Maximum 3SAT Decomposition

Jonah Sachs

February 2024

1 Problem Formulation

Maximum 3-Sat is a derivative of the satisfaction problem. A set of clauses is presented; each clause contains a set of binary variables separated by OR operations. A clause can be defined as follows:

$$C_i = x_1 \vee x_2 \vee ... \vee x_n \quad x_i \in \{0, 1\} \tag{1}$$

The 3SAT problem specifies that each clause contains exactly three variables. In addition, all satisfaction problem setups involve an applied AND operation between each individual clause. 3SAT demands we return a solution that fulfills every declared clause. Maximum 3SAT has a looser requirement; the returned solution is that which fulfills the most clauses. With this setup considered, we can define the amount of clauses satisfied as N and 3SAT as:

$$N = C_1 \wedge C_2 \wedge ... \wedge C_n \tag{2}$$

2 Logic to Algebra

The goal of this project is to decompose a classically understood optimization problem into the QUBO or Ising formulation. These forms allow for processing on quantum systems through either Adiabatic Quantum Computers or Gate-Based Quantum Systems. We will be considering a specific instance of the Maximum 3SAT problem, given below. The form involves 4 clauses with 3 terms each, and is a valid instance of the Maximum 3SAT problem.

$$N = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \tag{3}$$

Our goal is to decompose these logic-based expressions into binary arithmetic. We employ the following expressions in order to transform a 3-way OR expression and a typical not gate.

$$\neg x_i = (1 - x_i) \qquad (x_1 \vee x_2 \vee x_3) = x_1 + x_2 + x_3 - x_1 x_2 - x_2 x_3 - x_1 x_3 + x_1 x_2 x_3 \tag{4}$$

Both can be exhaustively proved for binary variables by examining all possible cases of { 0,1 }. The second expression is much more complicated, and very closely resembles the inclusion-exclusion principle from mathematics. The OR formulation was taken from other examples decomposing satisfaction problems into QUBO.

QUBO formulations require quadratic terms for valid solutions to be produced. This can be proved conceptually, but was also verified by attempting to solve the Maximum 3SAT using the above equations. A simplification is required for the last term of our inclusion-exclusion relation since it possesses degree 3. We can use a trick employed in other QUBO decompositions: declaring extra binary variables. We create the following definition which can also be proved equivalent exhaustively.

$$w_i \in \{0,1\} \quad x_1x_2x_3 = \max\ w_i(x_1 + x_2 + x_3 - 2) \tag{5}$$

Via some algebraic simplification, we are now left with (6) for any given clause. The max expression in (5) is absorbed into the overall equation since we are seeking the overall maxima. This term represents the truth of a specific clause. The term is 1 if the conditions of the clause are met and 0 otherwise. This clause is for the base expression involving no NOT operations. In order to apply a NOT, we can use the NOT expression given in (4).

$$C_i = (1 + w_i)(x_1 + x_2 + x_3) - x_1x_2 - x_1x_3 - x_2x_3 - 2w_i \tag{6}$$

Our goal is still to maximize the amount of satisfied clauses. Given the above truth expression, Maximum 3SAT decomposes to the following:

$$N = \sum C_i \qquad \max(N) \tag{7}$$

Each clause can be simplified given the presence of NOT operations. The calculations will provided externally. The final expression for N is as follows:

$$\begin{aligned} N = \sum C_i = 0x_1 + 2x_1x_2 - 2x_1x_3 + x_1w_1 - x_1w_2 + x_1w_3 - x_1w4 \\ -x_2 + 0x_2x_3 + x_2w_1 + x_2w_2 - x_2w_3 + x_2w_4 + x_3 \\ +x_3w_1 + x_3w_2 + x_3w_3 - x_3w_4 - 2w_1 - w_2 - w_3 + 0w_4 \end{aligned} \tag{8}$$

3 QUBO Matrix

In order to solve computationally, we need to provide a QUBO Matrix. This can be solved with either adiabatic or gate-based quantum systems. In a physical context, this can be understood as the Hamiltonian or as the Total Energy of the System. Given our declared binary variables, each row/column will represent a given variable. The entry of each point of the matrix represents the overlap term between the two binary variables. It is clear why this matrix will appear symmetric. Because of its symmetry, we only consider the upper triangular matrix and can declare all lower triangular entries null. Any constant is dropped considering it will have no efffect on the optimization problem. If neccessary, it can be added back later.

