# Efficient Combinatorial Optimization Using Quantum Annealing

**Team**: Hristo Djidjev, PI (CCS-3)

Guillaume Chapuis (CCS-3)

Georg Hahn (Imperial College, UK)

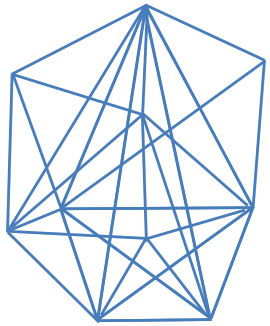
Guillaume Rizk (INRIA-Rennes, France)

October 6, 2016

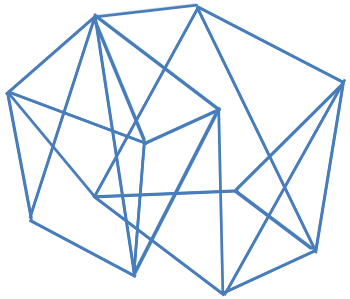
**LA-UR-16-27928**

# Objectives

- Develop algorithms for solving NP-hard optimization problems suitable for QUBO/D-Wave implementation
- Study scalability/accuracy issues and ways to mitigate them
- Characterize problem instances for which D-Wave may outperform classical alternatives
- Specific problems studied:



maximum clique



graph partitioning

balance parts sizes

minimize # cut edges

# Solving optimization problems on D-Wave

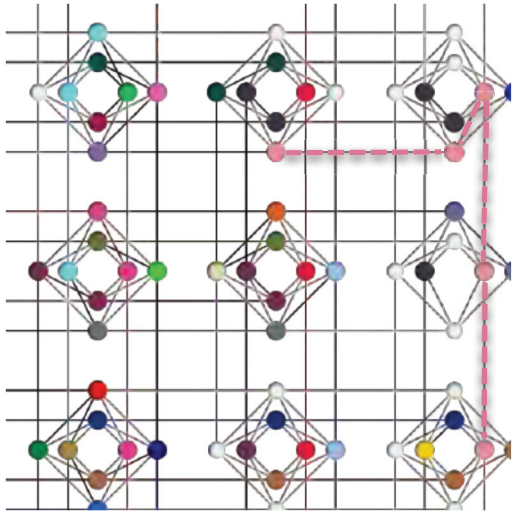
- Formulate as a quadratic integer problem

$$\min_{q_1,\ldots,q_n} \left( \sum_{i=1}^{n} a_i q_i + \sum_{1 \le i < j \le n} a_{ij}\, q_i q_j \right)$$

  - Ising formulation: $q_i \in \{-1, 1\}$
  - QUBO formulation: $q_i \in \{0, 1\}$
- Map problem onto DW hardware
  - Embed connectivity graph defined by $a_{ij}$ coefficients into DW hardware (Chimera) graph
  - Encode $a_{ij}$ and $a_i$ coefficients as DW hardware parameters
- Anneal, read solutions in a loop (~500 times)

# Challenges

- Number of qubits (~1000)
- Chimera graph connectivity
  - Variables mapped onto “chains” of qubits
  - Cannot embed complete graphs of more that 45 vertices
  - Chain qubit values can differ
- Limited precision for encoding qubit and coupler weights
- Environmental noise during annealing affects the outcome



# QUBO formulations for Max Clique

- Constrained formulation

$$\text{maximize:} \quad \sum_{i \in V} x_i$$

$$\text{subject to:} \quad \sum_{(i,j) \in \bar{E}} x_i x_j = 0, \quad x_i \in \{0, 1\}$$

- Unconstraint (QUBO) formulation

$$\text{minimize} \left( -\sum_{i \in V} x_i + M \sum_{(i,j) \in \bar{E}} x_i x_j, \quad x_i \in \{0, 1\} \right)$$

