Solving NP-complete Puzzles with Quantum Annealing

Collin Wilson, Haoping Xu, Yongtao Zhan, Yousuf Aborahama

Abstract

Several popular puzzles are known to be NP-complete. In this paper, we formulate four NP-complete puzzles, Light Up, Numberlink, Masyu and Instant Insanity as quadratic unconstrained binary optimization problems (QUBO) and solve them with the D-Wave 2000Q quantum computer as well as a classical solver. Results of the quantum and classical solvers are compared and limitations are addressed. (Source code available at https://github.com/Jack-XHP/QuantumPuzzle/tree/master)

1 Introduction

Puzzles such as Sudoku are immensely popular pastimes. Most newspapers include puzzle sections and these pencil-and-paper games are ubiquitous on store bookshelves and magazine racks. However, fascination with puzzles and games extends beyond viewing them as simple pastimes with much published research examining the complexity of various puzzles [1–7].

Efficiently solving NP-complete problems is a main area of research today. Quantum computing, whose supremacy over classical computing is still controversial, is an often considered method for speeding up computation of NP-complete problems [8, 9]. One of the near-term quantum computing solutions for these problems, D-Wave 2000Q, may provide a computational benefit and many NP-hard problems have been mapped to formats compatible with D-Wave hardware [10, 11]. In this paper we map the NP-complete puzzles Light Up, Numberlink, Masyu and Instant Insanity to quadratic unconstrained binary optimization problems and solve a few examples using D-Wave 2000Q and compare solutions to those given by a classical simulator.

1.1 Light Up

Light Up, also called Akari, is a pencil-and-paper game that originated in Japan. The board consists of a grid of black and white cells with some black cells labelled with numbers 0 to 4. An example of which is shown in Fig.1a. Placing a bulbs in the white cells on the board lights up the entire row and column of the chosen cell unless there is a black cell in the way. The goal is to place bulbs until the entire grid is lit up under the following conditions:

1. No two light bulbs shine on each other.
2. A labelled black cell must have the indicated number of bulbs adjacent to its four sides. Unnumbered cells may have any number of bulbs adjacent to it.

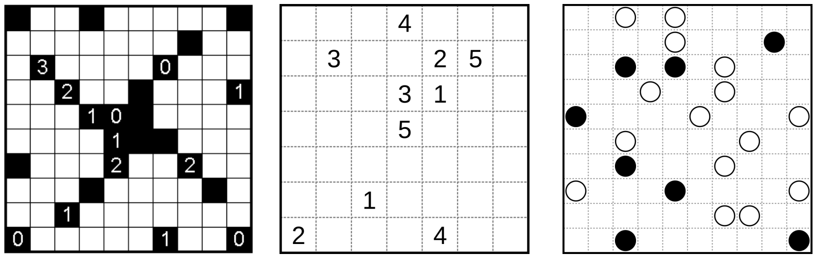
An online playable online version can be found here [12].

A polynomial reduction from Circuit-SAT to Light Up has been demonstrated [4] so the Light Up puzzle is NP-complete.

1.2 Numberlink

Numberlink is another popular logic puzzle. Its setup involves a grid with $N \times M$ size, in which several pairs of different numbers are placed in each cell (See Fig.1b). The main objective is to link each number pair with one continuous link along the grid cells following four rules [13]:

1. Identical numbers should be connected with a continuous path.
2. Paths must go through the center of a cell horizontally or vertically and never go through the same cell twice.
3. Paths cannot cross, branch off, or go through other numbered cells.



(a) Light Up (b) Numberlink (c) Masyu

Figure 1: An example of the puzzles Akari, Numberlink, and Masyu.

Solving Numberlink can be complicated. In fact, existing literature shows that the Numberlink puzzle is NP-complete by reformatting 3-SAT problem to the Numberlink problem [1]. Due to its computational complexity, traditional solver can only handle small scale grids and needs exponential computation time.

1.3 Masyu

The Masyu puzzle, also known as Evil Influence, consists of a $M \times N$ grid, and certain amount of white and black circles placed in grid cells (See Fig.1c). The goal of the puzzle is to connect all circles using a single continuous non-intersecting loop. We follow a modified version of the original Masyu rules:

1. Draw a single loop that passes through all cells containing white and black dots that never crosses itself, branches or goes though the same cell twice.
2. Lines passing through white circles must pass straight through the cell.
3. Lines passing through black circles must turn ninety degrees in the cell.

The Masyu puzzle belongs to a class of puzzles called Pearl puzzles which are NP-complete [14].

1.4 Instant Insanity

The Instant Insanity puzzle is a cube-stacking game which consists of n cubes where each face is coloured with one of n colours. The goal is to stack the cubes such that each of the four vertical faces of the stack contain all n colours. A typical Instant Insanity puzzle has $n = 4$, but the generalized puzzle has been shown to be NP-complete [7].

2 Related Work

Quadratic unconstrained binary optimization (QUBO) has grown in importance in the field of combinatorial optimization, many existing works are applying it to solve a wide range of problems [15]. A QUBO problem is defined by minimizing a weighted sum of all quadratic terms among n binary variables $x_i \in \{0, 1\}$.

$$\min_X X^T Q X \qquad \text{s.t. } X \in \{0, 1\}^n.$$

Since it is closely related to Ising problems in physics, quantum annealing, one area of quantum computing, uses QUBO as one of its problem formats and adapts $\{0, 1\}^n$ as the search space [16]. Quantum annealing is a process for discovering the global minimum of a given multidimensional quadratic function, the method achieves that via starting with a superposition of all possible states and slowly cool down to the ground state with the lowest energy [17]. The quantum annealing process is expected to find the minimal energy with certain probability given the problem setup and device's parameters, especially the QPU's noise level [18].

