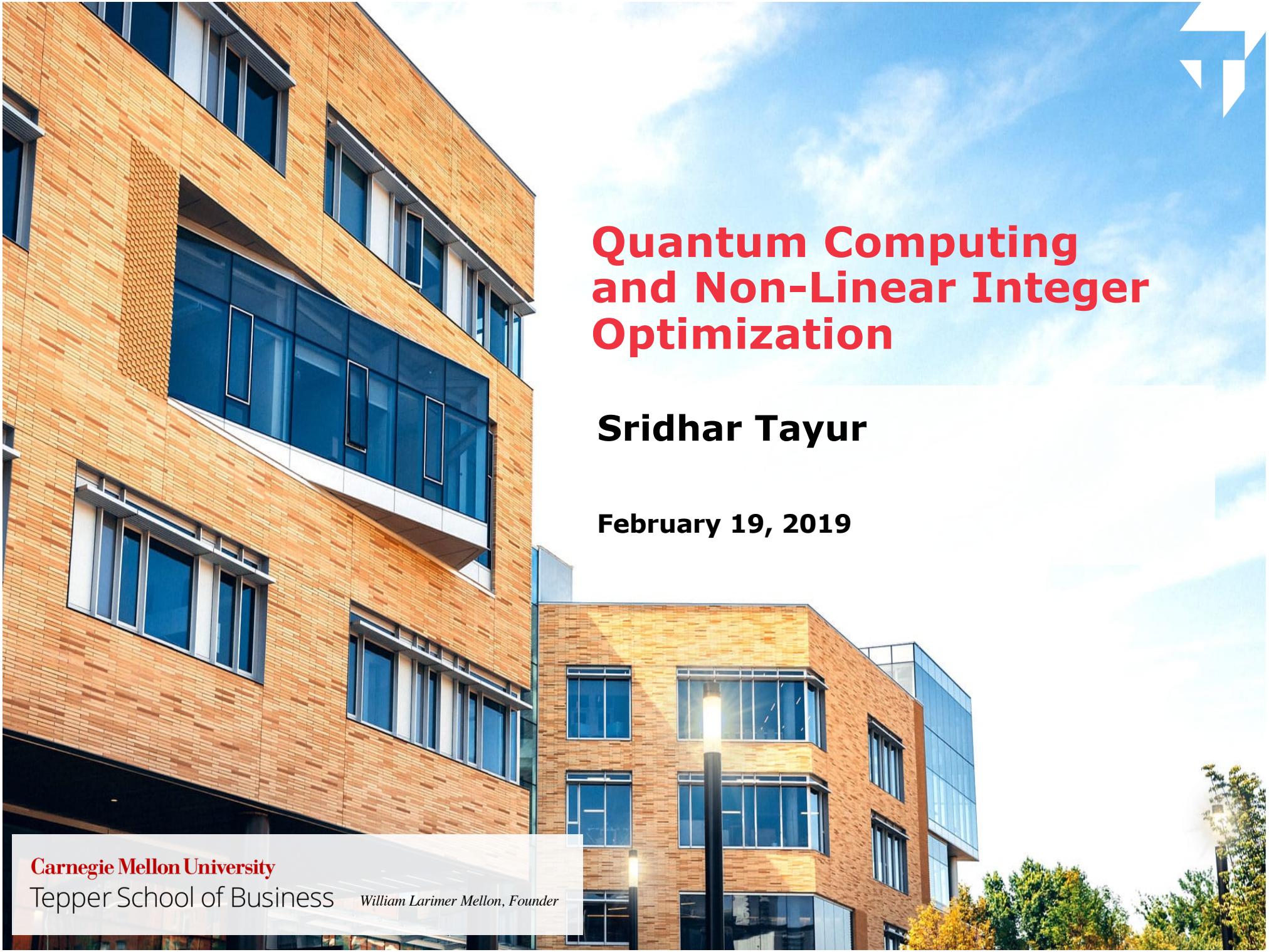


Quantum Computing and Non-Linear Integer Optimization

Sridhar Tayur

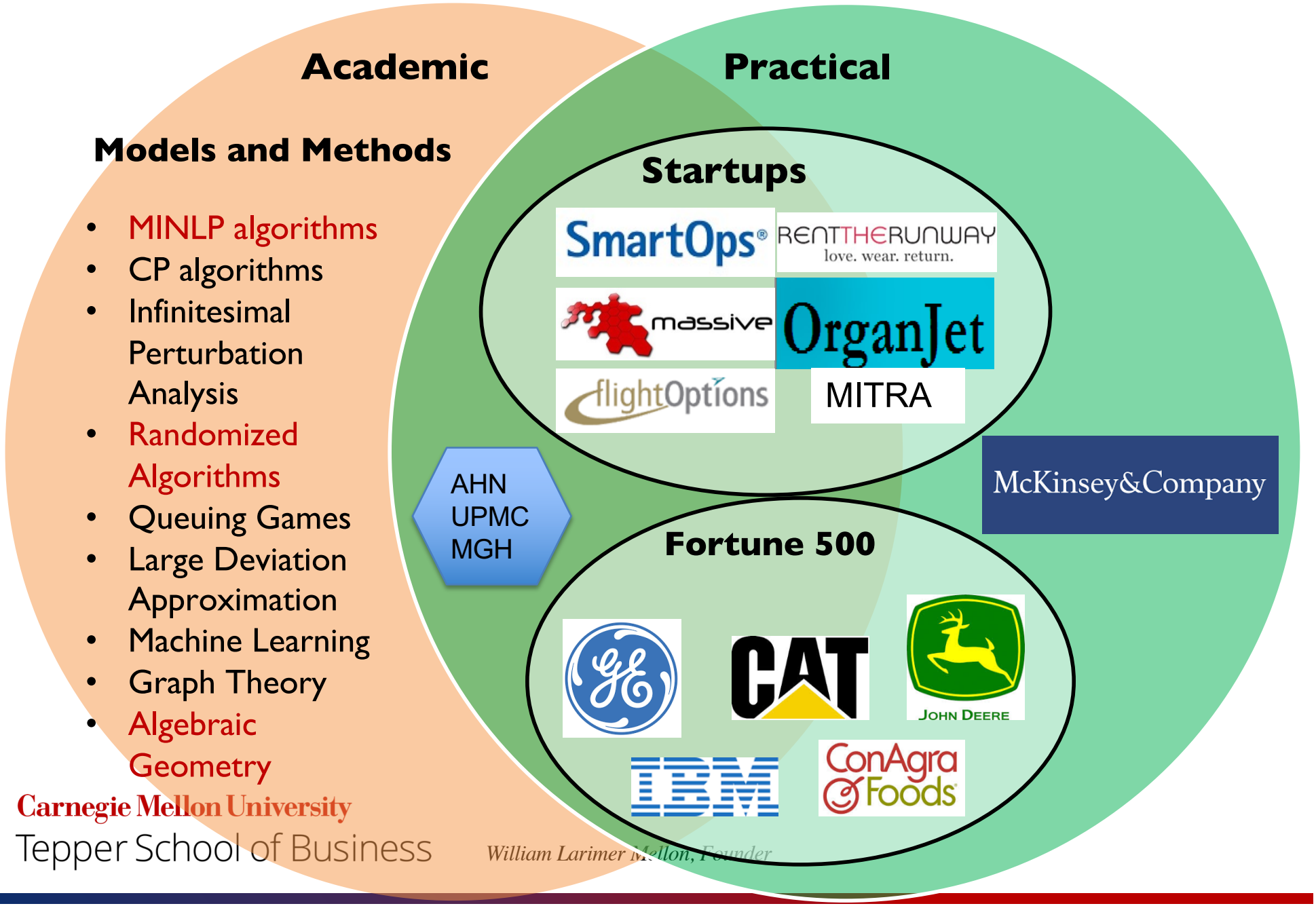
February 19, 2019

Carnegie Mellon University
Tepper School of Business
William Larimer Mellon, Founder

Agenda

- Orientation: Why, What, How
- Vocabulary and Getting Acquainted
- Background: Adiabatic Quantum Computing
- Background: Graver Basis
- Graver Basis via Quantum Annealing
- Non-linear Integer Optimization on D-Wave
- How to surpass Classical Best-in Class?
- References

Solve Real Problems Using Cool Math!



Academic Capitalist

For developing and commercializing innovative methods to optimize supply chain systems

NAE Election 2017 Citation

What Next?

What can quantum computing do for us in service of non-linear integer optimization?

Vocabulary

- Qubits, Spin, Superposition, Entanglement
- Adiabatic Quantum Computing, Gate/Circuit model
- Ising model, Quantum annealing
- Spectral gap (“anti-crossing’)
- Quadratic Unconstrained Binary Optimization (QUBO)
- Embedding and Compiling
- Test sets, Graver Basis, Augmentation
- Computational Algebraic Geometry, Toric Varieties
- Chimera, Pegasus

Getting acquainted

- Regan, Lipton. Quantum Algorithms via Linear Algebra: A Primer.
- Rieffel, Pollack. Quantum Computing: A Gentle Introduction.
- Nielsen, Chuang. Quantum Computation and Quantum Information.
- D-Wave. Introduction to the D-Wave Quantum Hardware.
- Papers by Farhi, Lloyd, Aaronson, Aharonov, Preskill…and their co-authors

Key concepts from my earlier research

- Kannan, Mount, Tayur (1995). A Randomized Algorithm for Optimizing over Certain Convex Sets, *Mathematics of Operations Research*, 20(3), pp. 529-549.

  Spectral Gap drives worst-case complexity of algorithm

- Tayur, Thomas, Natraj (1995). An Algebraic Geometry Algorithm for Scheduling in Presence of Setups and Correlated Demands, *Mathematical Programming*, 69(3), pp. 369-402.

- Bertsimas, Perakis, Tayur (2000). A New Algebraic Geometry Algorithm for Integer Programming, *Management Science*, 46(7), pp. 999-1008.

