

# Quantum Annealing Approaches to Graph Partitioning for Electronic Structure Problems

ISTI LDRD D-Wave Quantum Computing Efforts Debrief

S. M. Mniszewski, [smm@lanl.gov](mailto:smm@lanl.gov)

C. F. A. Negre, [cnegre@lanl.gov](mailto:cnegre@lanl.gov)

H. Ushijima-Mwesigwa, [hayato@lsnl.gov](mailto:hayato@lsnl.gov)

April 27, 2017



Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

LA-UR-17-23580

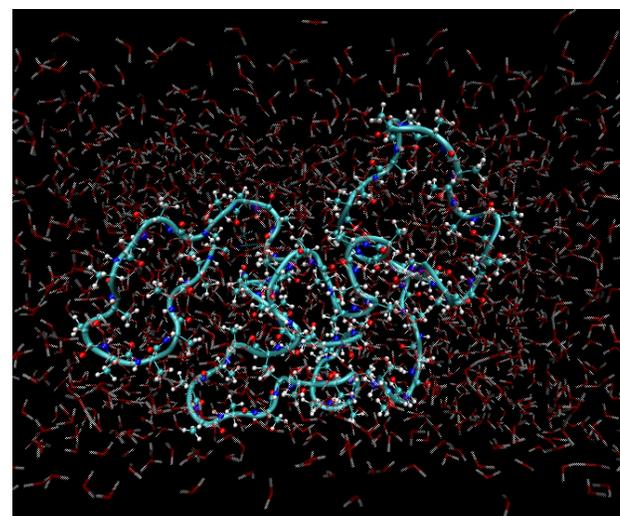
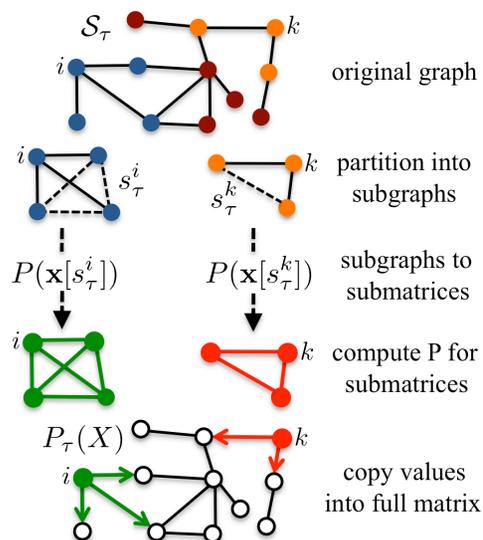
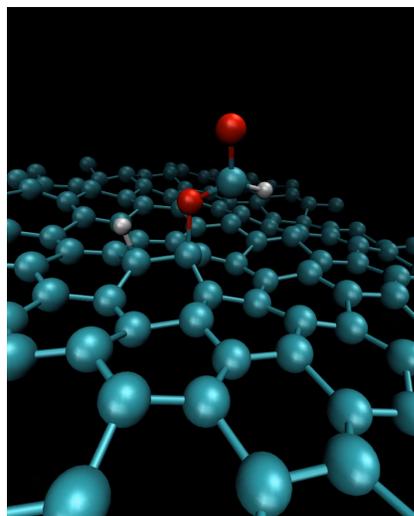
# Introduction

- **Motivated by graph-based methods for quantum molecular dynamics (QMD) simulations**
- **Explored graph partitioning/clustering methods and implementations that run on the D-Wave**
  - *k*-Concurrent graph partitioning into equal parts
  - *k*-Concurrent community detection using the modularity metric
  - Iterative multi-level graph partitioning with D-Wave refinement
- **Used *sapi* and hybrid classical-quantum *qbsolv* software tools**
- **Demonstrated “proof of principle” results on example graphs and material electronic structure graphs**
- **Results are shown to equal or out-perform current “state of the art” methods**

