jungsang Kim, Nima Alidoust, Arman Zaribafiyan & Takeshi Yamazaki

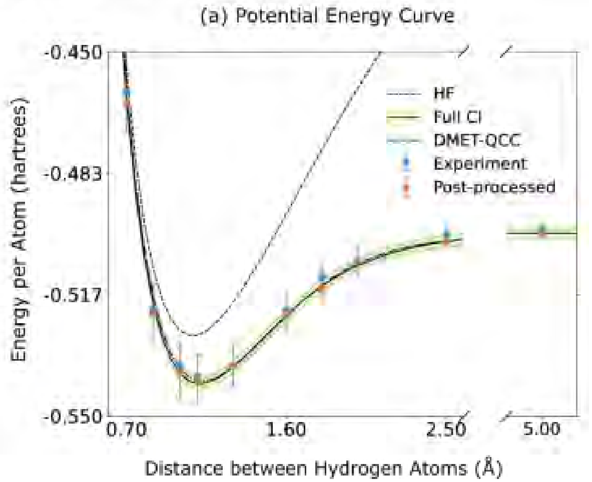
Communications Physics **4**, Article number: 245 (2021) | Cite this article

Fig. 1: Problem decomposition-based pipeline for efficient electronic structure simulation on a quantum computer.



a Schematic illustration of the pipeline. **b** The density matrix embedding theory (DMET) algorithm. "1- and 2-RDMs" refers to one- and two-particle reduced density matrices. **c** Pre- and post-optimizing compilation circuits. (i) Pre-optimizing compilation input, with the gates M_0 and M_1 chosen appropriately for the different measurement bases. (ii) Post-optimizing compilation outputs for the three output circuits for ZZ (left), XZ (middle), and XX (right). Note that XZ and ZX result in the same circuit, outside of the relabeling of the qubit indices. See Supplementary Tables 3 and 4 for the

Fig. 2: The potential energy curve of the sym hydrogen atoms (H_{10}).



For reference, we include classically simulated curves using several methods: Hartree–Fock (HF) shown with the dotted black line, full configuration interaction (full CI) shown with the solid black line, and density matrix embedding theory with the qubit coupled-cluster ansatz and variational quantum eigensolver (DMET-QCC) shown with the dashed black line. The full CI and DMET-QCC lines are surrounded by yellow and blue regions indicating chemical accuracy with respect to their values, and their overlap is indicated in green. The energies obtained by the experiment are shown in blue circles, and the post-processed energies are shown in red circles. The respective error bars for each are calculated using bootstrapping which is described in the "Methods" section. **a** The full potential energy

1. Kawashima, Yukio, et al. "Optimizing electronic structure simulations on a trapped-ion quantum computer using problem decomposition." Communications Physics 4.1 (2021): 245.

BONUS: Latest Hybrid Quantum-Classical Chemistry

arXiv > quant-ph > arXiv:2405.05068

Search...

Help | Adv

Quantum Physics

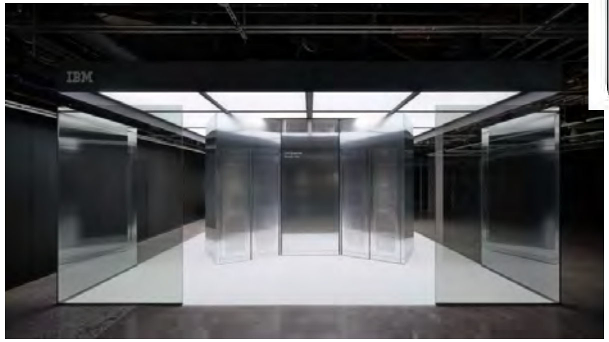
[Submitted on 8 May 2024]

Chemistry Beyond Exact Solutions on a Quantum-Centric Supercomputer

Javier Robledo-Moreno, Mario Motta, Holger Haas, Ali Javadi-Abh
Kunal Sharma, Sandeep Sharma, Tomonori Shirakawa, Iskandar S
Takita, Minh C. Tran, Seiji Yunoki, Antonio Mezzacapo

IBM Quantum

RIKEN



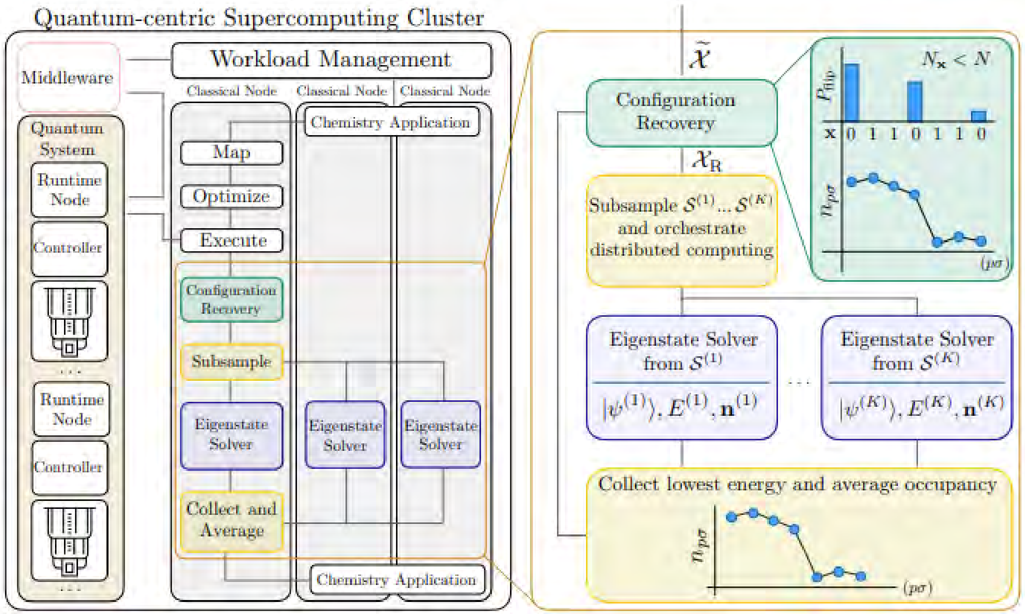


Figure 1. **Quantum-centric supercomputing architecture and workflow diagram. Left:** We illustrate a simplified architecture used to execute our workflow. The architecture has a cluster with a quantum system alongside classical runtime nodes within an isolated environment. A workload management system controls hybrid quantum-classical jobs through middleware. Our workflow is distributed on a set of classical nodes. It includes standard quantum chemistry application routines such as computing electronic integrals, mapping to qubits, and preparing circuits to be executed. **Right:** Details of the classical post-processing step. The input is a set of noisy samples $\widetilde{\mathcal{X}}$ from the quantum execution that are processed with our configuration recovery step, using information from a vector $\mathbf{n}$ of reference orbital occupancies. The green inset shows an example where a configuration with $N_x < N$ is corrected. The set of recovered configurations $\mathcal{X}_R$ is sub-sampled and distributed for projection and diagonalization on parallel classical nodes. A new average reference occupancy vector $\mathbf{n}$ is computed from the results, and the configuration recovery loop is repeated self-consistently until convergence.

