

Received: 30 November 2014 Accepted: 18 May 2015 Published: 08 June 2015

OPEN Experimental quantum annealing: case study involving the graph isomorphism problem

Kenneth M. Zick¹, Omar Shehab^{1,2} & Matthew French¹

Quantum annealing is a proposed combinatorial optimization technique meant to exploit quantum mechanical effects such as tunneling and entanglement. Real-world quantum annealing-based solvers require a combination of annealing and classical pre- and post-processing; at this early stage, little is known about how to partition and optimize the processing. This article presents an experimental case study of quantum annealing and some of the factors involved in real-world solvers, using a 504qubit D-Wave Two machine and the graph isomorphism problem. To illustrate the role of classical pre-processing, a compact Hamiltonian is presented that enables a reduced Ising model for each problem instance. On random N-vertex graphs, the median number of variables is reduced from N^2 to fewer than N log, N and solvable graph sizes increase from N=5 to N=13. Additionally, error correction via classical post-processing majority voting is evaluated. While the solution times are not competitive with classical approaches to graph isomorphism, the enhanced solver ultimately classified correctly every problem that was mapped to the processor and demonstrated clear advantages over the baseline approach. The results shed some light on the nature of real-world quantum annealing and the associated hybrid classical-quantum solvers.

Quantum annealing (QA) is a proposed combinatorial optimization technique meant to exploit quantum mechanical effects such as tunneling and entanglement¹. Machines purportedly implementing a type of quantum annealing have recently become available². While the extent of "quantumness" in these implementations is not fully understood, some evidence for quantum mechanical effects playing a useful role in the processing has been appearing³⁻⁵. Aside from the debate over quantumness, there are interesting questions regarding how to effectively solve a real-world problem using a quantum annealer. Quantum annealing-based solvers require a combination of annealing and classical pre- and post-processing; at this early stage, little is known about how to partition and optimize the processing. For instance, current quantum annealers have severe practical limitations on the size of problems that can be handled. Can the pre-processing algorithms be modified in order to improve scalability? A second question involves post-processing. Quantum annealers provide solutions to an "embedded" version of a problem involving physical qubits. Post-processing is generally required for translating these to solutions to the original problem involving logical qubits (aka variables). Occasionally, a chain of physical qubits representing a single variable resolves to an inconsistent state, a scenario known as a broken chain. Studies are needed regarding broken chains and the possibility of classical error correction during post-processing.

This article presents an experimental case study of quantum annealing and some of the factors involved in real-world solvers, using a 504-qubit D-Wave Two machine. An example of parsimonious pre-processing is considered, along with post-processing majority voting. Through experiments on a 504-qubit D-Wave Two machine, we quantify the QA success probabilities and the impact of the methods under study. We use the graph isomorphism (GI) problem as the problem of focus. The GI problem is to determine whether two input graphs $G_{1,2}$ are in essence the same, such that the adjacency matrices

¹University of Southern California Information Sciences Institute (USC ISI), Arlington, Virginia, USA. ²University of Maryland – Baltimore County, USA. Correspondence and requests for materials should be addressed to K.M.Z. (email: kzick@isi.edu)

can be made identical with a relabeling of vertices. This problem is an interesting candidate for several reasons. First, an accurate quantum annealing-based solver for GI has never been implemented. Second, quantum approaches can sometimes provide new insight into the structure of a problem, even if no speedup over classical approaches is achieved or even expected. Third, the GI problem is mathematically interesting; though many sub-classes of the problem can be solved in polynomial time by specialized classical solvers, the run time of the best general solution is exponential and has remained at $e^{O(\sqrt{N}\log N)}$ since 1983^{6,7}. The classical computational complexity of the problem is currently considered to be **NP**-intermediate⁸, and the quantum computational complexity of the problem is unknown. Graph isomorphism is a non-abelian hidden subgroup problem and is not known to be easy in the quantum regime^{9,10}. Lastly, the GI problem is of practical interest. It appears in fields such as very large-scale integrated circuit design, where a circuit's layout graph must be verified to be equivalent to its schematic graph¹¹, and in drug discovery and bio-informatics, where a graph representing a molecular compound must be compared to an entire database, often via a GI tool that performs canonical labeling⁶.