***Graph partitioning/clustering implementations on the D-Wave.***

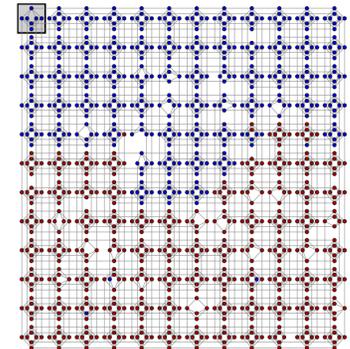
# Motivation

- Next Generation Quantum Molecular Dynamics LDRD-DR (PI:AMN Niklasson)
- Quantum-based models capture the making and breaking of covalent bonds, charge transfer between species of differing electronegativities, and long-range electrostatic interactions - reactions
- Graph-based methods for quantum molecular dynamics (QMD) simulations
  - A. M. N. Niklasson et al, Graph-based linear scaling electronic structure theory, *J. Chem. Phys.* **144**, 234101 (2016).
- Density matrix generated each timestep from many small sub-matrices (or sub-graphs)
- Shown to be equivalent to traditional methods (ex. diagonalization)



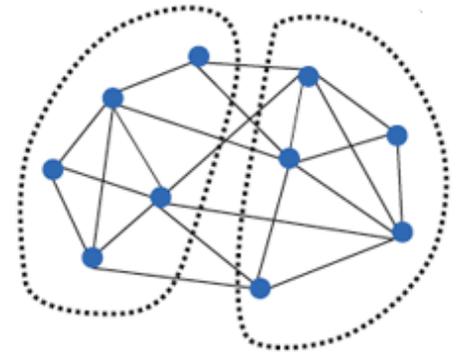
# Quantum Annealing on the D-Wave 2X System

- **Ising model formulation used for small graphs that are directly embeddable in the *Chimera graph using sapi***
  - Minimizes the objective function:  $O(h,J,s) = \sum h_i s_i + \sum J_{ij} s_i s_j$
- **Larger graphs are formulated as QUBOs for hybrid classical quantum *qbsolv* - 0/1 valued variables**
  - Minimizing the objective function:  $O(Q,x) = \sum Q_{ij} x_i + \sum q_{ij} x_i x_j$
- **Reduced qubit/coupler footprint achieved by thresholding and graph complement**
- **D-Wave 2X Architecture**
  - Fixed sparse graph  $G = (V,E)$  called the *Chimera graph*
  - 1095 qubits, 3061 couplers, with sparse bipartite connectivity
  - *Chimera graph* consists of a 12 x 12 array of 4 x 4 bipartite unit cells



# Graph Partitioning - Description

- **Definition:**
  - Given a graph  $G = (V, E)$ 
    - $V \sim$  nodes
    - $E \sim$  edges (possibly weighted)
- **Goal:** Partition  $V$  into  $k$  equal parts minimizing the number of cut edges between parts
- **Applications:**
  - Graph-based QMD simulations - Used as initial solution for core-halo partitioning
  - Physical network design
  - VLSI design
  - Telephone network design (original application with algorithm due to Kernighan)
  - Load balancing - minimize total communication between processors
  - Sparse matrix-vector multiplication - Partition the rows of a matrix to minimize communication during matrix-vector multiplication



# Graph Partitioning – Benchmarks and Random Graphs

- **Graphs partitioned:**
  - Using *sapi* for small graphs (up to 45 vertices)
  - Using *qbsolv* for large graphs (> 45 vertices, up to 1000s)
- **Results comparison: minimize number of cut edges between parts**
  - Quality
    - METIS
    - KaHIP (**winner 10<sup>th</sup> DIMACS challenge**)
    - Best known solution
  - Data
    - Walshaw benchmark archive (<http://chriswalshaw.co.uk/partition/>)
    - Random graph models (from NetworkX), e. g. Erdos-Renyi, PowerLaw graphs
    - Molecule electronic structure graphs from QMD simulations
- **Comparable to existing methods and sometimes better!**

# Graph 2-Partitioning Formulation

- **Exact Formulation**

- $\min x^T L x$ , such that  $\sum x_i = n/2$ ,  $x_i \in \{0,1\}$

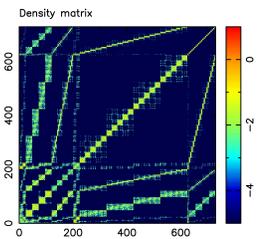
- **QUBO**

- $\min x^T (\beta L + \alpha 1_{n \times n}) x - \alpha n (\sum x_i) + \alpha n^2/4$ ,  $x_i \in \{0,1\}$