Given the promise of quantum speedup, for a wide ranges of problems, their QUBO formats have been studied and many existing works demonstrate the potential of solving them via quantum annealer. One popular topic in this area is solving NP-hard problems, as these problems require exponential amount of time to run on classical computers, any possible quantum speedup will be a large breakthrough. A list of famous NP-hard problems are solved by quantum annealing, namely, number partitioning[19], max 2-SAT [20], max independent set [21], max cut [22], and graph coloring [23]. Meanwhile, some pieces of literature demonstrate the possibility to rewrite famous puzzles, which are known to be NP-complete, into QUBO formats and solve them using a classical simulator or a actual quantum annealer[24, 25]. We want to build our solver based on these previous works and solve other NP-complete puzzles on quantum annealer using QUBO format.

3 Methods

In this section, we deal with the QUBO formulation of each puzzle one at a time. For each puzzle we will discuss first the qubits assignment, and then move to the QUBO coefficients where we divide the problem into different terms that deals with both the constraints and the winning condition.

3.1 Light Up

Given a Light Up puzzle on an $N \times M$ grid, we are going to assign a qubit to each white cell on the grid where the qubit $q_{i,j}$ is assigned to the ij-th cell (i.e., on the i-th row and j-th column) such that $q_{i,j} = 1$ when a bulb is placed there and $q_{i,j} = 0$ otherwise. Let q be the collection of all such qubits and q_i be the i-th element in q. Clearly, once the values of these qubits are found, a solution for the Light Up puzzle is obtained.

After describing the qubits assignment we move to the components of the QUBO Hamiltonian which are:

1. *Avoid having two bulbs shinning on each other while lighting up the entire grid:*

   We can convert the first rule of the Light Up puzzle to the problem of finding the maximum independent set. We can construct an undirected graph $G = (V, E)$ for a Light Up puzzle. Each white cell is a vertex and two vertices are connected if the two white cells they represented can "see" each other, (i.e., are in the same row or column without being blocked by black cells). The maximum independent set of this graph is equivalent to the placement of as many light bulbs as possible without shining each other.

   So to convert the independent set problem to QUBO we start by considering V_0 that is a subset of the vertices (none of which is connected) of the undirected graph $G = (V, E)$. This subset is called maximum if no other independent set is larger than it. A scheme for converting the problem of finding the maximum independent set to a QUBO problem is described in Ref.[26]. Here we briefly review this scheme. For each vertex $v_i \in V$, we assign a variable q_i to it and construct a $|V| \times |V|$ matrix Q that

   $$Q_{ij} = \begin{cases} 2, & \text{if } (v_i, v_j) \in E, \text{ (i.e., } v_i \text{ and } v_j \text{ are connected).} \\ -1, & \text{if } (v_i, v_j) \notin E, \text{ (i.e., } v_i \text{ and } v_j \text{ are not connected).} \\ 0, & \text{if } i = j \end{cases}$$

   The first Hamiltonian term $H_1 = q^T Q q$ is minimized if $V' = \{v_i | q_i = 1\}$ is the maximum independent set. If there was a larger independent set, then H_1 could be made smaller since the contribution of a pair of nonadjacent vertices to Q is -1. Conversely, if V' is a maximum independent set, then H_1 cannot be made smaller.

2. *Enforce the number of light bulbs adjacent to a black cell to match the cell number:*

   Lets consider the situation where the ij-th cell is a black cell labeled by the number n, then we just have to minimize the term $H_{B_{ij}} = (q_{i+1,j} + q_{i,j-1} + q_{i-1,j} + q_{i,j+1} - n)^2$ which can ensure that there is only n adjacent bulbs. This lead to the following term in the Hamiltonian

   $$H_2 = \sum_{(i,j) \in \mathcal{B}} (q_{i+1,j} + q_{i,j-1} + q_{i-1,j} + q_{i,j+1} - n_{i,j})^2, \tag{1}$$

where $\mathcal{B}$ is the set of ordered pair that represent the coordinates of numbered black cells. Finally, the complete QUBO Hamiltonian is a linear combination of the above two terms, which can be written as:

$$H = H_1 + \lambda H_2. \tag{2}$$

Note that λ is typically a large positive constant that ensures that the second term is identical to zero which implies that the imposed constraints is satisfied.

3.2 Numberlink

Now, lets consider a Numberlink puzzle on an $N \times M$ grid and K pairs of numbers. To be able to easily encode the puzzle as a QUBO problem we will represent each of these number as a vector of qubits where only one of the entry of the vector is assigned the value one. This will result in splitting the Numberlink grid into K grids where each grid now correspond to a different pair of numbers. After that, each cell in the k-th grid is assigned a qubit q_{kij} where $q_{kij} = 1$ if the ij-th cell is in the path of the link between the k-th pair of numbers, and $q_{kij} = 0$ otherwise. This can be easily understood from Fig. 2 where the solved puzzle presented in Fig. 2(a) is decomposed into two grids. Figure 2(b) shows the qubits related to the number 1 while Fig. 2(c) shows the ones associated with the number 2.

After describing the qubits assignment we can move to the terms needed to construct the Hamiltonian associated with the Numberlink puzzle as follows:

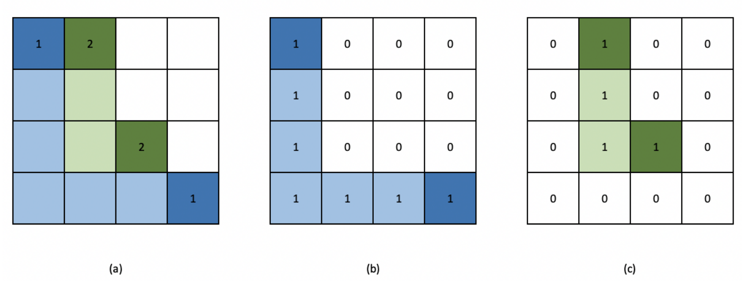


Figure 2: (a) sample of Numberlink with one correct solution (b) qubits assignments corresponding to the solution of layer 1 (c) qubits assignments for layer 2

1. *Enforce the solver to maintain the original puzzle:*

   Since each one of the original pairs of numbers is assigned to a vector of qubits, we need their cells to maintain the original number, otherwise, the solver will change the puzzle during the solving process. This leads to having the first term in the Hamiltonian as

$$H_1 = -\sum_{(k,i,j)\in\mathcal{O}} q_{kij}^2, \tag{3}$$

   where $\mathcal{O}$ is the set of indices of the numbers appearing in the given Numberlink puzzle.