This article relates to previous works as follows. A pre-print by King and McGeoch discusses tuning of quantum annealing algorithms, including the use of low-cost classical post-processing majority voting similar to what is evaluated in this article¹². Our study goes further regarding pre-processing (designing a Hamiltonian to generate compact Ising models) and covers graph isomorphism rather than problems such as random not-all-equal 3-**SAT**. A work by Rieffel *et al.* maps real-world problems such as graph coloring to a D-Wave quantum annealer¹³. Regarding the graph isomorphism problem in particular, multiple attempts have been made using adiabatic quantum annealing. One of the first attempts assigned a Hamiltonian to each graph and conjectured that measurements taken during each adiabatic evolution could be used to distinguish non-isomorphic pairs¹⁴. A subsequent experimental study using a D-Wave quantum annealer found that using quantum spectra in this manner was not sufficient to distinguish non-isomorphic pairs¹⁵. A second approach converted a GI problem to a combinatorial optimization problem whose non-negative cost function has a minimum of zero only for an isomorphic pair. The approach required $N \mid \log_2 N \mid$ problem variables and additional ancillary variables. It was numerically simulated up to N=7 but not validated on a quantum annealing processor¹⁶. An alternative GI Hamiltonian was proposed by Lucas¹⁷.

Proposed Solver

Preliminaries. Before proceeding, we briefly define some key concepts. In quantum annealing, a system is first initialized to a superposition of all possible states as dictated by Hamiltonian H_{init} , then slowly evolved such that the initializing function becomes weaker and the energy function of interest, defined by the problem Hamiltonian H_P , becomes dominant. The time-dependent combination of the two Hamiltonians is referred to as the system Hamiltonian $H(t)^{18}$.

$$H(t) = A(t)H_{init} + B(t)H_{p}, \quad t \in [0, T],$$
 (1)

where A(t) and B(t) are monotonic functions representing the annealing schedule up to time T. In this work, we focus on problem Hamiltonians H_p . The problem Hamiltonian represents an Ising model, where each spin variable $s_i \in \{-1, 1\}$ is subject to a local field h_i and where variables interact pairwise with coupling strength J_{ij}^{19} :

$$H_P = \sum_i h_i s_i + \sum_{i,j} I_{ij} s_i s_j. \tag{2}$$

Problems can alternatively be represented as quadratic unconstrained binary optimization (QUBO) problems, with binary variables $x_i \in \{0, 1\}$ instead of spin variables, and with the following cost function¹:

$$f_{qubo}(x) = \sum_{i,j} Q_{ij} x_i x_j.$$
(3)

In some cases, a QUBO representation is a more convenient starting point. A QUBO problem can be converted to an Ising problem using the relation $x_i = \frac{(s_i + 1)}{2}$.

It is often the case that an Ising problem [h, J] involves variable interactions not supported by a quantum annealing processor graph. An example is a variable with degree higher than the maximum supported by the D-Wave Chimera architecture (six). One common strategy is to find a *minor embedding* of the problem graph in the processor graph¹⁹, in which a set of physical qubits is used to represent each variable. Each qubit is strongly coupled to at least one other qubit in the set, in an effort to keep the entire set in a consistent state. These sets are commonly referred to as chains. We refer to a minor-embedded version of a problem as an embedded Ising problem [h', J'].

Baseline Hamiltonian. We first describe a baseline penalty Hamiltonian for the GI problem, building upon the Hamiltonian described in Lucas¹⁷. The problem input is a pair of simple, undirected N-vertex graphs $G_{1,2}$. Penalties are applied such that the ground state energy is zero if the pair is isomorphic and

greater than zero otherwise. The intent is for an energy minimization process (such as quantum annealing) to provide a solution to this decision problem.

The baseline Hamiltonian includes binary variables $x_{u,i} \in \{0, 1\}$ for every possible mapping of a vertex u in G_2 to a vertex i in G_1 ; in a solution, the variable is 1 if u is mapped to i, and 0 otherwise. There are two types of penalties that can be applied to a vertex set mapping. One type penalizes mappings that are not bijective. For instance, a mapping in which vertices 1 and 2 in G_2 are both mapped to vertex 1 in G_1 is invalid; the term $x_{1,1}x_{2,1}$ is assigned a positive penalty G_1 such that if G_2 are both mapped to vertex 1, the solution energy will necessarily be greater than zero. The second penalty type involves edge inconsistencies. An example of an edge inconsistency is when vertices g_1 are mapped to g_2 and an edge exists in one input graph and not the other (e.g. g_2 while g_3 while g_4 is Edge inconsistencies are assigned a positive penalty g_3 . Additional details regarding this style of Hamiltonian can be found in Lucas¹⁷.