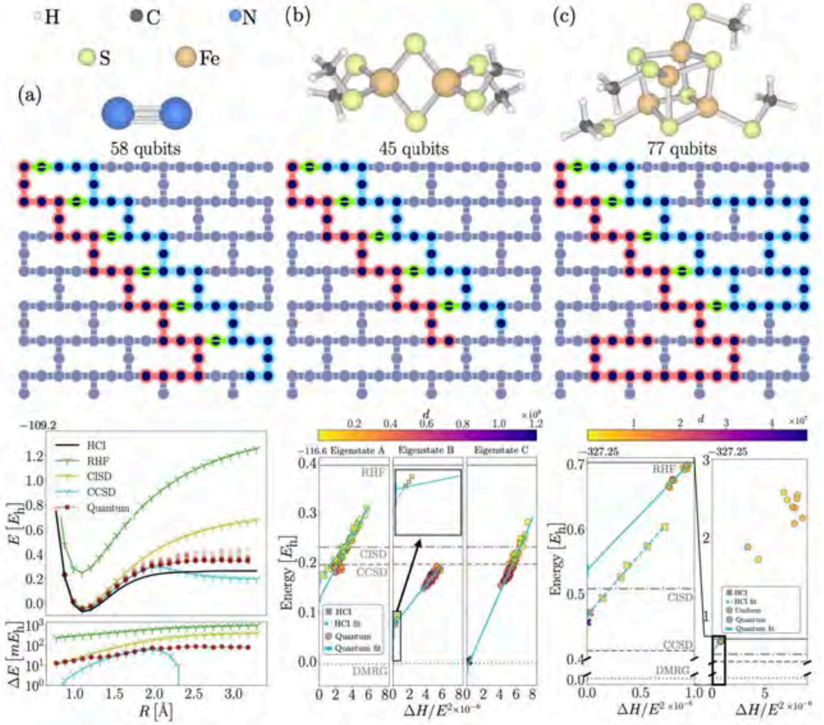


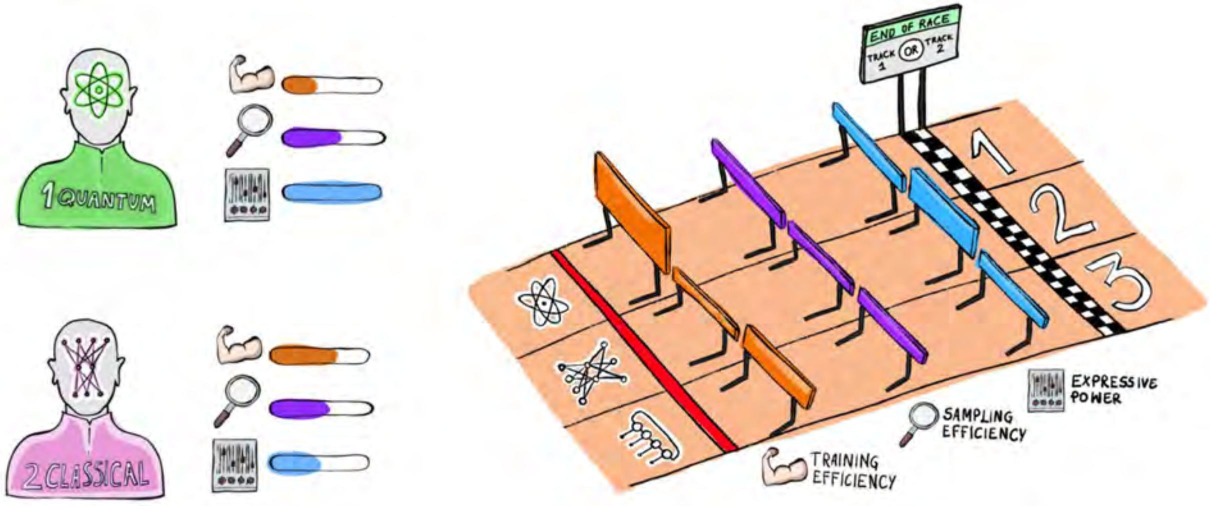
Figure 4. **Experiments: Chemistry on large basis sets. (a)** 58 qubits are used to model the N_2 dissociation (cc-pVDZ basis set). **(b)** 45 qubits are used for the [2Fe-2S] cluster (TZP-DKH basis set) and **(c)** 77 qubits for the [4Fe-4S] cluster (TZP-DKH basis set). The top panels show a 3-dimensional representation of the geometry of each molecule. The middle panels show the qubits selected on a Heron quantum processor layout, following the same color convention as panel (b) in Fig. 3. The **bottom panel in (a)** shows the potential energy surface comparison, as well as the energy difference ΔE between the Heat-Bath Configuration Interaction (HCI) energy and the energies obtained from different methods, including the quantum estimator. The brown scatter plot shows the value of $E^{(k)}$ for all batches of configurations and the connected dots show $\min_k(E^{(k)})$. The **bottom panel** in **(b)** shows the energy-variance analysis for three different eigenstates that both HCI and our method find upon increasing the value of d, as labeled by the color bar. For each approximate eigenstate $|\psi^{(k)}\rangle$, the horizontal axis $\Delta H = \langle\psi^{(k)}|\hat{H}^2|\psi^{(k)}\rangle - \langle\psi^{(k)}|\hat{H}|\psi^{(k)}\rangle^2$. The **bottom panel in (c)** shows a comparison of the energy-variance analysis applied to quantum measurement outcomes and bitstrings (with the correct particle number) sampled from the uniform distribution. The DMRG energy in panels (b) and (c) is from Ref. [45].

1. Robledo-Moreno, Javier, et al. "Chemistry beyond exact solutions on a quantum-centric supercomputer." arXiv preprint arXiv:2405.05068 (2024).

Decomposition approaches for arbitrary constraints

Using techniques from machine learning

Quantum within a generative framework for optimization – Cool idea!



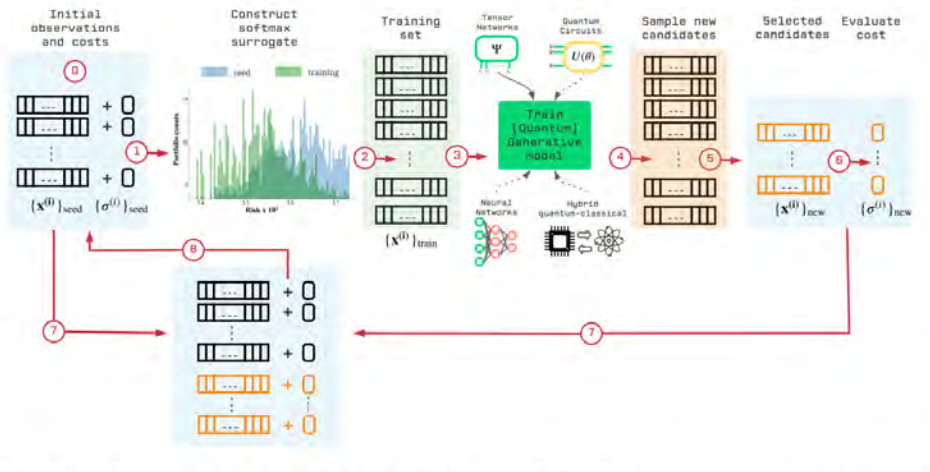
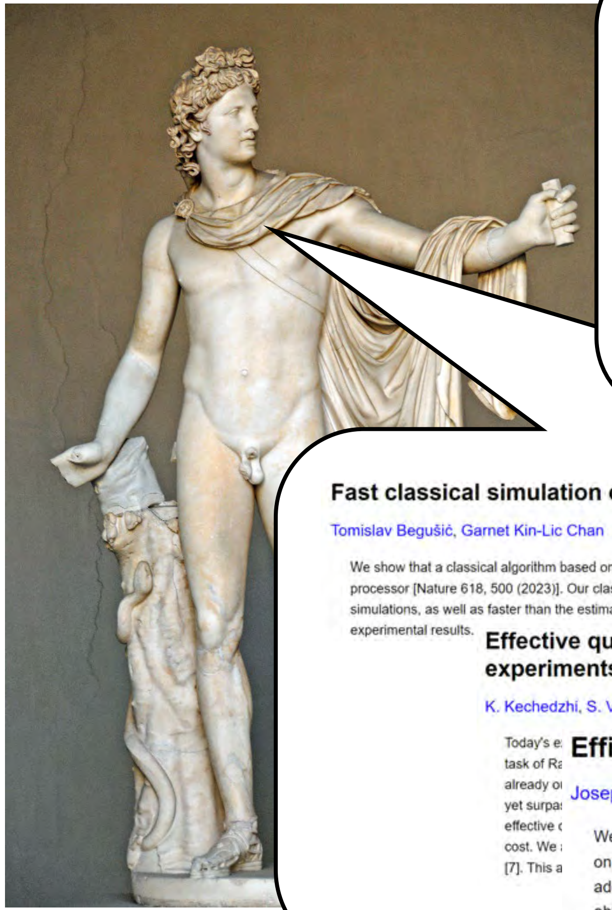
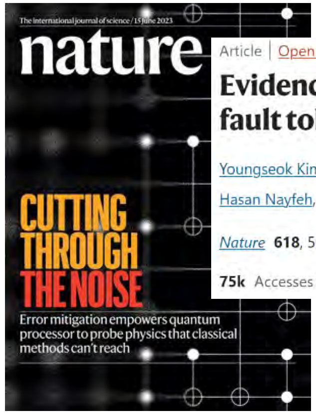


FIG. 1. **Scheme for our Generator-Enhanced Optimization (GEO) strategy.** The GEO framework leverages generative models to utilize previous samples coming from any quantum or classical solver. The trained quantum or classical generator is responsible for proposing candidate solutions which might be out of reach for conventional solvers. This *seed data set* (step 0) consists of observation bitstrings $\{x^{(i)}\}_{\text{seed}}$ and their respective costs $\{\sigma^{(i)}\}_{\text{seed}}$. To give more weight to samples with low cost, the seed samples and their costs are used to construct a *softmax* function which serves as a *surrogate* to the cost function but in probabilistic domain. This softmax surrogate also serves as a prior distribution from which the *training set* samples are withdrawn to train the generative model (steps 1-3). As shown in the figure between steps 1 and 2, training samples from the softmax surrogate are biased favoring those with low cost value. For the work presented here, we implemented a tensor-network (TN)-based generative model. Therefore, we refer to this quantum-inspired instantiation of GEO as TN-GEO. Other families of generative models from classical, quantum, or hybrid quantum-classical can be explored as expounded in the main text. The quantum-inspired generator corresponds to a tensor-network Born machine (TNBM) model which is used to capture the main features in the training data, and to propose new solution candidates which are subsequently post selected before their costs $\{\sigma^{(i)}\}_{\text{new}}$ are evaluated (steps 4-6). The new set is merged with the seed data set (step 7) to form an updated seed data set (step 8) which is to be used in the next iteration of the algorithm. More algorithmic details for the two TN-GEO strategies proposed here, as a *booster* or as a *stand-alone* solver, can be found in the main text and in A 5 and A 6 respectively.

1. Hibat-Allah, Mohamed, et al. "A framework for demonstrating practical quantum advantage: comparing quantum against classical generative models." Communications Physics 7.1 (2024): 68.
2. Alcazar, Javier, et al. "Enhancing combinatorial optimization with classical and quantum generative models." Nature Communications 15.1 (2024): 2761.

Foreseeable future





Article | Open Access | Published: 14 June 2023

Evidence for the utility of quantum computing before fault tolerance

Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout van den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme & Abhinav Kandala

Nature 618, 500–505 (2023) | Cite this article

75k Accesses | 1 Citations | 607 Altmetric | Metrics

Fast classical simulation of evidence for the utility of quantum computing before fault tolerance

Tomislav Begušić, Garnet Kin-Lic Chan

We show that a classical algorithm based on sparse Pauli dynamics can efficiently simulate quantum circuits studied in a recent experiment on 127 qubits of IBM's Eagle processor [Nature 618, 500 (2023)]. Our classical simulations on a single core of a laptop are orders of magnitude faster than the reported walltime of the quantum simulations, as well as faster than the estimated quantum hardware runtime without classical processing, and are in good agreement with the zero-noise extrapolated experimental results.

Effective quantum volume, fidelity and computational cost of noisy quantum processing experiments

K. Kechedzhi, S. V. Isakov, S. Mandrà, B. Villalonga, X. Mi, S. Boixo, V. Smelyanskiy

Efficient tensor network simulation of IBM's kicked Ising experiment

Joseph Tindall, Matt Fishman, Miles Stoudenmire, Dries Sels

We report an accurate, memory and time efficient classical simulation of a 127-qubit kicked Ising quantum system on the heavy-hexagon lattice. A simulation of this system on a quantum processor was recently performed using noise mitigation techniques to enhance accuracy (Nature volume 618, p. 500-505 (2023)). Here we show that, by adopting a tensor network approach that reflects the qubit connectivity of the device, we can perform a classical simulation that is significantly more accurate than the results obtained from the quantum device in the verifiable regime and comparable to the quantum simulation results for larger depths. The tensor network approach used will likely have broader applications for simulating the dynamics of quantum systems with tree-like correlations.