2. *Propagating the link between similar pairs:*

   The constraint needed to force the solver to create a link between similar pairs can converted into two parts. Each of the qubits assigned to the numbers in the puzzle must have only one adjacent non-zero qubit, while all other qubits are either zero or one and have two adjacent non-zero qubits. This constraint can be encoded as

$$H_2 = \sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(\sum_{s\in\mathcal{D}_{kij}} q_s - 2 + f(q_{kij})\right)^2 q_{kij}^{1-f(q_{kij})}, \tag{4}$$

   where $\mathcal{D}_{kij}$ is the set of adjacent indices to (k, i, j), and $f(q_{kij}) = \mathbb{1}((k, i, j) \in \mathcal{O})$.

   However, such a formula does not comply with QUBO format when q_{kij} is not an original number in the grid as there will be cubic terms in $(q_1 + q_2 + q_3 + q_4 - 2)^2 q_{kij}$, while QUBO only consist of quadratic terms. Each of the cubic terms can be handled by introducing an additional auxiliary binary variable z and preform a reduction by substitution, where in QUBO, minimizing the three qubit expression $q_1q_2q_{kij}$ is equivalent to minimizing the four qubit expression $zq_{kij} + U(q_1q_2 - 2(q_1 + q_2)z + 3z$ where $U > 1$ [27].

3. *Prevent paths crossing:*

   The constraint needed to prevent the paths of different numbers in a Numberlink puzzle from crossing can be achieved by allowing only one of the grids to have a non-zero qubit for a given cell which will in turn associate that cell with a single path only. In other words, the value of all qubits assigned to the ij-th cell must add up to one. Similar to H_2, we can write the constrain in QUBO format as:

$$H_3 = \sum_{i=1}^{N}\sum_{j=1}^{M}\left(\sum_{k=1}^{K} q_{kij} - 1\right)^2 \tag{5}$$

Finally, the complete QUBO Hamiltonian is just a linear combination of the previous "sub-Hamiltonians" mentioned above as

$$H = \sum_{i=1}^{3}\lambda_i H_i, \tag{6}$$

where λ_i's are positive constants that determines the trade-off among the three formulae, which are tuned during the simulation to achieve better solutions.

3.3 Masyu

Given a Masyu puzzle with a $M \times N$ grid and a set of b black circles denoted by B and a set of w white circlesdenoted by W, one can visualize the cells as vertices in a graph $G = (V, E)$ and the problem becomes finding a cycle within the graph that satisfied the previously described constraints. To tackle the issue of having a single cycle that passes through all the spots, the problem is converted into finding a map $F : I \to V$ where $I = \{0, 1, \ldots, R-1\}$ and R is the length of the cycle. The map F should be injective and enforce a cycle to satisfy the puzzle constraints. Since R is unknown one need to repeat the process at most MN times since $(b + w) \leq R \leq MN$. Although one can define such a map as an indexing for the chosen subset of V which can be done in $O(MN \log(MN))$ qubits where each qubit acts as a digit. However, reaching such limit is quit difficult since we are restricted by the problem form as well. Nevertheless, one can easily define the whole map using MNR qubits. This is done by defining $q_{i,j}$ to represent that $i \in I$ is mapped to vertex j in the graph. Similar assignment mechanism was used in [28] to find Hamiltonian cycles within a graph. With such representation, the remaining constraints can be formulated in a QUBO form as follows:

1. *F is a valid map:*

   Each element in I is mapped to exactly one element in V which can be achieved by forcing each element in I to be related to a single element in V as follows:

$$H_1 = \sum_{i=1}^{R} \left(\sum_{j=1}^{MN} q_{i,j} - 1 \right)^2 . \tag{7}$$

2. *F is injective:*

   For each pair of elements $i_1, i_2 \in I$, $F(i_1) = F(i_2) \implies i_1 = i_2$, which can be achieved by allowing each element in V to be related to at most one element in I as follows

$$H_2 = \sum_{j=1}^{MN} \left(\sum_{i=1}^{R} q_{i,j} - 0.5 \right)^2 . \tag{8}$$

3. *The sequence defined by F is a cycle:*

   To achieve this we need to have $F(i), F(i+1)$ to be adjacent vertices where $F(R) = F(0)$, which can be done by penalising the consecutivity of non-adjacent vertices in the sequence as follows

$$H_3 = \sum_{(j_1, j_2) \in V \times V - E} q_{i,j_1} q_{i+1,j_2}, \tag{9}$$

   where $q_{R,k} = q_{0,k}, \forall k$.

4. *The path through a white circle must be straight:*

   This condition can be restated as: If the k-th white circle w_k is the image of $i \in I$ then either $f(i-1) = w_k \pm 1$ and $f(i+1) = w_k \mp 1$ or $f(i-1) = w_k \pm N$ and $f(i+1) = w_k \mp N$ which can be encoded as a part of the Hamiltonian as follows:

$$H_4 = \sum_{i=1}^{R} \sum_{j \in W} C_{i,j}, \tag{10}$$

   were $C_{i,j} = q_{i-1,j-1} q_{i+1,j+1} + q_{i+1,j-1} q_{i-1,j+1} + q_{i-1,j-n} q_{i+1,j+n} + q_{i-1,j+n} q_{i+1,j-n}$.

5. *The path through a black circle must follow a right angle:*

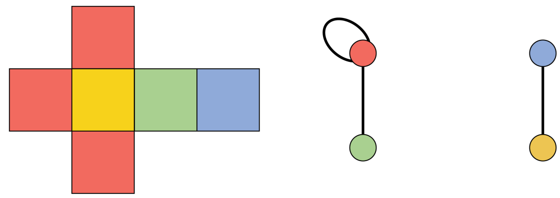
   This condition can be restated as: If the k-th black circle b_k is the image of $i \in I$ then

$$f(i-1) = b_k - N \text{ and } f(i+1) = b_k \pm 1 \text{ or}$$

$$f(i-1) = b_k + N \text{ and } f(i+1) = b_k \pm 1 \text{ or}$$

$$f(i-1) = b_k - 1 \text{ and } f(i+1) = b_k \pm N \text{ or}$$

$$f(i-1) = b_k + 1 \text{ and } f(i+1) = b_k \pm N$$



(a) (b)

Figure 3: (a) Unfolded cube. (b) The graph representation of the cube shown in (a).