In Lucas¹⁷, couplings can incur zero, one, or two penalties. Here, we require that couplings be penalized no more than once, in order to achieve a simple set of coupler values (e.g. $\{0, 1\}$ instead of $\{0, 1, 2\}$) amenable to quantum annealing. Specifically, an edge-related penalty is applied to a coupling only if there is not a vertex mapping penalty. As an example, if vertices u,v are mapped to i,j and i=j, then the bijection has been violated and coupling $x_{u,i}x_{v,j}$ will incur a vertex mapping penalty (C_1) ; therefore, an additional edge inconsistency penalty (C_2) need not be applied. Edge inconsistency penalties are only applied if $i \neq j$ and $u \neq v$. The set of required coupler values can be further simplified by setting $C_1 = C_2 > 0$. The complete Hamiltonian is

$$H_{1} = C_{1} \sum_{u} \left(1 - \sum_{i} x_{u,i} \right)^{2} + C_{1} \sum_{i} \left(1 - \sum_{u} x_{u,i} \right)^{2} + C_{2} \sum_{\substack{i,j \notin E_{1} \\ i \neq j}} \sum_{u,v \in E_{2}} x_{u,i} x_{v,j} + C_{2} \sum_{\substack{i,j \in E_{1} \\ u \neq v}} \sum_{\substack{u,v \notin E_{2} \\ u \neq v}} x_{u,i} x_{v,j}, \tag{4}$$

where the $i \neq j$ and $u \neq v$ conditions represent the main modifications to Lucas¹⁷.

Compact Hamiltonian. The approach embodied in the baseline Hamiltonian H_1 suffers from a severe lack of scalability. For N-vertex input graphs, it requires N^2 logical variables. Moreover, due to the limited direct connections between qubits in the D-Wave Chimera architecture, problems are often given a minor embedding into the processor working graph. This typically involves replicating variables across multiple qubits. Thus, the qubit requirements can reach $O(N^4)$. Problems mapped in this way to a ~500-qubit processor tend to be limited to N=5 or 6. We now investigate whether a more effective Hamiltonian can be designed. The idea is that many variables and interactions are unnecessary, and information indicating so can be leveraged up front during the requisite pre-processing. Note that an isomorphic mapping requires the vertices in each matched pair to have the same degree. Thus, degree information can be used to decide whether two vertices are eligible to be matched. We propose a compact Hamiltonian H₂ that avoids creating variables for vertices of different degree. A second, minor simplification deals with isolated vertices (degree = 0). If $G_{1,2}$ each have k isolated vertices, an isomorphic mapping of such vertices is trivial and thus no variables or penalties for those vertices need be modeled. If $G_{1,2}$ have a different number of isolated vertices, then they also have a different number of non-isolated vertices and existing variables and penalties for those will suffice. Thus we only create variables and penalties for vertices with degree greater than zero.

Given the two enhancements, the total number of variables required is $\sum_{i=1}^{\max} |d_i|^2$ where $|d_i|$ is the multiplicity of the set d_i containing vertices of degree i. In the worst case of regular graphs of degree r greater than zero, all nodes have the same non-zero degree and thus the simplifications provide no benefit— N^2 variables are still required. Many real-world graphs are not regular and for these the benefits can be large. The entire compact Hamiltonian H_2 is

$$\forall (\deg(i) = \deg(u) > 0; \deg(v) = \deg(j) > 0):$$

$$H_{2} = \sum_{u} \left(1 - \sum_{i} x_{u,i} \right)^{2} + \sum_{i} \left(1 - \sum_{u} x_{u,i} \right)^{2} + \sum_{\substack{i,j \in E_{1} \\ i \neq j}} \sum_{u,v \in E_{2}} x_{u,i} x_{v,j} + \sum_{\substack{i,j \in E_{1} \\ u \neq v}} \sum_{u,v \notin E_{2}} x_{u,i} x_{v,j}.$$
(5)

Figure 1 shows an example of problem instances generated using the baseline and the compact Hamiltonians on the same input; it illustrates how the number of variables and the number of non-zero interactions—as seen in the off-diagonal entries in the associated QUBO matrix Q—can be much smaller when using the compact Hamiltonian. The variable scaling of each Hamiltonian is quantified and compared in Results.

 H_2 was validated using a software solver (D-Wave System's ising-heuristic version 1.5.2) that provides exact results for problems with low tree width. In exhaustive testing of all 2^{12} N=4 pairs and 2^{20} N=5 pairs, the ground state energy of H_2 was confirmed to be zero for isomorphic cases and greater than zero for non-isomorphic.

As a part of a quantum annealing-based solver, an algorithm can be employed that accepts a graph pair as input and uses the Hamiltonian to generate an associated QUBO problem (later converted to

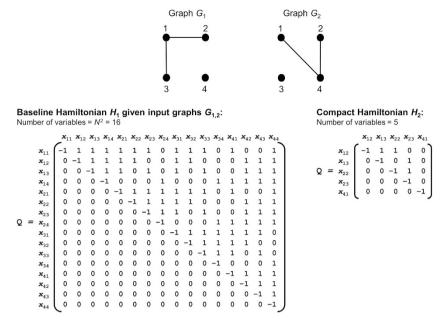


Figure 1. Illustration of problem instances generated using baseline Hamiltonian H_1 and compact Hamiltonian H_2 on the same input. For many inputs, H_2 requires fewer variables and non-zero interactions.