1. https://en.wikipedia.org/wiki/Classical_sculpture
2. https://en.wikipedia.org/wiki/Quantum_Man
3. Youngseok, et al. "Evidence for the utility of quantum computing before fault tolerance." Nature 618.7965 (2023)
4. https://arxiv.org/abs/2306.15970
5. https://arxiv.org/abs/2306.16372
6. https://arxiv.org/abs/2306.14887

PURDUE UNIVERSITY

Quantum-Classical Hybrid Methods for Optimization*

David E. Bernal Neira

May 28th, 2024

Principal Investigator of SECQUOIA

Assistant Professor - Davidson School of Chemical Engineering, Purdue University

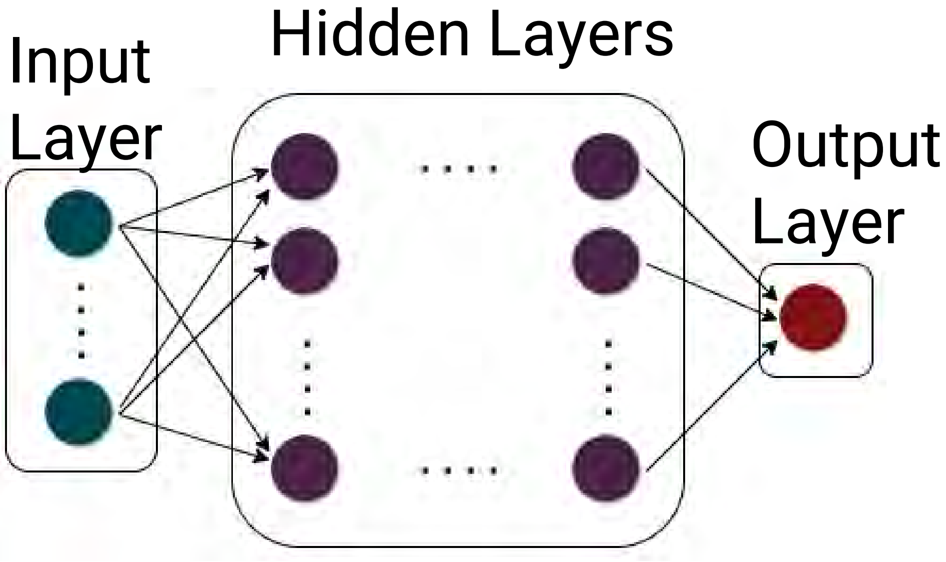
Master Class - Quantum Computing for CP, AI, and OR, and vice-versa

CPAIOR 2024, Uppsala, Sweden

Copositive decomposition using QUBO

How did we get here?

Verification of Feed-forward Neural Networks



Verify: for all $x \in X$, $f(x) \in Y$

With Ising solvers?

- Large number of continuous variables
- Optimality matters!

? Is this still a stop sign?

1. Brown, Robin, et al. "A copositive framework for analysis of hybrid ising-classical algorithms." SIAM Journal on Optimization 34.2 (2024): 1455-1489.

Copositive decomposition using QUBO

What problems are we interested in solving? Why?

- Mixed-binary quadratic programming

$$
\begin{aligned}
\min \quad & x^\top Q x + 2c^\top x && \text{Nonconvex quadratic objective} \\
& a_i^\top x = b_i,\ i \in 1, \ldots, N && \text{Linear Equality Constraints} \\
& x \geq 0 && \text{Non-negativity Constraint} \\
& x_j \in \{0, 1\},\ j \in B && \text{Binary Variables}
\end{aligned}
\qquad \text{(MBQP)}
$$

<u>Representative of:</u>

- Neural Network Verification
- Model Predictive Control
- Many NP-hard problems (e.g., maximum clique)
- Scheduling and Routing

1. Brown, Robin, et al. "A copositive framework for analysis of hybrid ising-classical algorithms." SIAM Journal on Optimization 34.2 (2024): 1455-1489.

Copositive decomposition using QUBO

Where is the convex part of MBQP?

Mixed-binary quadratic program (MBQP)

$$
\begin{aligned}
\min \quad & x^\top Q x + 2c^\top x \\
& a_i^\top x = b_i,\ i \in 1,\ldots,N \\
& x \geq 0 \\
& x_j \in \{0,1\},\ j \in B
\end{aligned}
$$

Completely positive program (CPP)

$$
\begin{aligned}
\min \quad & \left\langle \begin{pmatrix} Q & c \\ c^\top & \cdot \end{pmatrix}, \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \right\rangle \\
\text{s.t.} \quad & \left\langle \begin{pmatrix} \cdot & \frac{1}{2}a_i \\ \frac{1}{2}a_i^\top & \cdot \end{pmatrix}, \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \right\rangle = b_i,\ i \in 1,\ldots,N \\
& \left\langle \begin{pmatrix} a_i a_i^\top & \cdot \\ \cdot & \cdot \end{pmatrix}, \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \right\rangle = b_i^2,\ i \in 1,\ldots,N \\
& \left\langle \begin{pmatrix} -e_j e_j^\top & \frac{1}{2}e_j \\ \frac{1}{2}e_j^\top & \cdot \end{pmatrix}, \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \right\rangle = 0,\ j \in B \\
& \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \in C^*
\end{aligned}
$$

[Bur]: (MBQP) and (CPP) are equivalent

1. Burer, Samuel. "On the copositive representation of binary and continuous nonconvex quadratic programs." Mathematical Programming 120.2 (2009): 479-495.

Copositive decomposition using QUBO

What is the completely positive cone?

Completely Positive Cone:

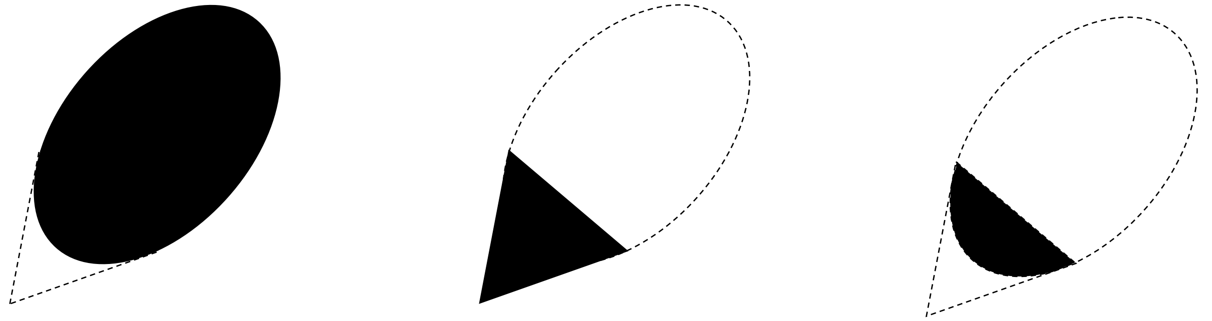
$$C_n^* = \{M \in \mathbb{R}^n \mid M = \sum_k v_{(k)} v_{(k)}^\top, v_{(k)} \in \mathbb{R}^n_{\boxed{+}}\}$$

Source of difference between complete positive and positive semidefinite

Positive Semidefinite S^+

Entrywise Nonnegative $\mathcal{N}$

Completely Positive C^*



1. Burer, Samuel. "On the copositive representation of binary and continuous nonconvex quadratic programs." Mathematical Programming 120.2 (2009): 479-495.

Copositive decomposition using QUBO

Is “copositive” a new way of saying “completely positive”?

Completely positive program (CPP)

$$
\begin{aligned}
\min \quad & \left\langle \begin{pmatrix} Q & c \\ c^\top & \cdot \end{pmatrix}, \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \right\rangle \\
\text{s.t.} \quad & \left\langle \begin{pmatrix} \cdot & \frac{1}{2}a_i \\ \frac{1}{2}a_i^\top & \cdot \end{pmatrix}, \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \right\rangle = b_i,\ i \in 1,\dots,N \\
& \left\langle \begin{pmatrix} a_i a_i^\top & \cdot \\ \cdot & \cdot \end{pmatrix}, \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \right\rangle = b_i^2,\ i \in 1,\dots,N \\
& \left\langle \begin{pmatrix} -e_j e_j^\top & \frac{1}{2}e_j \\ \frac{1}{2}e_j^\top & \cdot \end{pmatrix}, \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \right\rangle = 0,\ j \in B \\
& \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \in C^*
\end{aligned}
$$

[Bur]: (MBQP) and (CPP) are equivalent

Copositive program (CPP)

$$
\max \quad \gamma + \sum_i \mu_i^{(\mathrm{lin})} b_i + \mu_i^{(\mathrm{quad})} b_i^2
$$

$$
M(\mu, \lambda, \gamma) \in \mathcal{C}
$$

$$
M(\mu,\lambda,\gamma) := \begin{pmatrix} Q & c \\ c^\top & \cdot \end{pmatrix} - \sum_i \mu_i^{(\mathrm{lin})} \begin{pmatrix} \cdot & \frac{1}{2}a_i \\ \frac{1}{2}a_i^\top & \cdot \end{pmatrix} - \sum_i \mu_i^{(\mathrm{quad})} \begin{pmatrix} a_i a_i^\top & \cdot \\ \cdot & \cdot \end{pmatrix} - \sum_{j\in B} \lambda_j \begin{pmatrix} -e_j e_j^\top & \frac{1}{2}e_j \\ \frac{1}{2}e_j^\top & \cdot \end{pmatrix} - \gamma \begin{pmatrix} \cdot & \cdot \\ \cdot & 1 \end{pmatrix}
$$

Our contribution

Theorem: If (MBQP) is feasible with bounded feasible region, then (CPP) and (COP) are equivalent

1. Burer, Samuel. "On the copositive representation of binary and continuous nonconvex quadratic programs." Mathematical Programming 120.2 (2009): 479-495.

Copositive decomposition using QUBO

What is the completely positive cone?

Copositive: $C_n = \{M \in \mathbb{R}^n \mid x^\top M x \geq 0, \forall x \in \mathbb{R}^n_{\boxed{+}}\}$

Completely Positive: $C_n^* = \{M \in \mathbb{R}^n \mid M = \sum_k v_{(k)} v_{(k)}^\top, v_{(k)} \in \mathbb{R}^n_{\boxed{+}}\}$

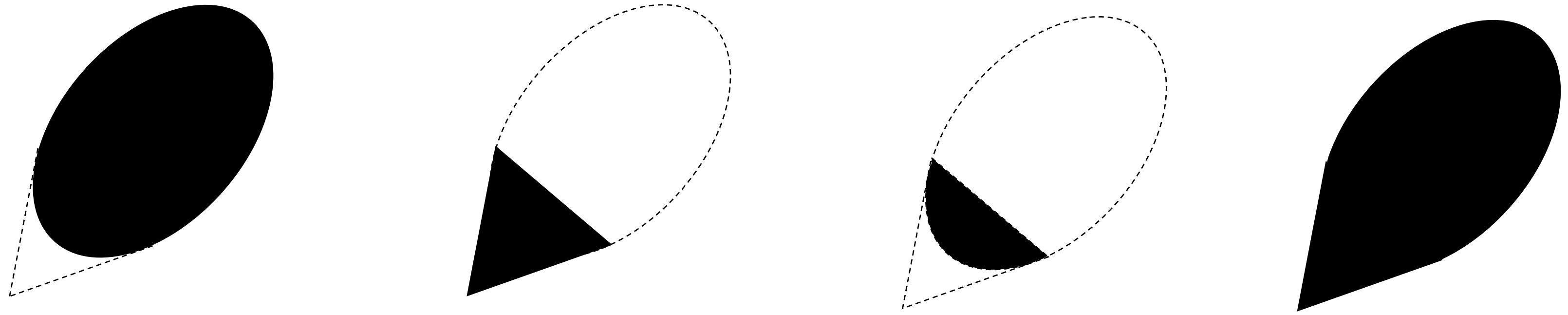
Source of difference between co/complete positive and positive semidefinite

Positive Semidefinite S^+

Entrywise Nonnegative $\mathcal{N}$

Completely Positive C^*

Copositive C



1. Burer, Samuel. "On the copositive representation of binary and continuous nonconvex quadratic programs." Mathematical Programming 120.2 (2009): 479-495.

