Visualization Tools for Quantum Annealing Researchers

Austin Wallace

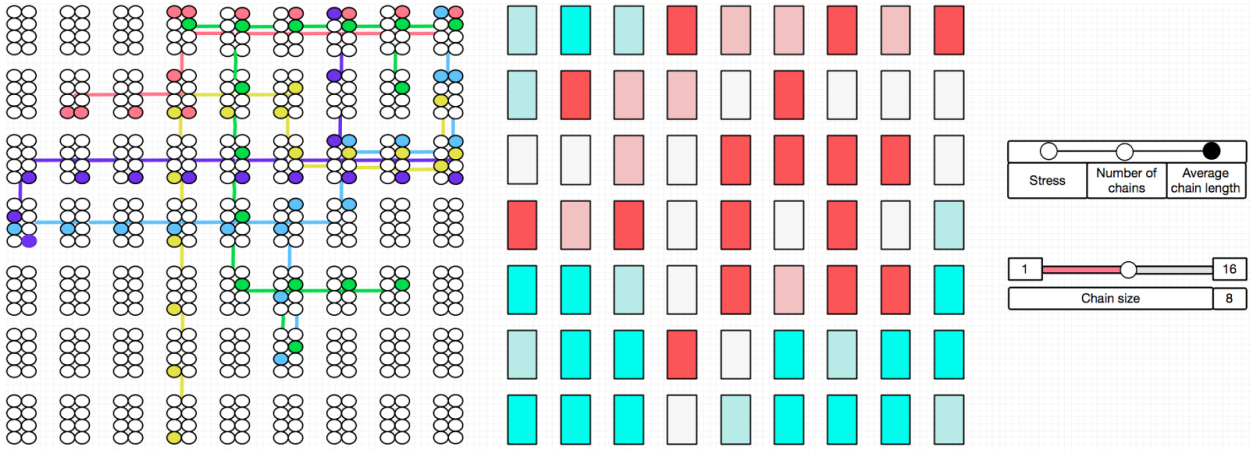


Fig. 1. Broken Chain Vis a system for examining the embedding of a problem onto DWave's hardware.

Abstract—We present an information visualization investigation into the quantum annealing domain. We conduct original research to characterize the data and task abstractions in this field. The task and data abstractions that we explore are used as the basis of a new representation of the Chimera graph, an important part of the current quantum annelaing process. Our redesign reduces the visual clutter often associated with representations of the Chimera graph, while remaining familiar to users. We propose a system for examining an embedding, using our redesigned Chimera graph represenation.

Index Terms— Quantum annealing, graph representation, heatmap, design paper, information visualization, survey

1 Introduction

Quantum Annealing is a burgeoning field that has experienced substantial growth as annealers begin to solve real world problems. As major powers such as Google, Lockheed Martin, and Los Alamos Laboratories continue to become more involved, the depth of literature continues to increase. Despite this growth, work on visualization has lagged behind the development of other tools. While a few core visualizations have become de-facto industry standards, a systematic review of visualization needs and best practices in the field has not been attempted.

This paper will attempt to characterize the existing visualizations in the domain, sourced from prominent papers and research groups. After characterizing the existing visualizations, it will then be possible to extract the best practices and make recommendations to improve or redesign those visualizations. In the case that there are gaps in the literature relative to the tasks accomplishable by visualization, this paper will propose new visualizations based on the best practices established beforehand.

1.1 Contributions

- Precisely characterize available data for visualization in the quantum annealing domain
- Through surveys of domain experts and existing literature, examine the scope of potentially useful visualization tasks
- Characterize existing visualizations in the quantum annealing domain
- Critique and offer recommendations for best practices on existing recommendations
- Propose new visualizations

• *Austin Wallace is the University of British Columbia. E-mail: austeane@gmail.com.*

This report is submitted in fulfillment of CPSC 547 Information Visualization, Computer Science, University of British Columbia, Fall 2015

2 Domain Overview: Quantum Annealing

A quantum annealer is a solver that takes in a weighted, undirected graph, where nodes are qubits (values that end up as 1 or 0), and edge values, known as couplers, represent the correlation between qubits. Quantum annealers and traditional computers both attempt to solve combinatorial optimization solvers, however quantum annealers may have a substantial runtime advantage over classical computers. Graph similarity problems and scheduling problems are both common combinatorial optimization problems faced in industry, and are each NP-hard. Quantum annealing is designed for these types of problems.

A quantum annealer takes a physical representation of the problem and samples from low energy states of that physical system which correspond to good solutions of the optimization problem. That physical system can be thought of as a graph and is made up of qubits, representing variables or nodes, and couplers, representing edges. A qubit is a quantum bit that can be conceptualized as being in both the 1 and 0 states at the same time. A coupler is a link between two qubits that influences whether the connected qubits are the same or different values. To be represented as a physical graph of qubits and couplers, the problem must be first formulated in a specific way, called a quadratic binary optimization problem (QUBO).

The QUBO specifies the local bias for each qubit between -1 and 1, with a positive bias pushing the qubit to end at a value of one and a negative bias pushing the value to zero. It also specifies the bias of each coupler between -1 and 1, with a positive value indicating that the two qubits connected by the coupler are biased to take the same value. Fig. 1 shows a representation of DWave hardware that a QUBO could be mapped onto.

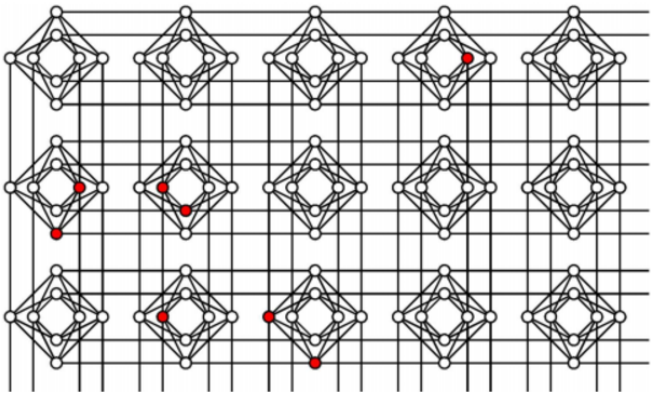


Fig. 1. A representation of the DWave hardware. Each qubit, a circle, is connected by a coupler, a line, to four qubits in its local block, as well as a qubit in each of two neighboring blocks. White qubits are functional, while red qubits are faulty. A QUBO is mapped into a graph with this pattern of nodes and edges. Each unique QUBO will place different biases on each node and edge shown above.

To be solved by a quantum annealer, not only does the problem need to be a QUBO, but it must fit on the available physical hardware. A general QUBO can be represented as a fully connected graph which allows for coupling between any two nodes. However, currently available hardware is not fully connected. The only commercially available quantum annealer, the DWave 2000Q, only allows for connections in a specific way, called the Chimera architecture. Complicating things further, in a given physical machine, some small portion of the qubits will be known to malfunction, and thus are not used. An embedded problem is represented by a graph known as the physical graph, since it encodes information the way it is physically connected in the hardware. The original problem is represented by a graph known as the logical graph, since it represents information as it most logically maps to the variables of the problem. The transition from a logical graph to a physical graph is shown in Fig. 3 [1], where the square-shaped graph is the logical graph and the minor embedding is the physical graph.

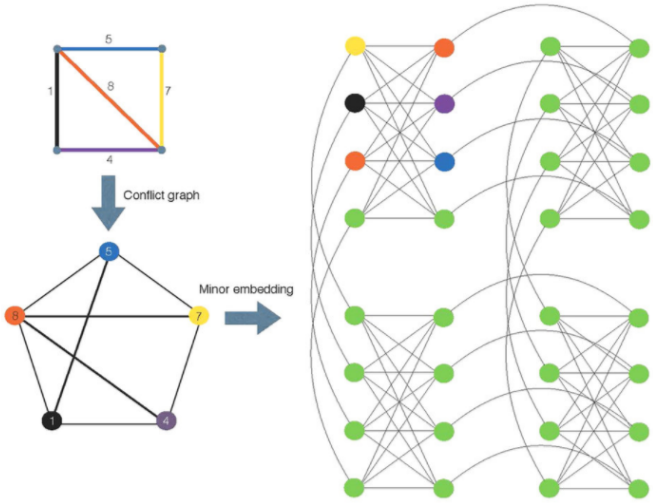


Fig. 3. The process of transforming a logical graph into a physical graph. A problem with 4 nodes and 5 edges is reformulated as a conflict graph containing one node for each of the original edges. The conflict graph is then modified to fit onto the Chimera architecture in the embedding stage as depicted by the square and pentagon shapes on the left of the diagram.

The problem of fitting a general, potentially fully connected QUBO onto the physical hardware is known as embedding and is currently one of the biggest hurdles to solving real-world problems. In the most general case, this is known to be NP-hard [2]. However, embedding onto a Chimera graph can be done in polynomial time, though the time to find an exact embedding can still be prohibitive even for 10 qubits. Thus, embedding is done heuristically and is the subject of active research. Embedding involves finding correspondence for nodes in the original formulation of the problem, called logical qubits, onto the variables of the hardware graph, known as the physical graph. To make that possible on a less connected graph, a logical qubit is often represented as a chain of physical qubits. An example mapping is shown in Figure 4 [3], with logical variables $\tilde{\imath}$ and $\tilde{\jmath}$ each being mapped to a chain containing two physical nodes, i_1 and i_2, and j_1 and j_2, respectively.