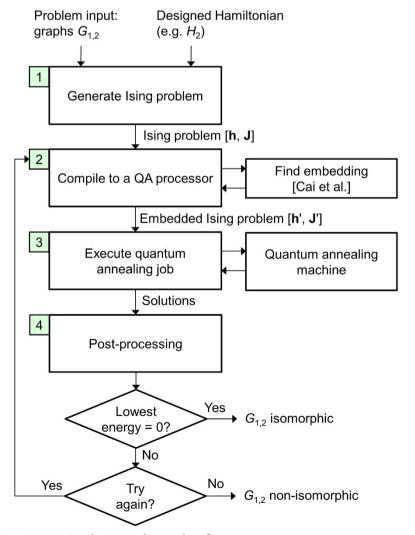


Figure 2. Graph isomorphism solver flow.

an Ising problem). The proposed algorithm using the compact Hamiltonian H_2 is presented (in pseudo code) in Algorithm 1.

Algorithm 1 | High-level algorithm for generating a QUBO problem using the proposed compact Hamiltonian H_2 .

```
Input: adjacency matrices G_{1,2}
Output: QUBO matrix
// Define QUBO variables
for each node v in G_2 do
   for each node i in G_1 do
      if deg(v) = = deg(i) and deg(i) > 0 then
        create QUBO variable x_{v,i}
      end
   end
end
// Populate QUBO matrix with penalty terms
for each pair of different QUBO variables x_{u,i} and x_{v,j} do
   // Assign penalty for node mapping conflict
   if i = i then
     penalize mapping u \rightarrow i, v \rightarrow j // Two nodes in G_2 map to the same node in G_1
   else if u = v then
      penalize mapping u \rightarrow i, v \rightarrow j // A node in G_2 maps to two nodes in G_1
   // Assign penalty for edge discrepancy
   else if G_1(i, j) != G_2(u, v) then
     penalize mapping u \rightarrow i, v \rightarrow j // Edge in one graph, non-edge in other
   end
for each QUBO variable do
   assign value along diagonal of Q
end
```

Complete Solver Flow. The high-level flow of the proposed GI solver is shown in Fig. 2. The problem input is a graph pair $G_{1,2}$. In this article, the graph types considered are random, simple, undirected graphs. Graphs are generated using the Erdős-Rényi model²⁰ G(N, p) where we set the probability of an edge being present p = 0.5. An advantage of G(N, 0.5) graphs for an initial study is that all graphs are equally likely. This type has been used in classical graph isomorphism work as well. In step 1, the input graphs and the Hamiltonian formulation of interest (e.g. H_1 or H_2) are used to generate a QUBO problem which is then converted to an Ising problem [h, J]. An example of an algorithm for generating the QUBO problem is shown in Algorithm 1. The Ising problem is then compiled to a specific quantum annealing processor in step 2. A main task is to find sets of physical qubits to represent the problem variables (aka logical qubits); this is achieved by providing the J matrix and the processor working graph to the D-Wave findEmbedding() heuristic²¹. Subsequently, the parameters of the embedded Ising problem [h', J'] are set following certain strategies such as the use of a random spin gauge (see Methods). The embedded Ising problem, sometimes referred to as a machine instruction, is submitted to the quantum annealing machine along with several job parameters. The quantum annealing job is executed in step 3 and solutions are returned in the form of strings of two-valued variables. These solutions and energies are associated with the embedded problem, not the original Ising problem. Therefore a post-processing step is necessary (step 4), in which the state of each qubit chain is plugged into the cost function of the Ising problem. A difficulty arises when the states of the qubits in a chain are inconsistent, a case referred to as a broken chain. In the proposed solver, broken chains can be handled by either discarding the associated solution, or by performing majority voting over each chain. The two strategies are compared empirically in Results. Given a solution to the original Ising problem, the solution energy can be calculated. If the

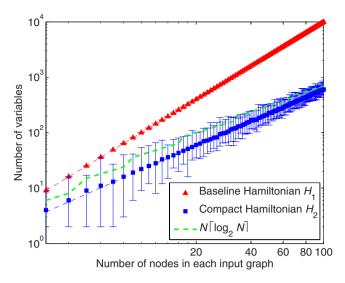


Figure 3. Scaling of the number of Ising model variables. H_1 requires N^2 variables by construction; H_2 scales more efficiently—the median variable requirement fits to $y = 0.748 \ N^{1.45} \ (R^2 = 0.9995)$. Bars indicate the maximum and minimum. For reference, the green dotted line represents $N \lceil \log_2 N \rceil$. Inputs are 50 isomorphic and 50 non-isomorphic pairs of G(N, 0.5) graphs.

lowest energy is zero, then the input pair can be declared isomorphic and no further jobs are necessary. Otherwise, a decision must be made whether to repeat the process from step 2 or to stop and declare that isomorphism could not be established.

