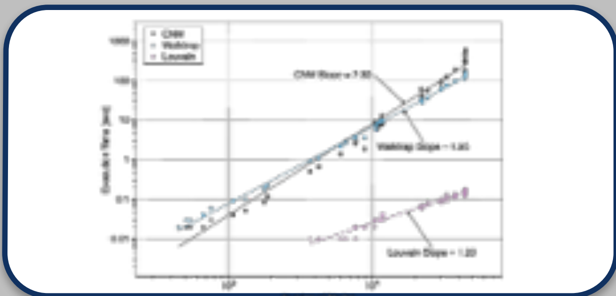
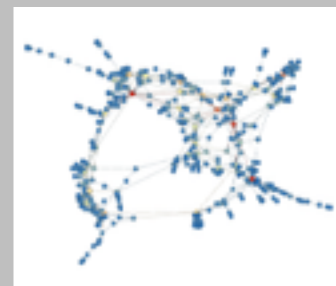
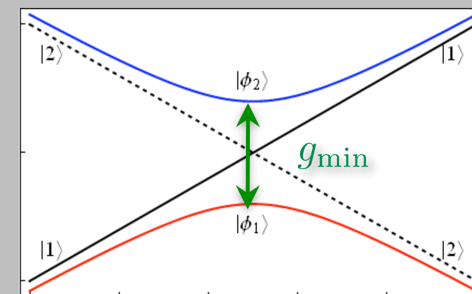
SAND2014-17408C

Exceptional service in the national interest

Sandia National Laboratories







Benchmarking quantum annealing for community detection on synthetic social networks

Ojas Parekh

Sandia National Labs

AQC, 2014

A question?

Which is harder (pick a reasonable definition of "hard")?

Finding an optimal solution 99% of the time

Finding a solution within 99% of optimal all the time

Which is more practically relevant?

A question?

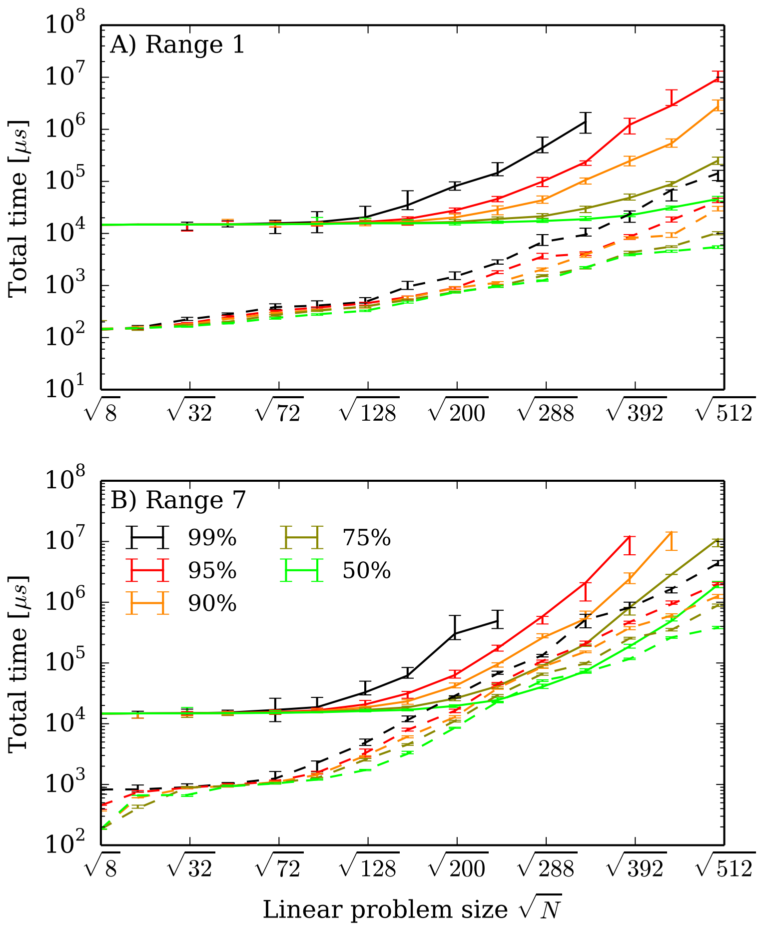
Which is harder (pick a reasonable definition of “hard”)?

Finding an optimal solution 99% of the time	Finding a solution within 99% of optimal all the time
“Average”-case analysis, rather than worst-case	Approximation algorithm, or approximation scheme

How should we measure success?

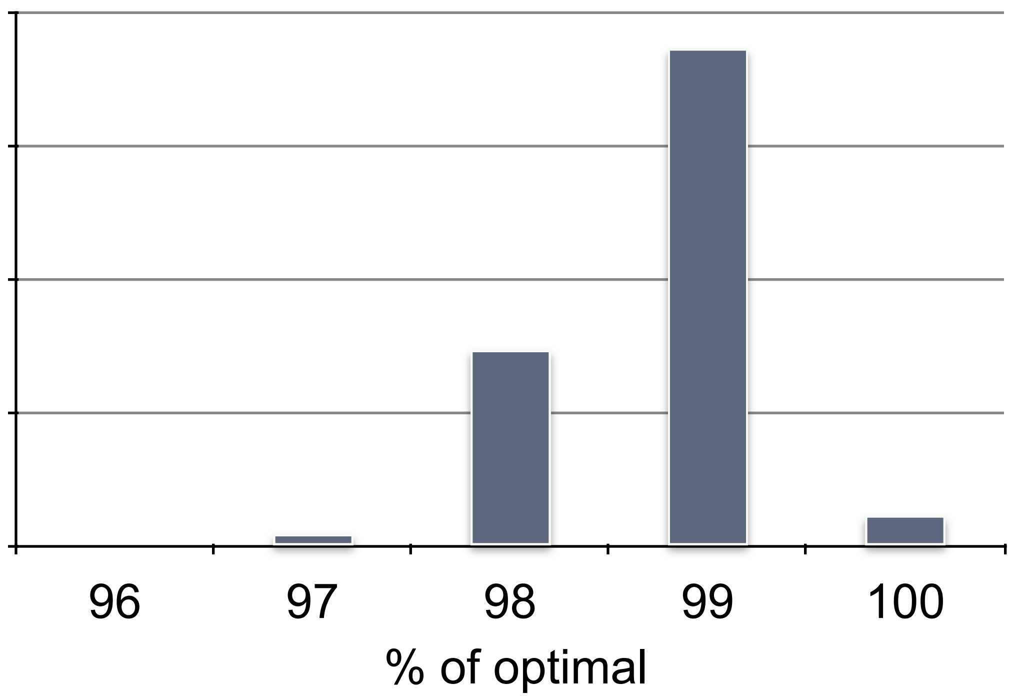
Sandia National Laboratories

Optimal 99% of the time



Within 99% of optimal all of the time

55,000 random {-1,1}-weight instances on 509-qubit D-Wave Two



Was always within 96% of optimal!

Left: Ronnow *et al.*, *Defining and detecting quantum speedup*, arXiv:1401.2910v1 (2014)

We ask

- What is an appropriate measure of success?
- What classical algorithm(s) should be used for comparison?
- How should one select appropriate benchmark instances?