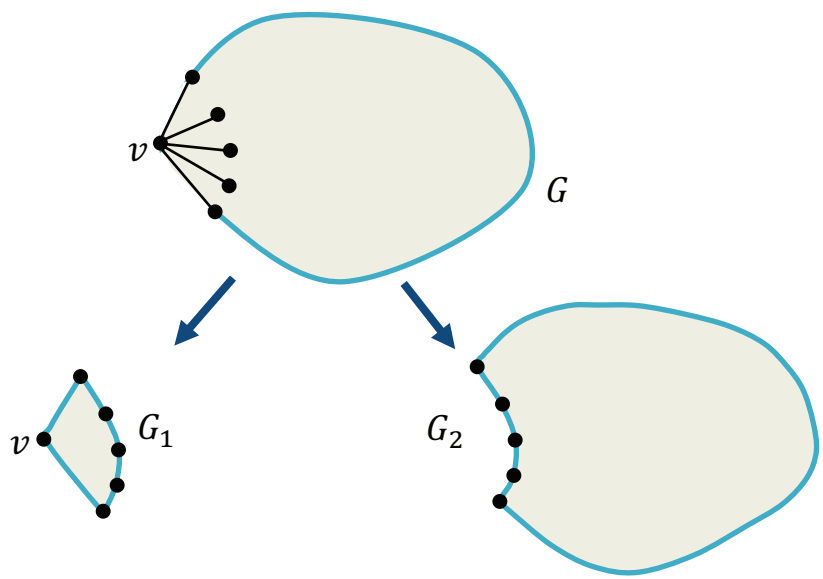
  - dense $\{a_{ij}\}$ matrix for sparse input graphs
  - problem sizes limited to ~45

# Methods for solving larger problems

- Extracting the $k$-core
  - If looking for a clique of size $\geq k$+1, remove all vertices of degrees less than $k$ and iterate
  - the resulting subgraph is a $k$-core
  - find a max clique on the $k$-core, rather than the original graph
- Subgraph partitioning
  - divide $G$ into overlapping subgraphs of “small” sizes (≤ 45)
  - find the max clique in each subgraph
  - one of these cliques is a max clique of original graph

# Methods for solving larger problems (cont.)

- Single-vertex partitioning
  - pick a vertex $v$
  - solve problem for graphs $G_1 = v + N(v)$ and $G_2 = G - v$ (recurs.)
  - return the larger of the cliques



# D-Wave tools for solving QUBO/Ising

- sapi
  - input in Ising or QUBO format
  - allows control over quantum annealing phases and parameters
- qSage
  - Input any objective functions of *n* binary variables
  - hybrid method - uses a combination of tabu search and quantum sampling
- qbsolve
  - problem size not limited by hardware parameters
  - hybrid solver (not clear how much DW annealing contributes)

# Sapi implementation

- Graphs of size ≤45
- Use precomputed $K_{45}$ embedding
- Do 500 annealings, evaluate quality/optimality
- Total solution runtime ~0.15 sec.
- Broken chains

  $x_2 = 1$: 000000000000000000000001111

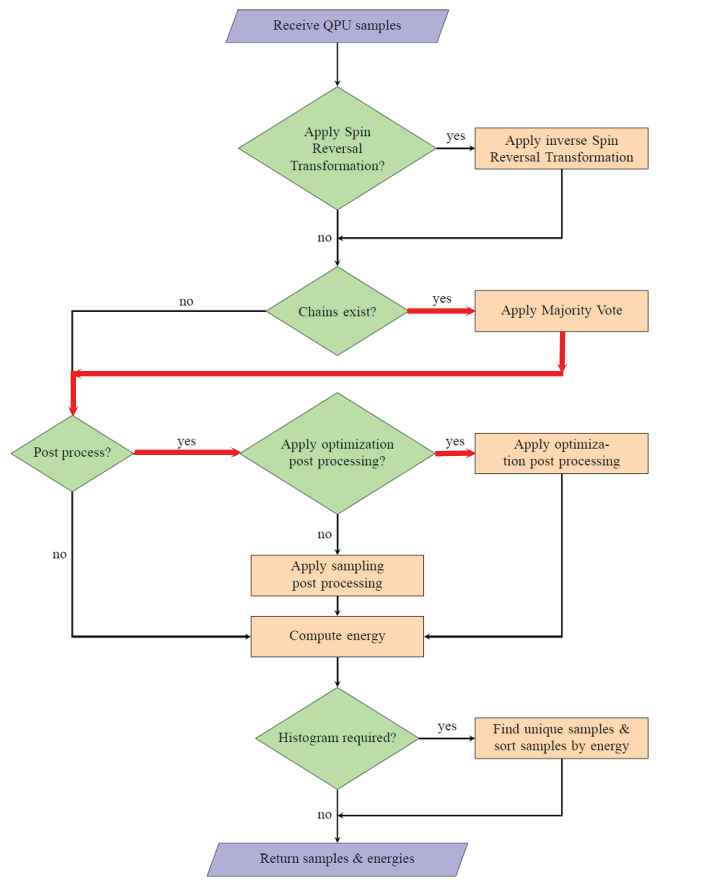
  $x_4 = 1$: 000000000000000000000001111

