SOLVING 2-SAT INSTANCES VIA “QUANTUM” ANNEALING

Presented by: Amit Choubey, Siddhartha Santra, and Greg Quiroz
Outline

• Motivation
• The Satisfiability Problem
• Quantum Annealing and Adiabatic Quantum Computing (AQC)
• MAX-SAT Problem
• Classical Algorithms
• 2-SAT on the D-Wave chip
Motivation

- Quantum information processing (QIP) offers advantages over classical processing techniques in terms of complexity
  - Deutsh-Jozsa Algorithm
- D-Wave chip may be the first quantum processing chip
  - Is it quantum?
  - Does it outperform efficient classical solvers?
- 3-SAT problem has been studied on the D-Wave chip, but not the 2-SAT.
Satisfiability Problem (SAT)

Given a set of $N$ variables $\{x_i\}_{i=1}^{N}$, a problem instance of $K$-SAT is defined in conjunctive normal form (CNF) as

$$Q = A_1 \land A_2 \land \ldots \land A_M$$

where $K$ variables are included in each of the $M$ clauses denoted by

$$A_j = x_{j1} \lor \neg x_{j2} \lor \ldots \lor \neg x_{jK}.$$ 

The total number of possible problem instances is $\left( \binom{N}{K} \right)^M$.

The focus of our study will be the 2-SAT problem ($K=2$).
Quantum Annealing

- **Classical Annealing**
  - Utilizes temperature annealing
  - Probability of transition:
    \[ P(x'|x) \propto e^{-\beta[H(x') - H(x)]} \]
  - Energy:
    \[ H(x) = K(x) + V(x) \]

- **Quantum Annealing**
  - Fixed temperature
  - Adjusts energy parameters
  - Probability of transition:
    \[ P(x'|x) \propto \left\langle x'|\psi(x) \right\rangle, \quad \partial_t\left|\psi(x)\right\rangle = -iH(\Gamma)\left|\psi(x)\right\rangle \]
  - Energy:
    \[ H(\Gamma) = H_{potential} + \Gamma H_{kinetic} \]
Adiabatic Quantum Computation (AQC)

- Consider the QA Hamiltonian
  \[ H(t) = f_1(t)H_0 + f_2(t)H_P \]

  - Ground state: \( |\Phi_0(t)\rangle \)
  - First excited state: \( |\Phi_1(t)\rangle \)
  - Energy gap: \( \Delta(t) \)

- Computation:
  - \( H_0 \) is easily prepared
  - \( H_P \) ground state encodes the solution to the problem of interest

- Adiabatic Theorem: state evolved according to the Schrodinger equation to remains sufficiently “close” to the instantaneous ground state of \( H(t) \) provided that

\[
T > \frac{a}{q} \frac{\left( \max_s \frac{\|dH(s)/ds\|}{\min_t \Delta(t)} \right)^{-1}}{s = t/T}
\]
AQC on D-Wave Chip

• Implements AQC using superconducting flux qubits

Implementable Hamiltonian:

\[ H(t) = f_1(t) \sum_j \sigma_j^x + f_2(t) \left( \sum_j h_j^z \sigma_j^z + \sum_{i,j} J_{ij} \sigma_i^z \sigma_j^z \right) \]
Advantageous of Quantum Annealing

- Provably more efficient than classical algorithms
  - Grover’s search problem: locate a marked file in an unsorted database containing $N$ items
    - Classical run time scales as $T \propto O(N)$
    - Quantum run time scales as $T \propto O\left(\sqrt{N}\right)$
  - Perhaps more easily implementable than standard circuit model algorithms
- Possible to map classical problems to quantum Hamiltonians, e.g. Grover, 2-SAT, 3-SAT
  - Caveat: Hamiltonian may contain highly non-local interactions
MAX 2-SAT Problem Definition