Copositive decomposition using QUBO

Since the problem is convex now, is it easy to solve?

$$\max \quad \overbrace{\gamma + \sum_i \mu_i^{(\text{lin})} b_i + \mu_i^{(\text{quad})} b_i^2}^{\text{Linear objective}}$$

$$\left.\underbrace{M(\mu, \lambda, \gamma)}_{\text{Linear combination of constraint matrices}} \in \mathcal{C}\right\} \text{Convex cone}$$

Key insight:
Checking membership of the copositive cone is co-NP-complete, but can be solved via a non-convex quadratic minimization:

$$\underset{||z||\leq 1,\, z\in\mathbb{R}^n_{\geq 0}}{\text{minimize}} \quad z^\top M(\mu,\, \lambda,\, \gamma) z$$

Any negative solution can be used to construct a feasibility cut

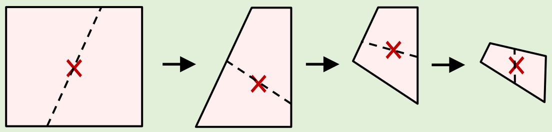
1. Burer, Samuel. "On the copositive representation of binary and continuous nonconvex quadratic programs." Mathematical Programming 120.2 (2009): 479-495.

Copositive decomposition using QUBO

How do you solve the copositive program?

Solution via cutting-plane algorithms

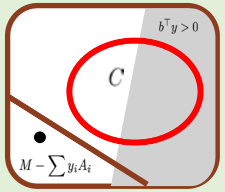
Compute center:
Selection of test points as some sort of center of the remaining search space ensures consistent progress when reducing the search space.



Checking copositivity:
Checking membership in the copositive cone is co-NP-complete, but can be solved via a non-convex quadratic minimization:

Add feasibility cut:
Convexity of the copositive cone means that an infeasible point can be extrapolated to rule out an entire halfspace.

A certificate of non-copositivity immediately gives rise to a separating hyperplane.



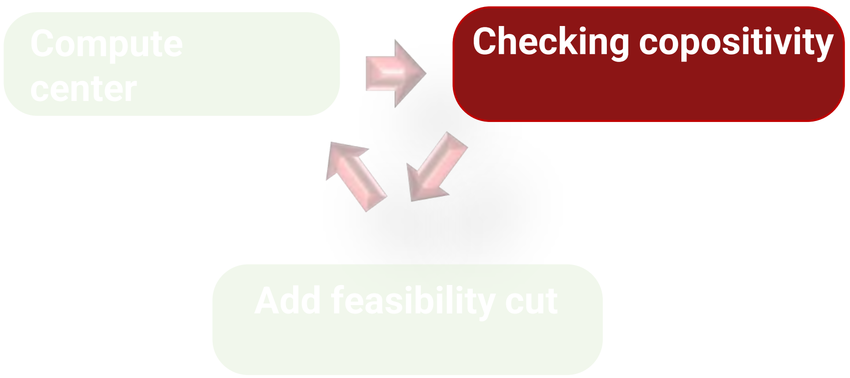
Ising solver

Carried out classically

1. Burer, Samuel. "On the copositive representation of binary and continuous nonconvex quadratic programs." Mathematical Programming 120.2 (2009): 479-495.

Copositive decomposition using QUBO

What does Ising/QUBO have to do with this?



<u>Non-convex quadratic minimization:</u>

$$\underset{||z||\leq 1,\, z\in\mathbb{R}^n_{\geq 0}}{\text{minimize}} \quad z^\top M(\mu,\, \lambda,\, \gamma) z$$

- Approximate as an Ising problem
- Does not require global minimum
- **Any** $z^\top M(\mu,\, \lambda,\, \gamma) z < 0$

  can be turned into a feasibility cut

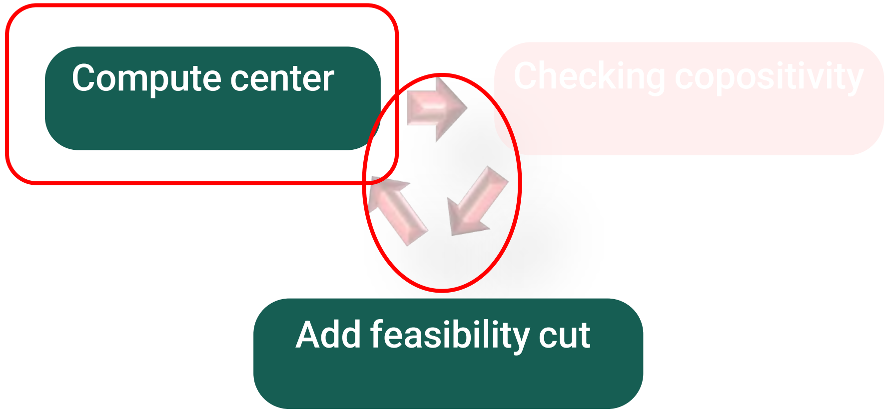
1. Burer, Samuel. "On the copositive representation of binary and continuous nonconvex quadratic programs." Mathematical Programming 120.2 (2009): 479-495.

Copositive decomposition using QUBO
Is the rest of the problem easy?

Cutting-plane algorithms shift complexity onto the copositivity checks

Variations in cutting plane algorithms trade-off:
- Complexity of computing a center
- The number of outer iterations

All are polynomial time in number of decision variables



Name	Oracle Queries	Total Run-time (excluding oracle queries)
Center of Gravity	$\mathcal{O}(m\log(\frac{R}{r}))$	#P-hard [42]
Ellipsoid	$\mathcal{O}(m^2\log(m\frac{R}{r}))$	$\mathcal{O}(m^4\log(m\frac{R}{r}))$
Inscribed Ellipsoid	$\mathcal{O}(m\log(m\frac{R}{r}))$	$\mathcal{O}((m\log(m\frac{R}{r}))^{4.5})$
Volumetric Center	$\mathcal{O}(m\log(m\frac{R}{r}))$	$\mathcal{O}(m^{1+\omega}\log(m\frac{R}{r}))$
Analytic Center	$\mathcal{O}(m\log^2(m\frac{R}{r}))$	$\mathcal{O}(m^{1+\omega}\log^2(m\frac{R}{r}) + (m\log(m\frac{R}{r}))^{2+\frac{\omega}{2}})$
Random Walk	$\mathcal{O}(m\log(m\frac{R}{r}))$	$\mathcal{O}(m^7\log(m\frac{R}{r}))$
Lee, Sidford, Wong	$\mathcal{O}(m\log(m\frac{R}{r}))$	$\mathcal{O}(m^3\log^{\mathcal{O}(1)}(m\frac{R}{r}))$

1. Burer, Samuel. "On the copositive representation of binary and continuous nonconvex quadratic programs." Mathematical Programming 120.2 (2009): 479-495.

Copositive decomposition using QUBO

Does this actually work?

$$\begin{aligned} \underset{x_1, x_2}{\text{minimize}} \quad & \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ -1 & \cdot \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ \text{subject to} \quad & x_1 + x_2 = 1, \\ & x_1,\, x_2 \in \mathbb{R}_{\geq 0}. \end{aligned}$$

$$x_1^* = \frac{1}{3}, \quad x_2^* = \frac{2}{3} \qquad \begin{pmatrix} X^* & x^* \\ x^{*\top} & 1 \end{pmatrix} = \begin{pmatrix} x_1^* \\ x_2^* \\ 1 \end{pmatrix} \begin{pmatrix} x_1^* & x_2^* & 1 \end{pmatrix} = \begin{pmatrix} 1/9 & 2/9 & 1/3 \\ 2/9 & 4/9 & 2/3 \\ 1/3 & 2/3 & 1 \end{pmatrix}$$

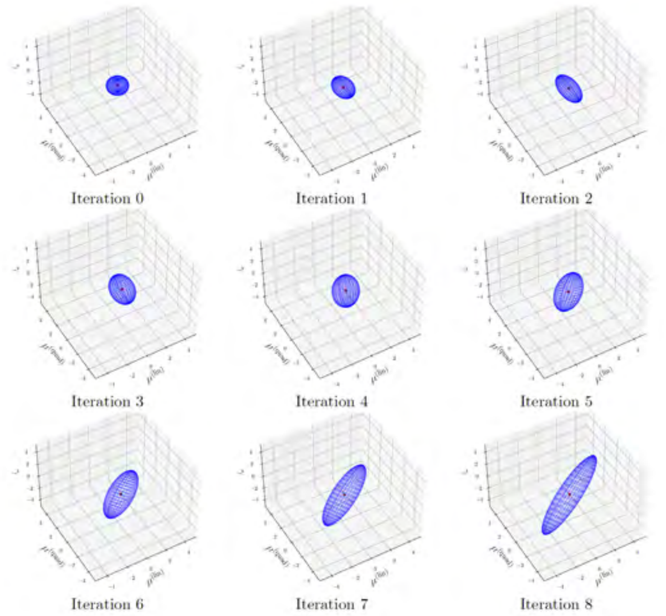
CPP Reformulation

$$\begin{aligned} \underset{X \in \mathbb{R}^{2\times 2}, x \in \mathbb{R}^2}{\text{minimize}} \quad & \left\langle \begin{pmatrix} 1 & -1 & \cdot \\ -1 & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}, \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \right\rangle \\ \text{subject to} \quad & \left\langle \begin{pmatrix} \cdot & \cdot & 1 \\ \cdot & \cdot & 1 \\ 1 & 1 & \cdot \end{pmatrix}, \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \right\rangle = 2, \\ & \left\langle \begin{pmatrix} 1 & 1 & \cdot \\ 1 & 1 & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}, \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \right\rangle = 1, \\ & \left\langle \begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 \end{pmatrix}, \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \right\rangle = 1, \\ & \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \in \mathcal{C}_3^*. \end{aligned}$$

COP Dual

$$\begin{aligned} \underset{\mu, \lambda, \gamma}{\text{maximize}} \quad & \gamma + 2\mu^{(\text{lin})} + \mu^{(\text{quad})} \\ \text{subject to} \quad & M(\mu, \lambda, \gamma) \in \mathcal{C}_3 \end{aligned}$$

$$M(\mu, \lambda, \gamma) = \begin{pmatrix} 1 - \mu^{(\text{quad})} & -1 - \mu^{(\text{quad})} & -\mu^{(\text{lin})} \\ -1 - \mu^{(\text{quad})} & -\mu^{(\text{quad})} & -\mu^{(\text{lin})} \\ -\mu^{(\text{lin})} & -\mu^{(\text{lin})} & -\gamma \end{pmatrix}$$