Problem definitions

- Ising:

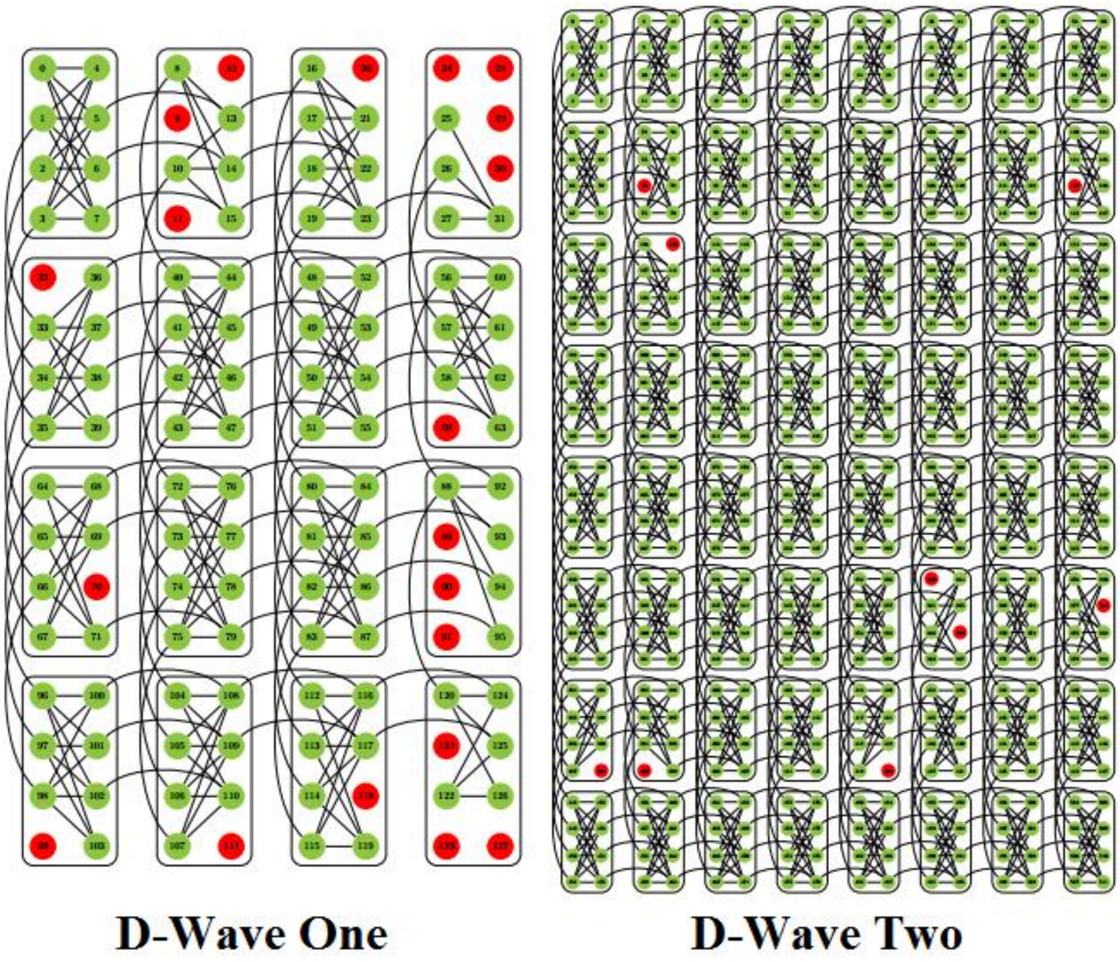
$$\min_{x_i \in \{-1,1\}} \sum_{ij} J_{ij} x_i x_j + \sum_i h_i x_i$$

- Quadratic binary unconstrained optimization (QUBO):

$$\min_{x_i \in \{0,1\}} \sum_{ij} A_{ij} x_i x_j + \sum_i c_i x_i$$

Comparison of Rainier and Vesuvius chips

Rainier
108/128
spins



Vesuvius
506/512
spins

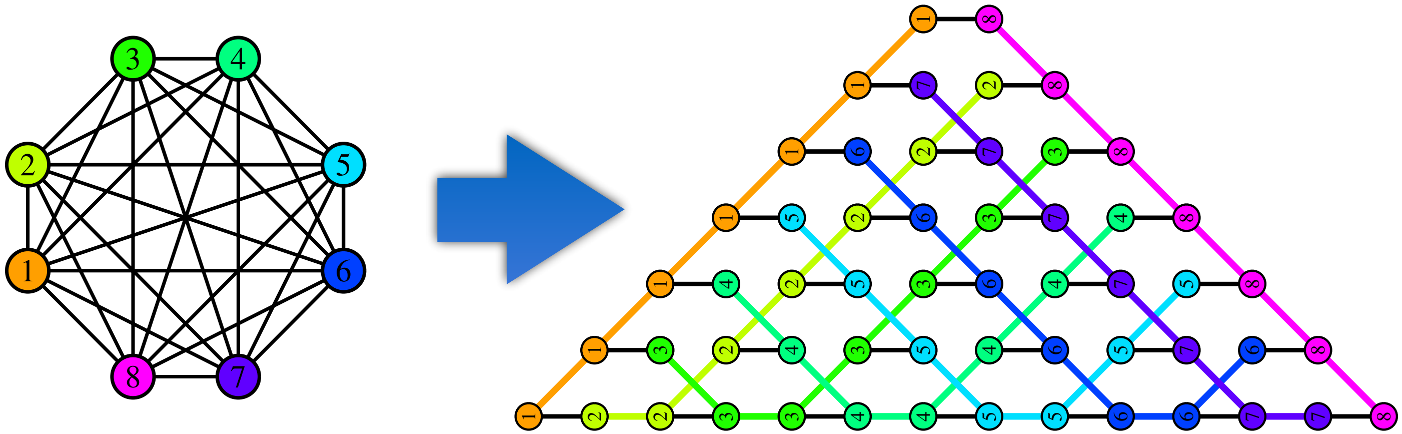
Images from D-Wave Systems: http://www.dwavesys.com .

Complexity of Ising on Chimera

- (Decision version) NP-complete even with no linear term and {-1, 0, 1} weights [Barahona, 1982]
- We show NP-complete with no linear term and {-1,1} weights
  - Instances used in D-Wave benchmarking studies [with Benjamin Moseley at Washington University]
- Tree-width (path-width) is $\Theta(\sqrt{n})$, yielding $O(2^{\sqrt{n}})$ algorithm
  - "Subexponential" exact algorithm even though NP-hard
- Approximation complexity?
  - Polynomial-time approximation scheme (PTAS) [Saket, 2013, arXiv:1306.6943]
  - PTAS's are rarely efficient; theory vs practice?
  - Efficient approx algorithm for say, getting within 75%?