  $x_6 = 0$: 000000001111110000000000

  $x_7 = 0$: 000000000000000000000011111

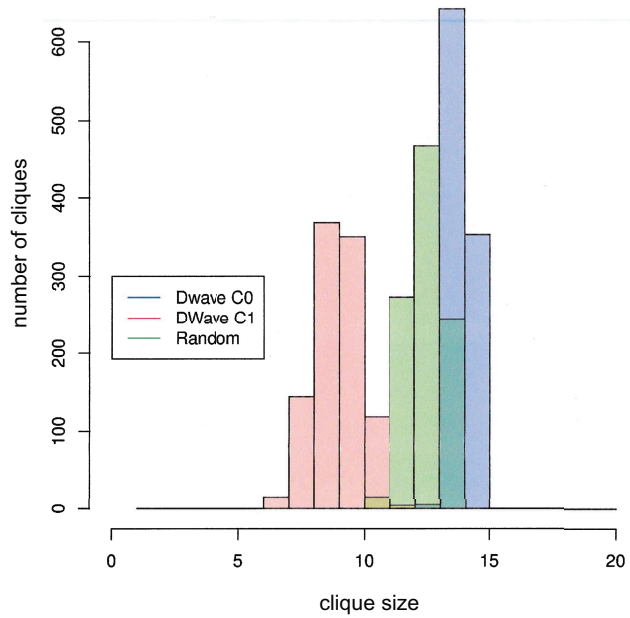
  ... (16 broken chains out of 41)