1. Burer, Samuel. "On the copositive representation of binary and continuous nonconvex quadratic programs." Mathematical Progran

Copositive decomposition using QUBO

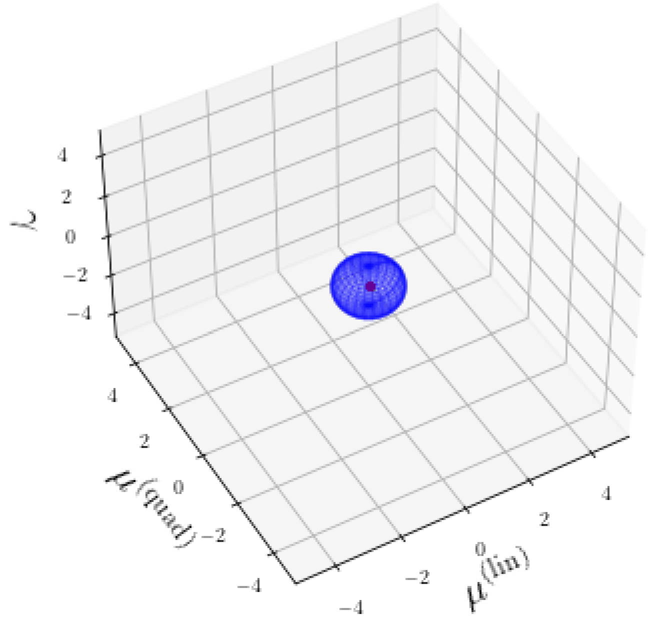
Does this actually work (x2)?

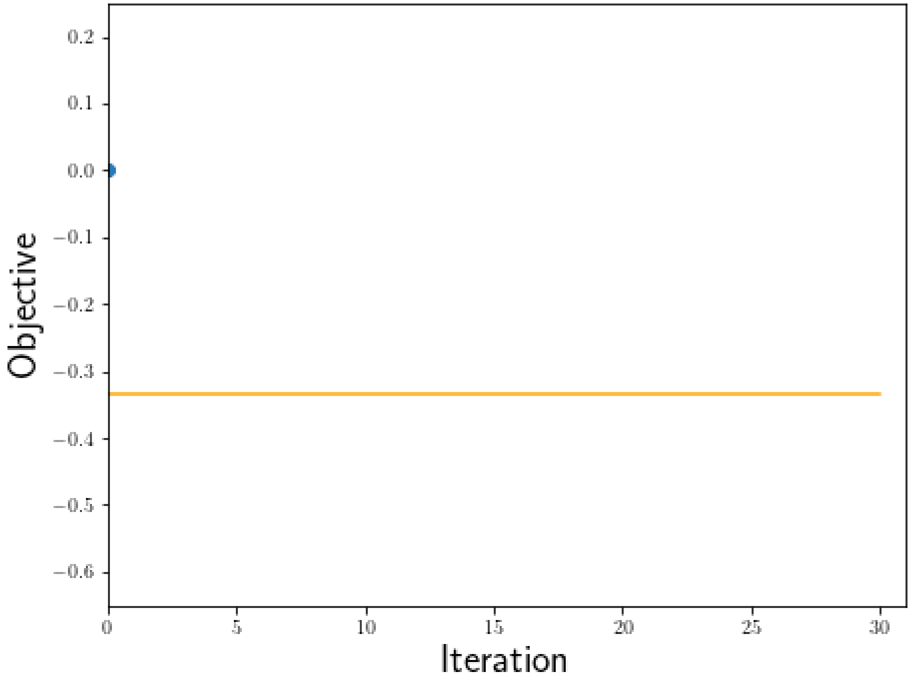
$$\begin{array}{ll} \underset{x_1, x_2}{\text{minimize}} & \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ -1 & . \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ \text{subject to} & x_1 + x_2 = 1, \\ & x_1,\ x_2 \in \mathbb{R}_{\geq 0}. \end{array}$$

$$x_1^* = \frac{1}{3}, \quad x_2^* = \frac{2}{3}$$

$$\begin{array}{ll} \underset{\mu, \lambda, \gamma}{\text{maximize}} & \gamma + 2\mu^{(\text{lin})} + \mu^{(\text{quad})} \\ \text{subject to} & M(\mu, \lambda, \gamma) \in \mathcal{C}_3 \end{array}$$

$$M(\mu, \lambda, \gamma) = \begin{pmatrix} 1 - \mu^{(\text{quad})} & -1 - \mu^{(\text{quad})} & -\mu^{(\text{lin})} \\ -1 - \mu^{(\text{quad})} & -\mu^{(\text{quad})} & -\mu^{(\text{lin})} \\ -\mu^{(\text{lin})} & -\mu^{(\text{lin})} & -\gamma \end{pmatrix}$$

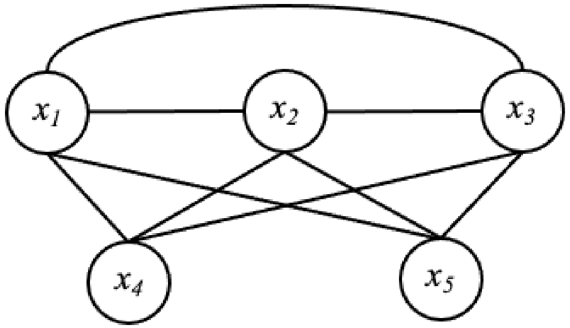




1. Burer, Samuel. "On the c

Copositive decomposition using QUBO

How can we assess the efficiency of our method?



$$\overline{A} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & 1 & \cdot \end{pmatrix}$$

$$\min_{\lambda} \quad \lambda$$

$$\begin{pmatrix} \lambda - 1 & -1 & -1 & -1 & -1 \\ -1 & \lambda - 1 & -1 & -1 & -1 \\ -1 & -1 & \lambda - 1 & -1 & -1 \\ -1 & -1 & -1 & \lambda - 1 & \lambda - 1 \\ -1 & -1 & -1 & \lambda - 1 & \lambda - 1 \end{pmatrix} \in C_5$$

Completely positive program (CPP)

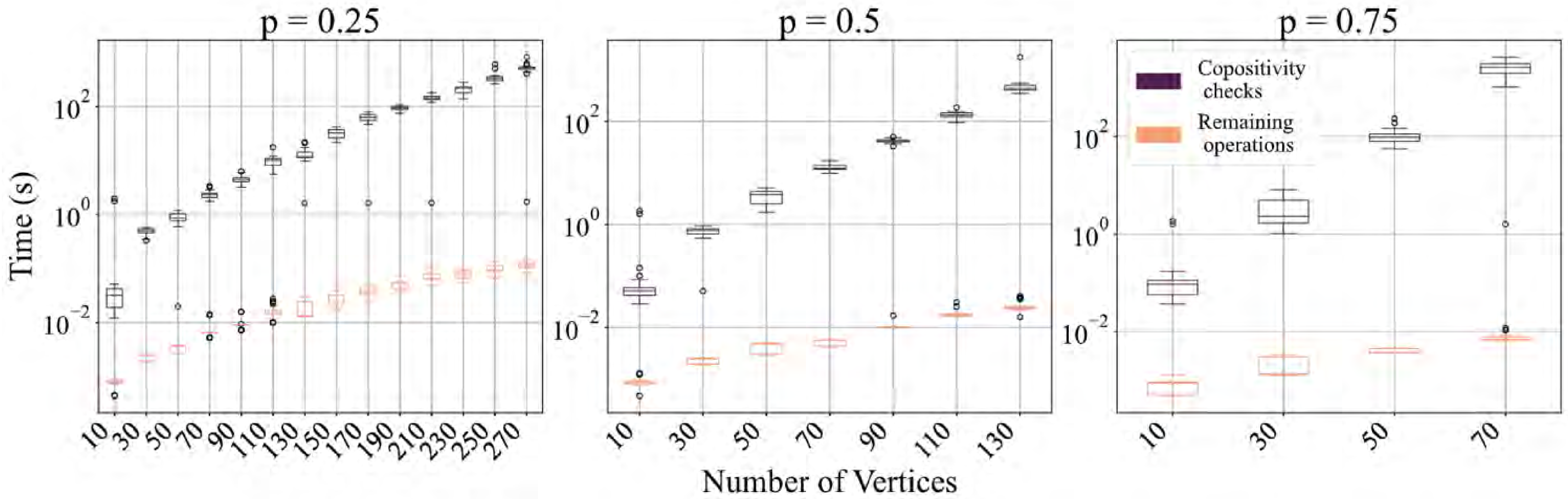
$$\begin{aligned} \underset{X \in \mathbb{R}^{n \times n}}{\text{maximize}} \quad & \langle \mathbb{1}\mathbb{1}^\top, X \rangle \\ \text{subject to} \quad & \langle \overline{A} + I, X \rangle = 1, \\ & X \in \mathcal{C}_n^*, \end{aligned}$$

Copositive program (COP)

$$\begin{aligned} \underset{\lambda \in \mathbb{R}}{\text{minimize}} \quad & \lambda \\ \text{subject to} \quad & \lambda(I + \overline{A}) - \mathbb{1}\mathbb{1}^\top \in \mathcal{C}_n \end{aligned}$$

Copositive decomposition using QUBO

What problems are we interested in solving? Why?

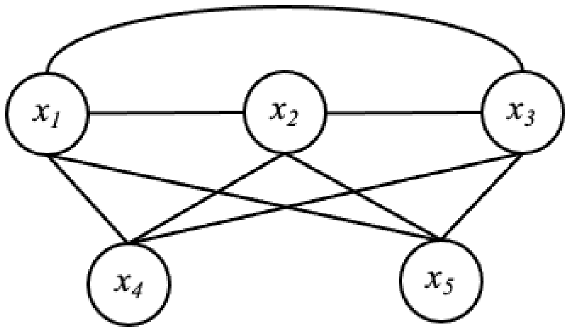


Time it takes to solve MaxClique problems using cutting plane algorithm and solving copositivity check using 1 thread in a computer running Gurobi 9.0.3

1. Anstreicher, Kurt M. "Testing copositivity via mixed–integer linear programming." Linear Algebra and its Applications 609 (2021): 218-230.

