Multilevel Quantum Annealing For Graph Partitioning

Hayato Ushijima-Mwesigwa Christian F. A Negre, Susan M. Mniszewski, Ilya Safro

Qubits 2018
D-Wave Users Conference

September 25, 2018

Motivation

Challenge:

- D-Wave 2X
  - $\approx 45x45$ arbitrary QUBO
- D-Wave 2000Q
  - $\approx 65x65$ arbitrary QUBO

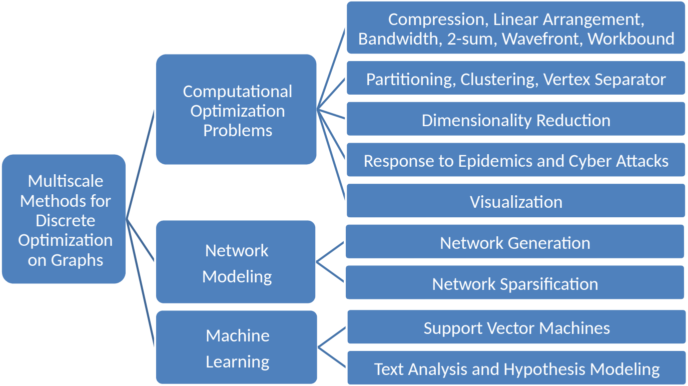
Question:
How can we efficiently use near-term D-Wave computers for solving large-scale problems?
Approach:
Hybrid classical-quantum algorithms within the multilevel framework

Multilevel Methods For Combinatorial Optimization

Multilevel Methods:

- Technique useful for problems with multiple scales of behavior
- Major phases:
  - Coarsening Phase
  - Initial Solution
  - Uncoarsening Phase
    - Interpolation
    - Refinement

Multilevel Methods For Combinatorial Optimization



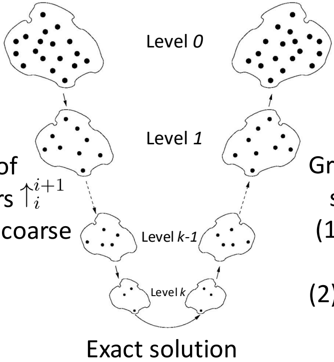
Applications of Multilevel Methods

Multilevel Methods For Optimization

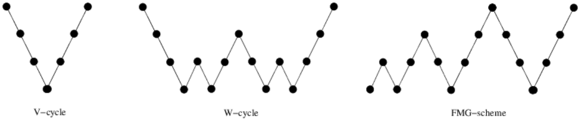
- Line search multigrid for convex optimization
  (Goldfarb, Wen)
- PDE-constrained optimization
  (Borzi, Nash, Toint, ...)
- Multilevel trust-region methods
  (Gratton, Mouffe, Sartenaer, Toint, ...)
- Non-convex non-linear optimization for VLSI placement
  (Chan, Cong, Sze, ...)
- Linear programming - multilevel iterative methods
  (Gelman, Mandel, ...)
- Derivative-free multilevel optimization
  (Mendonca, Peckman, Toint, ...)

Multilevel Methods For Combinatorial Optimization

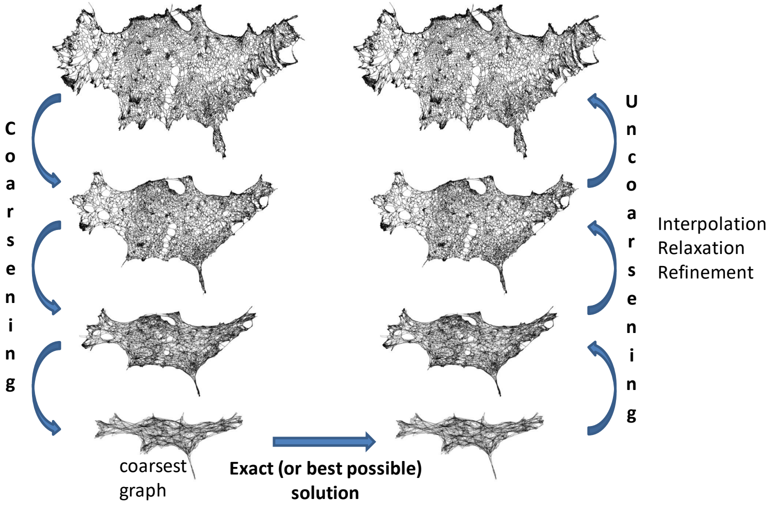
Coarsening
Create a hierarchy of restriction operators $\uparrow_i^{i+1}$ and corresponding coarse problems $P_0,\dots,P_k$