  Test sets, deep connection between integer programming and computational algebraic geometry

Hybrid Quantum Classical Algorithms

- Step 0 Think Differently

  Reframe OR models (INLP) to Physics models (Ising)

- Step 1 (Classical)

  Embed QUBO into Quantum Hardware

- Step 2 (Quantum)

  Degenerate solution(s) found

- Step 3 (Classical)

  Recover from chain breaks and post-process

Background Material

Adiabatic Quantum Computing (AQC)
Ising Model
D-Wave 2000Q

Test Sets in Optimization
Graver Basis

Carnegie Mellon University
Tepper School of Business

William Larimer Mellon, Founder

Adiabatic quantum computation (AQC) solves optimization problems of the form

$$(\mathcal{P}) : argmin_{(y_0,\cdots,y_{m-1})\in\mathbb{B}^m} f(y_0,\cdots,y_{m-1}),$$

where **B = {0, 1}** and ***f*** is a polynomial function.

Each binary variable ***y*** is mapped into a quantum spin (or a qubit).

Each monomial in ***f*** defines a many body interaction (or coupling) between the involved spins.

When ***f*** is quadratic, it is a quadratic unconstrained binary optimization (QUBO).

So: the solution of the problem *(P)* sits on the ground state (the eigenvector of lowest energy) of the quantum system.

AQC finds the ground state through adiabatic quantum evolution that slowly evolves the ground state of the initial known system into the sought ground state of the problem:

$$H(s) = (1 - s)H_{initial} + sH_{problem}.$$

The **slow time** $s = t/T$ goes from 0 to 1 (where T is the effective total time of the adiabatic evolution).

In practice, s could be replaced by a good $g(s)$, for improved performance as it affects the minimum spectral gap of the resulting $H(g(s))$.

Only the restricted class of Ising spin glass Hamiltonians is currently physically realized.

The quantum system is constituted of a set of spins that are arranged in a graph X where only local 2-body interactions, along the edges of X, are allowed.

Note that, with the restriction to Ising architecture, AQC is no longer universal.

$$H_{(\mathcal{P})} = \sum_{i \in \mathbf{Vertices}(X)} h_i \sigma_i^3 + \sum_{ij \in \mathbf{Edges}(X)} J_{ij} \sigma_i^3 \sigma_j^3,$$

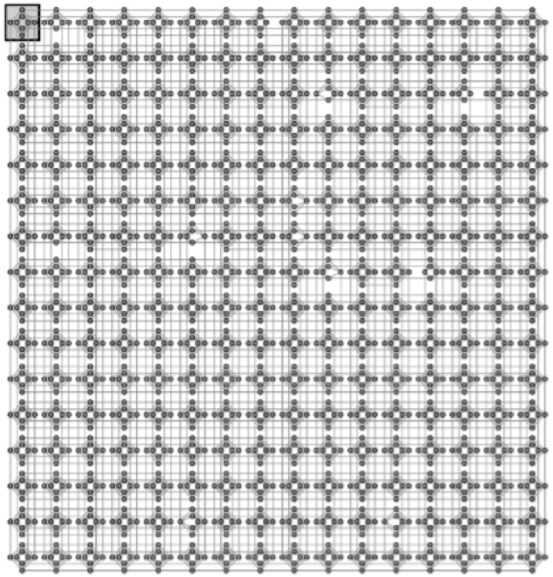
with

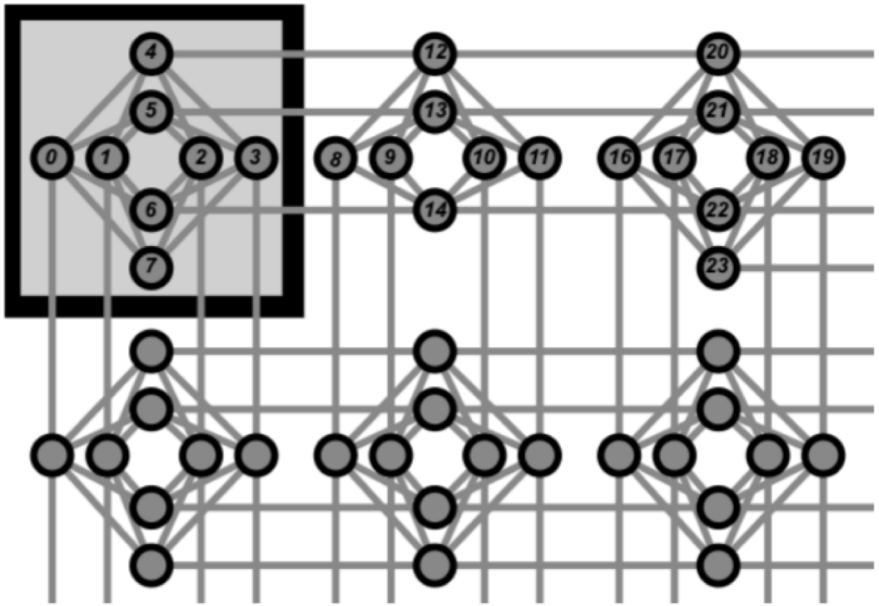
$$\sigma_i^3 = I \otimes \cdots \otimes \sigma^3 \otimes \cdots \otimes I \quad \text{where } I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \sigma^3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

With different coupling strengths *J*, the aggregated interaction is, in general, very complicated.

Computing the ground state of *H(P)* is *hard* but computing the ground state of the initial Hamiltonian is trivial.