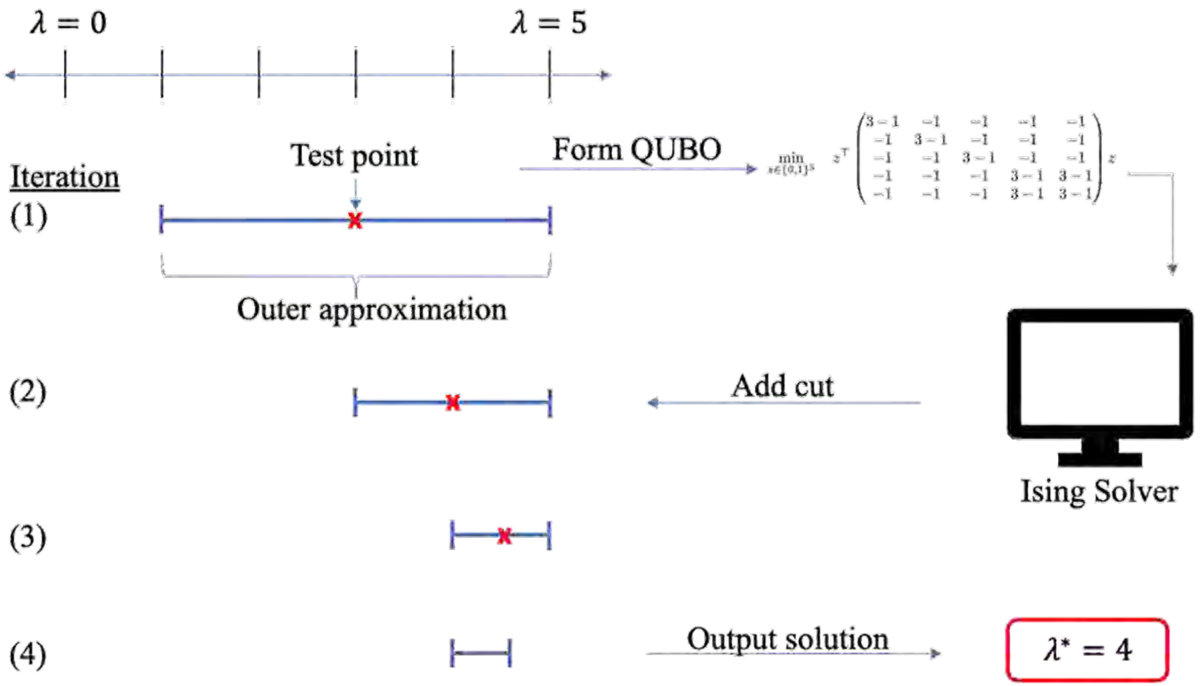
Copositive decomposition using QUBO

Where comes the Ising solver?



$$\overline{A} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & 1 & \cdot \end{pmatrix}$$

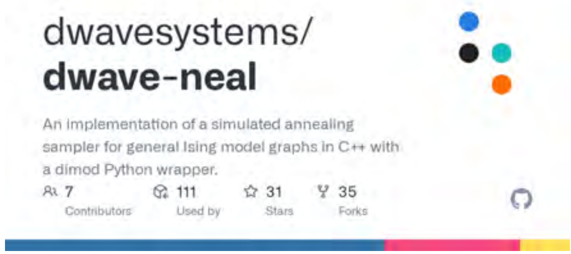
$$\min_{\lambda} \quad \lambda$$
$$\begin{pmatrix} \lambda-1 & -1 & -1 & -1 & -1 \\ -1 & \lambda-1 & -1 & -1 & -1 \\ -1 & -1 & \lambda-1 & -1 & -1 \\ -1 & -1 & -1 & \lambda-1 & \lambda-1 \\ -1 & -1 & -1 & \lambda-1 & \lambda-1 \end{pmatrix} \in C_5$$



Copositive decomposition using QUBO

What if you used an Ising solvers for the copositive program?

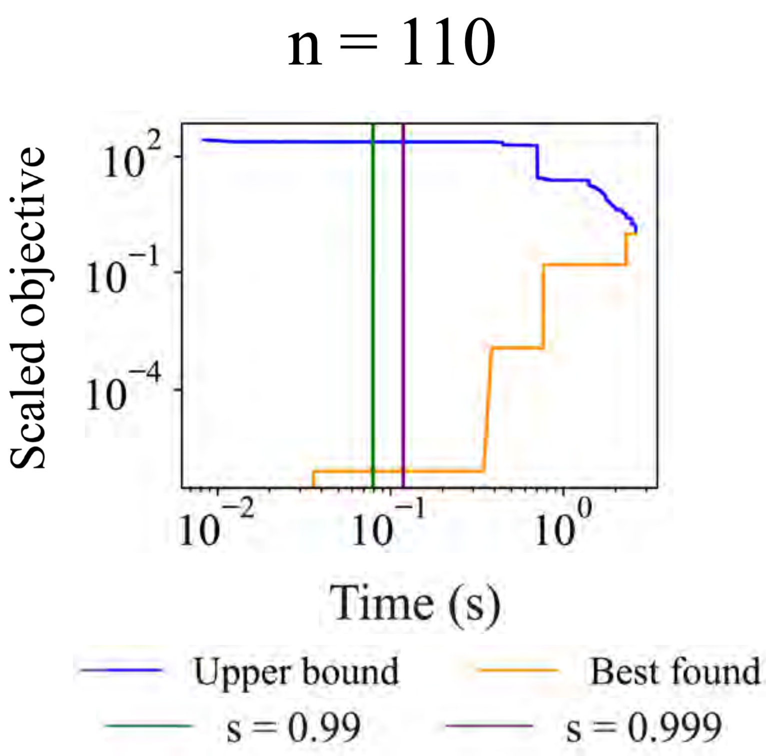
Sample trajectories of Gurobi's upper and lower bounds of scaled objective (optimal solution=1) against the time to solution to 99% and 99.9% certainty using a simulated annealing code dwave-neal.



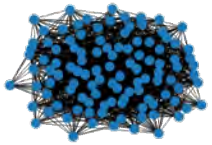
docs.ocean.dwavesys.com/projects/neal/en/latest/

$$p_{\text{success}} = \frac{\text{num}_{\text{OK}}}{\text{num}_{\text{trials}}}$$

$$\text{TTT} = t\frac{\log(1 - s)}{\log(1 - p_{\text{sucess}})}$$



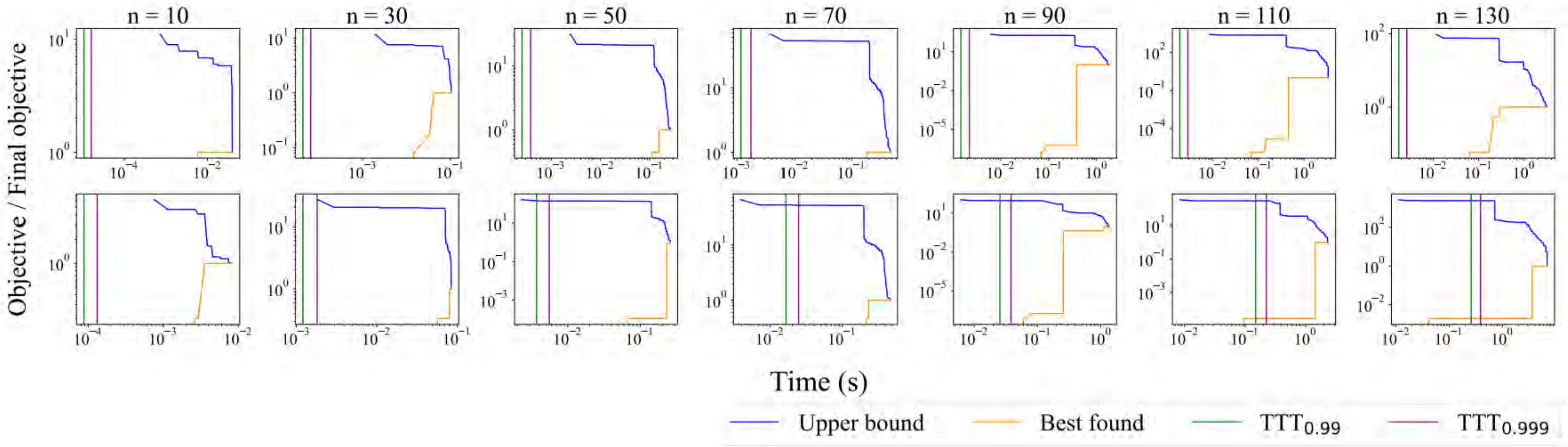
Maxclique problem of random Erdos-Renyi graph



- Density of 25%
- Size of 110
- Subproblem in cutting plane algorithm whose optimal solution is negative
  - num_{OK} are samples with negative objective.
  - $\text{num}_{\text{trials}}$ usually 1000

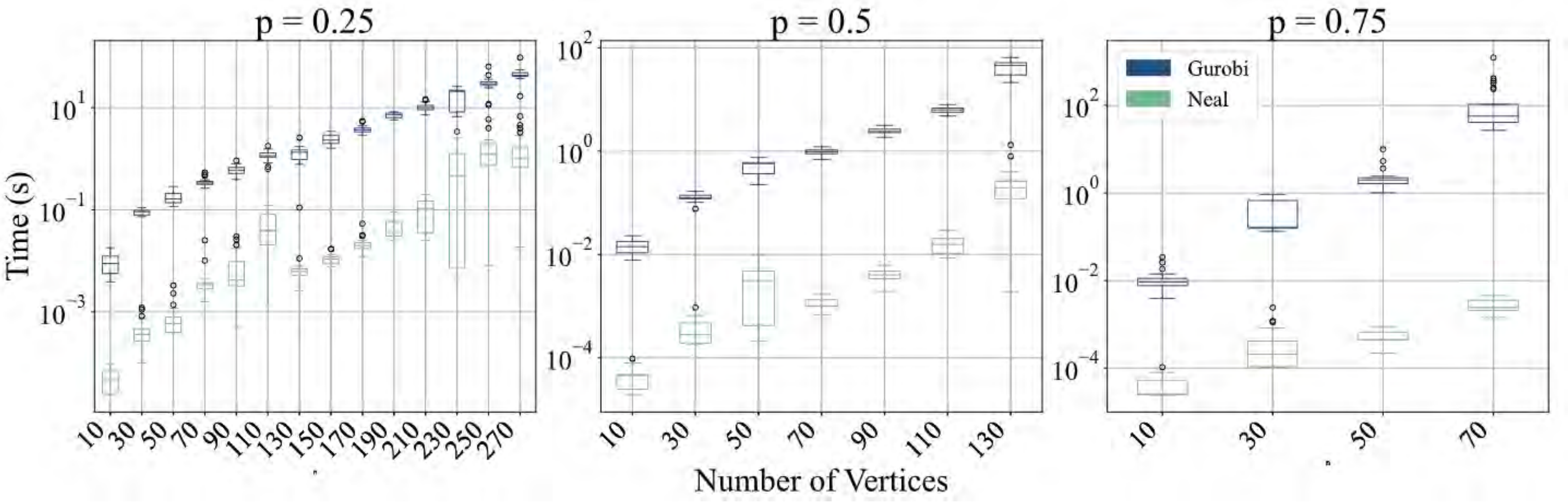
Copositive decomposition using QUBO

Does it work on a single instance?