Uncoarsening
Gradually approximate solutions $S_{k-1},\dots,S_0$ by
(1) interpolation from previous level, and
(2) further refinement



Multilevel Methods For Combinatorial Optimization



Multilevel Methods For Combinatorial Optimization

- Examples: VLSI Placement, Partitioning, Minimum Linear Arrangement, Minimum Bandwidth, Clustering, TSP, Community Detection, Segmentation, Visualization, ...
- Quality: Usually exhibit superior results to other methods on practical test suites.

  Why? Because it is easy to combine the multiscale frameworks with other methods.
- Time: Usually exhibit **linear** time complexity with no hidden coefficients.
- Technical advantage: Admits parallelization. Suitable for various HPC configurations.

Multilevel Requirements

Question: Is the multilevel approach suitable for my problem, P?

Refinement Requirements:

- Refinement algorithm - Does a refinement algorithm exist?
- Can refinement algorithm handle additional restrictions caused by coarsening phase?
  - e.g., coarser graphs are weighted in GP
- For some problems, only known heuristics are based on construction rather than refinement
  - Not clear if multilevel can be applied

Multilevel Requirements

Coarsening Requirements:

- Solution in any of the coarsened spaces should induce a solution on the original space
  - current solution could be extended through all levels to a solution of the original problem
  - coarse solution should have the same cost with respect to objective function
  - goal is to find set of coarse variables that in future would interpolate their solution to fine variables

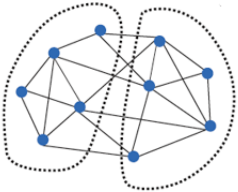
Graph Partitioning

Graph Partition Problem:

- Given $G = (V, E)$
  - $V \sim$ nodes, $E \sim$ edges
- **Goal:** Partition V into k approximately equal parts minimizing the number of cut edges between parts

Applications:

- Graph-based QMD simulations
- VLSI design
- Load balancing - minimize communication between processors
- Sparse matrix-vector multiplication - Partition rows to minimize communication
- Social networks, cyber networks, ...



Graph Partitioning

Partitioning large graphs is often an important subproblem for complexity reduction/parallelization

Research in Graph partitioning

- NP-hard: uses heuristics and approximation algorithms
- Very active area of research spanning over 50 years
- Most successful practical methods use multilevel paradigm
- Popular mutlilevel tools:
  - CHACO by Hendrickson and Leland, since 1993
  - METIS by Karypis and Kumar, since 1995
  - SCOTCH by Pellegrini, since 1996
  - JOSTLE by Walshaw, since 1995
  - KAHIP by Schulz, since 2013

Solving Optimization Problems on D-Wave 2X

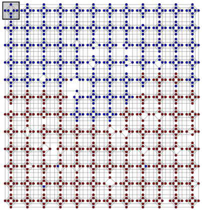
- Formulate as unconstrained quadratic integer problem

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

  - Ising formulation if $q_i \in \{-1, 1\}$
  - QUBO formulation if $q_i \in \{0, 1\}$
- Map problem onto D-Wave hardware
  - Embed graph defined by a_{ij} into D-Wave hardware (Chimera) graph

Challenges:

- Sparse connectivity of chimera graph
- Limited precision
- Max size arbitrary QUBO $\approx$ 45 variables



QUBO formlations for Graph Partitioning

Constrained formulation for 2 parts:

$$\begin{array}{ll} \text{minimize} & \mathbf{x}^T L \mathbf{x} \\ \text{subject to} & \sum x_i = n/2 \\ & x_i \in \{0,1\},\ \ i = 1, \ldots, n \end{array}$$

Unconstrained (QUBO) formulation for 2 parts:

$$\begin{array}{ll} \text{minimize} & \mathbf{x}^T L \mathbf{x} + \alpha(\sum_i x_i - n/2)^2 \\ & x_i \in \{0,1\},\ \ i = 1, ..., n \end{array}$$

$\alpha \sim$ penalty constant (balanced parts)

QUBO formlations for k-Graph Partitioning

Constrained formulation for k parts:

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{k} \mathbf{x}_j^T L \mathbf{x}_j \\ \text{subject to} & \displaystyle\sum_i x_{i,j} = n/k,\ j = 1, \ldots, k \\ & \displaystyle\sum_j x_{i,j} = 1, \quad i = 1, \ldots, n \\ & x_{i,j} \in \{0,1\}, \quad i = 1, ..., n,\ j = 1, \ldots, k \end{array}$$

Unconstrained (QUBO) formulation for k parts:

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{k} \mathbf{x}_j^T L \mathbf{x}_j + \sum_{j=1}^{k} \alpha_j (\sum_{i=1}^{n} x_{i,j} - \frac{n}{k})^2 \\ & + \sum_{i=1}^{n} \gamma_i (\sum_{j=1}^{k} x_{i,j} - 1)^2 \\ & x_{i,j} \in \{0,1\} \end{array}$$

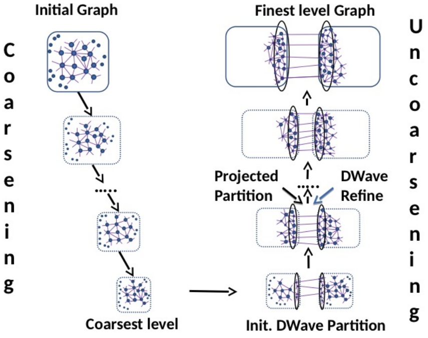
- $\alpha_j, \gamma_i \sim$ penalty constants

Multilevel Graph Partitioning with Quantum Annealing

Current work:

1. Coarsening Phase
   - Max edge weight matching
   - Algebraic Multigrid
   - Future work: coarsening with quantum device
2. Initial Partition
   - Exact solver
   - D-Wave
3. Uncoarsening/Refinement:
   - Kernighan-Lin and it's variations
   - D-Wave refinement

Multilevel Graph Partitioning with D-Wave



Multilevel Quantum Annealing for GP

D-Wave is used for

- Initial Partitioning
- Refinement

Initial Partitioning with D-Wave

Question: How good is D-Wave for initial partitioning?

Approach: We study the following,

1. Quality of partitioning **unweighted** graphs
2. Quality of partitioning **weighted** graphs with uniform volume

Initial Partitioning with D-Wave

1. **Quality of partitioning unweighted graphs:**
   - Graph data:
     - Walshaw benchmark archive (http://chriswalshaw.co.uk/partition/)
     - Molecule electronic structure graphs from QMD simulations
     - Random graph models
   - Tools:
     - SAPI, D-Wave API
     - qbsolv: hybrid method with D-Wave and tabu search
   - **Experiment**:
     - D-Wave Vs KaHIP, (solution quality)
     - D-Wave Vs METIS, (solution quality)

Initial Partitioning: k- graph partitioning

- Dense random graphs
- Using sapi for embedding and solving
- Limited to ≈ 45 node graph
- 15-node graph into 4 parts and 20-node graph into 3 parts used 900+ qubits
- Results comparable for SAPI, METIS and qbsolv
- Results using SAPI are typically equal to qbsolv

n	k	SAPI	METIS	qbsolv
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

Initial Partitioning: *k*- Graph Partitioning

- Dense random graphs
- Using qbsolv for large graphs
- Produces $kn \times kn$ QUBO
- 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**

Initial Partitioning with D-Wave

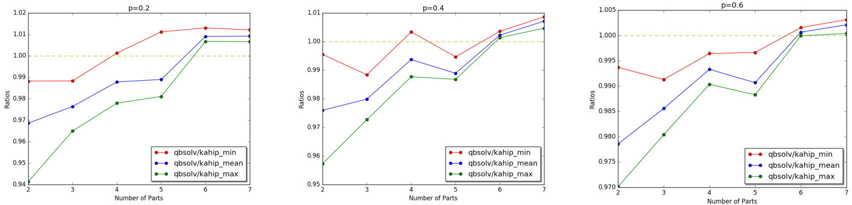
Quality of partitioning weighted graphs:

- Graph data:
  - Random graph models
    - 420 nodes
    - Vary edge probability p
    - Edge weight $\sim$ *uniform*$(1, 100)$
- Tools:
  - qbsolv
- **Experiment**:
  - D-Wave Vs KaffpaE, (solution quality)
    - Partition into $k = 2, 3, 4, 5, 6, 7$
    - KaffpaE run 20 times for each k
    - Save KaffpaE best, mean and worst cut value
    - Compare quality

Initial Partitioning: Weighted Graphs

Experiment:

- D-Wave Vs KaffpaE, (solution quality)
  - Partition into $k = 2, 3, 4, 5, 6, 7$
  - KaffpaE run 20 times for each k
  - Save KaffpaE best, mean and worst cut value
  - Compare quality



- Smaller than 1 means qbsolv was better

Conclusion: Positive results for initial partitioning

Uncoarsening Phase: Refinement

Question:

How to refine (improve) a given partition with D-Wave?