D-Wave 2000Q





Cross representation of Chimera 4 x 16 x 16, the graph of D-Wave Systems 2000Q processor. Qubits are arranged in 16 x 16 blocks (or cells). Each block is a 4 x 4 bipartite graph. Qubits have restricted connectivity along the edges where each qubits can interact with at most six neighbors. The missing vertices or edges are faulty qubits or couplers.

$$(\mathcal{P}_\star): \; argmin_{(y_0,\cdots,y_{m-1})\in\mathbb{B}^m} \quad y_0 \sum_{i=1}^{8} c_i y_i .$$

Vertex ***y0*** has degree 8; Chimera's max degree is 6.

We **blow up** the *singular* vertex ***y0*** into a *chain.*

Need many such chains, making embedding a hard problem.

Chains have a tendency to break in practice.

Two choices are (a) what annealing time to use (b) how best to post-process broken chains (*majority vote* or *minimize energy*)?

Problem Embedded into Chimera

Figure 2: The logical graph of the objective function in $(\mathcal{P}_\star)$

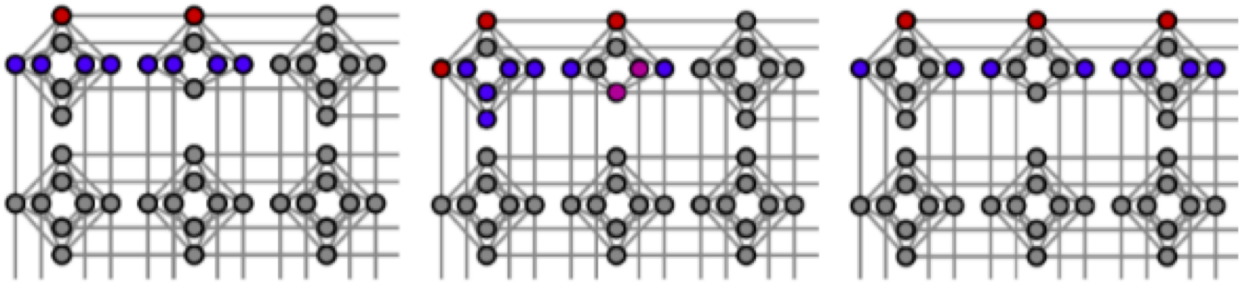


Figure 3: Three different examples of embeddings of the logical graph 2 inside Chimera. In all cases, the red chains of qubits represent the logical qubit y_0. The remaining qubits are represented with 1-chain (i.e., a physical qubit) in blue, except for the middle picture, where y_7 is represented by the purple 2-chain.

Chain Breaking

Figure 2: The logical graph of the objective function in $(\mathcal{P}_\star)$

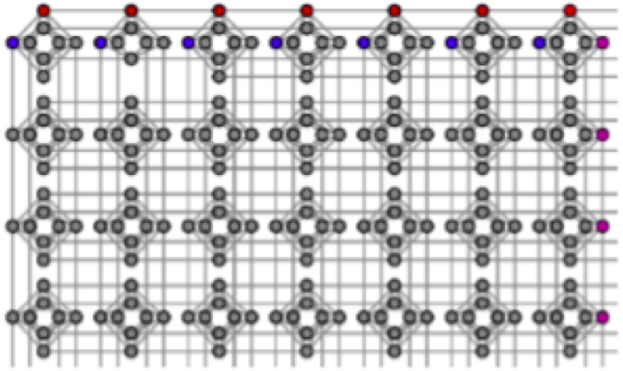


Figure 6: The depicted minor embedding (for the problem $(\mathcal{P})_*$) has two long chains that don't persist through the adiabatic evolution (in D-Wave). In this case, the quantum processor fails to return an answer.

Test Sets in Optimization

- Nonlinear integer program:

$$(IP)_{A,b,l,u,f}: \quad \min\left\{ f(x) \;:\; Ax = b, \quad x \in \mathbb{Z}^n \quad , \quad l \le x \le u \right\}$$

$$A \in \mathbb{Z}^{m \times n} \quad , \quad b \in \mathbb{Z}^m \quad , \quad l,u \in \mathbb{Z}^n, \quad f : \mathbb{R}^n \to \mathbb{R}$$

- Can be solved via *augmentation procedure*:

1. Start from a feasible solution
2. Search for augmentation direction to improve
3. If none exists, we are at an optimal solution.

Graver Basis of Matrix A

- Graver Basis is the finite set of conformal minimal elements of:

$$\mathcal{L}^*(A) = \left\{ x \;\middle|\; Ax = \mathbf{0}, \quad x \in \mathbb{Z}^n \quad , \quad A \in \mathbb{Z}^{m \times n} \right\} \setminus \{\mathbf{0}\}$$

- Partial order:

$$\forall x, y \in \mathbb{R}^n \quad x \sqsubseteq y \quad s.t. \quad x_i y_i \geq 0 \quad \& \quad |x_i| \leq |y_i| \quad \forall \quad i = 1, \ldots, n$$

- x is conformal minimal to y: $x \sqsubseteq y$

Graver Basis is Test Set for:

- $\min cx$, Linear
- $\max f(Wx), W \in \mathbb{Z}^{d\times n}$, f *convex on* $\mathbb{Z}^d$
- $\min \sum f_i(x_i)$, f_i convex (separable convex)
- $min \left|x - x_0\right|_p$
- Some other nonlinear costs