Instances or random Erdos-Renyi graphs with density 25%.
Sample trajectories of Gurobi's upper and lower bounds of scaled objective against the time to solution to 99% and 99.9% confidence

Copositive decomposition using QUBO
Does it only work on sparse graphs?

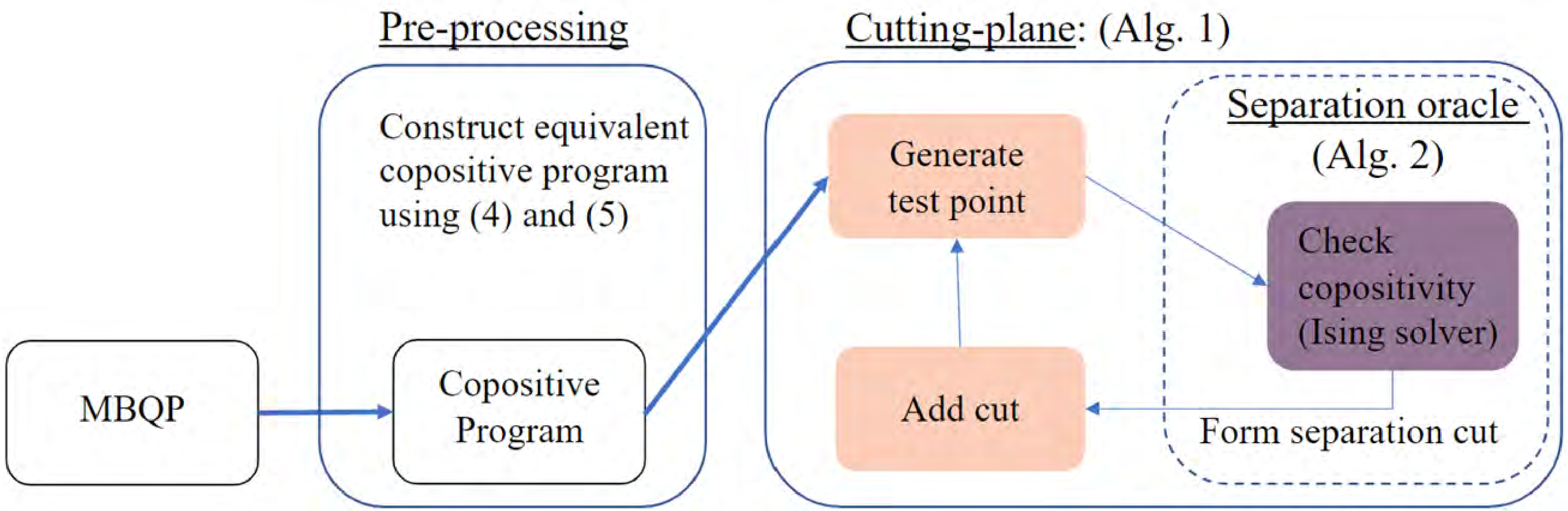


Simulated annealing TTT against solution time of Gurobi for 2 discretization points in the copositivity checks. For all densities, both methods scale exponentially with the number of vertices in the graph; however, SA is several orders of magnitude faster than Gurobi.

Copositive decomposition using QUBO

How about the remaining of the algorithm beyond copositivity?

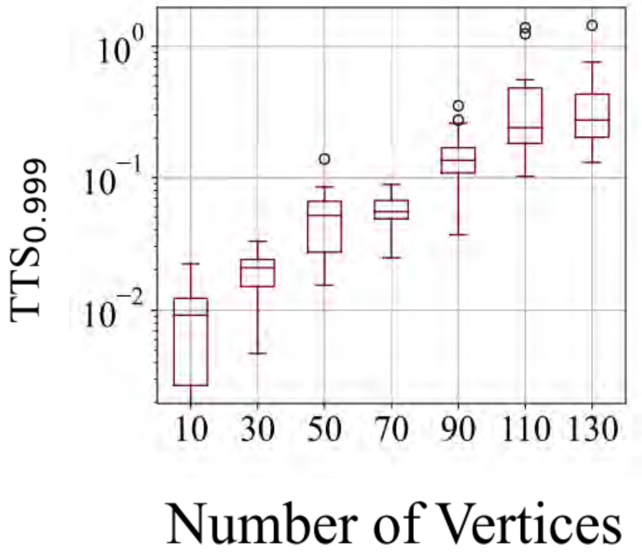
Solution process in a nutshell



Copositive decomposition using QUBO

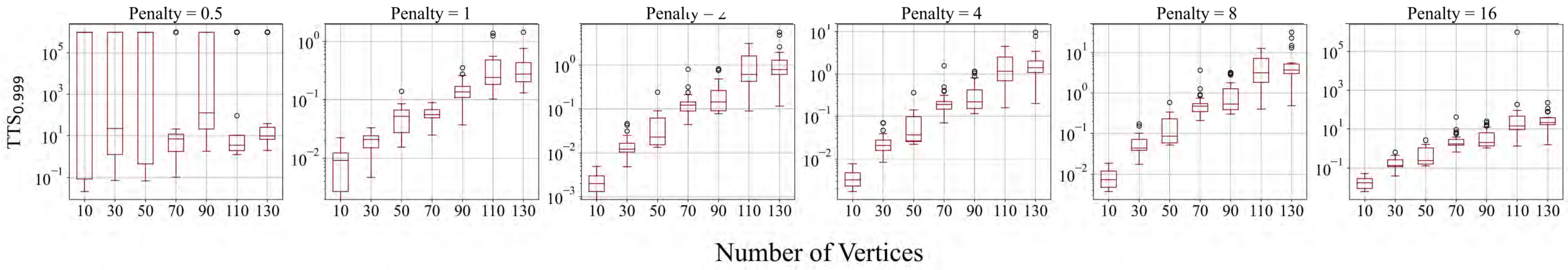
What if we had used a QUBO reformulation of MAXCLIQUE?

Using the provable tightest penalty (ρ =1)



Instances or random Erdos-Renyi graphs with density 25%

Performance varies with ρ!



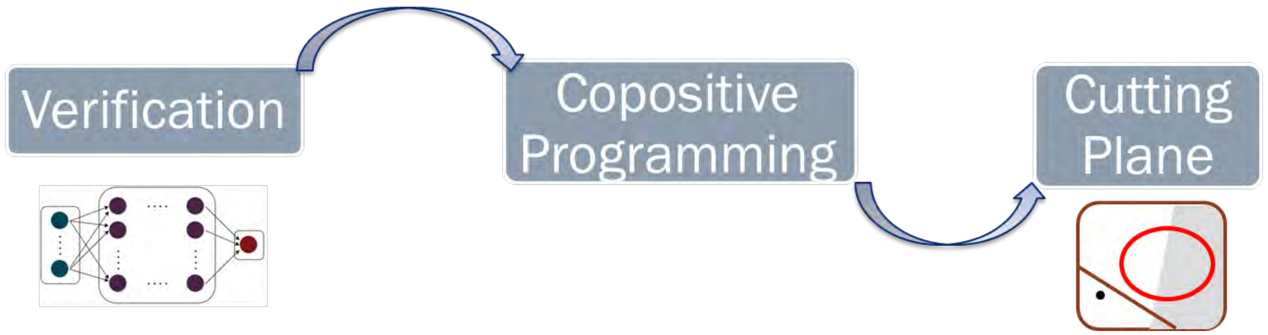
Copositive decomposition using QUBO

What is left to do?

Implementation and next steps

www.github.com/SECQUOIA/copositive-cutting-plane-max-clique

<u>Our Contribution:</u> Implementation of the proposed solution method



- Try other Ising solvers – Great results using the not most competitive alternative
- Apply to MBQPs and benchmark it against full MINLP solvers
- Refinement in cutting-plane algorithm, Ising non-copositivity certificate, …
- Discretization to be tackled via randomization algorithms

Copositive decomposition using QUBO

Proof of strong duality

6.1 Proof of strong duality

Problem (CPP) is equivalent to the following homogenous form completely positive program (i.e., min (CPP) = min (Hom-CPP)):

$$\begin{aligned}
\underset{x \in \mathbb{R}^n, X \in \mathbb{R}^{n \times n}}{\text{minimize}} \quad & \left\langle \begin{pmatrix} Q & c \\ c^\top & \cdot \end{pmatrix}, \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \right\rangle \\
\text{subject to} \quad & \left\langle \begin{pmatrix} A_{i,*}^\top A_{i,*} & -b_i A_{i,*}^\top \\ -b_i A_{i,*} & b_i^2 \end{pmatrix}, \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \right\rangle = 0, \\
& \left\langle \begin{pmatrix} -\mathbb{1}_{\{j\}} \mathbb{1}_{\{j\}}^\top & \frac{1}{2}\mathbb{1}_{\{j\}} \\ \frac{1}{2}\mathbb{1}_{\{j\}}^\top & \cdot \end{pmatrix}, \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \right\rangle = 0, \ \forall j \in B, \\
& \begin{pmatrix} X & x \\ x^\top & 1 \end{pmatrix} \in \mathcal{C}_{n+1}^*.
\end{aligned} \qquad \text{(Hom-CPP)}$$

This form will be useful for proving strong duality. Because the homogenized from of the equality constraints form a cone, this perspective will help prove strong duality between Problem (CPP) and its dual. The Lagrangian dual of (Hom-CPP) is the following copositive optimization problem:

$$\begin{aligned}
\underset{\mu, \lambda, \gamma}{\text{maximize}} \quad & \gamma \\
\text{subject to} \quad & \hat{M}(\mu, \lambda, \gamma) \in \mathcal{C}_{n+1},
\end{aligned} \qquad \text{(Hom-COP)}$$

where $\hat{M}(\mu, \lambda, \gamma)$ is defined as

$$\begin{aligned}
\hat{M}(\mu, \lambda, \gamma) := & \begin{pmatrix} Q & c \\ c^\top & \cdot \end{pmatrix} - \sum_i \mu_i \begin{pmatrix} A_{i,*}^\top A_{i,*} & -b_i A_{i,*}^\top \\ -b_i A_{i,*} & b_i^2 \end{pmatrix} \\
& - \sum_{j \in B} \lambda_j \begin{pmatrix} -\mathbb{1}_{\{j\}} \mathbb{1}_{\{j\}}^\top & \frac{1}{2}\mathbb{1}_{\{j\}} \\ \frac{1}{2}\mathbb{1}_{\{j\}}^\top & \cdot \end{pmatrix} - \gamma \begin{pmatrix} \cdot & \\ & 1 \end{pmatrix}.
\end{aligned} \qquad (23)$$