Approaches to problem embedding

- Embedding is hard: $O(n^n)$ vs $O(2^n)$
- Even harder when optimizing # qubits
- Choi: worst case $O(n^2)$ qubits for *n* vars
  - Requires (linearly) large coupler weights



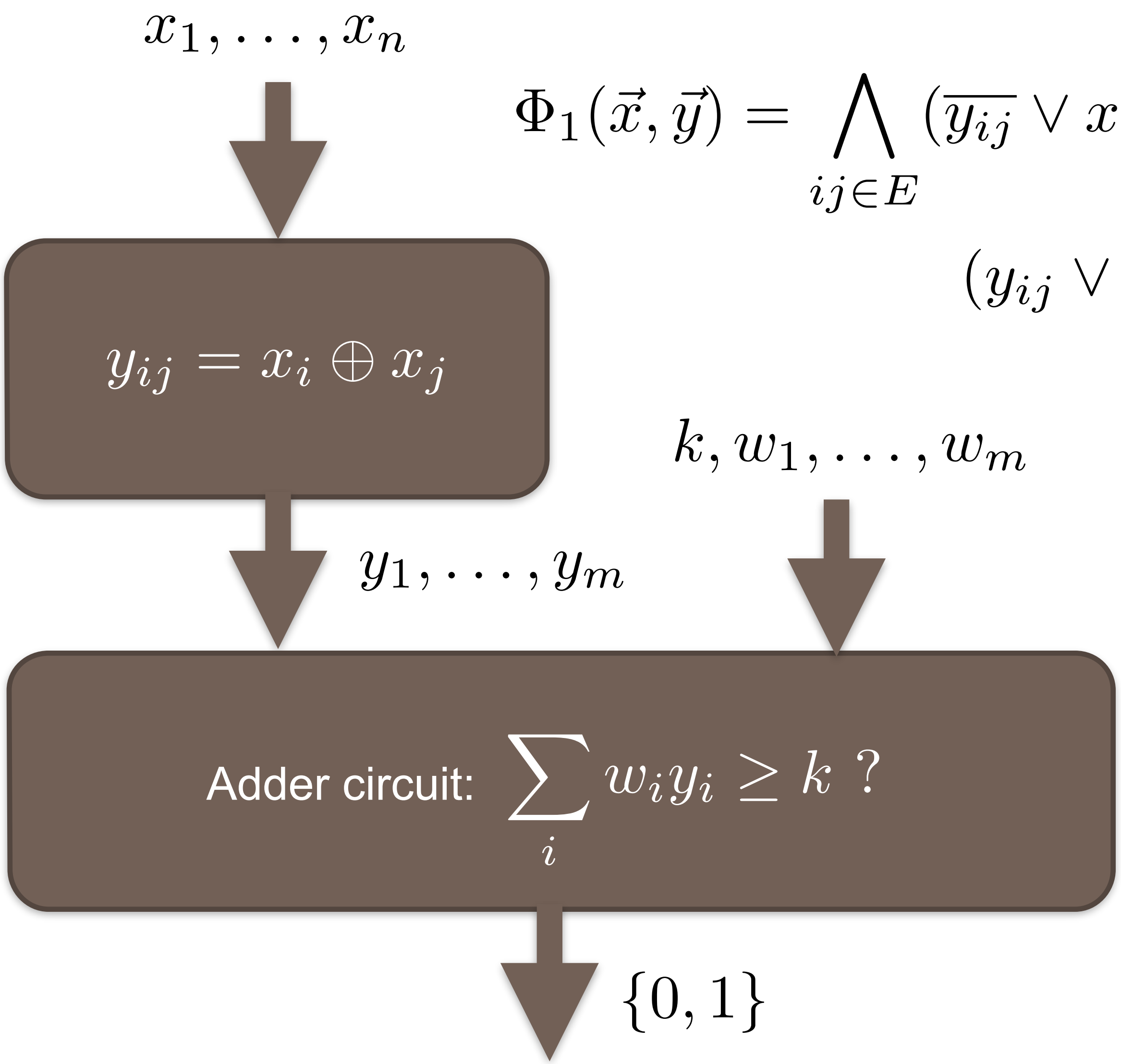
Limits of reducing to Chimera

- Can we do better than a quadratic blowup in qubits?
- Probably not, due to Exponential Time Hypothesis
  - Problems like Max-Cut on general graphs are conjectured to require $O(2^n)$ time
  - But we have a $O(2^{\sqrt{n}})$ time algorithm for Chimera Ising
  - So in some sense quadratic factor is artifact of Chimera
- Weights make this worse: Choi embedding assumes (linearly) large weights
- Reduction better than $O(n^2)$ for Max-Cut on bounded-degree graphs would improve best-known classical algorithm
  - Applies to **any** reduction, not just minor embeddings

Max-Cut as model problem

- NP-complete on bipartite graphs with weights {-1,1}
  - Replace each edge by a path of +1 and -1 edge
- Max-Cut essentially equivalent to Ising problem
  - Can use an apex vertex to model linear term
- We give reduction from weighted QUBO to unweighted QUBO
  - Unroll and optimize existing chain of reductions
  - Weights are a significant barrier in D-Wave benchmarking
- Our reduction only incurs linear blowup on bounded-degree graphs
  - However, does not preserve Chimera structure

Weighted QUBO as 3-SAT



$$\Phi_1(\vec{x}, \vec{y}) = \bigwedge_{ij \in E} (\overline{y_{ij}} \vee x_i \vee x_j) \wedge (\overline{y_{ij}} \vee \overline{x_i} \vee \overline{x_j}) \wedge$$

$$(y_{ij} \vee x_i \vee \overline{x_j}) \wedge (y_{ij} \vee \overline{x_i} \vee x_j)$$

$$\Phi_2(\vec{y}, \vec{w}, k)$$

$$\Phi = \Phi_1 \wedge \Phi_2$$

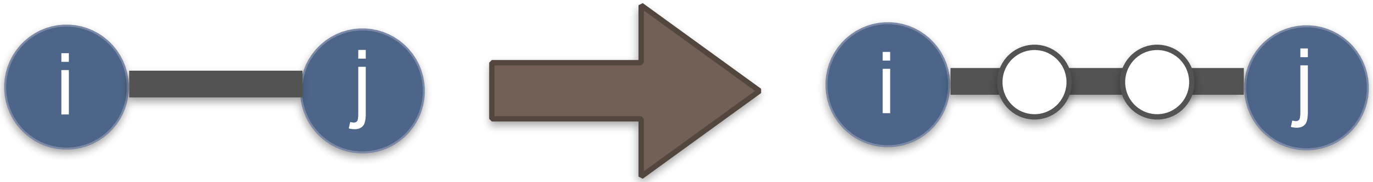
3-SAT as Unweighted Ising

- Standard reduction from 3-SAT to Independent Set
- Usual Ising formulation of Independent Set:

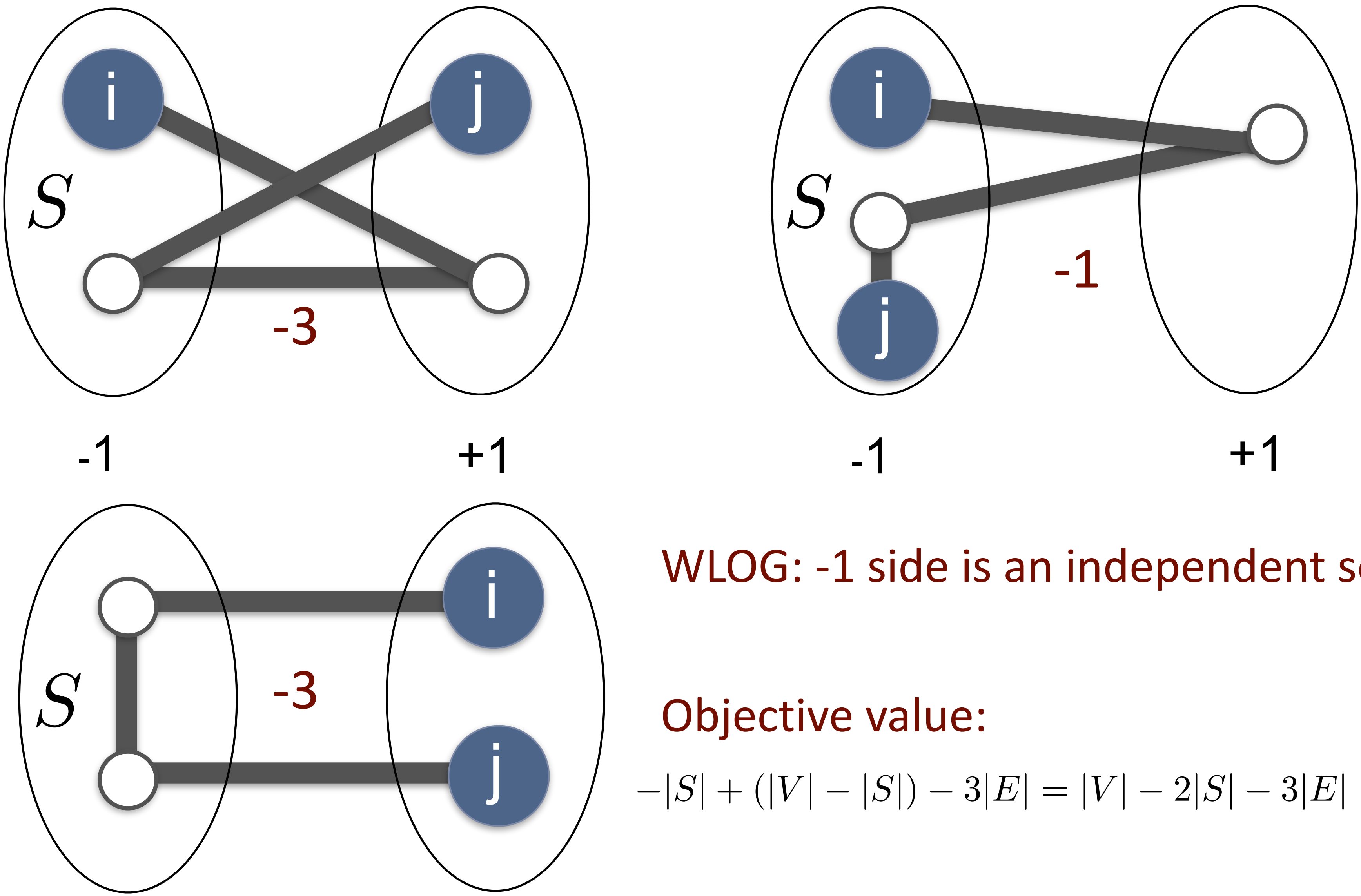
$$\min_{x_i \in \{-1,1\}} \sum_{ij \in E} x_i x_j - \sum_{i \in V} d_i x_i$$

- New formulation on graph where each edge replaced by 3-path circumvents linear weights above:

$$\min_{x_i \in \{-1,1\}} \sum_{ij \in E'} x_i x_j - \sum_{i \in V'} x_i$$



Independent set as Ising



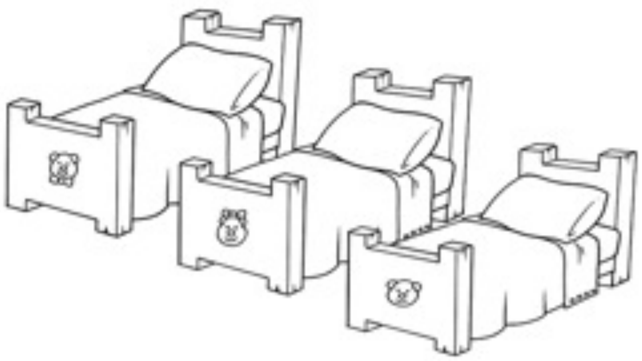
WLOG: -1 side is an independent set

Objective value:

$$-|S| + (|V| - |S|) - 3|E| = |V| - 2|S| - 3|E|$$

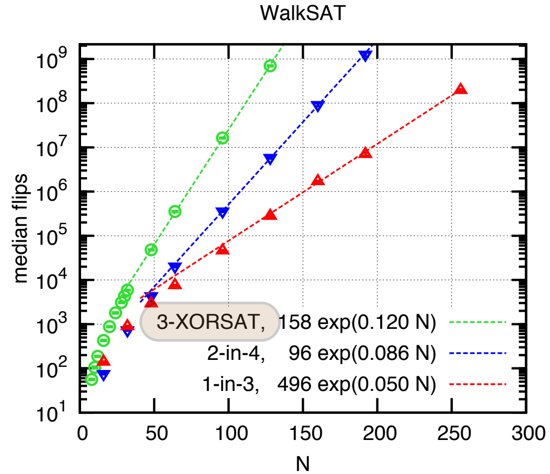
Instance selection: challenges

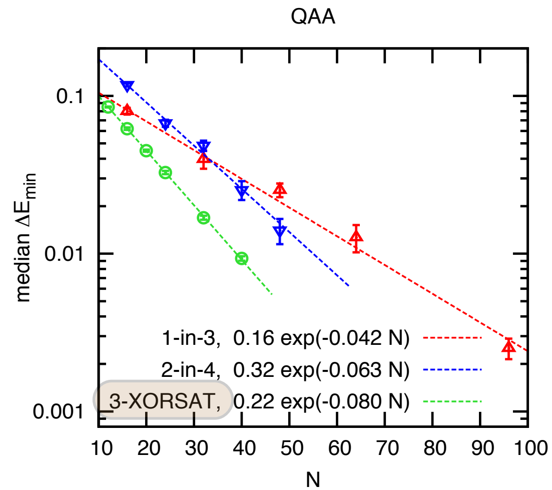
- Desire scalable family of related instances
- Real world often lacks well-defined optimality
- Random instances too easy or hard
- Known hard instances may be artificial
- Fairness to all benchmark algorithms
- Resources: time, memory, etc.



Picking the right algorithm

- Random Ising instances on D-Wave hardware
  - {-1,1} coupler values
  - Are such instances hard?
  - Hen and Young observed problem in P may appear hard (3-XORSAT in figure) [Phys. Rev. E 84, 061152 (2011)]
  - Random vs hard instances is a tricky issue





Configuration matters

- Random instances on D-Wave hardware
  - {-1,1} coupler values
  - D-Wave Two finds optimal in 0.5 sec, while classical algorithms scale poorly [McGeoch and Wang, Conf. Computing Frontiers 2013: 23]
    - Claimed 3600x speedup
  - We observe classical Integer Program solvers match performance with appropriate model [also Dash, arXiv:1306.1202v2 (2013)]
    - Speedup vanishes with proper configuration/usage
  - QUBO vs Ising
    - Random instance with linear term appear easy for former but hard for latter!

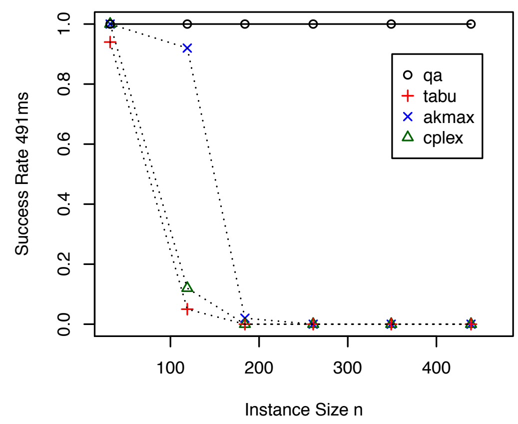


Figure 1: Success rates: proportion of best solutions found in 491ms CPU time (tabu, amax, cplex software) and exclusive access time (QA hardware).