Furthermore, unlike the case of white circles an extra term is needed here to guarantee that the path go through the black spots since this condition can be satisfied without it. This leads to the following term in the Hamiltonian

$$H_5 = \sum_{i=1}^{R} \sum_{j \in B} C_{i,j} + \sum_{j \in B} \left(\sum_{i=1}^{R} q_{i,j} - 1 \right)^2 \tag{11}$$

were $C_{i,j} = (q_{i-1,j-N} + q_{i-1,j+N})(q_{i+1,j-1} + q_{i+1,j+1}) + (q_{i-1,j-1} + q_{i-1,j+1})(q_{i+1,j-N} + q_{i+1,j+N})$.

Finally, One can obtain the final Hamiltonian by summing over all previously mentioned terms as

$$H = \sum_{i=1}^{5} H_i \tag{12}$$

3.4 Instant Insanity

To solve this puzzle, it is convenient to create a graph formulation of the problem. With the cubes labelled from 1 to n, construct a graph for each cube by letting the set of n colours be the vertices, V, and then for each pair of opposing faces in the cube, draw an edge between the appropriate colours in V. An example is shown in Fig.3.

A complete problem graph is formed by taking the set of vertices V and adding the edges from each cube, labelled by the cube label (See Fig.8e). The problem has a solution if two subgraphs, $A = (V_A, E_A)$ and $B = (V_B, E_B)$ can be identified from the resulting graph such that

1. $V_A = V$ and $V_B = V$, (i.e. both subgraphs contain all n colour vertices).
2. Each vertex is of degree two.
3. The edges of each subgraph are labelled by each cube exactly once.
4. The sets of edges E_A and E_B are disjoint (i.e. the subgraphs have no edges in common).

These requirements are made clearer by noting that the subgraphs A and B correspond to the opposing faces of the stack of cubes. Each face of the stack has to have all n colours. Each cube can only be in the stack once and if two faces of the cube are on the front and back of the stack, they cannot also be on the left and right faces of the stack and vice versa.

Given an Instant Insanity puzzle with n cubes and n colours, for each cube graph $G_i = (V_i, E_i)$, two qubits are assigned to each vertex and each edge. So each cube has six edge qubits and $2n$ vertex qubits. $12n^2$ total qubits are needed for the problem. The first qubit assigned to an edge or node is associated with a subgraph A and the second with subgraph B. Each qubit is assigned one if its associated edge/vertex is in its associated subgraph, and zero otherwise.

The indexing of qubits is arranged so that even indexed qubits are the subgraph A decision variables and odd indices are the subgraph B decision variables. In the formulation given below, the qubits are arranged in $6 + 2n$ columns ans n rows, one for each cube, so q_{ij} refers to qubit j from cube i.

After describing the qubits assignment we move to the components of the QUBO Hamiltonian which are:

1. *Edge inclusion implies vertex inclusion:*

   If an edge, e from a cube is included in subgraph $S \in \{A.B\}$ then the colour vertices linked by e must be included in S. Denote the edge represented by qubit $q_{i,j}$ as $e_{i,j}$ and the index set of colour vertices linked by edge $e_{i,j}$ as $I_{e_{i,j}}$. The $I_{e_{i,j}}$'s are defined by the colouration of the cubes in the puzzle. Inclusion is enforced a negative coefficient.

$$H_1 = -\sum_{i=1}^{n}\sum_{j=0}^{5}\sum_{k\in I_{e_{i,j}}} q_{i,j}q_{i,k}. \tag{13}$$

2. *Each edge can appear in only one subgraph:*

   Recalling that even indices are associated with subgraph A. The constraint is given by penalizing, with a large coefficient, the inclusion of both even/odd pairs.

$$H_2 = \sum_{i=1}^{n}\sum_{\substack{j=0\\ j\text{ even}}}^{5} q_{i,j}q_{i,j+1}. \tag{14}$$

3. *The edges of each subgraph are labelled by each cube exactly once:*

   This is enforced by ensuring that the sum over even/odd indices within a cube is equal to one.

$$H_3 = \sum_{i=1}^{n}\left(\left(\sum_{\substack{j=0\\ j\text{ even}}}^{5} q_{i,j} - 1\right)^2 + \left(\sum_{\substack{k=0\\ k\text{ odd}}}^{5} q_{i,k} - 1\right)^2\right). \tag{15}$$

4. *Each subgraph contains all n colours (vertices) and each vertex has degree 2:*

   If this condition is satisfied, then the sum over the qubits for a given colour and subgraph should be two. The condition is enforced by penalizing assignments that don't satisfy this condition.

$$H_4 = \sum_{j=6}^{6+2n}\left(\sum_{i=1}^{n} q_{i,j} - 2\right)^2. \tag{16}$$

The full Hamiltonian can finally be written as

$$H = \sum_{i=1}^{4}\lambda_i H_i, \tag{17}$$

where λ_i's are positive constants.

4 Experiment

4.1 Solve QUBO via Simulator

The proposed conversions from puzzles to QUBO coefficients are tested on a simulator to verify their correctness. For these experiments, D'Wave's qbsolv package is used as a simulator, it is a decomposing solver that looking for the minimum for a QUBO problem. For large problems, it will split them into sub QUBOs and solve each one using a tabu algorithm [29]. For each puzzle configuration, all the formulas are converted into the following format:

$$Q = \{(i,i) : c_i, (j,j) : c_j, (i,j) : c_{ij}, \ldots\}$$

Where $(i,i) : c_i$ corresponds to $a_i q_i + b_i q_i^2 = c_i q_i^2$ in the sum of all formulas. As $q_i \in \{0,1\}$, $q_i = q_i^2$ and c_i is the sum of all linear and quadratic terms coefficients of q_i. Similarly, for $(i,j) : c_{ij}$, c_{ij} is the sum of all coefficients associate with term $q_i q_j$.