# How D-Wave handles broken chains

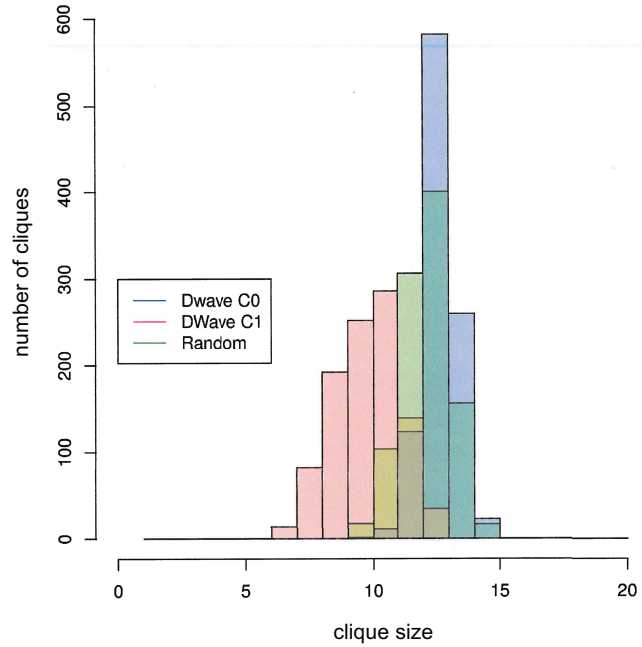



source: D-Wave

# The effect of postprocessing

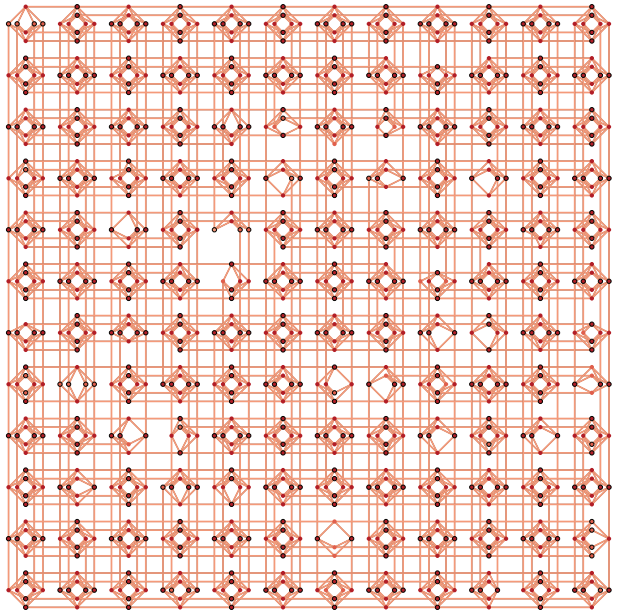



Stored $K_{45}$ embedding

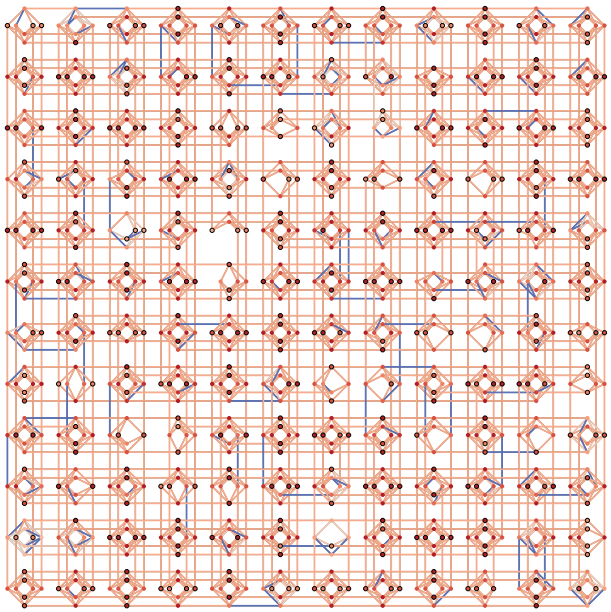



Computed embedding

# Using bigger input graphs

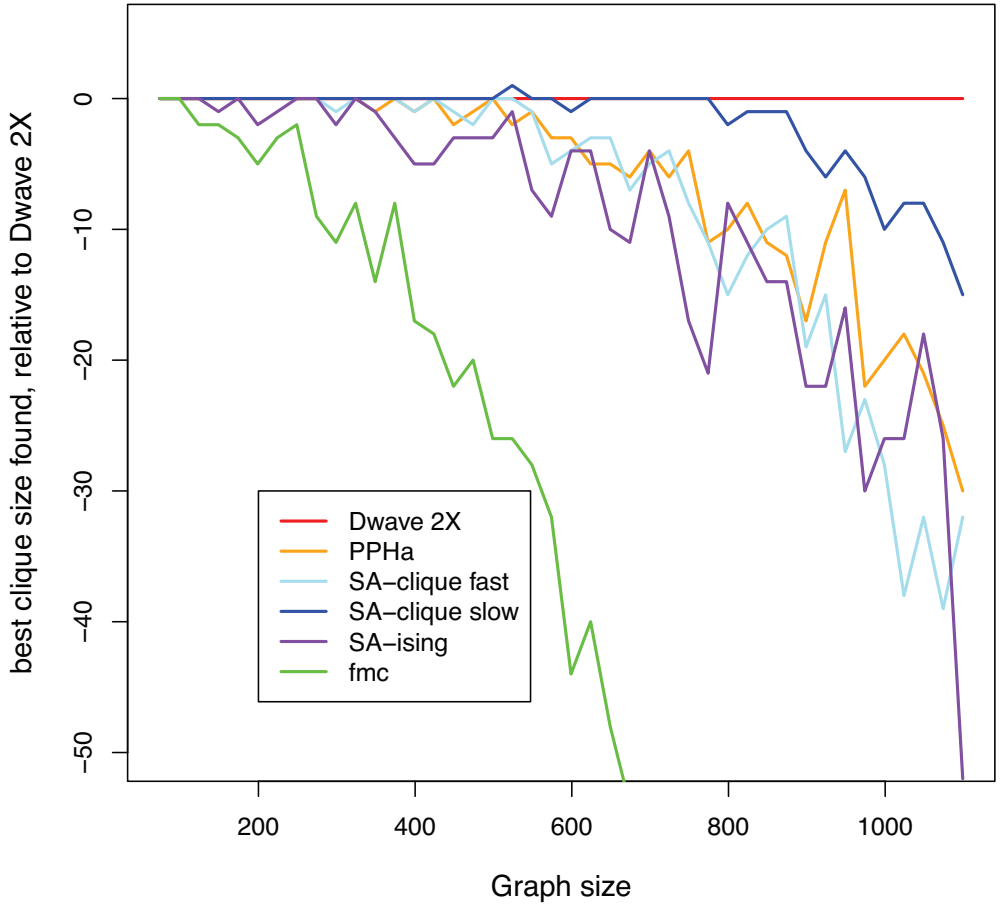


Chimera graph (LANL)