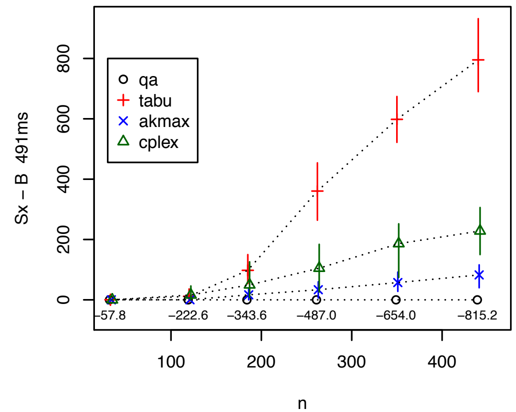
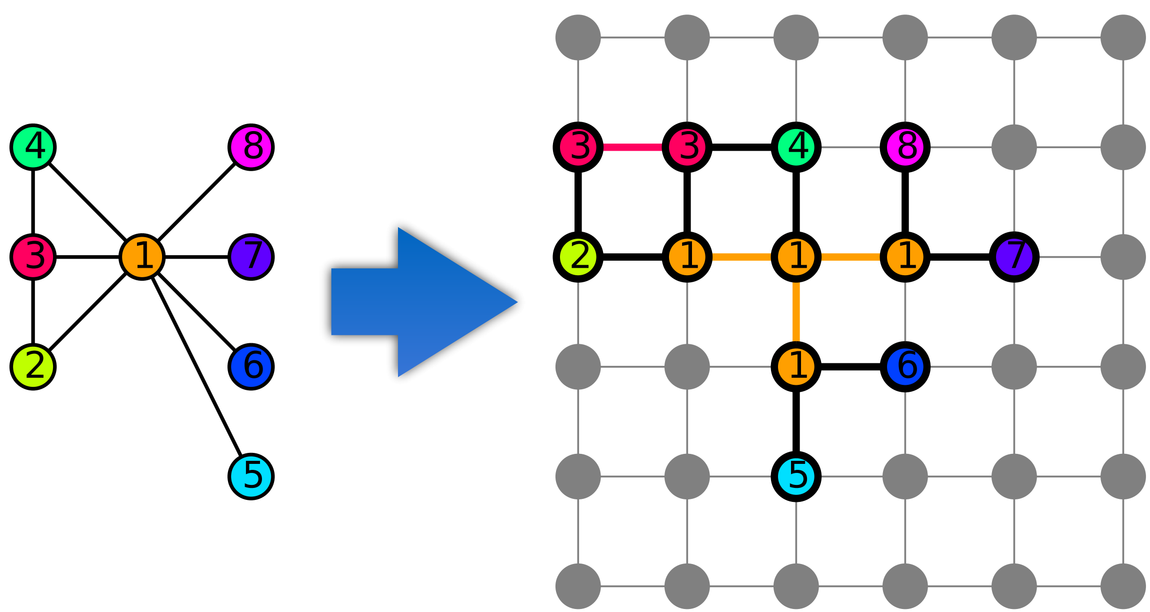


Figure 2: Differences between solution cost S_x in 491ms runtime, and best solution B for each input. Dotted lines connect means for each solver; vertical bars show the range of observations. The numbers at bottom are means $\overline{B}$ for each problem size.

Solving problems with D-Wave: challenges

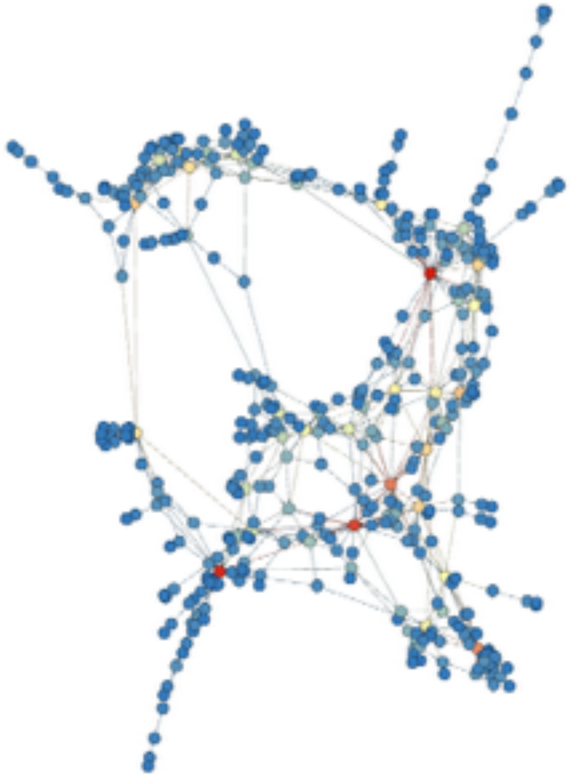
- Application graph must be embedded within Chimera graph
  - Requires extra qubits; worst case approx n^2 qubits for n nodes
  - Very hard to determine a good embedding for a given graph
  - Typical approaches to embedding require large weights to force all qubits corresponding to a node have same spin
  - Efficiency: #(Application graph nodes)/#(Qubits in Chimera graph)