4.2 Use Quantum Annealer and Minor Embedding

For solving QUBO problems on actual Quantum Annealer, we use a DW2000Q from D'Wave. Similar to qbsolv simulator, we input the converted QUBO dictionary and instead using classical simulation, assign D'Wave's QPU as the solver in qbsolv package.

Additional constraints on DW2000Q means more prepossessing are needed. As the quantum annealer only has 2048 qubits available in Chimera Graph [30], yet all formulas so far assume to have a full connected topology, minor embedding are needed to convert the QUBO to desired layout[30]. There are works about using a fully connected graph

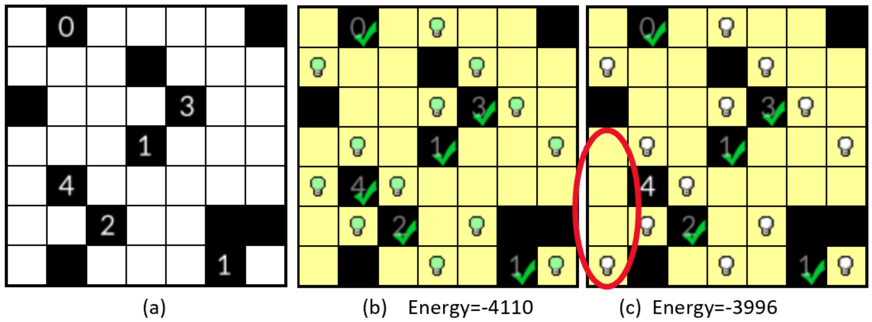


Figure 4: A Light Up puzzle on 7×7 grid. Panel.(a) is the original puzzle. The solution is given in panel.(b).(Energy=-4110) Panel.(c) gives the solution with a bit higher energy (Energy=-3996), which is slightly different with the correct solution(panel.(b)). Clearly we can see the position of one light bulb is shifted in the red circle.

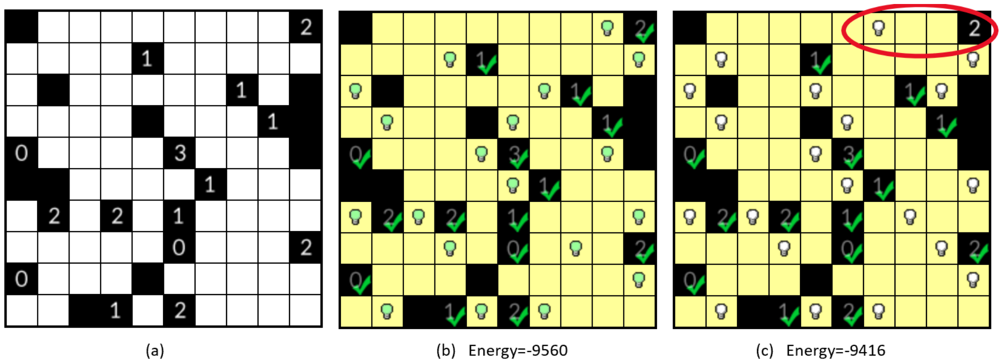


Figure 5: A Light Up puzzle on a 10×10 grid. Panel.(a) is the original puzzle. The solution is given in panel.(b).(Energy=-9560) Panel.(c) gives the solution with a bit higher energy(Energy=-9416), which is slightly different with the correct answer(panel.(b)).

and created a fixed embedding fits for all possible quadratic terms, yet such approach can only fit about 50 qubits in DW2000Q and requires splitting QUBO into small pieces [24]. Our experiments and other works indicates that such method is more time consuming in terms of QPU access time [24].

Given QUBO coefficients matrix sparse nature, fully connected assumption is too strong to solve the minor embedding efficiently. Instead of splitting QUBO into small fully connected graphs, a heuristic based algorithm, minorminer from D'Wave [30], is used to convert whole QUBO graph into Chimera Graph. Based on different natures of each puzzle and approaches of constrains, the maximum puzzle that can be solved varies, however, in general, a QUBO with 300 qubits is solvable by minorminer. Comparing to fully connected graphs, it can fit a bigger problem, yet our testing indicates that it come with the cost of significantly higher lowest energy. As most of the puzzles require a close to ground energy configuration to find a solution, the heuristic based miner embedding is no suited for our use case, and all the following experiments on QPU have a fully connected embedding.

5 Results

5.1 Light Up

In Fig.4, we give an example of the Light Up puzzle on a 7×7 grid. The original puzzle is shown in panel.(a). The quantum annealer and classical annealer gives the same solution, which is shown in panel.(b). A solution with a bit higher energy is given in panel.(c) at the bottom. It is slightly different from the correct solution, as we labelled with the red circle. In Fig.5, we give a more complex example of Light Up puzzle on a 10×10 grid. Panel.(a) is the original puzzle and the correct solution and the solution with a bit higher energy is shown in panel.(b) and (c). In this case, the grid is too complicated for the quantum annealer to solve. The solution is given only by the classical annealer.

5.2 Numberlink

Two examples of solving Numberlink puzzle on a 4×4 grid is shown in Fig.6. The left two panels are two puzzles. With the classical annealing algorithm, the solutions of the two puzzles are shown in the two panels in the middle.