Results

Ising Model Scaling. To compare the resource requirements of the two proposed Hamiltonians, 100 pairs of graphs are used as inputs to Step 1 of the solver flow (Fig. 2), where 50 pairs are isomorphic and 50 are non-isomorphic for each size up to N=100. Since H_1 models a variable for each possible vertex pair, N^2 variables are required by definition. Ising problems generated using H_2 are found to use fewer variables than H_1 ; scaling of the median problems fits to 0.748 $N^{1.45}$. Incidentally, this indicates that most problems have fewer variables than with the Gaitan *et al.* approach, which entails $N \mid \log_2 N \mid$ plus ancillary variables 16 . The variable scaling is illustrated in Fig. 3. In addition to the number of variables, a second resource metric is the number of non-zero interactions between variables; dense interactions make the minor embedding problem more difficult. We find that the scaling of variable interactions has been improved from $O(N^4)$ for H_1 to $O(N^{2.9})$ for H_2 (where $R^2=0.9991$).

Embeddability. Next, we compare the embeddability of the two approaches, in other words the extent to which Ising problems can be minor-embedded in a given processor graph. The processor of choice is the D-Wave Two Vesuvius-6 processor housed at USC ISI. At the time of this writing, the working graph contains 504 qubits and 1427 couplers. Embedding is attempted using the D-Wave findEmbedding() heuristic²¹ with default parameter values such as 10 "tries" per function call. As shown in Fig. 4a, embeddings are found for the majority of problems only for sizes $N \le 6$ when using H_1 , but sizes $N \le 14$ with H_2 (Fig. 4a). The median number of qubits across all problems scales as $O(N^{4.22})$ for H_1 and has been reduced to $O(N^{3.29})$ for H_2 (Fig. 4b).

Experimental Quantum Annealing for Graph Isomorphism. The accuracy of the solver described in the previous section was measured via trials conducted on a D-Wave Two Vesuvius quantum annealing processor. Several alternative strategies were compared—the use of Hamiltonians H_1 vs. H_2 , running a single job per problem vs. multiple jobs, and the use of chain majority voting during post-processing. Note that by construction of the Ising models using a penalty Hamiltonian, problems with non-isomorphic input graphs cannot achieve a zero energy state, regardless of annealing results. The main challenge for the solver is to find the zero energy state for isomorphic pairs. Thus, we first focus on the isomorphic case. One hundred isomorphic pairs were input into the solver for each size N from 3 to 20.

For one strategy in particular the zero energy state was always eventually achieved—the use of Hamiltonian H_2 combined with multiple jobs and chain majority voting. Thus, with this strategy there were no false negatives and classification accuracy reached 100% of the embeddable problems, as shown in Table 1. For the most difficult problem, the zero energy state was achieved on the 9th job. All other

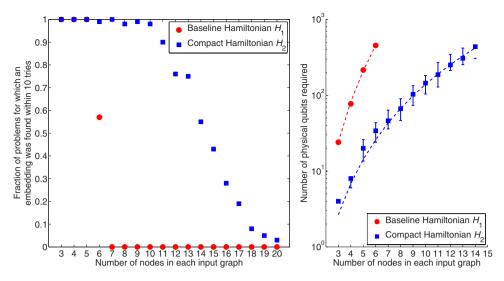


Figure 4. Embeddability when targeting the USC-LM Vesuvius processor's 504-qubit, 1427-coupler working graph. The inputs are 50 isomorphic and 50 non-isomorphic G(N, 0.5) pairs at each size. (a) Fraction of problems for which an embedding was found within 10 tries, using the D-Wave findEmbedding() heuristic. (b) Median number of utilized qubits among all 100 problems. H_1 data points fit to $y = 0.235 \ N^{4.22}$ (dotted line; $R^2 = 0.9996$); H_2 fits to 0.0723 $N^{3.29}$ ($R^2 = 0.9955$). A missing data point indicates no embedding was found for the median instance. The 65th and 35th percentiles are marked with bars (not always visible).

strategies incurred false negatives. For the successful strategy, the expected total annealing time was calculated (as described in Methods). Results are shown in Fig. 5.

For completeness, non-isomorphic pairs were run as well, using Hamiltonian H_2 and chain majority voting. Since in the worst case nine jobs were required to correctly classify the isomorphic pairs above, nine jobs were submitted for each non-isomorphic problem. One hundred non-isomorphic G(N, 0.5) problems were tested at each size between N=3 to 14; of the 1200 problems, embeddings were found for 1186. In addition, pairs of isospectral non-isomorphic graphs (PINGs) were tested. All N=5 PINGs were tested (150 permutations), as well as 100 random N=6 PINGs. As expected, none of the non-isomorphic problems achieved a zero energy state and thus none were classified as isomorphic. In other words, there were no false positives.