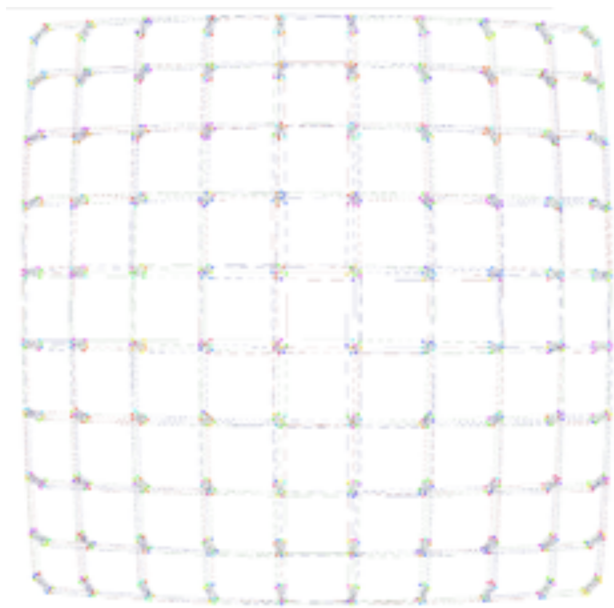
Application Graph

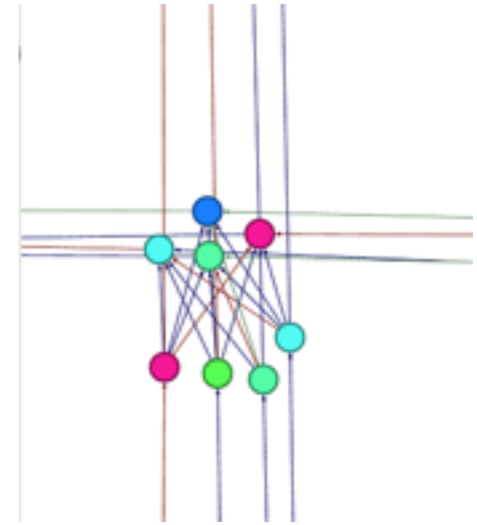


Qubit interaction Graph

Complex networks on the Chimera graph



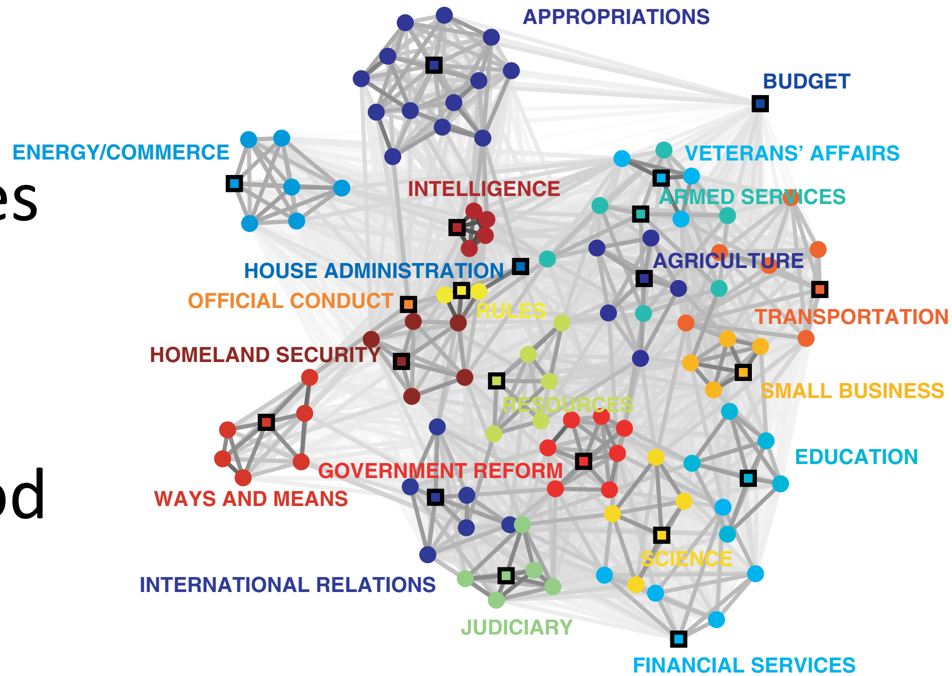




- New approach to circumvent embedding [with Jeremy Wendt]
- Generate complex network simultaneously while embedding it
- Efficiency for 512-node Chimera around 40% vs 6.25% worst case

Community detection

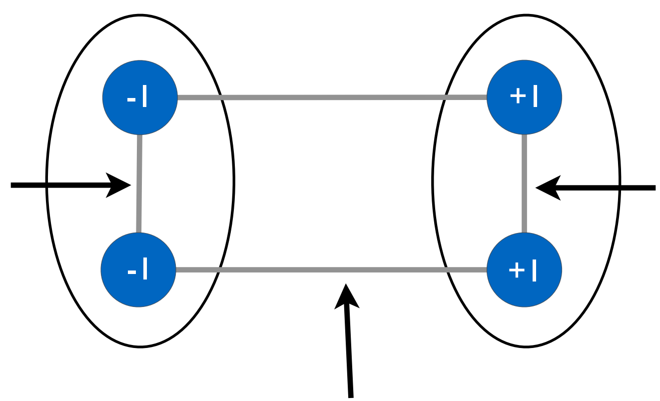
- Find natural communities or clusters in complex network
- Many measures of a good cluster (e.g., modularity, conductance)



2003-2004 House Committees
"Communities in Networks", AMS

Community detection via Ising

- Modularity
  - Measures strength of a partition of a graph into clusters/communities
  - Fraction of edges within clusters minus expected number within clusters for a random graph with a given degree distribution
  - Example: partitioning into 2 clusters
    - Each node i assigned $x_i=-1$ or $x_i=+1$, based on cluster in which it lives
    - $(1 + x_i x_i)/2$ indicates edges within clusters



$(1 + x_i x_i)/2 = (1 + -1*-1)/2 = 1$

$(1 + x_i x_i)/2 = (1 + 1*1)/2 = 1$

$(1 + x_i x_i)/2 = (1 + -1*1)/2 = 0$

Community detection via Ising

- Modularity
  - Fraction of edges within clusters minus expected number within clusters for a random graph with a given degree distribution
  - Example: partitioning into 2 clusters
    - Each node i assigned x_i=-1 or x_i=+1, based on cluster in which it lives
    - $(1 + x_i x_i)/2$ indicates edges within clusters
  - QUBO formulation for finding 2-clustering that maximizes modularity
    - m - number of edges
    - a_{ij} - 1 if edge ij is in the graph; 0 otherwise
    - d_i - degree of node i, i.e., number of edges incident to i
  - Note: this QUBO implementation requires a dense graph

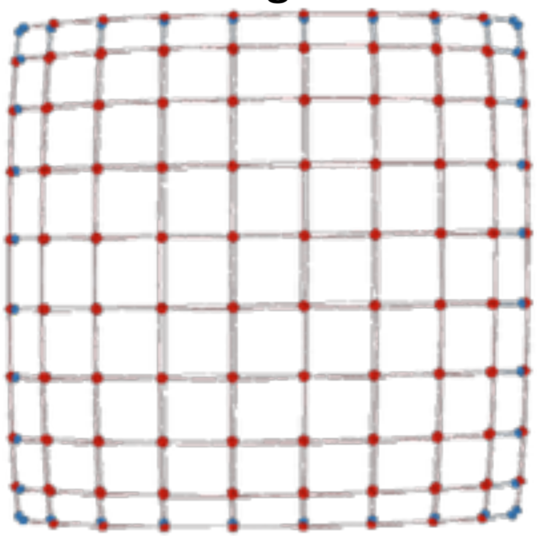
$$\sum_{i,j}\left(a_{ij}-\frac{d_i d_j}{2m}\right)\frac{1+x_i x_j}{2}=\underbrace{\sum_{i,j}\frac{1}{2}\left(a_{ij}-\frac{d_i d_j}{2m}\right)}_{\text{Constant term}}+\sum_{i,j}\underbrace{\frac{1}{2}\left(a_{ij}-\frac{d_i d_j}{2m}\right)}_{\text{(i,j) entry of QUBO matrix}}x_i x_j$$