Kernighan-Lin algorithm review:

- Kernighan and Lin, "An efficient heuristic procedure for partitioning graphs,"" The Bell System Technical Journal, vol. 49, no. 2, Feb. 1970.
- An **iterative**, 2-way, balanced partitioning heuristic
- Each iteration:
  - Vertex pairs with the largest decrease or the smallest increase in cut size are exchanged
  - These vertices are then locked
  - locked vertices do not participate in any further exchanges
  - Process continues until all the vertices are locked

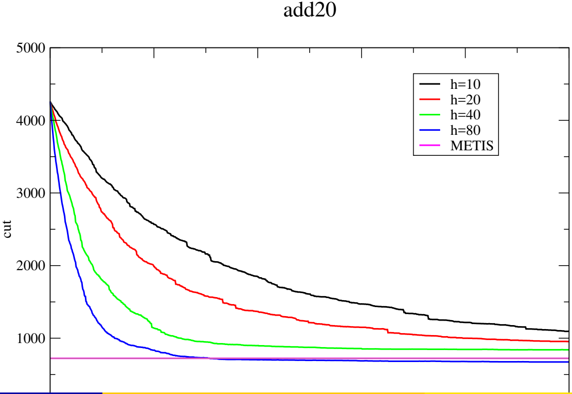
Refinement: D-Wave

KL Refinement Summary:

- At each pass, two nodes are swapped and gain function updated
- Developed for 2-way partitioning

D-Wave Refinement:

- Use D-Wave to swap set of **free nodes** $V' \subset V$ at once!

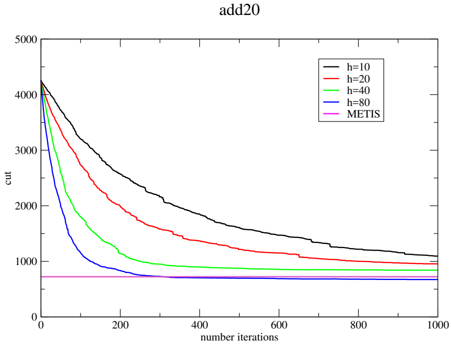


D-Wave Refinement with no Multilevel Framework

Question: How powerful can quantum annealing be for refinement?

Experiment:

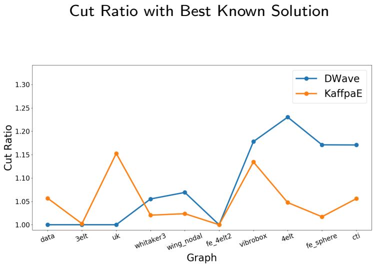
- Assume h is size of quantum annealing hardware
- Start at random solution
- Choose h nodes at random
- Optimize h nodes at each iteration (system call)
- One iteration = One system call
- $h \approx 45$ for D-Wave 2X

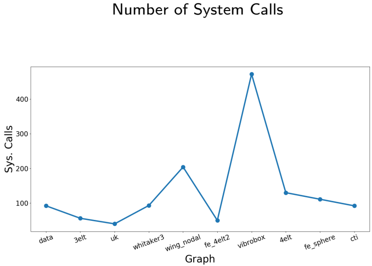


Experiments: Final Partitioning Results

- Graph data
  - Walshaw benchmark graphs with less than 20k nodes
- Experiment
  - One V-cycle D-Wave Vs One V-cycle KaHip
  - Compare with best known solution

Results: Walshaw Graphs





- Graphs between 2000 – 17000 nodes
- Achieved best known value for 3 graphs with less than 80 system calls
- Results comparable with known solvers

Summary

- Multilevel framework ideal for near-term quantum computing hardware
- D-Wave gives high quality initial partitions
- Archived best known results with for 3 graphs with less 50 systems calls on average

Future Work:

- Coarsening for GP with quantum annealing
- Improved choice of free nodes in refinement algorithm
- Quantum enhanced coarsening for other combinatorial optimization problems