- **where**

- $L \sim$  Laplacian matrix;  $1_{n \times n} \sim$  all ones matrix

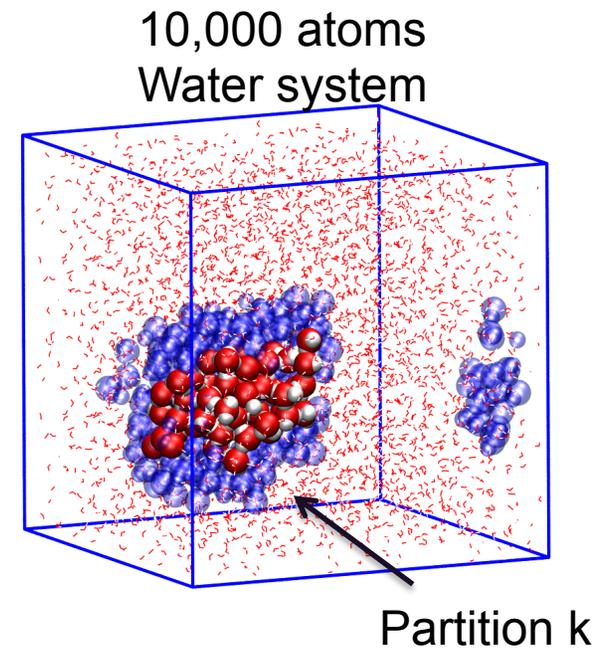
- $\alpha, \beta \sim$  penalty constants



Density matrix (for QMD)

$$\rho_{ij} = \langle i | f(H) | j \rangle$$

METIS



# Graph 2-Partitioning Results using *qbsolv*

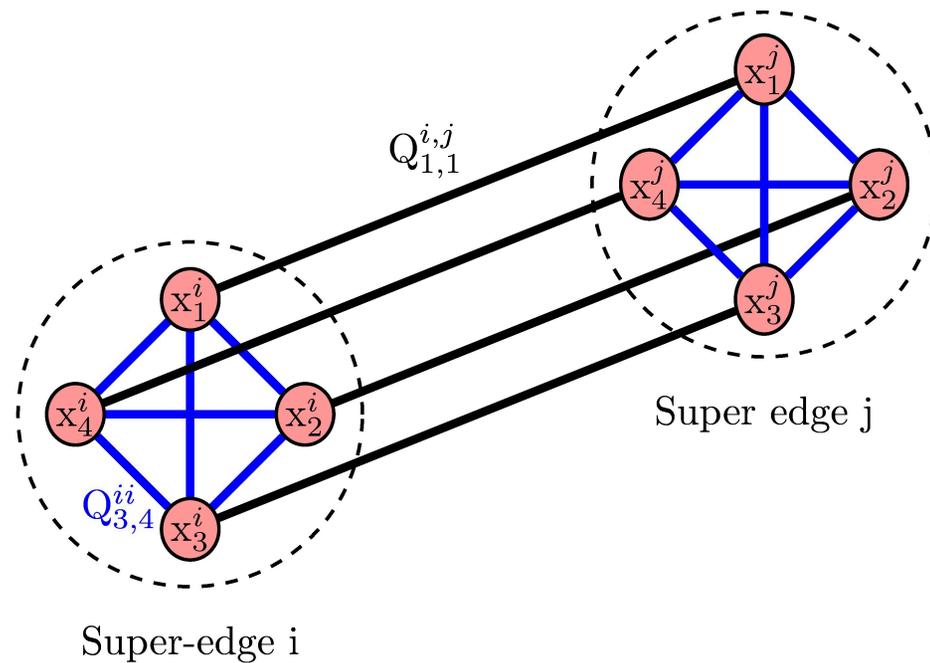
Walshaw GP benchmark graphs:

Graph	n	best	METIS	KaHIP	qbsolv
add20	2395	596	723	760	647
data	2851	189	225	221	191
3elt	4720	90	91*	92	90
bcsstk33	8738	10171	10244*	10175	10171

\*: not 0% imbalance

# ***k*-Concurrent Graph Partitioning – Multiple parts in parallel**

- Partition into  $k$  parts in parallel
- Uses super-node concept
- Unary embedding
- $k$  logical qubits per vertex
- New formulation requires a  $kn \times kn$  QUBO
- Results in 1 of  $k$  qubits set on for each vertex
- Similar to graph coloring problem
- Useful for graph partitioning and community detection