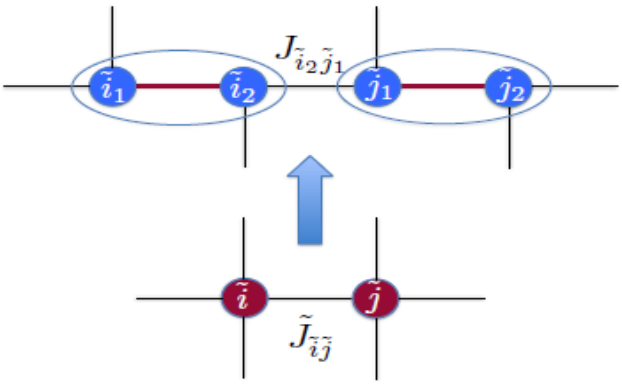


Fig. 4. A mapping from a logical graph (bottom) to a physical graph (top). The two chains have been circled to show that they represent a single, logical value, and it can be seen that the number of edges leaving each variable in the logical graph is equal to the total number of edges leaving its corresponding chain.

After embedding the QUBO, if needed, the quantum annealer solves the problem and returns a solution, which is a configuration of values on the nodes of the physical graph. The QUBO formulation, given this physical solution, returns the associated energy; a lower energy corresponds to a better solution. To find the low energy system, the annealer must attempt to simultaneously satisfy every constraint, with each qubit wanting to be up or down, and each edge wanting to share qubits with either the same or opposite value. The more constraints that are broken, and the strong the broken constraints, the higher the system's energy will be. The probability of the annealer returning a configuration is inversely proportional to the energy of the configuration, so better solutions are more likely to be returned, and equal energy solutions should be found at equal rates.

For all problems on a current quantum annealer, even NP-hard problems, the time it takes to arrive at a solution is on the order of microseconds. However, as it is a probabilistic solver, no solution is guaranteed to be anywhere close to the global minimum energy, i.e. the best solution. Thus, a quantum annealer is usually run on the order of 1000 times for a given problem before the solutions are returned en masse. A problem's difficulty for a quantum annealer can be characterized in terms of how many iterations are required before having a 99% chance of finding the true solution.

Annealing is not a solely quantum algorithm, and there are two major other classes of annealing: Simulated Annealing (SA) and Simulated Quantum annealing (SQA). SA is a classical algorithm run on non-quantum computers that also solves QUBO problems. SA models a problem as a system with temperature, and simulates what happens when the temperature is slowly decreased, eventually finding a local minimum. The advantage of QA over SA lies in quantum phenomena such as quantum tunnelling. Specifically, quantum tunnelling allows a quantum annealer in a given state of qubit values to transition to another, potentially lower, energy state by flipping multiple qubit values at once. This can occur even when the in-between states have an associated temperature that would be prohibitive for SA.

Simulated Quantum Annealing (SQA) attempts to take advantage of these effects, however it is only able to simulate the phenomena that drive QA. While SQA is often slower than QA, and has different underlying processes, it can run a greater variety of problems since it is not constrained by current quantum hardware. SQA is therefore the algorithm of choice when attempting to predict how an algorithm may perform on QA, or on how architectures other than the Chimera graph may affect QA algorithms.

3 Task and data characterization

During my literature search, I found little to no relevant research on how to characterize the tasks and data associated with the field. Therefore, I needed to conduct research with domain researchers to fill in this knowledge gap. I conducted this research with 1QBit as they are the leading commercial software organization in field of quantum annealing. I started with a survey because I wanted to gain a broad overview of the problems faced and solutions used by researchers in this field. After doing the survey, it was necessary to gain richer and more detailed data so face-to-face interviews were conducted with selected survey responses to understand the complexities and nuances of the problems faced.

To begin characterizing the visualization tasks important to domain experts, we prepared a mixed qualitative and quantitative survey for employees of 1QBit, a company that does research on quantum annealing. The survey (see Appendix 1) solicited feedback from 20 researchers and six software developers on their visualization needs. That data was analysed by abstracting the requirements of each domain expert and counting the occurrences of each task mentioned.

The results of the survey made it clear that the needs of the researchers were diverse, however it was not immediately conclusive how they related to each other or how to prioritize them. Many tasks were articulated in different ways, and it was not clear how many were genuinely separate tasks, or whether they were just different approaches to the same task.

The results of the initial survey helped contextualize and direct our focus for what tasks are worthwhile, but further interviews were needed to gather specific requirements. Researchers identified through the survey were interviewed, as well as 1QBit's software development lead. Interviews were conducted in an unstructured style (see Appendix 2). Notes were taken during the interview for later analysis. We conducted six face-to-face interviews on-site at 1QBit to finalize the concrete list of visualization tasks that were important to these domain experts, as well as understand how they were approaching these problems currently. The result of those interviews was a list of tasks and tools, as well as researcher input on potential improvements. We use the results of those interviews to inform the remainder of this paper.

From those interviews as well as the initial survey, we found that there was a meaningful segmentation of both data and tasks into four categories based on the workflow of research in quantum annealing.

1. The original problem (OP)
2. The embedding of the problem (EP)
3. The parameters and properties of the solver (PPS)
4. The results of the solver (RS)

These are the four main sources of data that can be used for visualization, and a majority of the tasks identified deal with exactly one of those, although some tasks connect two categories.

4 Data Abstraction

4.1 Original Problem

All QUBO problems can be abstracted as a graph. A graph representation allows for a wide variety of data from different sources to be abstracted in terms of objects and connections. To represent graphs, a node-link representation is common and is what we will use for the remainder of this paper. This representation encodes objects as nodes, and connections as edges, as in Fig. 2.

These graphs represent an optimization problem to be solved. There are a wide variety of problems which can be represented in this way such as the comparison of two graphs, the finding of semi-prime numbers, object recognition, etc [4].

Each graph can be represented as G=(V,E, V_{labels}, E_{labels}), where V and E are the vertex biases and edge biases of the graph at some index. V and E can reach 2048 and 5600 respectively on the most recent DWave hardware [5]. In practice, due to the limited connectivity of DWave's Chimera architecture, we can often only use on the order of 50 vertices. An example can be seen in Fig. 14 where all 2048 nodes are displayed.

The values on each node are called biases and are given as part of the QUBO problem. A positive value on a node indicates that that node should end up as a one when that problem is solved. The size of a bias determines how strongly solutions with the opposite assignment are penalized. The maximum values for a bias are determined by the hardware, however are usually on the order of plus/minus one.

The values on each edge are also referred to biases. Instead of biasing a node to a specific value, it biases a pair of qubits. A positive value expresses a preference for the two connected nodes to have the same value. A negative value expresses a preference for the two connected nodes to have different values. As with the bias on a node, the size determines the strength of this preference.

V_{labels} and E_{labels} are optional label attributes, giving additional information about each node and edge which is not necessarily passed on to the DWave solver, but which may be important in other ways, such as specifying a way to display the graph with coordinates or colours for each vertex.

For the remainder of this paper, we will refer to the above as the logical or original problem, as well as the logical nodes and edges. We will refer to a bias on an edge as conforming or anti-conforming.

4.2 Embedded Problem

As described in the Domain Overview above, it is not a trivial process to embed a problem onto the DWave hardware. The final physical graph, with the logical variables represented as chains, can be considered derived data. This derived data includes the original edges and nodes, as well as new edges and nodes. As a contrast to the original problem, this can be expressed as $G_{Physical}$=(V,E, $E_{Chain,ID}$). V and E are the same as in the original problem, while E_{Chain} is new. E_{Chain} are edges that are a part of a chain, which means that they are all strongly prefer their connected vertices to conform to each other. All vertices connected by E_{Chain} edges will be biased in the same direction.

We will refer to this as the physical problem or graph, with its constituent parts referred to as physical nodes and edges, specifying when important if physical edges are a part of the chain.

4.3 Parameters and Properties of the Solver

A solver will have its own graph structure. For DWave, that is currently the Chimera structure.

The Chimera architecture is defined by regular blocks of eight qubits; each qubit is connected to four intra-block qubits, as well as two more inter-block qubits.