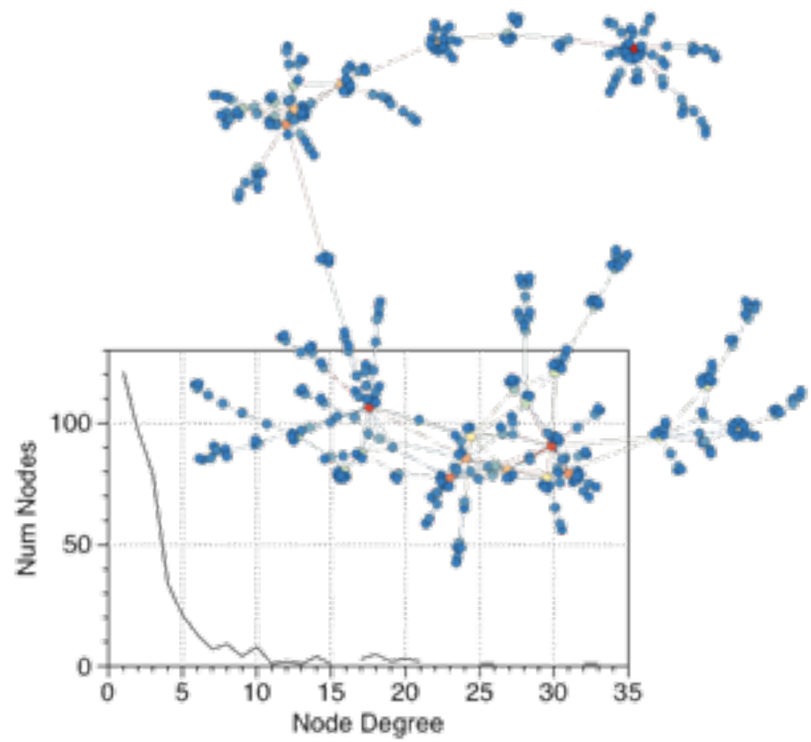
Constant term

(i,j) entry of QUBO matrix

Generating complex networks on a Chimera graph

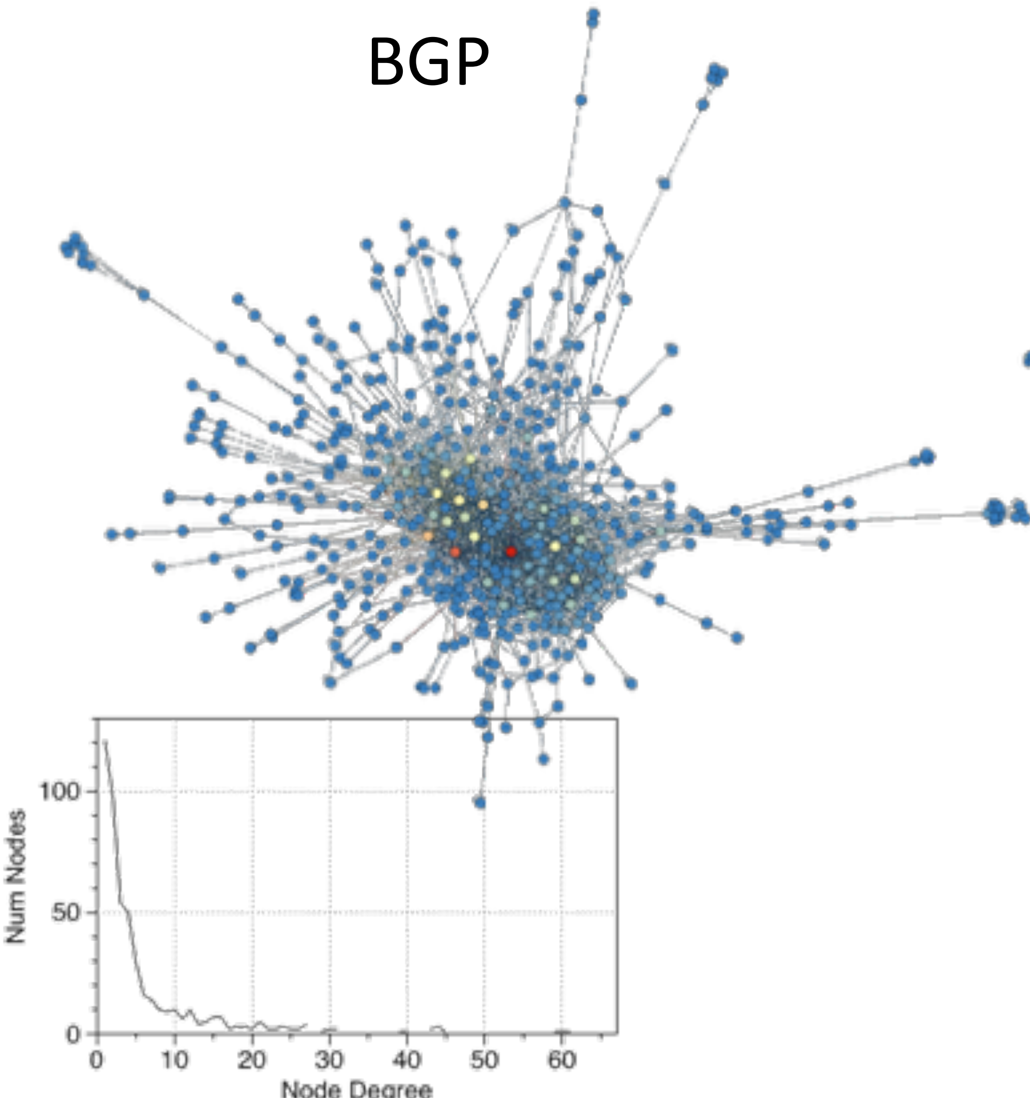
- Mapping arbitrary graph to Chimera is hard
- Instead, alter Chimera graph to have “real-world” properties
  - Merge nodes to increase node degree
  - Remove edges between nodes to decrease degree