Discussion

Several observations can be made from this case study. First, the formulation of the cost function (Hamiltonian) can have a noticeable impact on quantum annealing results. For the graph isomorphism problem, the baseline approach (embodied in Hamiltonian H_1 and in [Lucas]¹⁷) blindly creates QUBO variables for every possible vertex pair, whereas the proposed Hamiltonian H_2 is more parsimonious. Variable requirements decreased from N^2 to fewer than $N \log_2 N$ (Fig. 3) on the graph type under study, allowing larger problems to be solved (Fig. 4 and Table 1). Along with Rieffel¹³, this is one of the first quantum annealing studies to experimentally quantify the effect of alternative Hamiltonian formulations. One of the impacts of this observation is increased appreciation for the fact that all quantum annealing-based solvers are actually classical-quantum hybrids and that focus must be placed on effectively partitioning the processing and optimizing the classical portion. A caveat is in order—if the classical side is made to do too much work then the quantum annealing aspect becomes trivial and of little value. Further work is needed in identifying the specific strengths of annealing processors, and in leveraging the two sides appropriately.

A second observation is that using chain majority voting during post-processing can in some cases provide a benefit. Previously, such majority voting was evaluated for a different set of problems (scheduling) and was not found to provide a significant benefit¹³. In our context, there were many problems for which the zero energy ground state solution was only achieved when using this post-processing; without this form of error correction (in other words, when all solutions containing a broken chain were discarded), false negatives occurred. For instance, at N=12, 53 of 83 embedded problems were solved on the first job without using chain majority voting, and an additional 12 problems were solved by applying chain majority voting (Table 1). Classical error correction strategies other than majority voting should be explored and assessed in future studies, and their costs quantified.

To our knowledge, the evaluated solver is the first validated, experimental implementation of a QA-based graph isomorphism solver. While it ultimately classified every embeddable problem correctly and demonstrated clear advantages over the baseline approach, it has serious limitations as a graph

	Size of input graphs (number of vertices)															
	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Number of problems for which an embedding was found using H_1	100	100	100	74	0	0	0	0	0	0	0	0	0	0	0	0
Number of problems for which an embedding was found using H_2	100	100	100	99	99	96	95	93	87	83	54	39	20	7	4	1
Number of problems solved using H_1	100	99	98	13	0	0	0	0	0	0	0	0	0	0	0	0
Number of problems solved using H_1 and chain majority voting	100	100	99	43	0	0	0	0	0	0	0	0	0	0	0	0
Number of problems solved using H_2	100	100	100	98	96	93	86	75	67	53	34	23	9	1	2	0
Number of problems solved using H_2 and chain majority voting	100	100	100	99	99	96	90	80	73	65	40	25	12	3	2	1
Number of problems solved using H_2 , chain majority voting, and multiple jobs	100	100	100	99	99	96	95	93	87	83	54	39	20	7	4	1

Table 1. Number of isomorphic-input problems embedded and correctly classified as isomorphic via quantum annealing. One hundred problems were attempted at each problem size. All embedded problems were solved when using H_2 , chain majority voting, and multiple jobs.

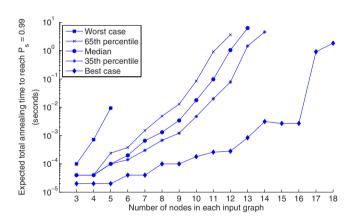


Figure 5. Total expected annealing time when using Hamiltonian H_2 , multiple jobs, and classical majority voting. Based on the quantum annealing results for 100 isomorphic-input problems. Missing data points indicate that no embedding was found and quantum annealing was not attempted.

isomorphism solver. The problem sizes are not competitive with those handled by classical solvers, which can handle G(N, 0.5) graphs with thousands of vertices⁶ and even for the hardest graph types can handle hundreds of vertices before running into difficulty²². Similarly, the scaling of the total annealing times (Fig. 5) is not competitive with classical scaling⁶. Ultimately, new approaches are likely needed if quantum annealing is to contribute to graph isomorphism theory or practice. Fortunately, the case study provides some new insight into experimental quantum annealing, and contributes methods that have relevance beyond the GI problem. It is hoped that the experimental evaluation of alternative Hamiltonian formulations adds to the understanding of the factors affecting quantum annealing performance, and that the demonstration of majority voting raises new questions about the role of post-processing for a variety of problems.