The two most important parameters are number of samples and anneal time per sample. Number of samples determine the number of times that DWave is independently run on the same problem. Anneal time per sample determines the amount of time that the solver spends during the process of finding the solution. The underlying physics of this annealing process is beyond the scope of this paper, however the relationship between anneal time and performance is not trivial. An examination of the interpretation of anneal time as a parameter is

```
Number of solutions: 14
Best energy: 96.328463
Best configuration: {0: False, 1: False, 2: False, 3: False, 4: False, 5: False, 6: False, 7: False, 8: False, 9:
False, 10: False, 11: False, 12: False, 13: True, 14: False, 15: True, 16: False, 17: True, 18: False}


 All solutions:
Energy = 96.3285, Frequency: 140, Configuration = {x0 = 0, x1 = 0, x2 = 0, x3 = 0, x4 = 0, x5 = 0, x6 = 0, x7 =
0, x8 = 0, x9 = 0, x10 = 0, x11 = 0, x12 = 0, x13 = 1, x14 = 0, x15 = 1, x16 = 0, x17 = 1, x18 = 0}

Energy = 97.02, Frequency: 6, Configuration = {x0 = 0, x1 = 0, x2 = 0, x3 = 0, x4 = 0, x5 = 0, x6 = 0, x7 = 0, x8
= 0, x9 = 0, x10 = 1, x11 = 0, x12 = 0, x13 = 0, x14 = 0, x15 = 1, x16 = 0, x17 = 1, x18 = 0}

Energy = 97.5256, Frequency: 145, Configuration = {x0 = 0, x1 = 0, x2 = 0, x3 = 0, x4 = 0, x5 = 0, x6 = 0, x7 =
0, x8 = 0, x9 = 0, x10 = 0, x11 = 1, x12 = 0, x13 = 0, x14 = 0, x15 = 1, x16 = 0, x17 = 1, x18 = 0}

Energy = 99.307, Frequency: 1, Configuration = {x0 = 0, x1 = 0, x2 = 0, x3 = 1, x4 = 0, x5 = 0, x6 = 0, x7 = 0,
x8 = 0, x9 = 0, x10 = 1, x11 = 0, x12 = 0, x13 = 0, x14 = 0, x15 = 1, x16 = 0, x17 = 0, x18 = 0}

Energy = 220.103, Frequency: 6, Configuration = {x0 = 0, x1 = 0, x2 = 0, x3 = 0, x4 = 0, x5 = 0, x6 = 0, x7 = 0,
x8 = 0, x9 = 0, x10 = 1, x11 = 1, x12 = 0, x13 = 0, x14 = 0, x15 = 1, x16 = 0, x17 = 1, x18 = 0}

Energy = 222.431, Frequency: 16, Configuration = {x0 = 0, x1 = 0, x2 = 0, x3 = 0, x4 = 0, x5 = 0, x6 = 1, x7 = 0,
x8 = 0, x9 = 0, x10 = 1, x11 = 0, x12 = 0, x13 = 0, x14 = 0, x15 = 1, x16 = 0, x17 = 1, x18 = 0}

Energy = 223.152, Frequency: 2, Configuration = {x0 = 0, x1 = 0, x2 = 0, x3 = 1, x4 = 0, x5 = 0, x6 = 0, x7 = 0,
x8 = 0, x9 = 0, x10 = 1, x11 = 0, x12 = 0, x13 = 0, x14 = 0, x15 = 1, x16 = 0, x17 = 1, x18 = 0}

Energy = 225.8, Frequency: 1, Configuration = {x0 = 0, x1 = 0, x2 = 0, x3 = 1, x4 = 0, x5 = 0, x6 = 1, x7 = 0, x8
= 0, x9 = 0, x10 = 0, x11 = 0, x12 = 0, x13 = 0, x14 = 0, x15 = 1, x16 = 0, x17 = 1, x18 = 0}

Energy = 543.182, Frequency: 23, Configuration = {x0 = 0, x1 = 0, x2 = 0, x3 = 1, x4 = 0, x5 = 0, x6 = 1, x7 = 0,
x8 = 0, x9 = 0, x10 = 1, x11 = 1, x12 = 0, x13 = 0, x14 = 0, x15 = 1, x16 = 0, x17 = 0, x18 = 0}

Energy = 544.936, Frequency: 2, Configuration = {x0 = 0, x1 = 0, x2 = 0, x3 = 1, x4 = 0, x5 = 0, x6 = 1, x7 = 0,
x8 = 0, x9 = 0, x10 = 1, x11 = 0, x12 = 0, x13 = 0, x14 = 0, x15 = 1, x16 = 0, x17 = 1, x18 = 0}

Energy = 1054.44, Frequency: 9568, Configuration = {x0 = 0, x1 = 0, x2 = 0, x3 = 0, x4 = 0, x5 = 0, x6 = 1, x7 =
0, x8 = 0, x9 = 0, x10 = 1, x11 = 1, x12 = 0, x13 = 1, x14 = 0, x15 = 1, x16 = 0, x17 = 1, x18 = 0}

Energy = 1058.85, Frequency: 5, Configuration = {x0 = 0, x1 = 0, x2 = 0, x3 = 1, x4 = 0, x5 = 0, x6 = 1, x7 = 0,
x8 = 0, x9 = 0, x10 = 1, x11 = 1, x12 = 0, x13 = 0, x14 = 0, x15 = 1, x16 = 0, x17 = 1, x18 = 0}

Energy = 1059.24, Frequency: 5, Configuration = {x0 = 0, x1 = 0, x2 = 0, x3 = 1, x4 = 0, x5 = 0, x6 = 1, x7 = 0,
x8 = 0, x9 = 0, x10 = 1, x11 = 0, x12 = 0, x13 = 1, x14 = 0, x15 = 1, x16 = 0, x17 = 1, x18 = 0}

Energy = 1768.54, Frequency: 1, Configuration = {x0 = 0, x1 = 0, x2 = 0, x3 = 1, x4 = 0, x5 = 0, x6 = 1, x7 = 0,
x8 = 0, x9 = 0, x10 = 1, x11 = 1, x12 = 0, x13 = 1, x14 = 0, x15 = 1, x16 = 0, x17 = 1, x18 = 0}
```

Fig. 5. A sample solution obtained from DWave on a toy problem with 19 physical variables generated by 1QBit. (Above) is a summary of the solutions, along with the best solution found. (Below) is a list of each solution, how often it occurred, and the dictionary specifying the assignment of values to qubits.

outside the scope of this paper.

4.4 Results of the Solver

Each time a solver is run, it will return a list of solutions. energy as well as a dictionary containing assignments of one or zero to each node. An example set of results from DWave can be seen in Figure 4.

The cardinality of the solution list can range from one to thousands. The cardinality of a dictionary can be up to the size of the physical graph, 2000 in the case of the latest DWave machine. This is small compared to the bits on a computer, but large enough to be competitive with traditional computers on some problems.

5 Task abstraction

A typical researcher has a general workflow for analyzing or solving problems using quantum annealing. This workflow consists of moving the data through the four steps outlined in the data abstraction section. The previously mentioned survey of 1QBit researchers and software developers was consulted in order to list and characterized all tasks within this workflow in terms of the data that they required. These results are presented in Table 2. The current established visualization techniques associated with each task can be placed into one of the following three ranks:

1. Well illustrated by visualization techniques borrowed from fields outside the domain of quantum annealing. Lowest importance.
2. There is work to be done on this topic to bring tools up to best practices in information visualization. Moderate importance.
3. This is an important task with substantial needs unaddressed by the current state of visualization in the quantum annealing domain. Highest importance.

From now on, these will be described as Rank 1, Rank 2, and Rank 3 tasks.

	Task	Op	Ep	Pps	Rs	Rank
1	**Examine the logical graph**	✓				1
2	**Compare distributions of solutions to a known theoretical distribution**				✓	1
3	**Compare algorithm performance as problem size scales**				✓	1
4	**Compare the physical and logical graphs**	✓	✓			3
5	**Discover areas on original graph that require long chains**	✓	✓			3
6	**Summarize the ease of embedding for a logical problem**	✓				2
7	**Compare distribution of degenerate solutions to what is expected from fair sampling**				✓	2
8	**Discover the quality of an embedding**		✓			3
9	**Examine the effects of parameters such as anneal time and temperature on DWave performance**			✓		1
10	**Discover which nodes are candidates to be fixed to improve performance**				✓	2
11	**Discover patterns of broken chains**		✓		✓	3
12	**Inspect patterns of chains for stability**		✓		✓	3
13	**Understand the topology of a problem on the physical graph**		✓			3

In this section, we will examine each rank of tasks, with the rank of each task as listed in the above table.

5.1.1 Rank 1 Tasks

Some areas of the quantum annealing process are not so different from their classical computing counterpart. The data that comes from these tasks are similar to data seen from a classical computation, and thus a variety of good visualizations already exist to represent it. We will give a brief overview, but not discuss each task further.