Graver Basis via Quantum Annealing

QUBO for Kernel
Sampling the Kernel
Post-processing Near-Optimal Solutions
Computational Results

Carnegie Mellon University
Tepper School of Business

William Larimer Mellon, Founder

Hybrid Quantum- Classical Graver

1. Finding the lattice kernel $\mathcal{L}^*(A)$ using many reads of quantum annealer : need a QUBO
2. Filtering conformal $\sqsubseteq$ minimal elements by comparisons, using classical methods
3. Repeating (1) and (2) while *adjusting* the "QUBO" variables in each run *adaptively*

QUBO for Kernel

$$\mathbf{Ax} = \mathbf{0}, \quad \mathbf{x} \in \mathbb{Z}^n \quad , \quad \mathbf{A} \in \mathbb{Z}^{m \times n}$$

$$\min \quad \mathbf{x}^T \mathbf{Q}_\mathbf{I} \mathbf{x} \quad , \quad \mathbf{Q}_\mathbf{I} = \mathbf{A}^T \mathbf{A} \quad , \quad \mathbf{x} \in \mathbb{Z}^n$$

$$\mathbf{x}^T = \begin{bmatrix} x_1 & x_2 & \cdots & x_i & \cdots & x_n \end{bmatrix}, \quad x_i \in \mathbb{Z}$$

- Integer to binary transformation: $x_i = \mathbf{e}_i^T X_i$

$$X_i^T = \begin{bmatrix} \mathrm{X}_{i,1} & \mathrm{X}_{i,2} & \cdots & \mathrm{X}_{i,k_i} \end{bmatrix} \in \{0,1\}^{k_i}$$

- Binary encoding: $\mathbf{e}_i^T = \begin{bmatrix} 2^0 & 2^1 & \cdots & 2^{k_i} \end{bmatrix}$
- Unary encoding: $\mathbf{e}_i^T = \underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}}_{k_i}$

QUBO for Kernel….

$$\mathbf{x} = \mathbf{L} + \mathbf{E}\mathbf{X} = \begin{bmatrix} Lx_1 \\ Lx_2 \\ \vdots \\ Lx_n \end{bmatrix} + \begin{bmatrix} \mathbf{e}_1^T & \mathbf{0}^T & \cdots & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{e}_2^T & \cdots & \mathbf{0}^T \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}^T & \mathbf{0}^T & \cdots & \mathbf{e}_n^T \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}$$

(L is the lower bound vector)

- QUBO:

$$\min \quad \mathbf{X}^T \quad \mathbf{Q}_\mathbf{B}\mathbf{X} \quad , \quad \mathbf{Q}_\mathbf{B} = \mathbf{E}^T\mathbf{Q}_\mathbf{I}\mathbf{E} + diag\left(2\mathbf{L}^T\mathbf{Q}_\mathbf{I}\mathbf{E}\right)$$

$$\mathbf{X} \in \{0,1\}^{nk} \quad , \quad \mathbf{Q}_\mathbf{I} = \mathbf{A}^T\mathbf{A}$$

Sampling for Kernel

- Each anneal starts with an independent uniform superposition (10000 per D-Wave call):

$$|\hat{0}\rangle = \frac{1}{2^n} \sum_{i \in \mathbb{Z}_2^n} |i\rangle$$

- Symmetry in QUBO (for arbitrary A) implies similar spread in valleys

- Techniques:
  - Random column permutation
  - Adaptive resource allocation chases the non-extracted solutions

Post Processing

Experimental observation:

- Majority (~ 90%) of sub-optimal solutions have *small* overall sum-errors: most near-optimal!
- Post-processing: Systematic pairwise error vector addition and subtraction to yield zero columns of these near-optimal solutions
- Overall numerical complexity low (and polynomial) by limiting range of errors post-processed

Post Processing: Analysis

- Optimal and suboptimal percentages for various sizes

	(2x5)	(3x5)	(3x6)	(3x7)	(4x8)	(5x8)	Overall
0	15	12	10	13	12	9	**11.8**
1	24	28	26	31	30	33	**28.7**
2	31	28	27	24	21	25	**26.0**
3	20	21	25	19	20	17	**20.3**
4	7	8	10	9	9	8	**8.5**
5	2	1	0	3	4	5	**2.5**
≥ 6	1	2	2	1	4	3	**2.2**

Computational Experience

- For all matrices that have a small span of integral values (± 4) the Graver basis acquired with less than 7 calls.
- All embeddable (m x 16) *binary matrices* Graver basis acquired in less than 5 calls.
- The quantum approach performs very well for narrow truncated band, i.e. {-1, 0,1} , attractive as test set for nonlinear combinatorial or low span integer programming.