**Super-node Concept**

# Graph $k$ -Partitioning Formulations

- **QUBO Formulation 1 – Proof of Principle**

- Combines super-node concept with the 2-partitioning formulation and constraints on balanced part sizes and each node in 1 of  $k$  parts
- Penalty constants:  $\alpha$  (balancing),  $\beta$  (minimize cut edges)
- Requires tuning of node weights and and coupler strengths for super-node
- Resulting parts not always well-balanced

- **QUBO Formulation 2 –  $(k-1)n \times (k-1)n$  QUBO**

- Combines super-node, balancing constraints on part sizes, and constraint on each node being in 1 of  $k-1$  parts
- Penalty constants:  $\alpha$  (balancing),  $\beta$  (minimize cut edges),  $\gamma$  (each node in 1 part)
- Results not always balanced,  $k-1$  balanced parts and large  $k$ th part

- **QUBO Formulation 3 –  $kn \times kn$  QUBO**

- Combines super-node with explicitly enforcing balancing constraints on part sizes, and constraint on each node being in 1 of  $k$  parts
- Penalty constants, :  $\alpha$  ,  $\beta$ ,  $\gamma$
- Results in balanced parts

## ***k*-Concurrent Graph Partitioning – Directly on D-Wave**

- Dense random graphs with  $\rho=0.9$  (from NetworkX)
- Using *sapi* for embedding and solving
- Limited to ~45 node graphs
- A 15-node graph into 4 parts and 20-node graph into 3 parts use 900+ qubits
- Results comparable for *sapi*, METIS, and *qbsolv*
- Results using *sapi* are typically equal to *qbsolv*

<b>n</b>	<b>k</b>	<b><i>sapi</i></b>	<b>METIS</b>	<b><i>qbsolv</i></b>
10	2	19	19	19
	3	29	29	29
	4	32	33	32
15	2	45	47	45
	3	62	62	62
	4	70	73	70
20	2	83	83	83
	3	120	122	120
27	2	156	164	156
30	2	182	183	182

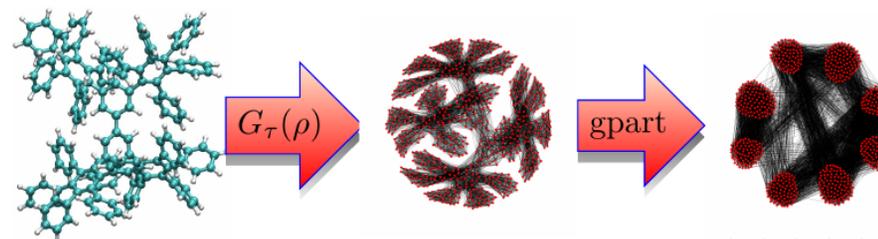
# ***k*-Concurrent Graph Partitioning – Large Random Graphs**

- Random graphs with  $\rho=0.9$  (from NetworkX)
- Using *qbsolv* for large graphs
- Penalty constants  $\alpha = 1000$ ,  $\beta = 1$ ,  $\gamma = 5000$
- Produces  $kn \times kn$  QUBO
- Results in equal sized parts
- Typically equal or better than METIS

n	k	METIS	qbsolv
250	2	13691	13600
	4	20885	20687
	8	24384	24459
	16	26224	26176
500	2	55333	54999
	4	83175	83055
	8	98073	97695
	16	105061	105057
1000	2	221826	221420
	4	334631	334301
	8	392018	392258
	16	421327	420970

# *k*-Concurrent Graph Partitioning – Molecule electronic structure graphs

Graph	k	METIS	qbsolv
Phenyl dendrimer n = 730	2	706	706
	4	20876	2648
	8	22371	15922
	16	28666	26003
	32	49732	49745
Peptide 1aft n = 384	2	12	12
	4	29	20
	8	121	66
	16	209	180
	32	495	425