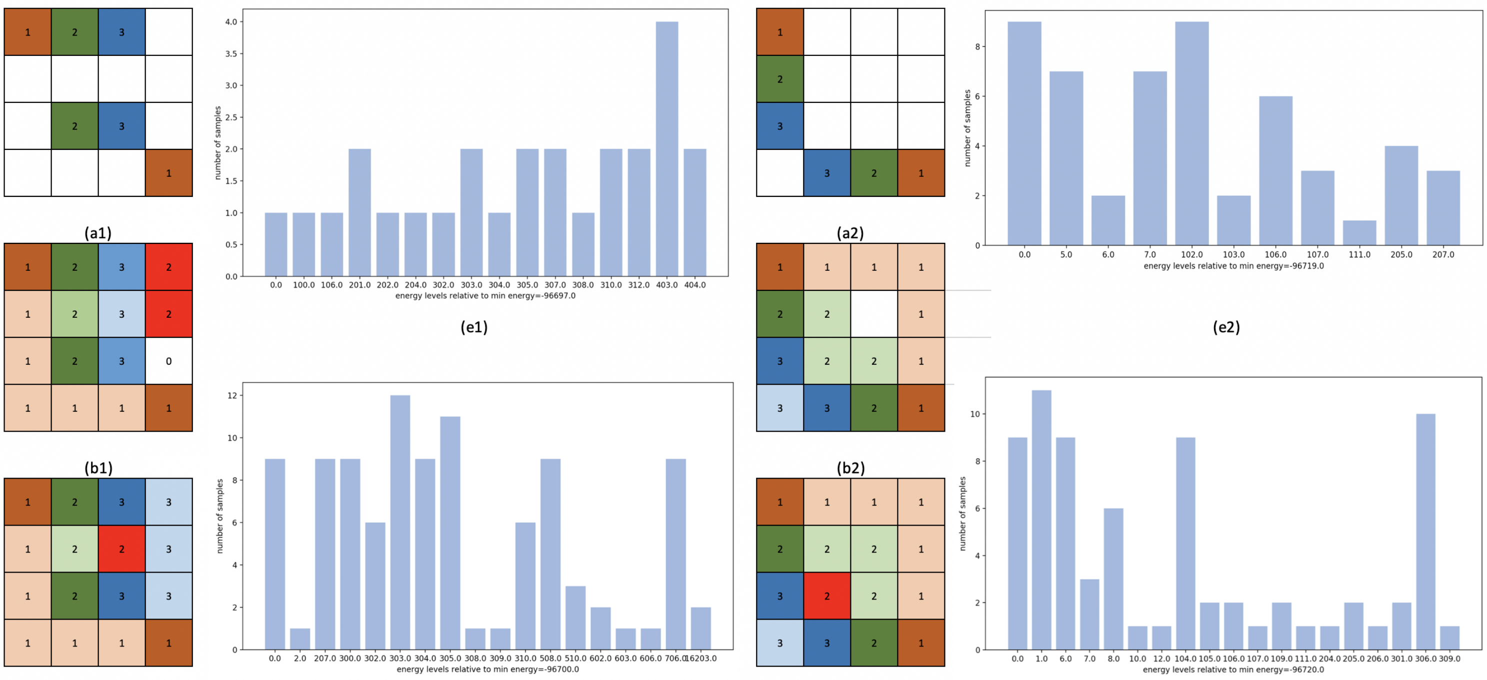
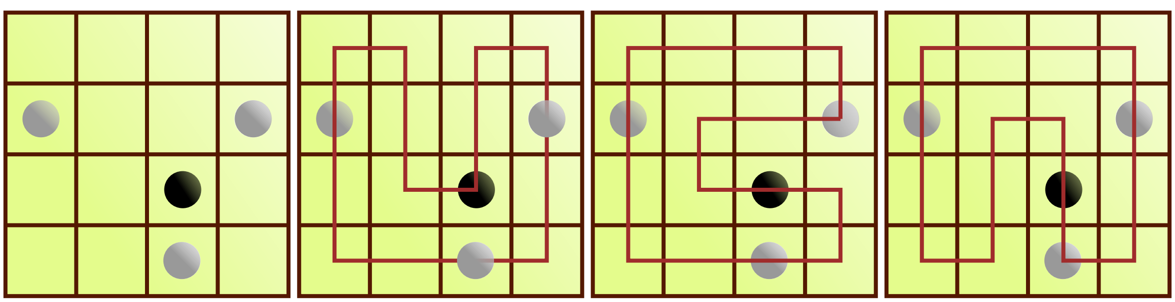


Figure 6: (a1) Numberlink puzzle setup. (b1) lowest energy solution (E=-96697.0) via simulator. (c1) lowest energy solution (E=-96700.0) via quantum annealer. (d1) distribution of energy levels visited by simulator. (e1) distribution of energy levels visited by QPU. (a2) another Numberlink puzzle setup. (b1) lowest energy solution (E=-96719) via simulator. (c1) lowest energy solution (E=-96720) via quantum annealer. (d2) distribution of energy levels visited by simulator. (e2) distribution of energy levels visited by QPU.



(a) (b) (c) (d)

Figure 7: The experimental result for Masyu (a) Unsolved Masyu puzzle (b) Correctly obtained solution (c) First incorrect solution by the simulator (d) First incorrect solution by the QPU

The two panels on the right gives the solutions of DWave quantum computer. Clearly we can see both the classical and quantum annealer gives the correct solutions although they are not exactly the same. In both puzzles, we use $\lambda_1 = -16000$, $\lambda_2 = 1$, $\lambda_3 = 100$, if $q_{kij} \in \wr$, $\lambda_2 = 100$, $\lambda_3 = 16000$.

5.3 Masyu

The experimental result of Masyu is shown in Fig.7. Panel.(a) is the original puzzle. Both classical and quantum annealer give correct solution with lowest energy, which is shown is panel.(b). Panel.(c) and panel.(d) show the solution with a bit higher energy given by classical and quantum annealer.