Non-Linear Integer Optimization on D-Wave

QUBO for Feasible Solution(s)
Hybrid Quantum-Classical Algorithm
Computational Results

Carnegie Mellon University
Tepper School of Business

William Larimer Mellon, Founder

QUBO for Feasible Solutions

$$\mathbf{Ax} = \mathbf{b} \qquad l \le \mathbf{x} \le u$$

$$\min \quad \mathbf{X}^T \mathbf{Q_B} \mathbf{X}, \quad \mathbf{Q_B} = \mathbf{E}^T \mathbf{Q_I} \mathbf{E} + 2diag\left[\left(\mathbf{L}^T \mathbf{Q_I} - \mathbf{b}^T \mathbf{A}\right)\mathbf{E}\right]$$

$$\mathbf{X} \in \{0,1\}^{nk}, \quad \mathbf{Q_I} = \mathbf{A}^T \mathbf{A}$$

- Using adaptive centering and encoding width for feasibility bound
- Results in many feasible solutions!

Hybrid Quantum-Classical Optimization

1. Calculate Graver Basis
2. Find Initial Feasible Solution(s) (Quantum)
3. Augmentation: Improve feasible solutions (Classical)

Capital Budgeting

- Important canonical Finance problem
- μ_i expected return
- σ_i variance
- ε risk

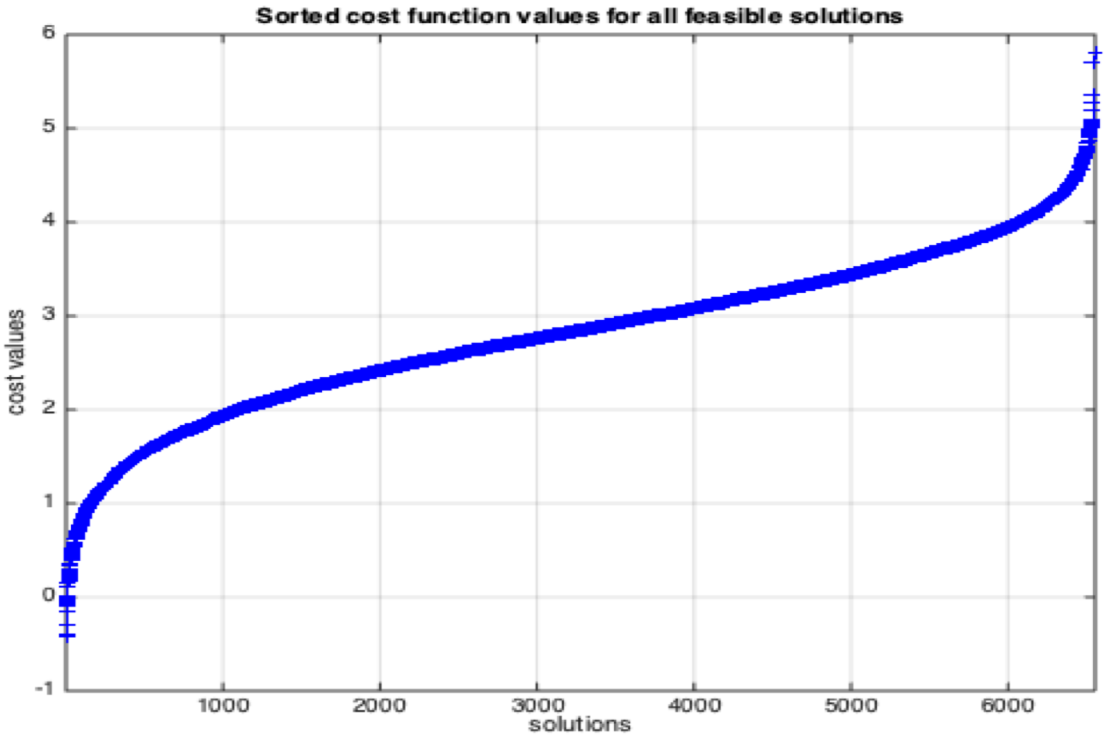
$$\begin{cases} \min \quad -\sum_{i=1}^{n} \mu_i x_i + \sqrt{\frac{1-\varepsilon}{\varepsilon} \sum_{i=1}^{n} \sigma_i^2 x_i^2} \\ Ax = b \quad , \quad x \in \{0,1\}^n \end{cases}$$

- Graver Basis in 1 D-Wave call (1 bit encoding)

$$A \in M_{5\times 50}(\{0,\cdots,t\}) \quad \mu \in [0,1]^{50\times 1} \qquad \sigma \in [0,\mu_i]^{50\times 1}$$

when t = 1 we have: $\mathcal{G}(A) \in M_{50\times 304}(\{-1,0,+1\})$

~ 6500 Solutions in One Call!



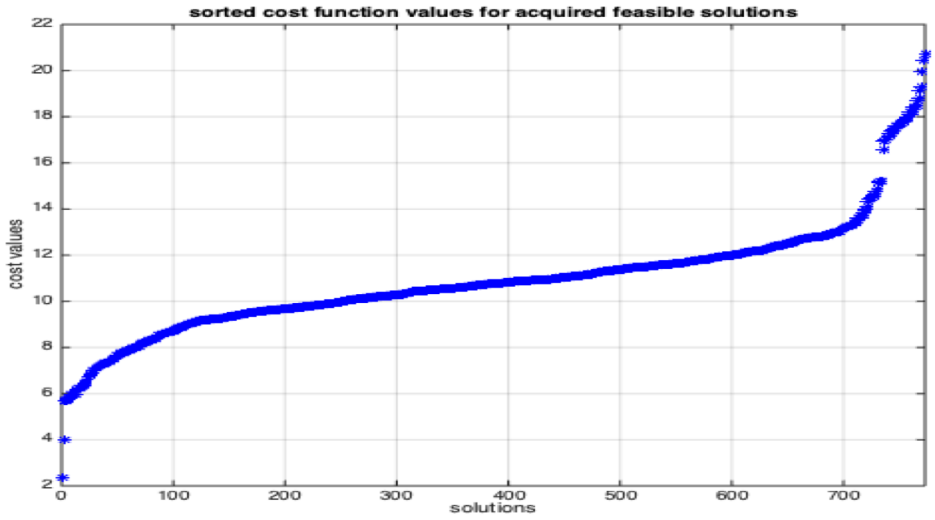
- From any feasible point in ~24-30 augmenting steps reach optimal cost = -3.69