# Graph Clustering using Community Detection

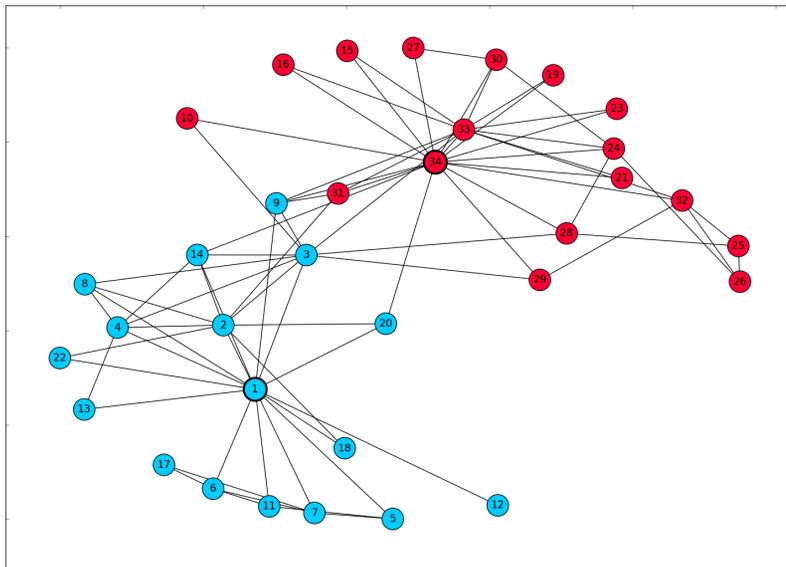
- **Using modularity or community network analysis for natural clusters**
  - M. E. J. Newman, Modularity and community structure in networks, 2006, *PNAS*, vol. 103, no. 23, 8577-8582.
- **For example, identifying secondary structures in proteins**
  - I Rivalta, MM Sultan, N-S Lee, GA Manley, JP Loria, VS Batista, Allosteric pathways in imidazole glycerol phosphate synthase, *PNAS*, vol. 109, no. 22, pp. 1428-1436 (2011).
- **Maximize modularity**
- **Use of thresholding to reduce edges – qubits/couplers**
- **Fits naturally on D-Wave machine, no reformulation required**
- **New  $k$ -Concurrent community detection formulation**

# 2-Clustering Community Detection with Thresholding

- Using modularity or community detection for natural clusters
- Compare for 2 clusters
- Effect of thresholds on weights
- Reduction in qubits/couplers

Karate club graph (n = 34) using qbsolv

Threshold	# edges	modularity
0	561	0.371794871795
0.02	544	0.371794871795
0.05	411	0.371466140697
0.07	300	0.371466140697
0.08	244	0.271449704142
0.1	227	0.271449704142
0.11	212	0.255095332018
0.12	194	0.255095332018
0.13	169	0



## 2-Clustering for Identifying Communities in Bio-Systems

- IGPS is an enzyme in bacteria consisting of 2 molecules (454 residues)
- Applying community detection using *qbsolv* resulted in communities corresponding to IGPS's 2 molecules
- The modularity matrix is calculated from a correlation matrix based on an MD simulation

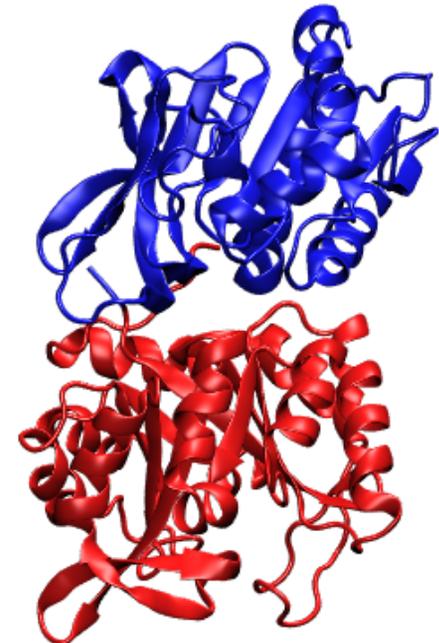
Correlation matrix (for CNA)