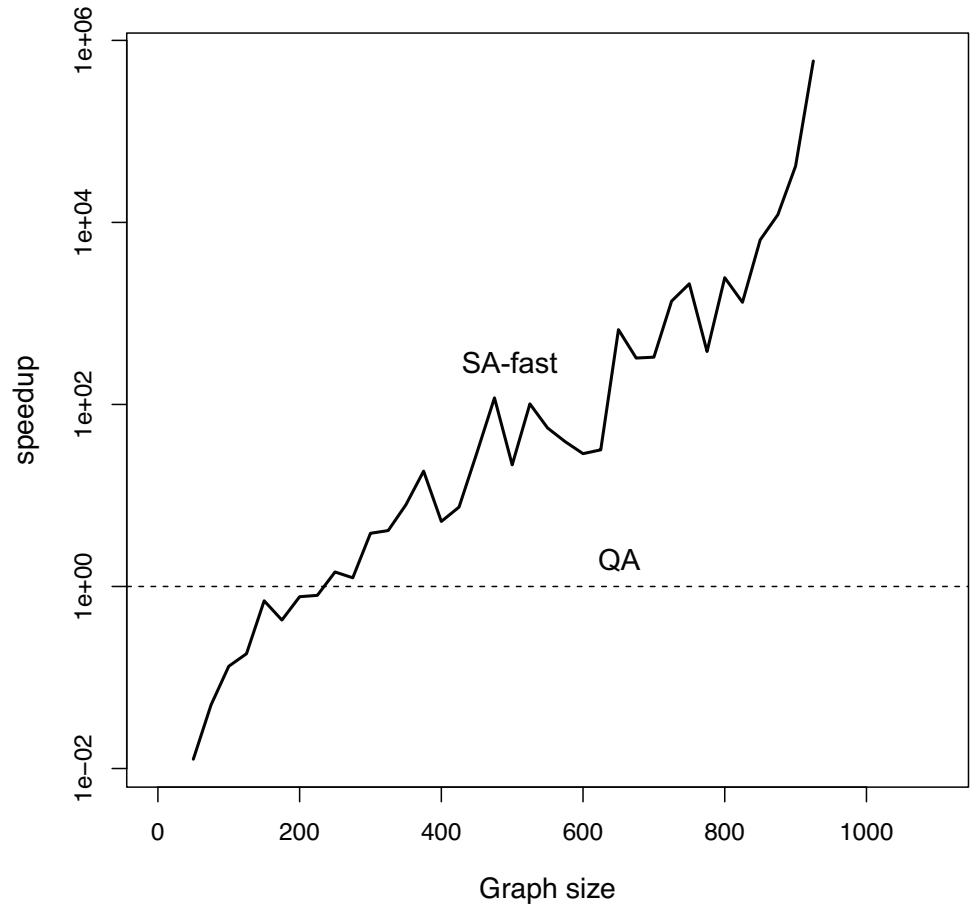


Chimera plus random chains

# Quality comparison



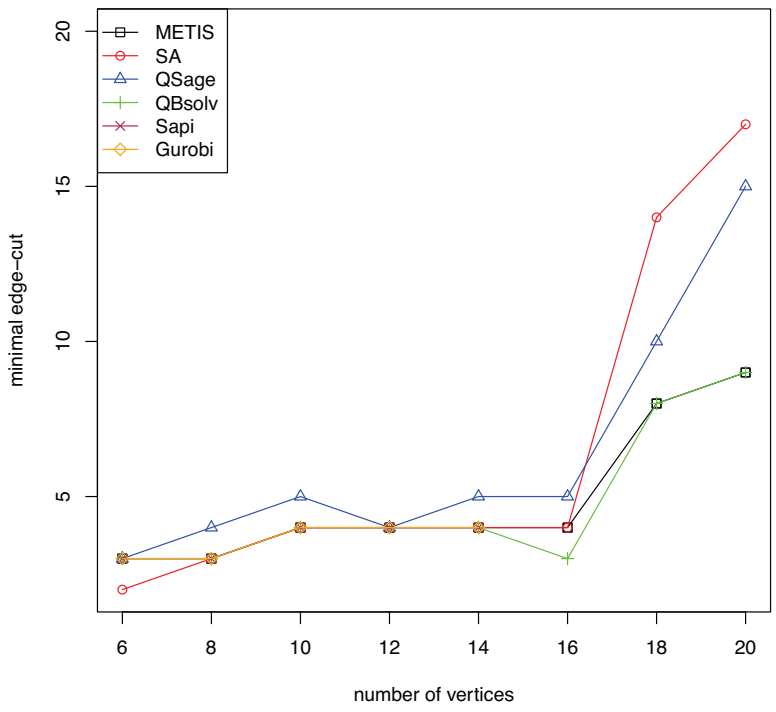
# Speed comparison

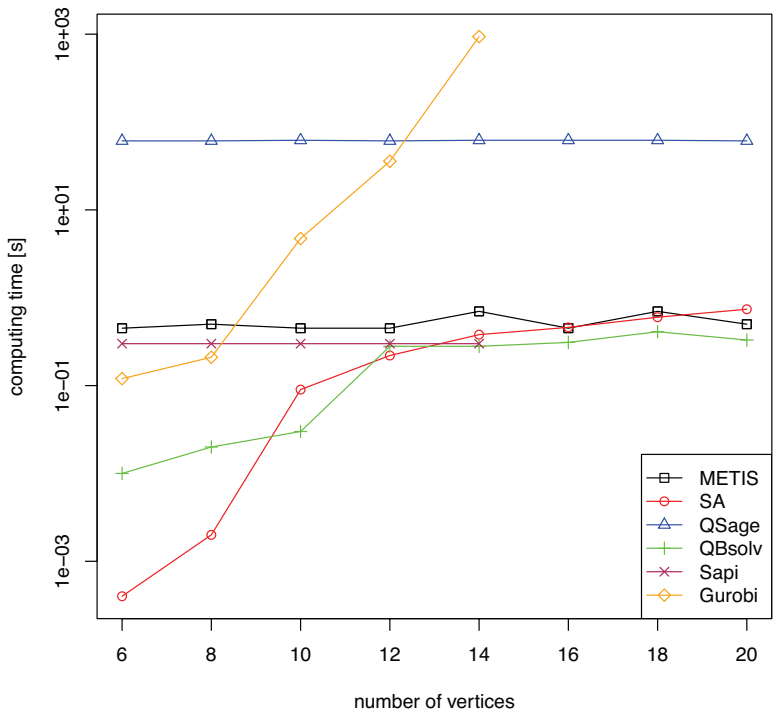


# Other results

- Probabilistic version of the single-vertex partitioning algorithm (about 5x speedup)
- Using an edge version of $k$-core technique
- Computing upper bounds by problem relaxation
- Studying other QUBO formulations
- Reducing the Hamiltonian's density by reformulation techniques (introduction of new variables) (work in progress)
- Studying the effects of spin reversals (work in progress)
- Results on solving graph partitioning on D-Wave

# Graph partitioning results



# Conclusions

- D-Wave uses very good heuristic classical algorithms (sapi postprocessing, qbsolv, qSage)
  - may mask shortcomings of the quantum phase
  - one should use sapi *without* postprocessing to study QA
- Can use graph decompositions to solve larger problems
- Topology of Chimera cause long chains that affect the accuracy
- Limited encoding precision of coefficients
- Quantum speedup over SA is real; could be huge if input graphs are large enough
- A lot of challenging research problems need to be solved in order to realize the potential of quantum annealing