5.4 Instant Insanity

The experimental result of Instant Insanity is shown in Fig.8. The solutions with the lowest energy given by the classical and quantum annealing are the same, which is shown in panel.(g). A solution with a bit higher energy is shown in

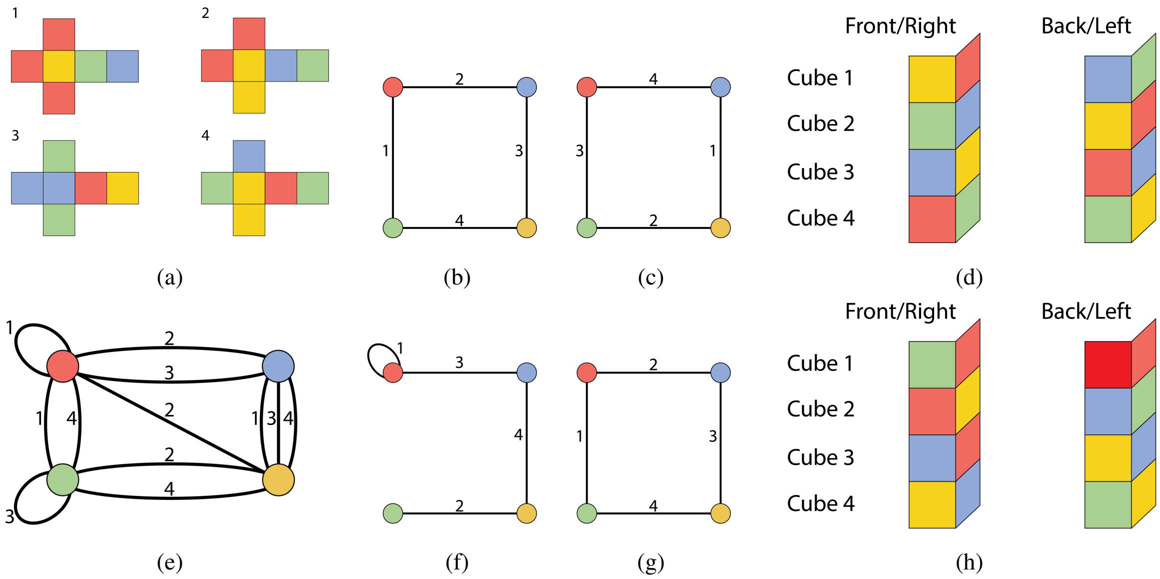


Figure 8: (a) Instant Insanity problem with $n = 4$ taken from [31] (b) The complete graph for the problem. (c and d) Lowest energy solution ($E = -96$) given by D-Wave which gives also gives a degenerate second solution with graphs A and B swapped. The stacked cubes corresponding to the $E = -96$ solution is shown in (e) The graphs in (f and g) represent a failed solution ($E = -95$). (h) The stacked cubes corresponding $E = -95$ sample.

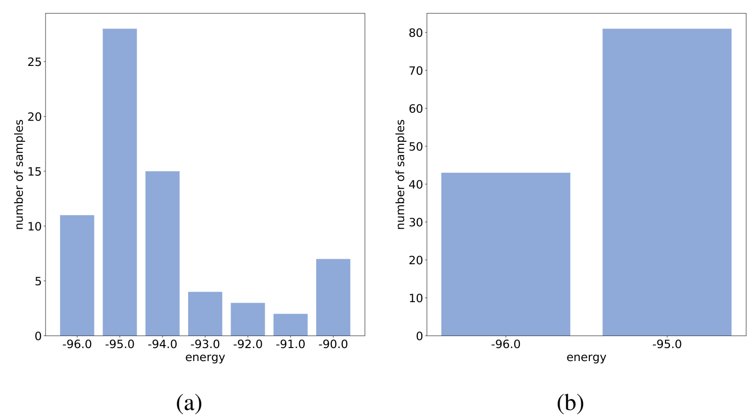


Figure 9: Sample distributions for sampling solutions with D-Wave (a) and a classical solver (b).

panel.(h). We can clearly see the color of the right face of cube 1 and cube 3 are the same in panel.(h), which means a wrong solution.

6 Discussion and Conclusion

For Light Up, as the formulation to QUBO use fewer qubits under the condition of same grid size, we can solve larger grid compared to other puzzles (like Masyu and Numberlink). Usually an $N \times N$ ($N < 8$) grid can be efficiently solve using our method on both the classical annealer and D-wave quantum computer. But for larger grid such as $N > 10$, the classical annealer can not search all the Hilbert space so usually goes to a local minimum rather than a global minimum. Besides, all the qubits are fully connected in the Hamiltonian of Light Up puzzle. As the D-Wave quantum computer can only solve partially connected Hamiltonian, the process of minor embedding becomes harder or even impossible as the grid size grows.

For Numberlink, the QUBO format constraints can effectively solve small configuration of puzzles both on classical simulator and actual quantum annealer. However, the QUBO still has several limitations. Firstly, as solutions are sampled from a distribution in both simulator and annealer cases, there is no guarantee to obtain a correct solution every time, instead, multiple experiments are needed to obtain a reasonable result. Shown in Fig.6(d) and Fig.6(e), there is no clear concentration of distribution near the lowest energy level, indicates the unstable nature of the solver. Because of that, solvers usually discover a local minimum with partially correct solution instead of the true global minimum, as shown in Fig.6, only one prefect solution is found, the rest are solutions with additional branches along the link path. Another drawback comes from the constraints formula itself, although with aforementioned QUBO, the ground energy level configuration is indeed the correct solution. Yet, in real world, especially for puzzle with large grid and few number pairs, parts of grids should be empty in the solution, but the vanilla QUBO we proposed will introduce some self looping path, similar in Fig.6(c2), to prevent that additional constraints are added. Namely, adding a universal penalty for each qubit on the grid to encourage the solver to find the minimal length solution. Additionally, cubic terms in the constraints increase the qubits count and QUBO complexity, and further limit QPU's ability to solve puzzles.

Since the problem mechanics of Masyu seems a lot similar to Numberlink, one may wonder why the formulation as a QUBO problem looks quite different when similar formulation could have been used to reduce the number of qubits (Note that in Masyu the number of qubits is $O(M^2N^2)$). The reason behind that is the issue of multiple cycles. By paying closer attention to the formulation of Numberlink, one can see that there is not any constraint that can prevent getting additional cycles along side the main path. These cycles do not form any issue for the case of Numberlink since they can be removed after the puzzle is solved (they do not affect the solution path and are merely additional). However, in Masyu the situation is different since these additional cycles may pass through some of the black and white circles; thus, when we remove them the solution will not be passing through all the circles. This issue was the reason behind using a more costly formulation in terms of qubits.