Additionally, QUBO problems rely on minimization in order to achieve accurate results. It is easy to transform the our Maximum 3SAT problem into the equivalent relation.

$$\min(N) = -\max(N) \tag{9}$$

Converting our factors accordingly, we come to the following expression for our QUBO matrix. It is a 7x7 matrix because we have 3 clauses, 4 terms, and 7 total binary variables. This matrix is equivalent to our original chosen instance of maximum 3SAT.

$$Q = \begin{bmatrix} 0 & -2 & 2 & -1 & 1 & -1 & 1 \\ 0 & 1 & 0 & -1 & -1 & 1 & -1 \\ 0 & 0 & -1 & -1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{10}$$

4 Classical Solution Set

The solution set to our chosen instance of Maximum 3SAT can be found classically by exhaustively choosing all possible combinations. This will be useful for determining whether our results using quantum systems are accurate. We know that the following states are solutions to the problem:

$$\{x_1, x_2, x_3\} = \{0,0,1\}, \{0,1,1\}, \{1,1,0\}, \{1,1,1\}, \tag{11}$$

5 Adiabatic Quantum Computers and DWave

Adiabatic Quantum Computers are a field separated from gate-based quantum computers. The evolution of the state within an adiabatic system is continuous as opposed to discrete. We start with a Hamiltonian which is easy to construct. If we evolve the state "slow enough," via the adiabatic theorem, we know that any state of the initial Hamiltonian must also be a state of the evolved Hamiltonian. Given a properly chosen adiabatic path, we can guide our initial Hamiltonian to our problem set solution without collapsing/destroying the state. These types of systems are not universal and are specifically designed to solve optimization problems.

DWave is a company that provides these types of quantum systems for solving optimization problems. DWave's current generation of quantum systems contain about 5000 qubits, much greater than any gate-based quantum computer. The binary quadratic matrix solver (BQM) was selected due to the

	0	1	2	3	4	5	6	energy	num_oc.	chain_b.
0	1	1	0	0	0	0	1	−1.0	74	0.0
1	0	0	1	0	0	0	0	−1.0	66	0.0
2	1	1	1	1	1	1	0	−1.0	56	0.0
3	1	1	1	1	0	0	0	−1.0	29	0.0
4	1	1	1	1	0	1	0	−1.0	38	0.0
5	0	1	1	0	1	0	1	−1.0	64	0.0
6	1	1	0	1	0	0	1	−1.0	71	0.0
7	0	1	1	0	1	0	0	−1.0	58	0.0
8	0	0	1	0	0	1	0	−1.0	63	0.0
9	1	1	1	1	1	0	0	−1.0	53	0.0
10	0	0	1	0	1	1	0	−1.0	99	0.0
11	0	1	1	1	1	0	0	−1.0	80	0.0
12	0	0	1	0	1	0	0	−1.0	93	0.0
13	1	1	0	1	0	0	0	−1.0	38	0.0
14	1	1	0	0	0	0	0	−1.0	43	0.0
15	0	1	1	1	1	0	1	−1.0	73	0.0
17	1	1	0	0	0	0	1	−1.0	1	0.142857
16	0	0	0	0	0	0	0	0.0	1	0.0

Figure 1: Output for 1000 shots for a Maximum 3SAT problem on DWave's BQM solver

problem structure. Other solvers are available for optimization problems of different structure. Embedding must be used in order to map our logical qubits to physical qubits. In other words, the embedder maps our QUBO matrix to the topology of DWave's Quantum Processing Unit (QPU).

The results below are from our QUBO problem when formulated with Dwave's BQM solver. We see high frequency for all expected solution values. Noise is also apparent through small instances of incorrect solutions. All values of -1 energy indicate a minimum and solution to the problem. These actually represent a minimum value of -4, for the case that all 4 clauses are satisfied. We removed a constant of 3 previously, so this discrepancy make sense. Overall, the results suggest that this formulation is correct. A previous attempt using 3rd order terms did not return a similar solution set. More than one case exists for any state given the introduction of additional binary variables. num_oc represents the amount of occurrences of any given state. A table of the results, given 1000 shots, are provided below.