Non-binary Integer Variables

- Low span integer $x \in \{-2,-1,0,1,2\}^n$

$$A \in M_{5\times 50}\left(\{0,1\}\right) \quad \mu \in [0,1]^{25\times 1} \quad \sigma \in [0,\mu_i]^{25\times 1}$$

- 2 Bit Encoding
- $\mathcal{G}(A) \in M_{25\times 616}\left(\{-4,\ldots,+4\}\right)$ in 2 D-Wave calls
- 773 feasible solutions in one D-Wave call

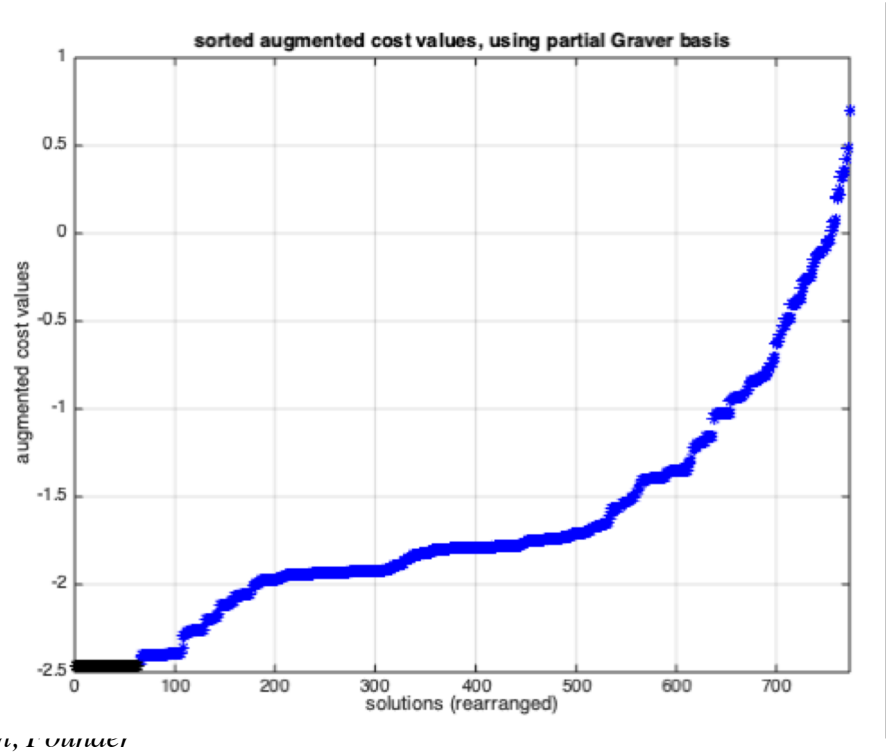


Augmenting…

- From any feasible points in ~20-34 augmenting steps, reach global optimal cost = -2.46
- **Partial Graver Basis:** One D-Wave call only

$$\mathcal{G}^P(A) \in M_{25\times 418}\left(\{-4,\ldots,+4\}\right)$$

- *64 out of 773 feasible starting points end up at global solutions.*



How to Surpass Best-in-Class Classical Methods?

Carnegie Mellon University
Tepper School of Business

William Larimer Mellon, Founder

Gurobi Optimizer 8.0

- Random $A \in M_{5\times 50}(\{0,\ldots,t\})$
- *"terms"* designates cardinality of set of **J** values

$$t=1 \Rightarrow \begin{cases} 0.2 & \text{sec} \\ 6 & \textit{terms} \end{cases} \qquad t=10 \Rightarrow \begin{cases} 16 & \text{sec} \\ 230 & \textit{terms} \end{cases} \qquad t=20 \Rightarrow \begin{cases} 3 & \text{min} \\ 620 & \textit{terms} \end{cases}$$

$$t=40 \Rightarrow \begin{cases} 21 & \text{min} \\ 1030 & \textit{terms} \end{cases} \qquad t=50 \Rightarrow \begin{cases} 75 & \text{min} \\ 1070 & \textit{terms} \end{cases} \qquad t=100 \Rightarrow \begin{cases} >8 & \textit{hours} \\ 1190 & \textit{terms} \end{cases}$$

- D-Wave: Chimera but improved coupler precision to handle more unique **J** elements for 0-1 matrices.