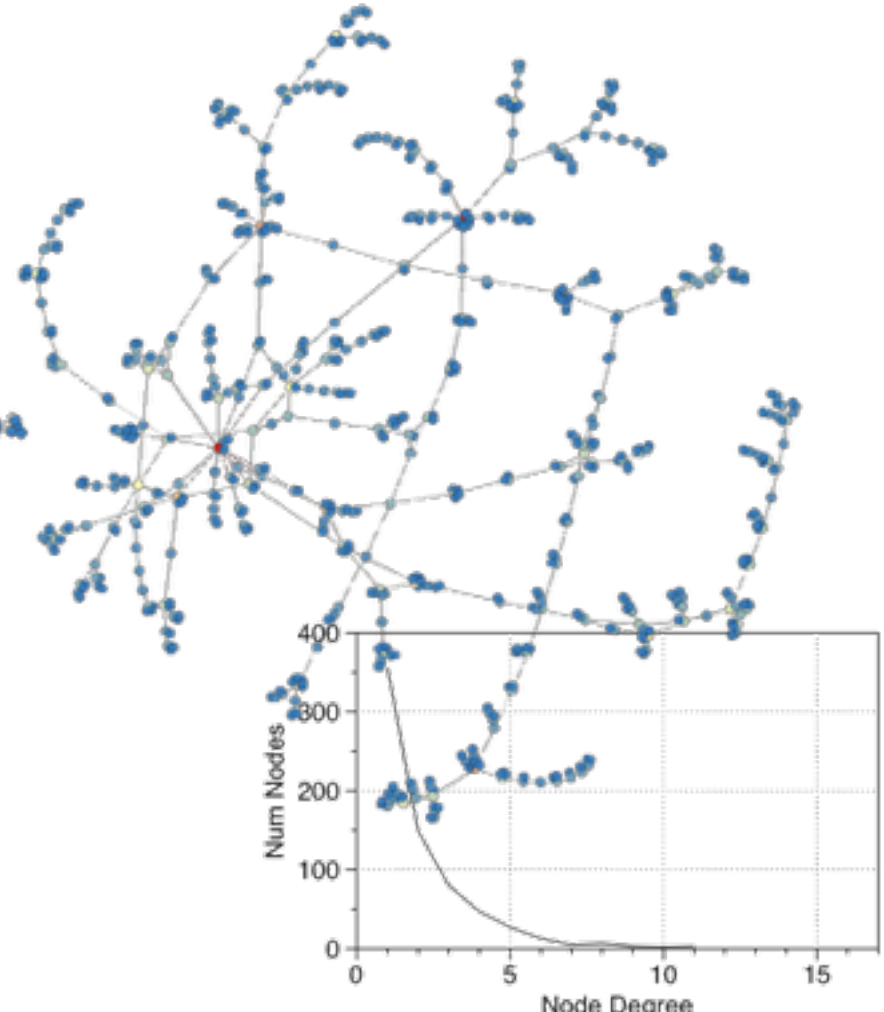


Real-world complex networks

BGP

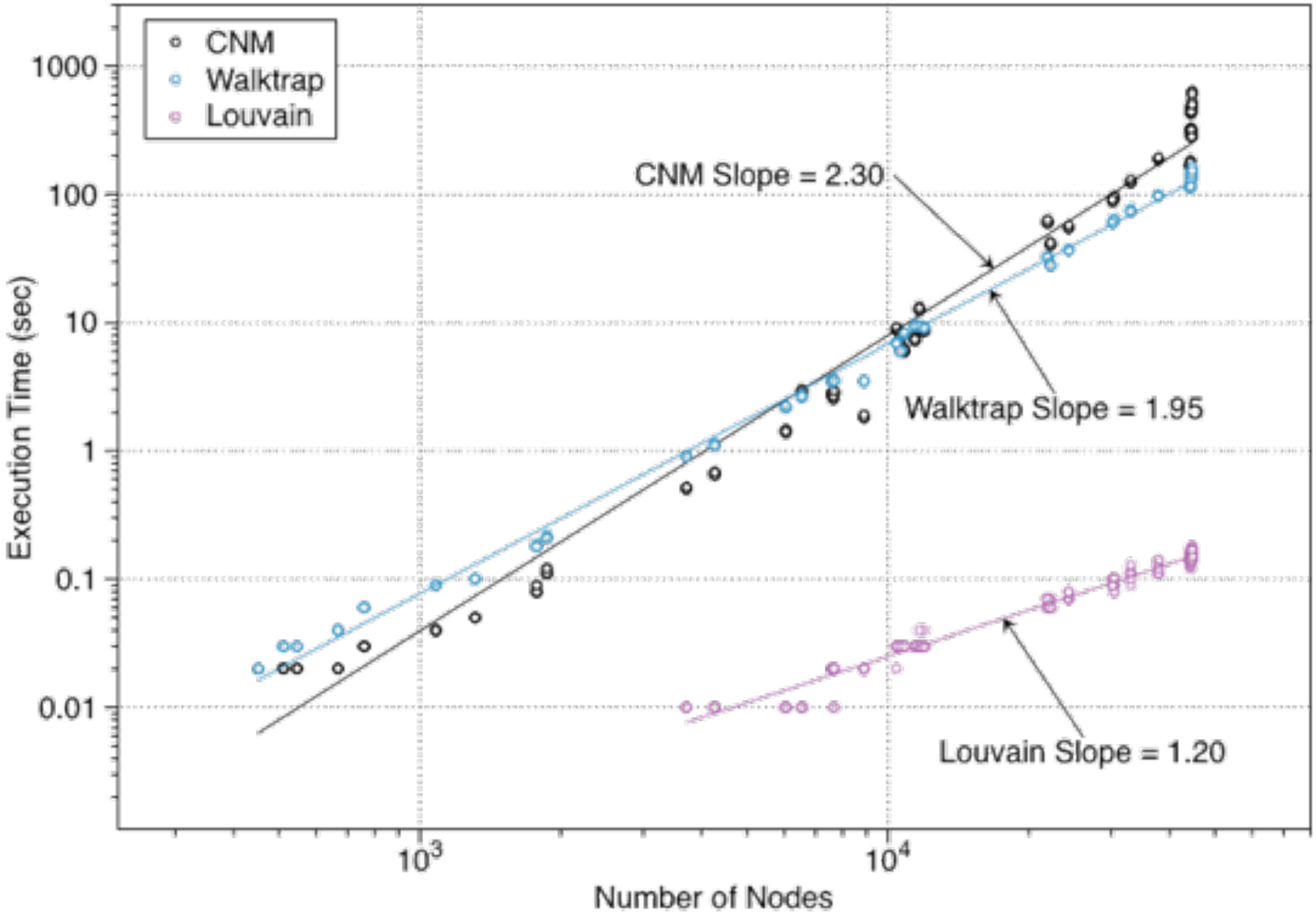


Twitter



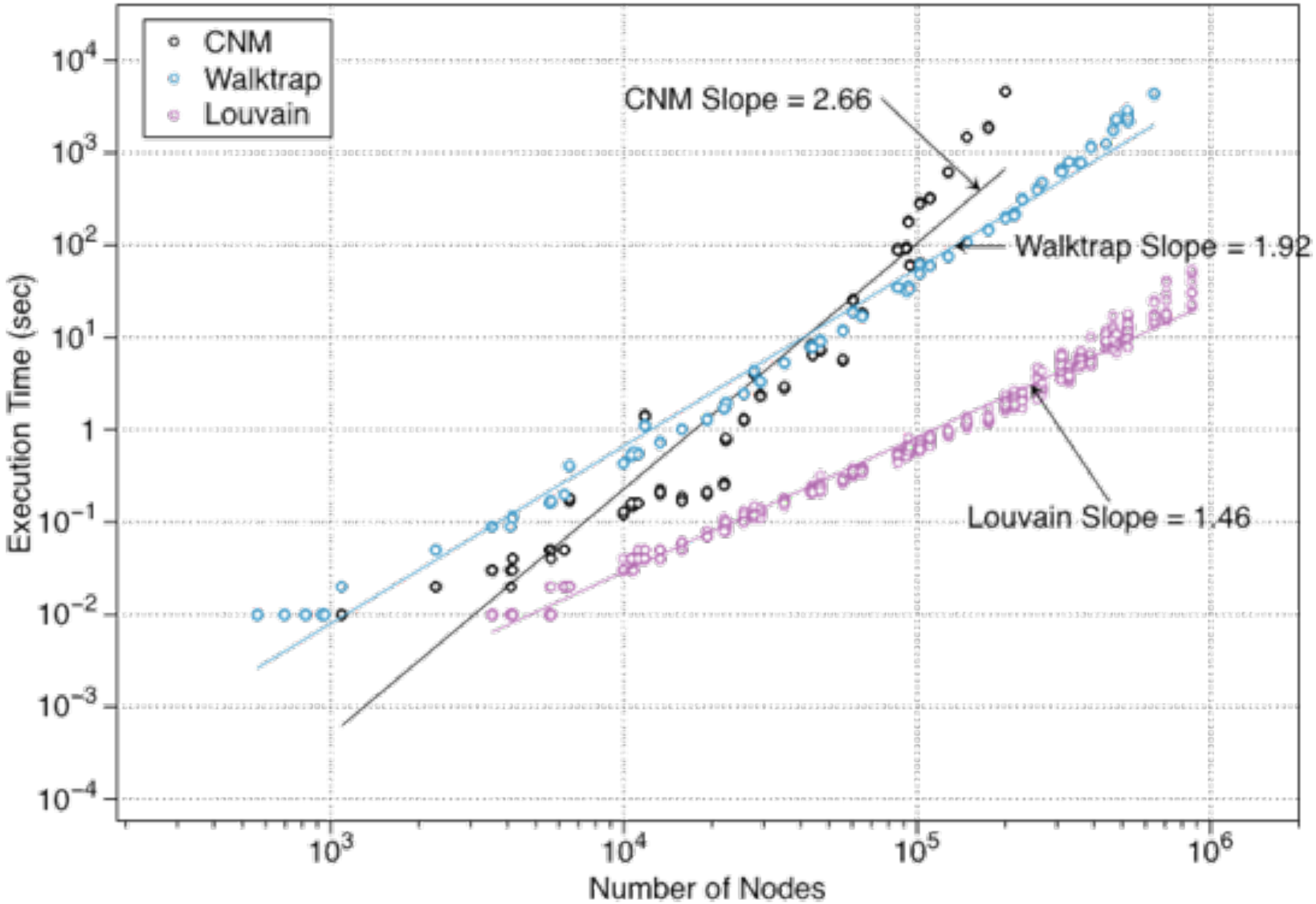
Classical Community Detection

- BGP Execution Time



Classical Community Detection

- Twitter Execution Time



Conclusion: We asked

- What is an appropriate measure of success?
  - Awareness of artificially hard problems for “natural” problems where close is good enough
  - Should try properly leverage hardness in defining success
- What classical algorithm(s) should be used for comparison?
  - Must select proper algorithm and configuration
  - Achieving fairness is tough when comparing an infant technology like quantum vs one honed over decades
- How should one select appropriate benchmark instances?
  - Balance between real-world, random, and hard is challenging