$$\mathbf{r}_{ij}^{MI} = g(\mathbf{I}[\mathbf{x}_i, \mathbf{x}_j])$$

Girvan-Newman



Modularity  
Matrix

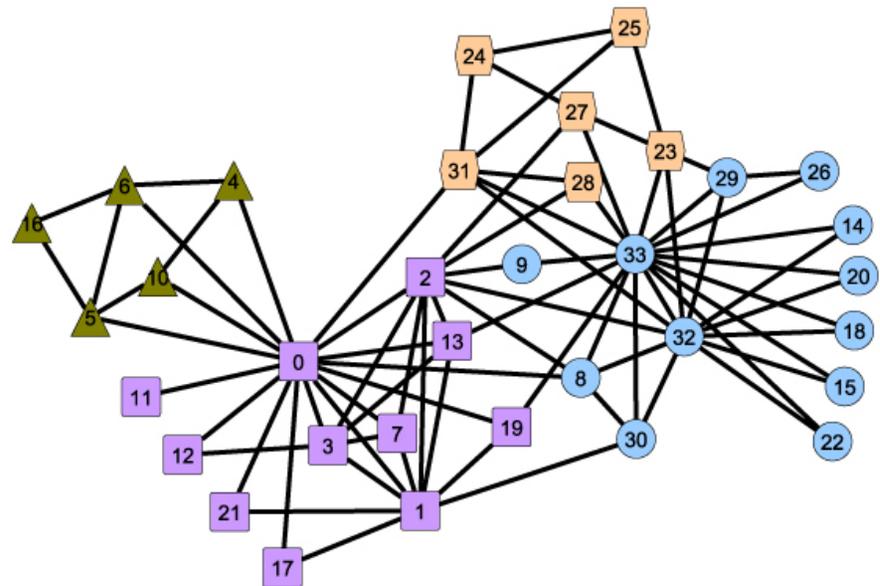


IGPS Protein Structure

# *k*-Concurrent Community Detection

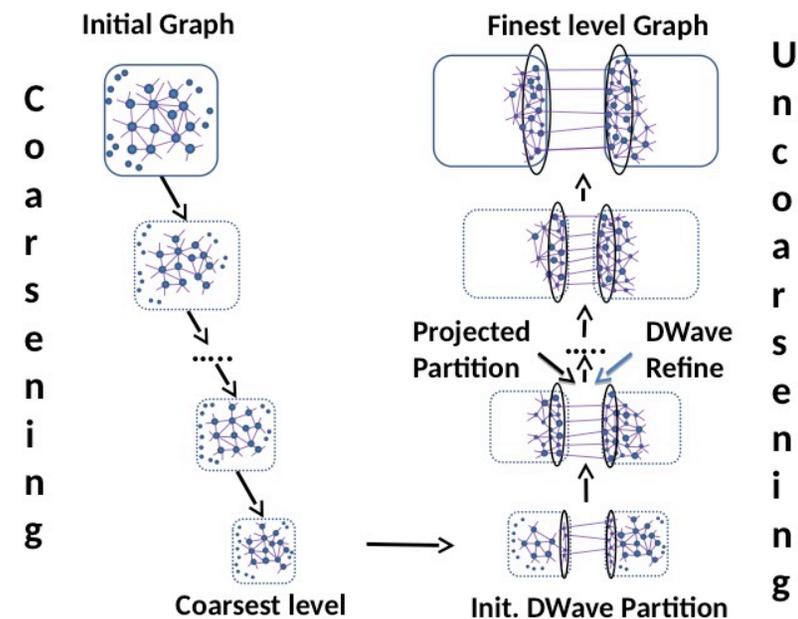
- Determine at most  $k$  communities, maximizing modularity in parallel
- Similar formulation as  $k$ -concurrent graph partitioning
  - $kn \times kn$  QUBO
  - Constraint: each node in only 1 community
  - Tunable penalty constants,  $\alpha$ ,  $\gamma$
- Using *qbsolv* on D-Wave
- Matches best known karate club graph results for communities and modularity
- Next, apply to more bio-systems and social networks

Karate club graph ( $N = 34$ )  
4 communities, modularity = 0.419789



# Multi-level Graph Partitioning with D-Wave Refinement

- **General solution strategy consisting of 3 stages**
  - Coarsening
  - Initial solution – on D-Wave
  - UnCoarsening – refinement on D-Wave
- **Successfully used in graph partitioning solvers**
  - METIS, KaHIP, and more
- **Explore KaHIP coarsening with D-Wave (*sapi*, *qbsolv*) refinement**
- **Develop prototype for specialized classical-quantum graph partitioning software**