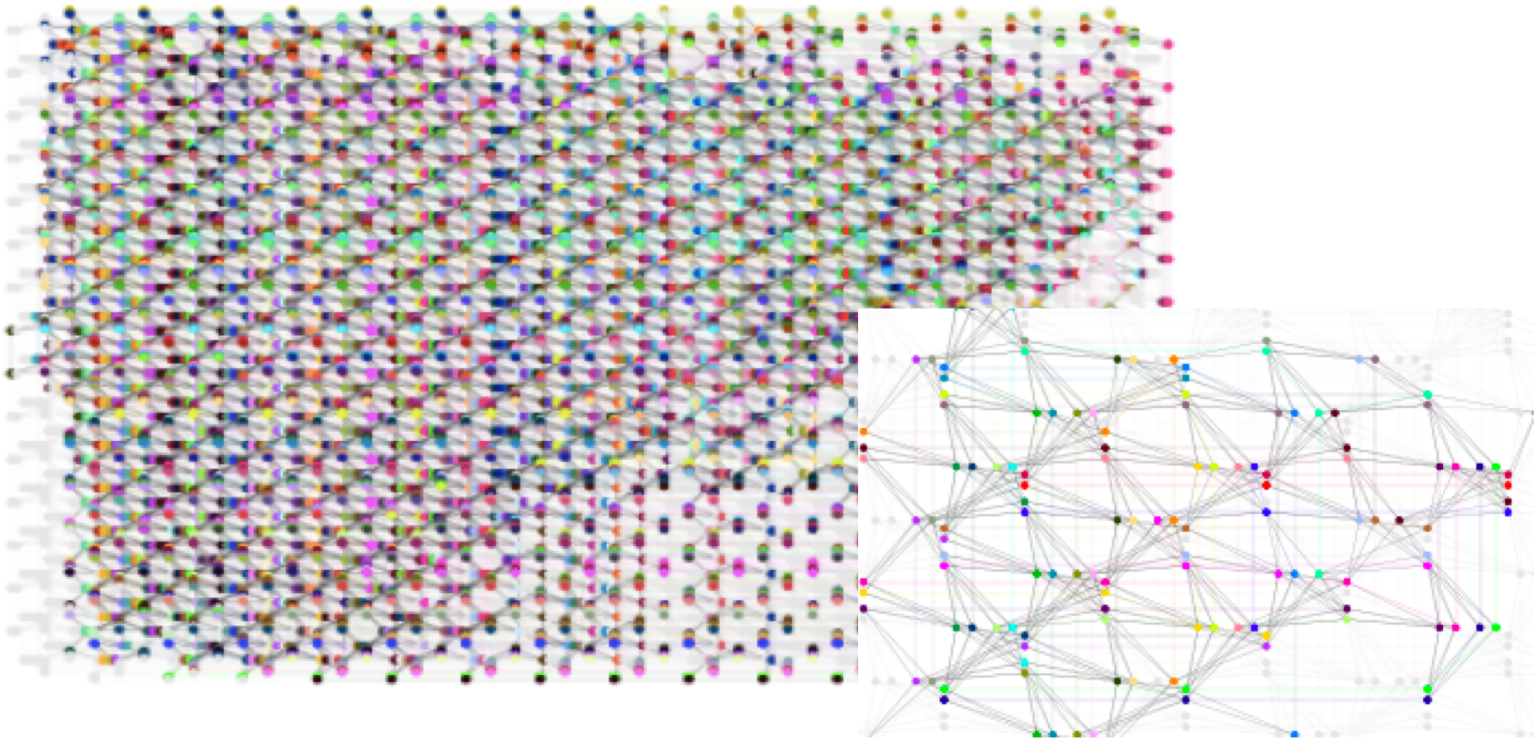
$$\begin{cases} t=1 \\ A^{20\times 80} \end{cases} \Rightarrow \begin{cases} 135 & \text{sec} \\ 13 & \textit{terms} \end{cases} \qquad \begin{cases} t=1 \\ A^{25\times 100} \end{cases} \Rightarrow \begin{cases} \sim 2 & \textit{hours} \\ 15 & \textit{terms} \end{cases} \qquad \begin{cases} t=1 \\ A^{30\times 120} \end{cases} \Rightarrow \begin{cases} >3 & \textit{hours} \\ 16 & \textit{terms} \end{cases}$$

How and Where to Surpass?

- If coupler precision doubles, with the same number of qubits and connectivity, we can be competitive on 0-1 problems and {0,...,t} matrices of size 50.
- Pegasus can embed a size 180 problem with shorter chains, should surpass Gurobi on {0,1} matrices of sizes 120 to 180, without an increase in precision.
- An order of magnitude increase in maximum number of anneals per call.
- Global optimization with difficult convex (and non-convex) objective functions?

Embedding in Pegasus

- Complete graph of size 170



References

- **Embedding and Compiling** (arXiv:1810.01440)

  Computational Algebraic Geometry, Grobner Basis of Toric Varieties, Fiber bundles…

- **Analyzing Quantum Speedup** (arXiv:1811.00675)

  Spectral gap in Adiabatic Evolution via Morse Homology. Gauss-Bonnet Theorem, Cerf Theory…

- **Non-linear Integer Optimization** (arXiv:1902.04215)

  Geometry of Numbers, Test-Sets, Graver Basis..

Thank you!

Email
stayur@cmu.edu