1: **Examine the logical graph**. Although not discussed in this paper, it is essential for a researcher to be able to examine and manipulate their original input graph. These graphs are of arbitrary shape and have arbitrary properties, and we refer you to the section on graph visualization in this survey paper for more information **[6].**

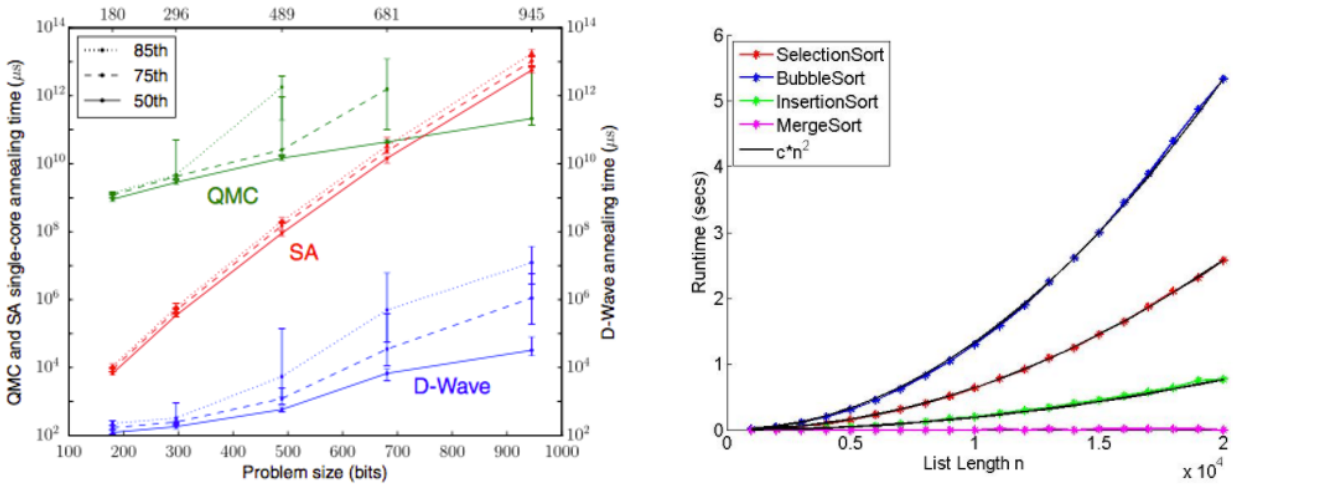


Fig. 6. Line graphs representing scalability of quantum annealing algorithms (left) and classical sorting algorithms (right). These address the comparison of algorithm performance task using similar idioms. The color identity channel is used to identify algorithms, while the position magnitude channel is used to encode performance of a single problem instance. Notice how the left graph encodes more information in terms of the variance in performance of each algorithm.

2: **Comparison of algorithm performance with respect to problem size**. This has been done repeatedly in the classical computing realm, with a very similar data format coming from quantum computations. **The figure [7][8]** shows a visualization used to compare the scalability of different QA algorithms next to a visualization comparing classical sorting algorithm runtimes. In both cases, each algorithm is represented by one or more lines of the same color with the height and slope of the trend line encoding information about the efficiency of the given algorithm.

3: **Examine the effects of parameters such as anneal time and temperature on DWave performance**. This may in fact be the task that is represented by the greatest number of figures in the literature. However, in general, almost all of these figures are some variation of a line chart where one parameter is plotted against another. Matsuda, Nishimori, and Katzgraber's paper is an excellent example of this type of analysis. [9]

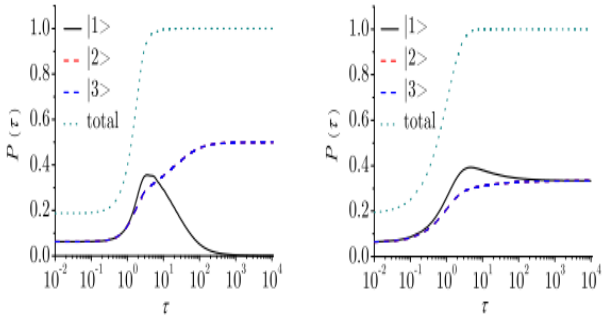


Fig. 7. A plot showing the probability of finding each of three possible solutions for different settings of the annealing time parameter. The total (dotted) line indicates the overall probability of finding a solution for a given annealing time. The three potential solutions are encoded with different colored lines, however both the second and third lines are encoded with the same pattern, and coincide completely. This reflects a poor design choice as the second, red, line is completely occluded, and it is up to the user to infer its position.

4: **Compare distributions of solutions to a known theoretical distribution**. Given a set of solutions, researchers want to see how the distribution of energies correspond to what they expected. Figure [9] is an example of such a distribution graph. There are aspects of this task related specifically to quantum annealing research, in that it is usually a Boltzmann distribution being compared to. It is an important task to consider while designing visualization systems, but as a separate task it is already well accomplished using standard plotting tools like MatPlotLib.

5.1.2 Rank 2 Tasks

Rank 2 tasks again are tasks that draw from the visualization literature. These tasks are at least somewhat important to examine and bring up to the standards of best practices in other fields.

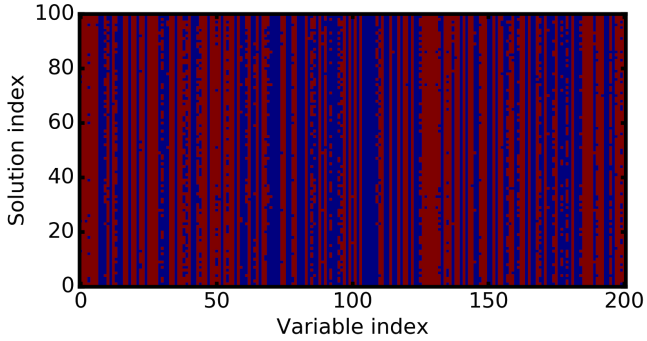


Fig. 8. A barcode graph showing whether a variable is assigned the value of one, blue, or zero, red, for a given solution. The vertical patterns of blue and red indicate that there is some stability in assignments of qubits across solutions. Qubits that are a single value for all solutions found so far may be considered strong candidates for fixing in a subsequent run of the same problem. This idiom does a good job of indicating overall whether a problem is a candidate for fixing qubits, however, it is not easy to make fine distinctions on a single variable level.

1: **Summarize the ease of embedding for a logical problem.** Before proceeding with further work, it is necessary to determine if the problem is feasible to solve on quantum hardware. This task is a first check on that feasibility. This is often done by examining the degree distribution of this problem. This can be done interactively without visualization, but visualization would provide a more nuanced understanding of the difficulty.

2: **Compare the distribution of degenerate solutions to what is expected from fair** It is important for future work on quantum annealers to characterize DWave in terms of how fairly it samples different solutions. For researchers interested in the properties of DWave, an important task is to understand how, why and when DWave deviates from fair sampling. This is important for some researchers, but it is not clear that it is immediately important for most problems.

3: **Discover which nodes are candidates to be fixed to improve performance.** A potential way to improve the performance of a solver is to reduce the space it has to search by fixing variables. Using visualization to is explored during the future work section.

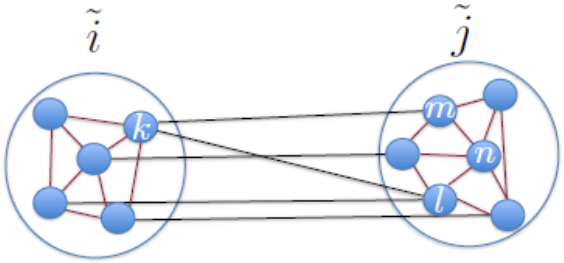


Fig. 9. Schematic of two logical groups (large enclosing circles) representing logical variables. Physical qubits are denoted by blue circles; physical problem couplings are denoted by black lines, chain couplings by red lines. This is used to show how a single logical variable can be represented as many connected physical variables. However, this does not scale well, as the visual clutter grows exponentially in the number of logical variables.

4 and 5: Compare the physical and logical graphs/discovering areas on the logical graph that require long chains. These are two interlinked tasks that both relate to comparing the physical and logical graphs. These are important tasks, and can be effectively accomplished with the linked interaction idiom. One current approach is shown in **figure x**, where every qubit contained in a chain is grouped together and the chain is circled. This however is prone to clutter for even moderate numbers of logical qubits, and alternative ideas will be discussed in the Future Work section of this paper.

5.1.3 Rank 3 Tasks

When evaluating which set of tasks to focus on for this paper, we took into consideration both how important each task is to the user, as well as how much room for improvement there was relative to the current best practices. Through our interviews with domain experts, it seemed that there were already adequate tools for many tasks such as plotting of distributions and logical graphs. However, that was not the case for the embedding process, or anything related to chains, as those are tasks unique to quantum annealing and without much obvious literature to draw from. Thus, we take the best practices from information visualization and apply them to the four following tasks related to chains and embedding:

1: Discover the quality of an embedding
A problem can be embedded using one of many possible embedding algorithms. Metrics can help to determine the best choice of embedding; however, a direct visualization will often reveal more than metrics or textual representations of the graph.
2 and 3: Discover patterns of broken chains; inspect patterns of chains for stability
As we will soon see, the visualization for broken chains is neither well designed nor fully featured. This makes it difficult for researchers to build an intuition and eventually a formal characterization of what patterns of chains contribute to a high proportion of broken chains. It is also not really possible to meaningfully compare patterns of chains, or closely inspect a single chain to determine its stability.
4: Understand the topology of a problem on the physical graph
While there are multiple representations of the physical graph which allow for micro-level understanding of topology, if one is willing to spend the time, the views that give an overview on the macro-level topology are lacking. This will be expanded on in the immediately that immediately follows.

6 Proposals for New Visualizations

One of the most important aspects of visualizing the current state of the quantum annealing is visualizing a physical graph. Currently, the only physical graph layout is the Chimera graph described in the domain overview, with blocks of eight nodes connected to neighboring blocks. While there are fewer degrees of freedom in the design space, the connections in a chimera graph are prone to visual clutter, as illustrated in **Figure [10]**. **Figure 5** shows the current most common representation of the Chimera structure.