**Definition 1:** Find the maximum number of satisfied clauses in a 2-SAT instance

**Definition 2:** Is there an assignment satisfying $W$ clauses in the formula

MAX 2-SAT is NP-Complete
MAX 2-SAT is polynomial time reducible to 2-SAT
3-SAT is known to be NP-Complete
A logical variable $v_j$ (TRUE or FALSE) is replaced by a corresponding integer variable $x_j$, that takes values 1 or 0. So:
- An unnegated literal is simply replaced by $x_j$
- A negated literal is replaced by the expression $1 - x_j$
- A clause is satisfied if and only if at least one of its $n_l$ literals is TRUE.
- For the integer problem, we sum the corresponding $n_l$ expressions. The clause is satisfied if and only if the sum is 1 or more.
IP Formulation

Let $Q$ be a CNF form:

$$Q = A_1 \land A_2 \land \cdots \land A_M$$

For all $i \leq M$, let $Y_i = \{j \leq n \mid x_j \in A_i\}$ and $\overline{Y} = \{j \leq n \mid \overline{x} \in A_i\}$

and $x \in \{0,1\}^n$, $y \in \{0,1\}^M$ be binary decision variables

TO:

$$\max \sum_{i \leq M} y_i$$

s.t. $\forall i \leq M \sum_{j \in Y_i} x_j + \sum_{j \in \overline{Y}_i} (1 - x_j) \geq y_i$
Classical Solver

- Based on IP formulation of MAX-2-SAT

- Branch and Bound algorithm used
  - Represent the space of all possible assignments for a CNF as a search tree
  - Internal nodes represent partial assignments
  - Leaf nodes represent complete assignments
  - Searches the tree in DFS (Pruned searching)
Random Clause Generation

- Number of variables: $N=32$ or 67
- $M=$ number of clauses in formula
- Clauses are randomly generated but are chip compatible
- $\alpha = \frac{N}{M}$ varied from 0.1 to 2 with step size 0.1

More about this from Sid
Results from Classical Solvers

![Graph 1: Average # of UNSAT CLAUSES vs $\alpha$](image1.png)

- $N=32$
- $N=67$

![Graph 2: Probability of SAT vs $\alpha$](image2.png)

- $N=32$
- $N=67$
Problem Statement

1. Map 2-SAT problems to the chip OR find what ensemble can be represented
2. Plot probability of a random instance being satisfied as a function of clause density
   • Random 2-SAT (solved exactly) – NA
   • Random over chip solvable instances (exact) ✔
   • Chip instances (solved w/ D-Wave) – in progress
   • Min number of violated clauses – exact vs chip (using local solver) ✔
Block Diagram of the Project

Generate chip compatible random problem instances

Convert each instance to their Ising equivalents

Call local solvers to solve the ensemble at fixed $\alpha$

Use state-of-the-art classical solvers to solve the same ensemble

Actual chip implementation – in progress
Restricted Random Ensemble

- Limited connectivity on chip: in all 108 working bits, 255 couplings
  - 32 working bits
  - 67 working bits
- Features of the program
  - Uniformly random negation
  - No repetition of clauses
Converting to Ising Equivalents and Hardware Implementation - Issues

• Accuracy of coupling, local fields:
  • h,J range on chip is limited, params.auto_scale
  • h,J precision limited, params.num_programming_cycles->increase mitigates hardware programming errors
  • For each programming cycle params.num_reads is the number of times the annealing process takes place – on chip this is supposed to yield a histogram, on local solvers it gives ‘num_reads’ number of lowest energy states

• What to expect? Least count of h,J come into play for large problem sizes?
More Hardware Parameters

- **annealing_time**: positive integer [5,20000] micro secs
- **Programming_thermalization**: wait time after programming to thermalize [0,10^7] micro secs
- **Readout_thermalization**: wait time after readout to thermalize [0,50000]
- **Auto_scale**

**Solver Choice**
- R4_7_C4_Zen2103_19091607_17_A7_C8R1-sw_optimize
- R4_7_C4_Zen2103_19091607_17_A7_C8R1-sw_sample
- c4-sw_optimize
- c4-sw_sample
- chimera
Prob of SAT and Avg # of UNSAT clauses for N=32 ensemble
Big Questions

• Phase transition in chip solvable instances?
  • Yes, at least according to local solvers :)”

• Is the chip good at counting max. # of solutions/min. # of violated constraints?
  • Still to see

• Time wise, will it outperform?
  • Even if yes, then on a random ensemble specifically tailored to its design