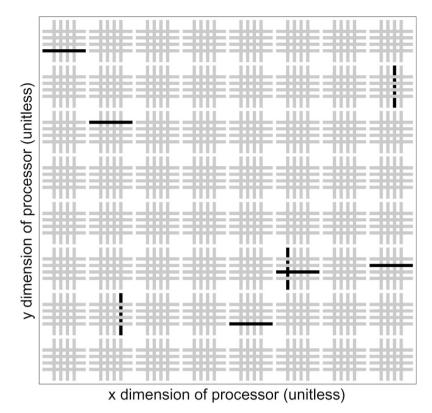


Figure 6. Physical layout of the working qubits in the USC-LM D-Wave Two Vesuvius-6 processor as of October 10, 2014. 504 working qubits (gray), 8 non-working (black). All 1427 couplers connecting working qubits are part of the working graph.

Methods

Quantum annealing experiments were performed on the D-Wave Two machine housed at USC ISI and operated by the USC-Lockheed Martin Quantum Computing Center. Experiments were conducted in October and November, 2014. The working graph of the machine's Vesuvius-6 quantum annealing processor consisted of 504 qubits and 1427 couplers during this period. The pattern of working qubits is shown in Figure 6. The qubit temperature was estimated by the manufacturer to be 16 ± 1 mK. Additional processor specifications include a maximum anti-ferromagnetic mutual inductance of 1.33, and 1/f amplitude of $7.5\pm 1~\mu\phi_0/\sqrt{\rm Hz}$.

Simple undirected N-vertex graphs were constructed according to the Erdős-Rényi G(n, p) model²⁰ with n = N and with the probability p of including each edge equal to 0.5. Non-isomorphic pairs were generated by creating two graphs as above and checking for non-isomorphism using the MATLAB graphisomorphism() function. Isomorphic pairs were generated by generating a single graph then applying a random permutation to arrive at the second graph. For each pair of input graphs, an Ising model was created using equation (4) or (5). Programming was performed using MATLAB R2014a win64 and the D-Wave MATLAB pack 1.5.2-beta2. The current version of the D-Wave sapiFindEmbedding() function cannot directly embed Ising models with more than one connected component (i.e. a set of variables that interact only with each other and not any of the remaining variables); therefore, models with this characteristic were not included in the input data. When attempting to generate 100 input pairs for each size, such disconnected models occurred no more than 4 times for each size $N \ge 14$. Similarly, the heuristic cannot accept models with fewer than two variables, so in the rare case of a trivial Ising problem with fewer than two variables (e.g. a non-isomorphic pair with no matching degrees), dummy variables were added to the problem.

The h_i values of the Ising problem were split evenly across each qubit in the associated chain in the embedded Ising problem. The I_{ij} values of the Ising problem were assigned to a single coupler connecting two variable chains in the embedded problem. The magnitudes of the embedded \mathbf{h}' and \mathbf{J}' were scaled together such that the maximum magnitude reached 20% of the full range supported by the processor; the range of the embedded h_i' values was [-0.4, 0.4] and the range of the embedded I_{ij}' values coupling different variables was [-0.2, 0.2]. This 20% value was determined empirically to provide good performance on the median difficulty problem at the largest sizes. Subsequently, the I_{ij}' values connecting physical qubits within a chain were set to the maximum ferromagnetic value (-1). A single random spin gauge transformation² was then applied to each embedded problem, with a gauge factor $a_i \in \{-1, 1\}$

associated with each qubit and transformation $h_i' \to a_i \ h_i'; J_{ij}' \to a_i \ a_j J_{ij}'$. One job was submitted to the quantum annealer per embedded problem; some Ising problems were associated with multiple embedded problems and jobs. After each programming cycle, the processor was allowed to thermalize for 10 ms (the maximum supported by the machine). The annealing time was set to the minimum value of $20\,\mu s$. The number of annealing and readout cycles per programming cycle was 40000, which allowed the total job time to be within the limits of the machine (1 s). The readout thermalisation time was set to the default value of 0. Regarding error correction through majority voting of chains of physical qubits, ties were broken by choosing the spin up state. The probability of achieving the zero energy state on job k is denoted

$$P_{0,k} = \frac{\text{number of annealing cycles achieving zero energy}}{\text{number of annealing cycles}}.$$
 (6)