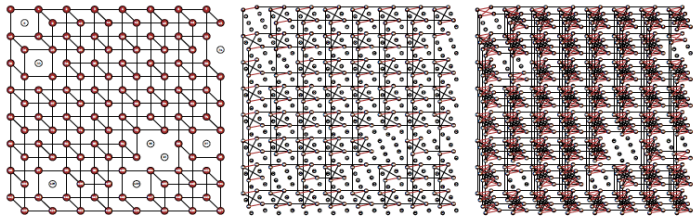


Fig. 10. Three graphs show different ways of representing the same problem. The logical graph (left) is the problem to be embedded. The other two images are physical graphs (middle and right) showing the results of two different embedding strategies onto a Chimera graph. As more edges are introduced into the graph, the level of visual clutter escalates rapidly.

To begin to redesign this idiom, we established the axes for our design space. These axes assume that we arrange nodes in blocks. One motivation for doing so is that each of the eight nodes in each block contains four edges within the block, with only two connections externally. This indicates that even when not constraining them to be together, most force directed algorithms would keep blocks together in an irregular way. This representation is further backed up by the fact that these qubits are physically grouped on the DWave chip.

By forcing this constraint, we open the door for more idioms that can rely on that assumption and a more thorough discussion of the design space. Throughout the rest of this paper, we consider qubits to be

nodes, however that is not the case in every design decision, so we will refer to them directly as qubits in the following section.

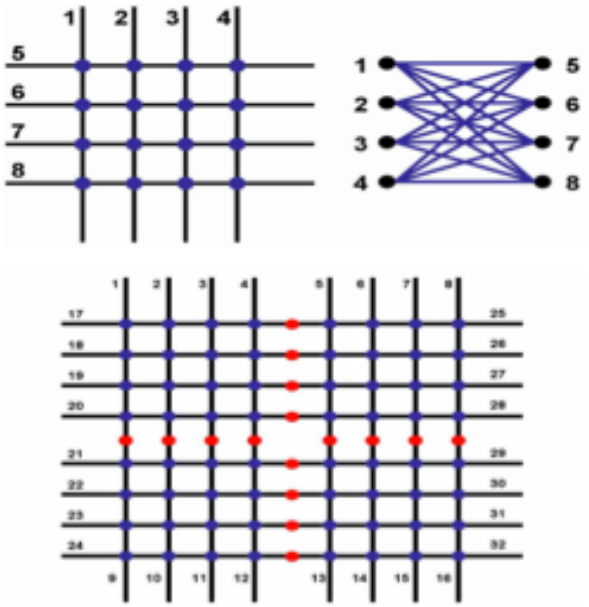


Fig. 11. Two representations of an 8-qubit block. (Top left) The variables are shown as black lines, with couplers denoted by the set of blue points where the lines intersect. (Top right) The variables are the black dots, and the couplers are shown as the blue connecting lines. (Bottom) The larger graph shows 4 blocks with a total of 32 variables. Red circles represent couplers between blocks.

We have determined that set of axes that describe the design space of the physical graph:

1. Representation of an individual qubit
2. Representation of an intrablock coupler
3. Representation of an interblock coupler
4. Arrangement of qubits in a block
5. Arrangement of blocks

The dominant representations are as follows:

6.1.1 Commonly Used Representation

See Fig. 2 (right)
Qubit: Point-Circle
Internal coupling: Connection mark (straight line)
External coupling: Connection mark (curved line)
Arrangement of nodes in a block: 2 by 4 block with each column fully connected to opposite column
Arrangement of blocks: Regular spaced grid with each block connected to its four neighbors, with the right column of each block having horizontal external edges, and the left hand column having vertical external edges.

The above is a natural set of design decisions, but one that creates visual clutter and reduces clarity, especially in inter-block edges.

Alternative design decisions include:

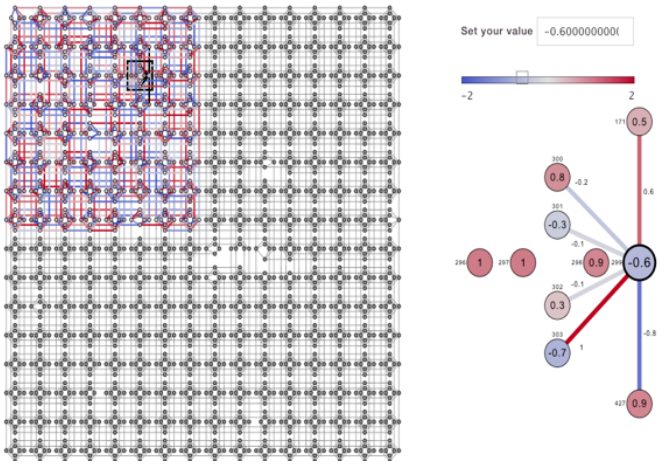


Fig. 12. The the overview in DWave's online API. (Left) DWave uses the diamond representation, and encodes conforming couplings as blue, with anti-conforming couplings as red. A translucent black box encloses the user's selection. Any encodings on the qubits themselves are cluttered. (Right) The detail view for the selected block of qubits. The values on each coupler and edge are specified, with the color encodings specified by the red-blue colormap.

6.1.2 Diamond Representation

Qubit: Point (circle)
Internal coupling: Connection mark (straight line)
External coupling: Connection mark (curved line)
Arrangement of nodes in a block:
A four high and four wide cross in the vertical and horizontal directions. Each node in the horizontal line is connected to each node in the vertical line, and visa versa. Nodes in the vertical and horizontal lines do not share an edge.
Arrangement of blocks:
Regular spaced grid with each block connected to its four neighbors, with the right column of each block having horizontal external edges, and the left hand column having vertical external edges.

This representation allows for more straight edges external to the qubit, at the cost of a slightly less regular arrangement. This is in general a better representation than the traditional one, however it do

6.1.3 Qubit as Edge Representation

Qubit: Connection mark (straight line)
Internal coupling: Point mark (circle)
External coupling: Point mark (circle)
Arrangement of nodes in a block:
A four by four grid of nodes representing couplers, with each connected to its four neighboring couplers by qubits.
Arrangement of blocks:
A regular square grid with connections between vertical and horizontal neighbors.
This is an excellent set of design choices for reducing visual clutter. However, having the qubits as edges is unintuitive. The purpose of a solver is to assign values to qubits, and representations that do not place them as the focus of the visualization are penalized for that. An inversion of nodes and edges would be jarring for people who are used to qubits as nodes. For certain applications, the tradeoff between parsimony and comfort may be worth it, and this representation should be kept in mind for idioms with a focus on couplers.

After considering the design decisions in present in the literature existence, we have decided to create a new idiom for use when visualizing larger graphs. The current qubit-as-node visualizations do not scale beyond the hundreds of nodes into the thousands. Past that point, it is difficult to discern the internal structure of the qubits, and there is a substantial amount of visual clutter.
So we set out to design something that would have the clean representation of the qubit-as-edge representation, but without the potentially confusing representation. To get there, we sacrificed information on internal connectivity.

6.1.4 Overview Representation

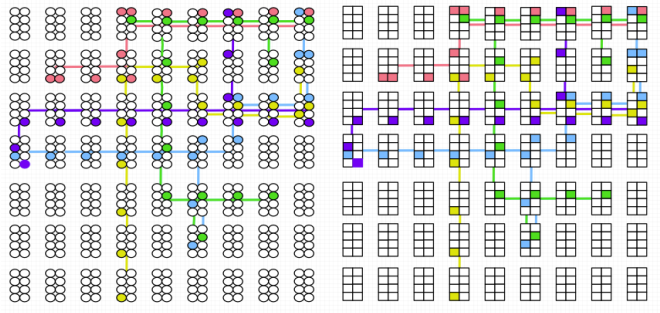


Fig. 12. The the overview in DWave's online API. (Left) DWave uses the diamond representation, and encodes conforming couplings as blue, with anti-conforming couplings as red. A translucent black box encloses the user's selection. Any encodings on the qubits themselves are cluttered. (Right) The detail view for the selected block of qubits. The values on each coupler and edge are specified, with the color encodings specified by the red-blue colormap.

Node: Point (ellipse)
Internal edge: Not represented
External edge: Connection mark (straight line)
Arrangement of nodes in a block: 2 by 4 block, no spacing
Arrangement of blocks: Regular spaced grid with each block connected to its four neighbors

This representation treats each block as a single entity, ignoring internal connections entirely. By doing this, we are able to increase the relative size of each qubit. This is important for scalability and for continuing to encode information in each qubit, even in a 2000 qubit overview as is the case for the current DWave annealer.

For connections between blocks, we decided to not tie the connection to a single qubit, but rather to the block as a whole. In the most widely used representation, external edges often cross as they go from the right column of nodes to the left to connect with the neighboring block's right column of nodes. Our representation avoids this, and is able to encode all external connections with straight lines. In addition, lines are laid out so that a line entering a block and exiting it from the opposite side will do so at the same location. This has the benefit of making lines easier to follow for users, especially across multiple blocks. The colors will need to be optimized to ensure that no two lines of the same colour enter the same block. Alternative identity channels to colour such as shape will also be considered.

We considered encoding each qubit as a rectangle instead of an ellipse, as that would again maximize the size of each qubit. However, this may be too jarring for users, as it veers from the traditional representation. As future work, we will gather feedback on this decision.

We considered encoding internal connections by coloring quarters of the internal semi circle of the node. Each quarter would represent whether a connection was present. This was not discernable at an overview and caused too much visual clutter to be worthwhile. We are not recommending this representation beyond as an overview. However, if this representation is extended for a more close-up view, encoding connections with quadrant will be explored further.

