ЕВРОПЕЙСКИ СЪЮЗ
ЕВРОПЕЙСКИ ФОНД ЗА
РЕГИОНАЛНО РАЗВИТИЕ

ОПЕРАТИВНА ПРОГРАМА
НАУКА И ОБРАЗОВАНИЕ ЗА
ИНТЕЛИГЕНТЕН РАСТЕЖ

# Solving large optimization problems on quantum annealing computers

**Hristo Djidjev**

October 28, 2021

ИКТ

ЦЕНТЪР ЗА ВЪРХОВИ ПОСТИЖЕНИЯ ПО
ИНФОРМАТИКА И ИНФОРМАЦИОННИ И
КОМУНИКАЦИОННИ ТЕХНОЛОГИИ

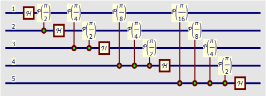
# Outline

- Brief review of quantum annealing basics
- Technique for <u>problem embedding</u> into the quantum hardware
- Optimizing the <u>spin reversal</u> transform
- Getting information about the <u>quantum annealing dynamics</u>
- Quantum annealing optimization using an <u>initial solution</u>

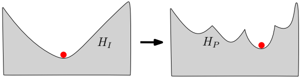

2/28

# Review: models for quantum computing

- Gate model quantum computing
  - Analog of a Turing machine, general-purpose
  - Programmed as a sequence of gates



- Adiabatic computing
  - Transition a time-dependent Hamiltonian towards one encoding the solution of the problem in its ground state
  - Theoretically equivalent to the gate-model




- Quantum annealing
  - Practical implementation of the adiabatic computing model
  - Due to very short execution times and hardware noise no optimality can be guaranteed


3/28

# Review: solving optimization problems on D-Wave

1. Reformulate the given optimization problem as:

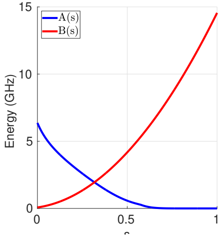
$$\min_{x_1,\ldots,x_n} \left( \sum_{i=1}^{n} a_i x_i + \sum_{1 \le i < j \le n} a_{ij}\, x_i x_j \right)$$

   - Ising formulation: $x_i \in \{-1, 1\}$
   - QUBO formulation: $x_i \in \{0, 1\}$

2. Map problem onto DW hardware
   - Embed connectivity graph defined by $a_{ij}$ coefficients into the quantum hardware
   - Encode $a_{ij}$ and $a_i$ coefficients in DW

3. Anneal, read solutions in a loop

$$\mathcal{H}(t) = A(t)\,\mathcal{H}_I + B(t)\,\mathcal{H}_P$$





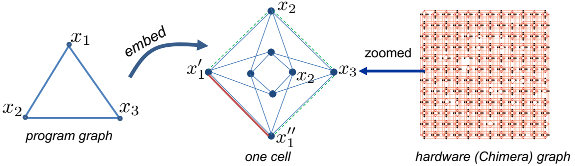
4/28

# Problem embedding techniques


5/28

# Chaining several physical qubits / broken chains

Example: $\min\{x_1 - x_2 - 3x_3 - x_1x_2 - 3x_1x_3 + 2x_2x_3\} \quad [1, 1, 1]$




Split $x_1$ into $x'_1$ and $x'_2$. Problem becomes

$$\min\{0.5(x'_1 + x''_1) - x_2 - 3x_3 + x'_1x_2 - 3x''_1x_3 + 2x_2x_3 \quad [-1, 1, -1, 1]$$

Add constraint: subject to $x'_1 = x''_1$.

Moving the constraint inside the objective, problem becomes

$$\min\{0.5(x'_1 + x''_1) - x_2 - 3x_3 + x'_1x_2 - 3x''_1x_3 + 2x_2x_3 + M(x'_1 - x''_1)^2\},$$

or

$$\min\{0.5(x'_1 + x''_1) - x_2 - 3x_3 + x'_1x_2 - 3x''_1x_3 + 2x_2x_3 - 2Mx'_1x''_1\}.$$

$M$ should be neither too small, nor too large.


6/28

# Broken chains are a common phenomenon

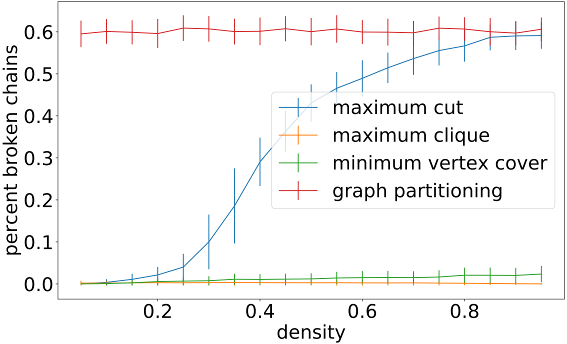



Figure 1: Proportion of broken chains for the Maximum Cut, Maximum Clique, Minimum Vertex Cover, and Graph Partitioning problems for 65 vertex graphs, as a function of the graph density using *uniform torque compensation* to calculate the chain strength for each problem.