Like Numberlink, the D-Wave solver was not able to consistently sample proper solutions to the Instant Insanity problem. The classical solver, however, was able to find the solutions in every run. If a problem has a solution, the formulation in fact has two solutions because in the formulation the subgraphs A and B have unique representations but interchanging them does not produce a new real solution (i.e. rotating the stack ninety degree is the same real solution but is represented by different solution to the QUBO formulation). Figure 9a shows the distribution of sample energies produced from 1000 reads using the D-Wave sampler. Many more poor quality solutions are produced.

Considering the results of running these four problems on the D-Wave sampler, it is evident that current limitations on the number of qubits as well as the noisiness of the hardware limit the near term application to large problems.

References

[1] Aaron Adcock, Erik D. Demaine, Martin L. Demaine, Michael P. O'Brien, Felix Reidl, Fernando Sánchez Villaamil, and Blair D. Sullivan. Zig-zag numberlink is np-complete. *Journal of Information Processing*, 23(3):239–245, 2015.

[2] Daniel Andersson. Hashiwokakero is np-complete. *Information Processing Letters*, 109(19):1145 – 1146, 2009.

[3] Jean-François Baffier, Man-Kwun Chiu, Yago Diez, Matias Korman, Valia Mitsou, André van Renssen, Marcel Roeloffzen, and Yushi Uno. Hanabi is np-hard, even for cheaters who look at their cards. *Theoretical Computer Science*, 675:43 – 55, 2017.

[4] Brandon Macphail. Light up is np complete. 2005.

[5] Robin Houston, Joseph White, and Martyn Amos. Zen puzzle garden is np-complete. *Information Processing Letters*, 112(3):106 – 108, 2012.

[6] Ziwen Liu and Chao Yang. Hanano puzzle is np-hard. *Information Processing Letters*, 145:6 – 10, 2019.

[7] Np-completeness, puzzles, and games. *Utilitas Mathematica*, 13:99–116, 1978.

[8] Frank et.al. Arute. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[9] Dmitri Maslov Edwin Pednault, John Gunnels and Jay Gambetta. *IBM Research Blog*.

[10] Phil Goddard, Susan Mniszewski, Florian Neukart, Scott Pakin, and Steve Reinhardt. How Will Early Quantum Computing Benefit Computational Methods? Technical report.

[11] Andrew Lucas. Ising formulations of many NP problems. feb 2013.

[12] Akari, 2001.

[13] Ed Pegg Jr. Number link. *The Mathematica Journal*, 10(3):469–73, 2007.

[14] Erich Friedman. Pearl puzzles are np-complete. *Unpublished manuscript, August*, 2002.

[15] Gary Kochenberger, Jin-Kao Hao, Fred Glover, Mark Lewis, Zhipeng Lü, Haibo Wang, and Yang Wang. The unconstrained binary quadratic programming problem: a survey. *Journal of Combinatorial Optimization*, 28(1):58–81, 2014.

[16] Fred Glover, Gary Kochenberger, and Yu Du. A tutorial on formulating and using qubo models. 2019.

[17] AB Finnila, MA Gomez, C Sebenik, C Stenson, and JD Doll. Quantum annealing: a new method for minimizing multidimensional functions. *Chemical physics letters*, 219(5-6):343–348, 1994.

[18] Davide Venturelli, D Marchand, and Galo Rojo. Job shop scheduling solver based on quantum annealing. In *Proc. of ICAPS-16 Workshop on Constraint Satisfaction Techniques for Planning and Scheduling (COPLAS)*, pages 25–34, 2016.

[19] Bahram Alidaee, Fred Glover, Gary A Kochenberger, and Cesar Rego. A new modeling and solution approach for the number partitioning problem. *Advances in Decision Sciences*, 2005(2):113–121, 2005.

[20] Gary Kochenberger, Fred Glover, Bahram Alidaee, and Karen Lewis. Using the unconstrained quadratic program to model and solve max 2-sat problems. *International Journal of Operational Research*, 1(1-2):89–100, 2005.

[21] Foad Mahdavi Pajouh, Balabhaskar Balasundaram, and Oleg A Prokopyev. On characterization of maximal independent sets via quadratic optimization. *Journal of Heuristics*, 19(4):629–644, 2013.

[22] Gary A Kochenberger, Jin-Kao Hao, Zhipeng Lü, Haibo Wang, and Fred Glover. Solving large scale max cut problems via tabu search. *Journal of Heuristics*, 19(4):565–571, 2013.

[23] Gary A Kochenberger, Fred Glover, Bahram Alidaee, and Cesar Rego. An unconstrained quadratic binary programming approach to the vertex coloring problem. *Annals of Operations Research*, 139(1):229–241, 2005.

[24] Asa Eagle, Takumi Kato, and Yuichiro Minato. Solving tiling puzzles with quantum annealing, 2019.

[25] Zahra Karimi, Kamran Zamanifar, Ahmad Baraani-Dastjerdi, and Nasser Ghasem-Aghaee. Sudoku using parallel simulated annealing. volume 6146, pages 461–467, 06 2010.

[26] Timothy Resnick. Sudoku at the intersection of classical and quantum computing.

[27] Chia Cheng Chang, Arjun Gambhir, Travis S. Humble, and Shigetoshi Sota. Quantum annealing for systems of polynomial equations. *Scientific Reports*, 9(1), Jul 2019.

[28] Anuradha Mahasinghe, Richard Hua, Michael J Dinneen, and Rajni Goyal. Solving the hamiltonian cycle problem using a quantum computer. *computing*, 29:40, 2019.

[29] Steven P. Reinhardt Michael Booth and A. Roy. Partitioning optimization problems for hybrid classical/quantum execution. Technical report, D-Wave Technical Report Series 1 4-1006A-A.

[30] Jack Raymond Aidan Roy Kelly Boothby, Paul Bunyk. Next-generation topology of d-wave quantum processors. Technical report, D-Wave Technical Report Series 14-1026A-C.

[31] Tai-Danae Bradley. Instant insanity puzzle | infinite series, 2019.