We propose that as an overview, this representation is much stronger than the alternatives at understanding the connectivity of the physical graph as a whole. We propose that it also requires fewer and simpler cognitive tasks, such as following straight lines versus curved lines. Finally, we propose that it accomplishes these things while remaining easily recognized and understood by researchers used to the common representation.

The remainder of this paper will discuss different multi-view visualizations that incorporate this overview representation of the physical graph.

6.2 Broken Chain Vis

As described in the domain overview, a single node in the original problem may be represented as multiple linearly connected nodes in the physical problem. This is because nodes in the original problem may have arbitrarily many edges, while a single node in the Chimera embedding may only have six edges. These nodes will all be biased towards the same outcome, either a one or a zero, and the edges will have strong conforming biases. These linearly connected nodes are referred to as a chain. Despite strong biases, these chains regularly break in real-world problems, and examining the reasons why is an area of active research. How prone a chain is to breaking can be conceptualized as stress in the same way as a stiff iron chain. If different links in the chain are being pulled opposite directions, then it creates areas of high stress and increases the likelihood of breaking.

The leading components of stress on a chain are chain length and chain pattern. Maximum chain length was believed by 1QBit researchers to be the most important metric to minimize during the embedding process, but it is now believed that it is not just the maximum chain length that matters and that certain patterns of chains are more likely to result in the breaking of chains. For example, because chains are not local, one end can be strongly biased towards being up, while the other is biased towards being down.

The key goal for this idiom will be to identify high-stress areas in an embedding, where stress is a combination of chain length and chain patterns.

For this, we will follow Shneiderman's helpful mantra of overview first, details on demand [11]. This means that the user should be given a helpful overview without overwhelming them with details. Only when the user asks for it should we give them details.

To begin, we examine the closest approximation to a solution being used in practice, DrawChimera, an internal tool at 1QBit.

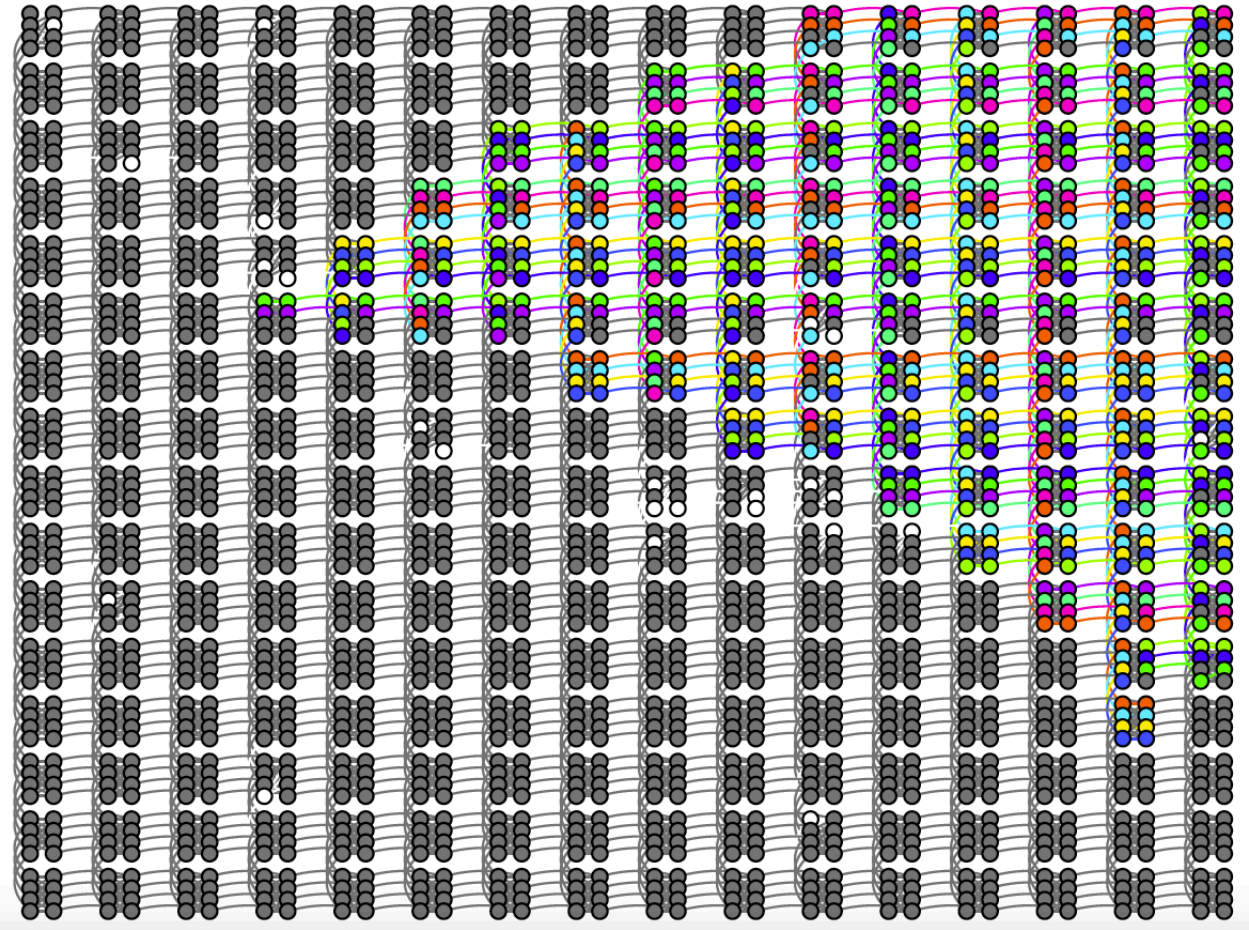


Fig. 14. DrawChimera is a node-link diagram with qubits encoded as nodes and couplers encoded as links. The colour channel is used for to encode all of the following: Whether a node is broke, white in this example; whether a node is in disuse, grey; and what chain each used node is in, encoded with other colors.

DrawChimera is a static idiom that attempts to both provide an overview along with the detail of which chain a given node belongs to. However, it is a poor idiom on many fronts. DrawChimera uses eight hues of similar saturations to encode chain membership, however there are very often n>>8 chains in the embedding. The decision to encode multiple chains with the same colour means this could lead to difficulty for users in identifying the entirety of a single chain.

Chain length is not directly encoded. For example, in Fig. 5 it is not at all obvious that every chain pictured has a length of 14.

Further, while DrawChimera allows the user to instantly recognize the areas of the solver being used, that is not a particularly important task, and it is hard to differentiate anything in the region that is being used. In Fig. 7, for example, it is difficult to perceive patterns within the top-right region of the graph.

A key flaw in DrawChimera as it relates to the workflow of a researcher is that it does not attempt to encode the stress on the given chains. Thus, it does not effectively assist the user in make judgments pertaining to the quality of the chosen embedding.

To improve on DrawChimera, we must first design a clear overview that accomplishes the following tasks.

- Summarize the chain lengths present in an embedding
- Identify areas of the graph for further investigation
- Provide an at a glance understanding of where the largest chains are

As with DrawChimera, we begin with a representation of the physical graph, however we will use the overview representation of the physical graph described in the previous section.

The overview system will utilize a heatmap view alongside a view that directly encodes chains, as shown in **Figure 7**. A heatmap shows the concentration of some property in each region of the graph relative to the average. The heatmap will encode information about the chains present in the embedding based on a criterion chosen by the user. The direct encoding view will represent the longest chains as default, and the chains of interest when indicated by interaction with the heatmap view.

6.2.1 Heatmap View

The Chimera graph lends itself to a natural and meaningful segmentation for the heat map; the stress on a chain comes from the direct connections to other chains, and connections are densest within a block. Taking advantage of this natural structure, we will encode a single value for each block of eight nodes.

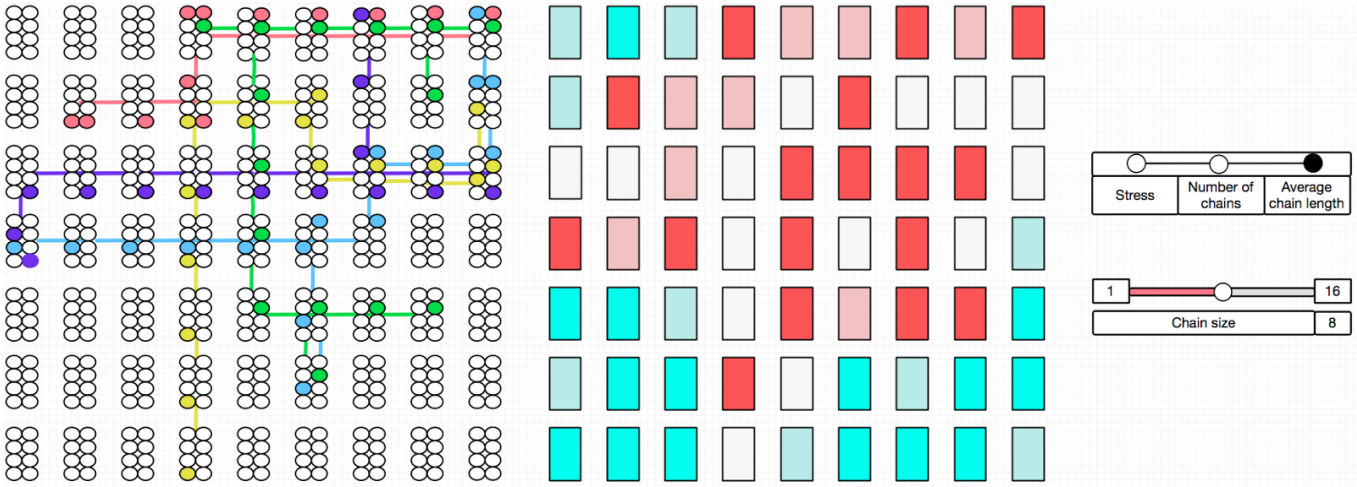


Fig. 15. The direct encoding view (left), the heatmap view (middle), and the controls (right) of the proposed Broken Chain Vis. The direct encoding view shows chains above a certain length and is designed so that chains are easy to follow. For example, it is easy to follow the chain encoded in yellow from the bottom-left up, to the right, down, to the right again, and finally finishing one block below the top right block. The heatmap view here shows the average chain length for the nodes in each block, with red being dense and blue being sparse. This allows a quick overview of which blocks contain many nodes belonging to long chains. The sliders control what is encoded in the heatmap (top) and the threshold for chain viewing (bottom).