Theorem 6.1 (Homogeneous Strong Duality). *If Problem (MBQP) is feasible with bounded feasible region, then strong duality holds between Problems (Hom-CPP) and (Hom-COP).*

Proof. Notice that the set of affine constraints,

$$\mathcal{NULL} := \left\{ \tilde{X} \mid \left\langle \begin{pmatrix} A_{i,*}^\top A_{i,*} & -b_i A_{i,*}^\top \\ -b_i A_{i,*} & b_i^2 \end{pmatrix}, \tilde{X} \right\rangle = 0, \left\langle \begin{pmatrix} -\mathbb{1}_{\{j\}} \mathbb{1}_{\{j\}}^\top & \frac{1}{2}\mathbb{1}_{\{j\}} \\ \frac{1}{2}\mathbb{1}_{\{j\}}^\top & \cdot \end{pmatrix}, \tilde{X} \right\rangle = 0 \right\}, \qquad (24)$$

forms a cone. So, we could express (Hom-CPP) as the following optimization problem:

$$\begin{aligned}
\underset{\tilde{X} \in \mathbb{R}^{(n+1) \times (n+1)}}{\text{minimize}} \quad & \left\langle \begin{pmatrix} Q & c \\ c^\top & \cdot \end{pmatrix}, \tilde{X} \right\rangle \\
\text{subject to} \quad & \left\langle \begin{pmatrix} \cdot & \\ & 1 \end{pmatrix}, \tilde{X} \right\rangle = 1, \\
& \tilde{X} \in \mathcal{C}_{n+1}^* \cap \mathcal{NULL}.
\end{aligned} \qquad (25)$$

As a quick aside, rewriting the problem in this way does not change the Lagrangian dual problem. To see this, we first write the Lagrangian dual of Problem (25) as

$$\begin{aligned}
\underset{\gamma \in \mathbb{R}, \tilde{M} \in \mathbb{R}^{(n+1) \times (n)}}{\text{maximize}} \quad & \gamma \\
\text{subject to} \quad & \tilde{M} = \begin{pmatrix} Q & c \\ c^\top & \cdot \end{pmatrix} - \gamma \begin{pmatrix} \cdot & \\ & 1 \end{pmatrix}, \\
& \tilde{M} \in \mathcal{C}_{n+1} + \mathcal{NULL}^*
\end{aligned} \qquad (26)$$

and notice that $\mathcal{NULL}^*$ is spanned by

$$\left\{ \begin{pmatrix} A_{i,*}^\top A_{i,*} & -b_i A_{i,*}^\top \\ -b_i A_{i,*} & b_i^2 \end{pmatrix} \right\} \cup \left\{ \begin{pmatrix} -\mathbb{1}_{\{j\}} \mathbb{1}_{\{j\}}^\top & \frac{1}{2}\mathbb{1}_{\{j\}} \\ \frac{1}{2}\mathbb{1}_{\{j\}}^\top & \cdot \end{pmatrix}] \right\}. \qquad (27)$$

Here, we take care to note that the *Lagrangian* dual may differ from the *conic* dual. In particular, feasibility of (MBQP) is sufficient for ensuring strong duality when $\tilde{M}$ is optimized over $(C^* \cap \mathcal{NULL})^*$

Copositive decomposition using QUBO

Proof of strong duality

[35, Prop 5.3.9], however it is not guaranteed that $(C^* \cap \mathcal{NULL})^*$ is equal to $C + \mathcal{NULL}^*$–this is the case if and only if $C + \mathcal{NULL}^*$ is closed.

To establish strong duality, we will first assert that if Problem (MBQP) is feasible with bounded feasible region, then Problem (Hom-CPP) has a nonempty and bounded set of optimal solutions. This follows directly from [8, Corollary 2.6], which states that for all optimal solutions, (x^*, X^*), of (Hom-CPP), x^* must lie within the convex hull of optimal solutions for (MBQP). If the set of optimal solutions for (MBQP) is non-empty and bounded, so is their convex hull. This establishes that the set of optimal solutions of (Hom-CPP) is also non-empty and bounded, allowing us to apply [63, Theorem 1.1].

□

Strong duality means that

$$\max(\text{Hom-COP}) = \min(\text{Hom-CPP}) \tag{28}$$

Theorem 6.2 (Inhomogeneous Lower Bound). *The optimal objective of Problem* (COP) *is at least that of Problem* (Hom-CPP) *(i.e.,* $\max(\text{COP}) \geq \max(\text{Hom-COP})$*).*

Proof. We will do this by showing that for each $(\hat{\mu}, \hat{\lambda}, \hat{\gamma})$ there exists (μ, λ, γ) such that

$$M(\mu, \lambda, \gamma) = \hat{M}(\hat{\mu}, \hat{\lambda}, \hat{\gamma}), \tag{29}$$

and

$$\gamma + \sum_i \mu_i^{(\text{lin})} b_i + \mu_i^{(\text{quad})} b_i^2 = \hat{\gamma}. \tag{30}$$

In other words, any feasible solution for (Hom-COP) can be transformed into a feasible solution for (COP) with equal objective value. To see this, we will suggestively break up $\gamma = \gamma^{(\text{res})} + \sum_i \gamma_i$ so equation (3) can be expanded as

$$\begin{aligned} M(\mu, \lambda, \gamma) =& \begin{pmatrix} Q & c \\ c^\top & \cdot \end{pmatrix} \\ & - \sum_i \left(\mu_i^{(\text{lin})} \begin{pmatrix} \cdot & \frac{1}{2} A_{i,*}^\top \\ \frac{1}{2} A_{i,*} & \cdot \end{pmatrix} + \mu_i^{(\text{quad})} \begin{pmatrix} A_{i,*}^\top A_{i,*} & \cdot \\ \cdot & \cdot \end{pmatrix} + \gamma_i \begin{pmatrix} \cdot & \cdot \\ \cdot & 1 \end{pmatrix} \right) \\ & - \sum_{j \in B} \lambda_j \begin{pmatrix} -\mathbb{1}_{\{j\}} \mathbb{1}_{\{j\}}^\top & \frac{1}{2} \mathbb{1}_{\{j\}} \\ \frac{1}{2} \mathbb{1}_{\{j\}}^\top & \cdot \end{pmatrix} - \gamma^{(\text{res})} \begin{pmatrix} \cdot & \cdot \\ \cdot & 1 \end{pmatrix} \end{aligned} \tag{31}$$

Then, the proposed (μ, λ, γ) is given by

$$\lambda_j = \hat{\lambda}_j \tag{32}$$

$$\mu_i^{(\text{lin})} = -2 b_i \hat{\mu}_i \tag{33}$$

$$\mu_i^{(\text{quad})} = \hat{\mu}_i \tag{34}$$

$$\gamma_i = b_i^2 \hat{\mu}_i \tag{35}$$

$$\gamma^{(\text{res})} = \hat{\gamma} \tag{36}$$

Then, notice that

$$\mu_i^{(\text{lin})} \begin{pmatrix} \cdot & \frac{1}{2} A_{i,*}^\top \\ \frac{1}{2} A_{i,*} & \cdot \end{pmatrix} + \mu_i^{(\text{quad})} \begin{pmatrix} A_{i,*}^\top A_{i,*} & \cdot \\ \cdot & \cdot \end{pmatrix} + \gamma_i \begin{pmatrix} \cdot & \cdot \\ \cdot & 1 \end{pmatrix} \tag{37}$$

$$= -2 b_i \hat{\mu}_i \begin{pmatrix} \cdot & \frac{1}{2} A_{i,*}^\top \\ \frac{1}{2} A_{i,*} & \cdot \end{pmatrix} + \hat{\mu}_i \begin{pmatrix} A_{i,*}^\top A_{i,*} & \cdot \\ \cdot & \cdot \end{pmatrix} + b_i^2 \hat{\mu}_i \begin{pmatrix} \cdot & \cdot \\ \cdot & 1 \end{pmatrix} \tag{38}$$

$$= \hat{\mu}_i \left(-2 b_i \begin{pmatrix} \cdot & \frac{1}{2} A_{i,*}^\top \\ \frac{1}{2} A_{i,*} & \cdot \end{pmatrix} + \begin{pmatrix} A_{i,*}^\top A_{i,*} & \cdot \\ \cdot & \cdot \end{pmatrix} + b_i^2 \begin{pmatrix} \cdot & \cdot \\ \cdot & 1 \end{pmatrix} \right) \tag{39}$$

$$= \hat{\mu}_i \begin{pmatrix} A_{i,*}^\top A_{i,*} & -b_i A_{i,*}^\top \\ -b_i A_{i,*} & b_i^2 \end{pmatrix} \tag{40}$$

Copositive decomposition using QUBO

Proof of strong duality

so by matching up terms in the sums, we see that $M(\mu, \lambda, \gamma) = \hat{M}(\hat{\mu}, \hat{\lambda}, \hat{\gamma})$. As for the objective value,

$$
\begin{aligned}
&\gamma + \sum_i \mu_i^{(\text{lin})} b_i + \mu_i^{(\text{quad})} b_i^2 && (41)\\
&= \gamma^{(\text{res})} + \sum_i \mu_i^{(\text{lin})} b_i + \mu_i^{(\text{quad})} b_i^2 + \gamma_i && (42)\\
&= \hat{\gamma} + \sum_i -2b_i^2 \hat{\mu}_i + b_i^2 \hat{\mu}_i + b_i^2 \hat{\mu}_i && (43)\\
&= \hat{\gamma} + \sum_i \hat{\mu}_i(-2b_i^2 + b_i^2 + b_i^2) && (44)\\
&= \hat{\gamma} && (45)
\end{aligned}
$$

so for each $(\hat{\mu}, \hat{\lambda}, \hat{\gamma})$ the proposed (μ, λ, γ) has equal objective value. $\square$

Corollary 6.2.1. *If Problem (MBQP) is feasible with bounded feasible region, then strong duality holds between Problems (CPP) and (COP).*

Proof. Theorem shows that max (COP) $\geq$ max (Hom-COP). So we have max (Hom-COP) $\leq$ max (COP) $\leq$ min (CPP) $=$ min (Hom-CPP). Combining this with max (Hom-COP) $=$ min (Hom-CPP) we get

$$\max(\text{Hom-COP}) = \max(\text{COP}) = \min(\text{CPP}) = \min(\text{Hom-CPP}). \tag{46}$$

Thus, strong duality must hold between (COP) and (CPP). $\square$