When multiple jobs are required, we calculate the geometric mean in the style of Boixo et al.²:

$$\overline{P_0} = 1 - \prod_{k=1}^K (1 - P_{0,k})^{1/K}. \tag{7}$$

The total annealing time required to reach 0.99 probability of success was calculated by multiplying the annealing time by the expected number of annealing cycles (repetitions R) using the formula²:

$$R = \left[\frac{\ln(1 - 0.99)}{\ln(1 - \overline{P_0})} \right]. \tag{8}$$

References

- 1. McGeoch, C. Adiabatic quantum computation and quantum annealing: Theory and practice (Morgan & Claypool Publishers, 2014).
- 2. Boixo, S. et al. Evidence for quantum annealing with more than one hundred qubits. Nat Phys 10, no. 3, 218-224 (2014).
- 3. Lanting, T. et al. Entanglement in quantum annealing processor. Phys. Rev. X 4, 021041 (2014).
- 4. Albash, T., Rønnow, T., Troyer, M. & Lidar, D. Reexamining classical and quantum models for the D-Wave One processor The role of excited states and ground state degeneracy. Eur. Phys. J. Special Topics 224 1, 111–129 (2015).
- 5. Boixo, S. et al. Computational role of collective tunneling in a quantum annealer. arXiv:1411.4036 [quant-ph] (2014).
- 6. McKay, B. & Piperno, A. Practical graph isomorphism, II. J. Symbolic Comput. 60, 94-112 (2014).
- 7. Babai, L. & Luks, E. Canonical labeling of graphs. Paper presented at ACM Symposium on Theory of Computing, Boston, MA, USA. doi:10.1145/800061.808746 (1983, April 25–27).
- 8. Reiter, E. & Johnson, C. Limits of computation: an introduction to the undecidable and the intractable (Chapman and Hall/CRC, Florida, USA, 2012).
- Moore, C., Russell, A. & Schulman, L. J. The symmetric group defies strong Fourier sampling. Paper presented at IEEE Symposium on Foundations of Computer Science, Pittsburgh, PA, USA. doi:10.1109/SFCS.2005.73 (2005, October 23–25).
- 10. Hallgren, S., Roetteler, M. & Sen, P. Limitations of quantum coset states for graph isomorphism. *J. ACM* 57, no. 6, article no. 34 (2010).
- 11. Kumar, Y. & Gupta, P. External memory layout vs. schematic. ACM T DES AUTOMAT EL 14, no. 2, article no. 30 (2009).
- 12. King, A. & McGeoch, C. Algorithm engineering for a quantum annealing platform. arXiv:1410.2628 (2014).
- 13. Rieffel, E. *et al.* A case study in programming a quantum annealer for hard operational planning problems. *QIP* **14**, no. 1, 1–36 (2015).
- 14. Hen, I. & Young, A. P. Solving the graph isomorphism problem with a quantum annealer. Phys. Rev. A 86, 042310 (2012).
- 15. Vinci, W. et al. Hearing the shape of the Ising model with a programmable superconducting-flux annealer. Sci. Rep. 4, article 4 (2014).
- 16. Gaitan, F. & Clark, L. Graph isomorphism and adiabatic quantum computing. Phys. Rev. A 89, 022342 (2014).
- 17. Lucas, A. Ising formulations of many NP problems. Front. Physics 2, no. 5 (2014).
- 18. Boixo, S., Albash, T., Spedalieri, F., Chancellor, N. & Lidar, D. Experimental signature of programmable quantum annealing. *Nat Comm.* 4, article no. 2067 (2013).
- 19. Choi, V. Minor-embedding in adiabatic quantum computation: I. The parameter setting problem. QIP 7, no. 5, 193-209 (2008).
- 20. Erdős, P. & Rényi, A. On the evolution of random graphs. Publ. Math. Inst. Hung. Acad. Sci. 5, 17-61 (1960).
- 21. Cai, J., Macready, W. & Roy, A. A practical heuristic for finding graph minors. arXiv:1406.2741 [quant-ph] (2014).
- 22. McKay, B. [Section 2.2] Handbook of graph theory, 2nd edition (Chapman and Hall/CRC, 2013).

Acknowledgements

We would like to thank Itay Hen for helpful discussions and suggestions, and Federico Spedalieri for feedback on an early version of the manuscript. O.S. would like to thank Professor Samuel J. Lomonaco, Jr. for insight into the graph isomorphism problem. This material is based in part upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001-11-C-0041. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. Distribution Statement "A" (Approved for Public Release, Distribution Unlimited).

Author Contributions

K.Z. and O.S. conceived the research; O.S. performed mathematical modeling and developed an algorithm for generating an Ising model; K.Z. designed and implemented the solvers, conducted experiments, and

wrote the manuscript text; M.F. and K.Z. supervised the research; M.F. reviewed the manuscript and contributed revisions.

Additional Information

Competing financial interests: The authors declare no competing financial interests.

How to cite this article: Zick, K. M. et al. Experimental quantum annealing: case study involving the graph isomorphism problem. Sci. Rep. 5, 11168; doi: 10.1038/srep11168 (2015).

This work is licensed under a Creative Commons Attribution 4.0 International License. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in the credit line; if the material is not included under the Creative Commons license, users will need to obtain permission from the license holder to reproduce the material. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/