There will be three toggleable options regarding what is encoded:

1. Chain-length density
2. Chains per block

For chain-length density, we simply average the lengths of the chains that each node in the block belongs to. A chain may have multiple nodes in a single block, and they will be treated as separate nodes for the purpose of the calculation of the average performed in the visualization.

Chains per block is a simple count of the number of separate chains in a block. A block that has a single chain making up four of its nodes will likely be less stressed than one where there are eight separate chains present.

6.2.2 Direct Encoding View

The second view will utilize the overview representation of the physical graph as described in the Overview Representation section. It is similar to the DrawChimera in that it directly encodes chains. However, it only encodes chains whose length are above a threshold which is determined by the user. By only showing the longest chains, we avoid the visual clutter of DrawChimera while retaining useful information. This is the at-a-glance overview of long, chains that was missing in DrawChimera.

In addition, links unused by the longest chains will not be shown, where in DrawChimera they were displayed but greyed out. When viewed this way, more attention is called to the important chains, and it has the benefit of removing information that is not relevant. Work by us still needs to be done on ensuring that chains are easily followable even at scale.

6.2.3 Summary Statistics

In a smaller view of ours on the same screen (not pictured), a simple histogram would encode the distribution of chain lengths in the embedded graph. This will quickly show the user if there is a diversity in chain lengths, with a few outlier chains taking up many nodes, or whether, like in Fig. 5, most chains are of similar length. This benefits the user by showing the overall distribution of chain length, even when they are only seeing the longest chains.

6.2.4 Interaction

Using the linked highlighting interaction idiom, when moussing over a block in the heatmap view, all of the chains that pass through that block would be selected and highlighted in the direct encoding view. Clicking on the block would grey out all of the other chains, while persistently highlighting the associated chains until deselected. Multiple blocks may be selected at once.

In addition, interaction would enhance the histogram. When blocks are selected, a second translucent histogram of a different color would be overlaid on the original histogram showing the distribution of chain lengths for the selected chains.

This interaction allows for easy and immediate inspection of chains associated with high stress areas identified through the heat map view. It would show immediately chains that correspond to a high stress area, and how their lengths differ from the general distribution.

6.2.5 Detailed view

Detailed views are used to give the user more information on demand. Once chains of interest are selected, a user can right click on a chain to enter a detailed view on that chain. An example of a detailed view is shown in Figure 8.

The proposed detailed view would feature a single horizontal chain, regardless of the chain's orientation in the overview, with nodes and edges represented as point and connection marks as usual. Branches in chains would be directed perpendicular to the main chain. This will reduce crowding on branching chains.

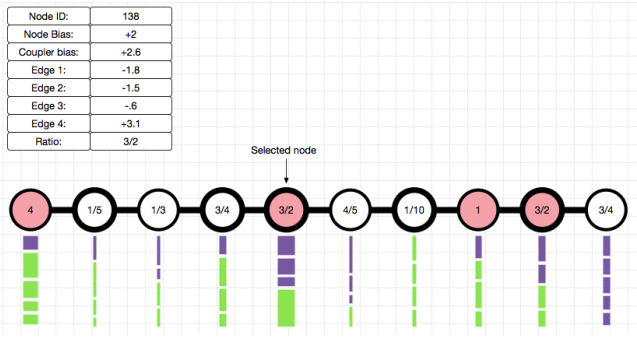


Fig. 15. The detailed view of the Broken Chain Vis. Each node in the chain is displayed in a horizontal line. Each node and link have a border thickness which represents the strength of the biases on them. The segmented columns underneath each node represent the external edges of each node. Here, each internal node has four external edge, while the two bookending nodes have five connections to non-chain nodes. The width of the segmented column represents the aggregate strength of the node's links to non-chain nodes. This allows the user to compare how at risk each node is for breaking the chain. The ratio of the strength of the external to the internal edge biases is directly shown on each node. A number above one means that the external edge biases are potentially stronger, which means that the node may be at risk for breaking the chain, and that is encoded with the color red. When a node is selected, summary statistics (top) show information about the biases which affect that node, as well as the id of the node. block. The heatmap view here shows the average chain length for the nodes in each block, with red being dense and blue being sparse. This allows a quick overview of which blocks contain many nodes belonging to long chains. The sliders control what is encoded in the heatmap (top) and the threshold for chain viewing (bottom).

The chain will have biases potentially of varying strength, but all in the same direction. So whether it is biased to 1 or 0 is immaterial. The border thickness of the node will indicate the strength of the bias. Similarly, for intra-chain links, the bias will be to conform, however the thickness of the line will encode the strength. The same border and edge thicknesses map to the same bias value. This allows for easy and accurate comparison of relative bias strengths.

The external edges will be represented as a single edge leaving each node. This collective edge would have the thickness equal to the total strength of the edge's couplings, regardless of whether they are conforming or anti-conforming couplings. The length of the edge would be broken up into blocks with equal width and length proportional to the relative strength of each separate edge. The total length of each edge would be the same, though some edges may have more components than others. This means that we are using the strong magnitude channel of length for mark comparison.

We emphasize that while it is possible for each extra-chain node to be pulling in directions that maximize the stress, this cannot yet be known. Instead, we concern ourselves with which node has the highest potential to be pulled. Each node will have written on it the ratio between the intra-chain bias strength and the extra-chain bias strength, with a higher ratio meaning that it has the potential to feel more stress from outside of the chain. In other words, the stronger the external strength, the more at risk the node is of breaking the chain

When clicking on a node, the user will be presented with the specifics regarding the index of the node as well as the strengths of all the biases on the node.

6.2.6 Scenario of Use

This view would be used by a researcher that has embedded their problem, and wishes to see the results of that embedding. After uploading the data, they would see many chains as well as the heatmap. They would select a chain-length threshold high enough that they can easily follow the longer chains across blocks. Looking at the heatmap, they would identify a block that is particularly dense, and investigate the longest chain in that block by clicking on it. After seeing the detailed view, they realize that five consecutive nodes in that chain are at very high risk of being broken. After seeing this, the researcher realizes that this is not an acceptable embedding, and chooses another algorithm to create a new embedding of their problem.

Overall, this proposed system accomplishes the Rank 3 tasks identified in the task abstraction relating to chains, **inspect patterns of chains for stability** and **discover the quality of an embedding.** It has the potential to be an effective tool to aid in investigating embedding strategies. Validation of this system's effectiveness will be conducted as future work by us at 1QBit.

7 Future Work: Views to utilize overview representation

The following are potential other use cases for the overview representation. They are not fully realized systems and have not been wireframed. However, they do highlight the wide applicability of the representation.

7.1 Linked highlighting between the physical and logical graphs

A relatively simple and oft requested visualization task is to understand the correspondence between a physical and logical graph. To accomplish this, we utilize the overview representation, as well as a representation of the logical problem.

For the original problem, the size of the graph is up to ~2000, but as the edge density in the original graph increases, the potential size decreases as the limit on edges of ~5000 is reached. For example, the upper bound on embedding a fully connected graph is no more than 100, and will often be much smaller. Due to these constraints on problem size, we can be comfortable using a simple force directed layout for the original graph. Users will also be able to upload their original graph with pre-specified two-dimensional node coordinates. Work into optimizing the planar embedding of an arbitrary graph is outside the scope of this paper.

For the physical problem, we use the overview idiom with in-use nodes in use filled in, as well as chains between blocks. This overview representation is preferable for large problems for the reasons specified above, as it abstracts away already incomprehensible details.

When some subsection of the original graph is highlighted, the chains corresponding to those logical nodes would be highlighted in the physical graph as well in the same way as in the broken chain vis. The converse is also true: Selecting nodes in the physical graph highlights nodes in the logical graph.

This simple design allows for responsive linked highlighting, and enables users to investigate how features of their logical problem map to the physical problem.

7.2 Fixing of qubits

As discussed in Rank 2 of the task abstraction, a visual representation is useful in determining which qubits to fix when running a problem on DWave in the future. An easy way to extend the overview representation to deal with this problem is to fill in nodes should be fixed, given a threshold of how stable each qubit is across solutions. The threshold would be specified via a slider in the view.