7/28

# D-Wave's default unembedding options

- **Majority vote**: Suppose qubit $x_i$ is mapped onto the quantum chip as a chain $x_i^{(1)}, \ldots, x_i^{(m)}$. The final value of $x_i$ is set to the most common value among the $m$ chained qubits.

- **Random weighted unembedding**: Calculate the proportion (empirical frequencies) of 0's and 1's in the chain $x_i^{(1)}, \ldots, x_i^{(m)}$, and set $x_i$ to the value of a coin flip with probability set to the empirical frequencies.

- **Minimize energy**: Calculate the value of the Hamiltonian based on all unbroken chains, iteratively probe values of broken chains, and assign final values based on a priority score.



8/28

# Tailored unembedding techniques

**General idea**:

- Determine set $U$ of unbroken chains and set $B$ of broken chains.
- The solution spanned by unbroken chains having value 1 is then used as a baseline solution (e.g., an initial clique).
- Main loop: iterate over broken chains in $B$ and use context specific knowledge of the problem to determine what value the variable corresponding to the broken chain should be assigned.
- Increase efficacy by using additional information provided by the annealer. In case of a tie, we take into account the percentages of 1 and 0 (or $-1$ for Ising) assigned to the qubits of each chain.

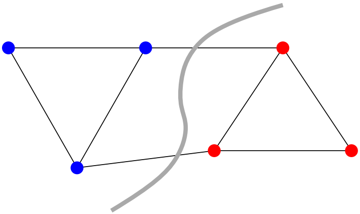

9/28

# Example: graph partitioning

**Graph Partitioning problem**: Minimize the cut of a graph, while keeping the two parts sizes balanced.
**Unembedding algorithm:**

Takes care of the balance requirement for the two parts. Construct the "blue" and "red" sets for unbroken ($U$) set only. Repeat the following two steps.

1. Pick arbitrary vertex belonging to broken chains in $B$, and allocate it to the part in which it has the lower degree.
2. If one part contains $\lceil 65/2 \rceil$ vertices, assign all remaining vertices belonging to broken chains in $B$ to the other part.




10/28

# Experiments: Graph Partitioning

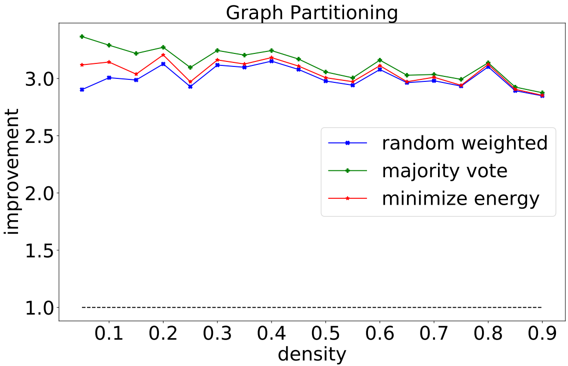



Figure 2: Graph Partitioning problem. Benchmark are the three default unembedding options (majority vote, random weighted, minimize energy) provided by D-Wave Systems, Inc.

# Optimizing the spin reversal


12/28

# The spin reversal transform

Ising problem: $Is = \sum_i h_i x_i + \sum_{i,j} J_{ij} x_i x_j \quad (x_i \in \{-1, 1\})$

**Idea**:

- Modify Ising $Is$ so that a certain variable sign gets flipped.
- Leave ground state of the Ising invariant.
- Potential to average out and reduce errors.

**To be precise**:

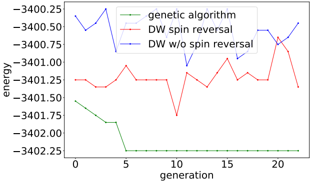
- Switch sign of $x_i$ by defining a new Ising model $Is'$ with $a_i' \to -a_i$ as well as $a_{ij} \to -a_{ij}$ and $a_{ji} \to -a_{ji}$.
- The minimum values of $Is$ and $Is'$ are equal, and a minimum of $Is'$ can be transformed into one of $Is$ by flipping $x_i$.
- Reverse set of variables and check if annealing solution better.