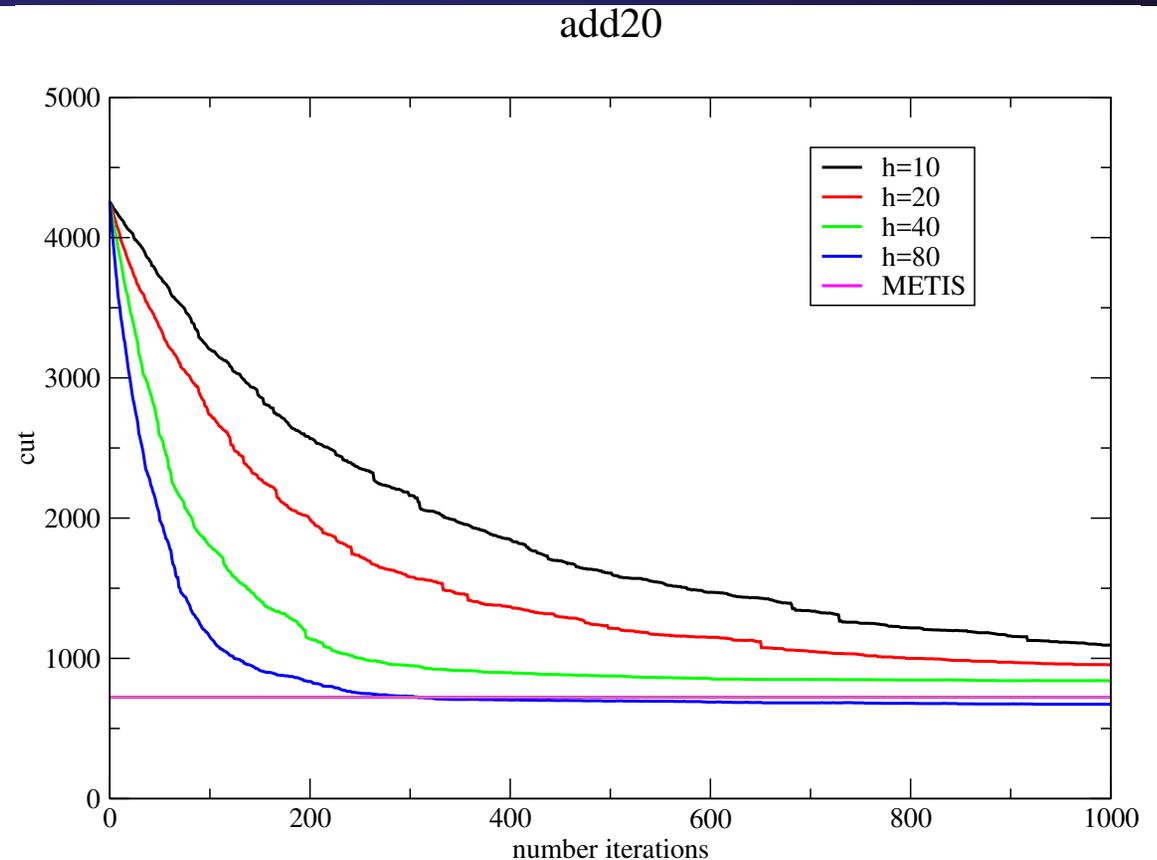
# Multi-level Graph Partitioning – Early results

- **Test graph from SuiteSparse Matrix Collection**
  - Delaunay\_n15, 32,768 vertices
- **Multi-level using KaHIP/*qbsolv* runs faster than *qbsolv* alone**
- **Results comparable to METIS and KaHIP**
- **Currently using 2-partitioning formulation for refinement**
- **Next, try k-concurrent partitioning and community detection**
- **More testing required**

Method	Timing	Quality
METIS	seconds	359 (16389, 16379)
KaHIP	seconds	364
<i>qbsolv</i>	hours	2888 (16384, 16384)
Multi-level KaHIP/ <i>qbsolv</i>	minutes	460

# Multi-level Graph Partitioning – Varying D-Wave embeddable graph size

- Partition  $h$  vertices at each iteration
- Walshaw benchmark graph
  - add20
  - 2395 vertices, 7462 edges
- Example runs of multi-level graph partitioning for add20 using different D-Wave maximum embeddable graph sizes ( $h \times h$ )
- Larger  $h$  results in less iterations required



## Next Steps

- **Publish graph partitioning and community detection results**  
Graph Partitioning using Quantum Annealing on the d-Wave System.  
H Ushijima-Mwesigwa, CFA Negre, SM Mniszewski
- **Apply k-concurrent graph partitioning to more examples**
- **Apply k-concurrent community detection to more bio-systems and social networks**
- **More development and testing for multi-level approach**
- **Explore reduced qubit approaches for current formulations**
- **Make code available**
- **Collaborate on relevant graph structure problems**

**The End**

**Thank You!**