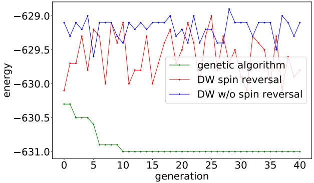
# Optimizing spin reversal

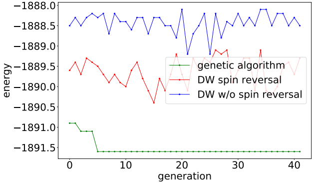
- Standard DW function: choose a random set of bits to flip.
- **How much can be gained if the set of flipped bits is optimized?**
- Search for an optimal set using a genetic algorithm:
  - Start with a random initial generation of bitstrings $S = \{010101\ldots,\ 101011\ldots,\ldots\}$;
  - Reverse qubits in $Is$ according to strings in $S$ and get the annealing results;
  - Randomly combine pairs from the lowest-energy top strings to get the new generation;
  - Flip some bits with probability the *mutation rate* parameter;
  - Repeat until halting condition met.

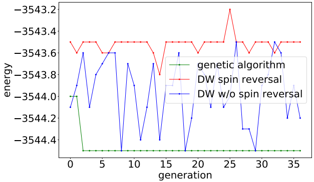

14/28

Optimization using a genetic algorithm.





15/28

# Using quenching for slicing the anneal process


16/28

# The annealing process and anneal schedule

- Input Ising problem:

$$Is = \sum_i h_i x_i + \sum_{i,j} J_{ij} x_i x_j \quad (x_i \in \{-1, 1\})$$

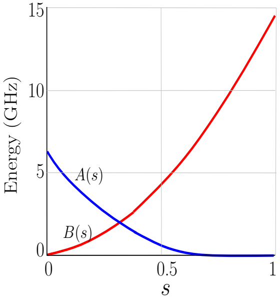
- Initial and Problem Hamiltonians:

$$H_I = \sum_i \sigma_i^x \qquad H_P = \sum_i h_i \sigma_i^z + \sum_{i,j} J_{ij} \sigma_i^z \sigma_j^z$$

- Combined Hamiltonian:

$$H(s) = A(s) H_I + B(s) H_P$$

$$s \in [0, 1]$$

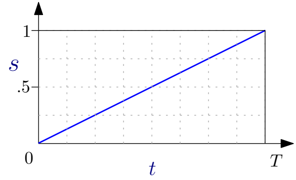




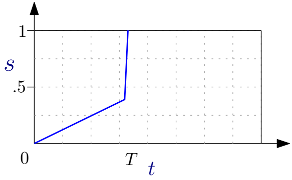
17/28

# Quenching

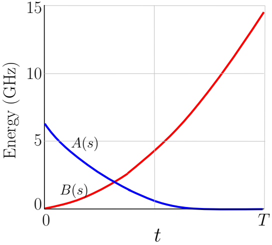
Specifying a schedule: specifying the anneal fraction function $s = s(t)$

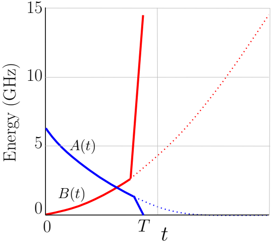



standard schedule




schedule with quenching

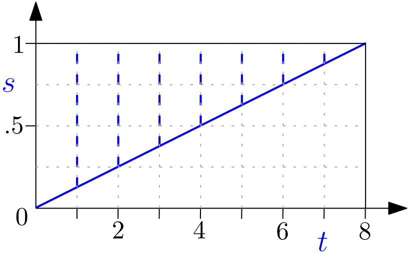


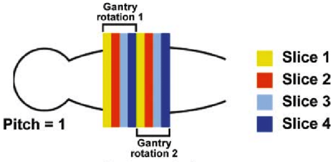




 18/28

# Using quenching to peek into the anneal process

- Idea: take "snapshots" of intermediary states




CT scan analogy

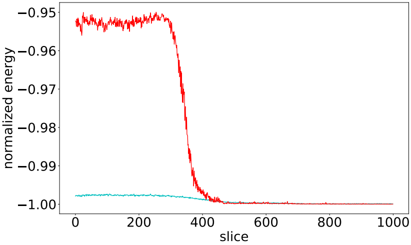
- Issue: hardware-imposed constraints on schedules
  - $s(0) = 0, s(T) = 1$
  - Bounded angle (quench cannot go completely vertical)
  - Implies that quench time is at least $\approx 1\mu s$ long

Challenge: $1\mu s$ taken by the quench may be too long (may significantly alter the ground state)


19/28

# Dealing with the problem: slow down the anneal

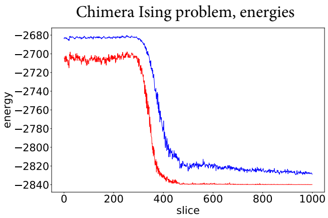
- Idea: Search for Ising/QUBOs that take much longer to anneal to near-optimality
- Implementation:
  - Use a genetic algorithm to "optimize" QUBOs
  - Fitness function: difference between $1000\mu s$ (total anneal time) and $1\mu s$ (quench time)
- Comparing slicing results for random vs optimized Ising models:

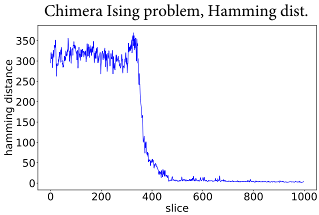


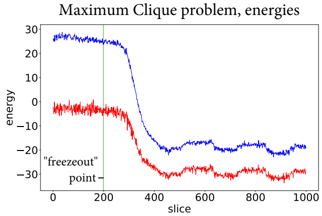


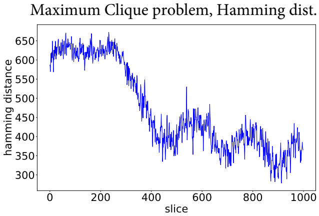
20/28

# Slicing results





21/28

# Using h-gain for planting an initial solution


22/28

# The h-gain schedule

- Hamiltonian for a regular annealing

$$H(s) = -\frac{A(s)}{2}\Big(\sum_{i=1}^{n}\hat{\sigma}_x^{(i)}\Big) + \frac{B(s)}{2}\Big(\sum_{i=1}^{n} h_i\hat{\sigma}_z^{(i)} + \sum_{i\leq j} J_{ij}\hat{\sigma}_z^{(i)}\hat{\sigma}_z^{(j)}\Big),$$

- Hamiltonian with an h-gain function, $g(t)$

$$H_{\mathrm{g}}(s) = -\frac{A(s)}{2}\Big(\sum_{i=1}^{n}\hat{\sigma}_x^{(i)}\Big) + \frac{B(s)}{2}\Big(\sum_{i=1}^{n} g(t) h_i\hat{\sigma}_z^{(i)} + \sum_{i>j} J_{ij}\hat{\sigma}_z^{(i)}\hat{\sigma}_z^{(j)}\Big),$$

- Our approach: use $g(t)$ and linear terms $h_i$ to encode an initial solution


23/28

# Encoding an initial solution

- Ising without a linear term
  - Example problems: Maximum Cut, Graph Partitioning
  - Add a linear term to encode an initial solution
    $\boldsymbol{x}^{(0)} = (x_1^0, \ldots, x_n^0)$ – initial solution
    $h(\boldsymbol{x}) = \sum_{i=1}^{n} (-x_i^0) x_i$ – corresponding linear term
  - Example: Initial solution $s = [-1, 1, 1, -1]$.
    Define linear term $x_1 - x_2 - x_3 + x_4$.
    The minimum of that linear function is $s$.
- Ising problems with a linear term
  - Add a new variable $z$ to homogenize the Ising

$$\sum_{i=1}^{n} h_i x_i + \sum_{i<j} J_{ij} x_i x_j \quad \Longrightarrow \quad \sum_{i=1}^{n} h_i x_i z + \sum_{i<j} J_{ij} x_i x_j$$

  - Ensure $z = 1$ in an optimal solution (e.g. by using a penalty)
  - Use the method for Ising problems without linear terms


24/28

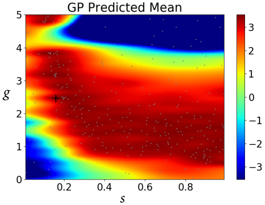
# Parameter setting procedure

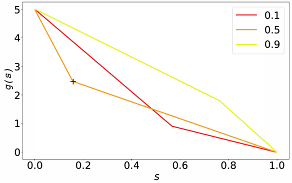
- Required parameters: annealing schedule (function $s(t)$), h-fain schedule (function $g(t)$), total anneal time $T$
- Large parameter space
- No guidelines about what shape for $g(t)$ may work
- Bayesian optimization used
  - can do black-box optimization
  - works with noisy objective functions
- Simplifications:
  - optimize for $T$ first
  - use the default $s(t) = t/T$
  - restrict $g(t)$ to have a single internal knot $(x, y)$


25/28

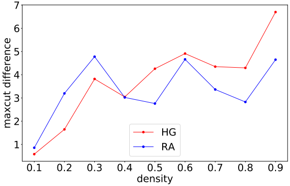
# Results

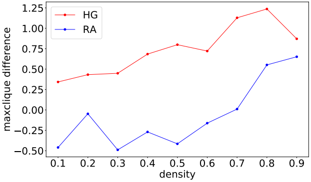
- Optimizing parameters (maximum cut)




- Comparison: h-gain vs reverse annealing





26/28

# Conclusion

- Quantum annealers can use tunable hardware parameters to improve their performance
- Finding optimal values of such parameters is itself a hard optimization problem
- Methods discussed in this talk include
  - Unembedding broken chains
  - Optimizing the spin reversal transform
  - Inferring information about the QA dynamics
  - Using h-gain for indicating an initial solution


27/28

# Thanks!


28/28