As nodes are fixed, their corresponding inter-group edges would be removed from the graph, highlighting areas that are still densely connected even for relatively fixed graphs. This allows for a simple way to see which areas of the graph are more amenable to being fixed, and since variables are fixed by coupler and not by qubit, the intra-block edges are not missed.

7.3 Energy histograms and fair sampling

Our proposed overview representation does not explicitly use information from the results of the solver, however the views which utilize it are often extensible to utilize solutions and that will be done as future work at 1QBit.

It is not particularly difficult to extend the above linked highlighting system to include solutions. With information on which chains were broken, instead of coloring the logical nodes by the length of the associated chain, we would color the node by the percentage of the time the associated chain broke in the solution set.

Additionally, we would display a histogram of the energy levels returned by the solver.
One of the tasks was to gain information on whether DWave behaved as a fair sampler. A simple check of fair sampling is to see if each solution at the same energy level is seen at similar rates. Thus, we would encode each bin in the histogram with a color designating how far it deviates from a fair sampler. Clicking on a bin associated with an energy level would expand the bin in a separate view and show a bar chart which indicates how often each solution with that energy level occurs. In a fair sampler, they would occur equally often. Clicking on a single solution in that bar chart would cause that solution to be represented on the overview representation.

Scaling in number of bins is not an issue, as generally there are less than 10 energy levels returned with any significant frequency. This view would give a quick idea of the energy distribution of the set of solutions, an important task in visualization.

8 Conclusion

Current methods of visualization are useful for gaining an intuition on the problems solved by quantum annealing researchers, however they are also problematic in the overwhelming amount of information presented at one time. Original research was used to guide this paper to improve on this visualization and give users an idea of the overall structure of the graph. At the same time, we allow the user to investigate areas of interest, providing them with even more detail through interactive layering of information in different views.

Quantum annealing is a field with specific and novel challenges in visualization. However, it is served well by drawing from other fields and from information visualization in general, where best practices can be learned and expanded on. We discussed the data available for visualization, as well as an overview of potential tasks in the field along with some examples of how they have been accomplished. As our main contribution, we proposed a new overview representation for a physical graph where a less cluttered approach was taken. We then showed how it could be used in an integrated system of views by domain experts. Finally, we described future uses of this view.

Visualization will continue to grow in importance in this domain, where intuition is still key to many aspects of research. Reducing the barriers to insight and understanding will be key focuses of visualization research in the field. The surface of the potential for visualization work is barely being scratched, as is the potential of the field in general.

Acknowledgments

The author wishes to thank 1QBit for their eager support during the research process for this paper. In addition, thanks to Professor Tamara Munzner for the feedback, insights, and forceful pushes that got this paper on track. Finally, thanks to Katie Olfert for the invaluable help in multiple aspects of this paper.

References

[1] C. Wang, H. Chen, and E Jonckheere, “Quantum versus simulated annealing in wireless interference network optimization,” Scientific Reports vol. 6, pp. 25797, May 2016.

[2] J. Cai, W. G. Macready, and A. Roy. "A practical heuristic for finding graph minors." *arXiv preprint arXiv:1406.2741* Jun. 2014.

[3] W. Vinci et al., "Quantum annealing correction with minor embedding," *Phys. Rev. A* vol. 92, no. 4 pp. 042310, Oct. 2015.

[4] Applications: Helping Solve Some of the Most Difficult Challenges, D-Wave, Burnaby, BC, pp. 1-12.

[5] The D-Wave 2000Q™ Quantum Computer Technology: Overview, D-Wave, Burnaby, BC, pp. 1-12.

[6] S. Liu *et al*., "A survey on information visualization: recent advances and challenges." *The Visual Computer* vol. 30, no. 12, pp 1373-1393, 2014.

[7] V. S. Denchev *et al*., "What is the Computational Value of Finite-Range Tunneling?." *Phys. Rev. X* vol. 6, no. 3, pp. 031015, 2016.

[8] A. Jepson, *Introduction to Sorting*. 2007. Retrieved from http://www.cs.toronto.edu/~jepson/csc148/2007F/notes/sorting.html

[9] Y. Matsuda, H. Nishimori, and H. G. Katzgraber. "Ground-state statistics from annealing algorithms: quantum versus classical approaches," *New J Phys* vol. 11, no. 7, pp 073021, 2009.

[10] K. Karimi *et al*., "Investigating the performance of an adiabatic quantum optimization processor." *QIP* vol. 11, no. 1, pp 77-88. 2012.

[11] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pp. 336-343. IEEE, 1996.

[12] B. Altshuler, H. Krovi, and J. Roland, “Anderson Localization Makes Adiabatic Quantum Optimization Fail,” *Proc. Natl. Acad. Sci. U.S.A.* vol. 107, no. 28, pp. 12446-12450, 2010.

[13] E. Farhi *et al*., “Performance of the quantum adiabatic algorithm on random instances of two optimization problems on regular hypergraphs,” *Phys. Rev. A* vol. 86, no. 5, Nov. 2012.

[14] Knysh, Sergey. "Computational Bottlenecks of Quantum Annealing," *arXiv preprint arXiv* 1506.08608, Jun. 2015.

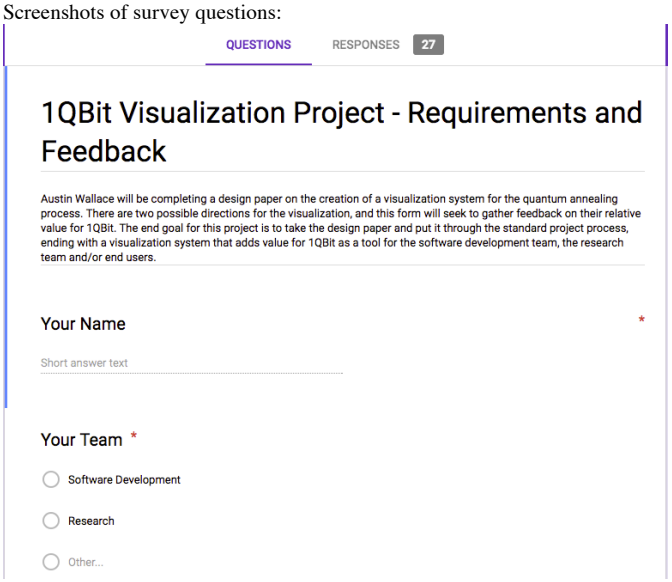
[15] E. Boyda *et. al*., “Deploying a quantum annealing processor to detect tree cover in aerial imagery of California,” *PLoS ONE* vol. 12, no. 2, Feb. 2017.

[16] M. Benedetti *et al*., "Estimation of effective temperatures in quantum annealers for sampling applications: A case study with possible applications in deep learning." *Phys. Rev. A* vol. 94, no. 2, pp. 022308, 2016.

[17] I. Adler *et al*., "Faster parameterized algorithms for minor containment," *Theor. Comput. Sci.* vol. 412, no. 50, pp. 7018–7028, 2011.
[18] *Qubist*. Burnaby: DWave, 2017.

APPENDIX 1

Screenshots of survey questions:



QUESTIONS RESPONSES 27

1QBit Visualization Project - Requirements and Feedback

Austin Wallace will be completing a design paper on the creation of a visualization system for the quantum annealing process. There are two possible directions for the visualization, and this form will seek to gather feedback on their relative value for 1QBit. The end goal for this project is to take the design paper and put it through the standard project process, ending with a visualization system that adds value for 1QBit as a tool for the software development team, the research team and/or end users.

Your Name *

Short answer text

Your Team *

- Software Development
- Research
- Other...

QA Result Visualization

Currently, there is little ability to visualize the results of quantum annealing.

This project would take the real results from an annealer like DWave's machine, over many anneals, and attempt to visualized those results. Possibilities exist for looking at an overview, an aggregation of all the anneals, correlations of qubits, etc.

This may assist in investigating errors, or gaining a better understanding of these processes.

I often work with the DWave results:

	1	2	3	4	5	
Never	○	○	○	○	○	Very Often

A tool that allows me to better visualize quantum annealing results would be valuable to my work

	1	2	3	4	5	
Not at all valuable	○	○	○	○	○	Very valuable

A tool that allowed me to better visualize the SQA process would be valuable to my work

	1	2	3	4	5	
Not at all valuable	○	○	○	○	○	Very valuable

What would you use a tool like this for?

Long answer text

What specific features would make a tool like this useful for you?

Long answer text

General Comments

Description (optional)

A project similar to the one outlined above is important for 1QBit to pursue in some form:

	1	2	3	4	5	
Unimportant	○	○	○	○	○	Critical

Do you have any further comments or suggestions?

Long answer text

Are you interested in us following up with you directly?

- Yes
- No

What would you use a tool like this for?

Long answer text

What specific features would you like this tool to have? Be as imaginative/descriptive as possible.

Long answer text

SQA Visualization

The first option for the visualization is a tool that helps to visualize the quantum annealing process, via quantum annealing.

By being able to sample the values of the SQA process, the visualization would be able to show how the qubit values evolve over time, either for a single run, or the usual trends averaged over multiple runs.

I often use Simulated Quantum Annealing in my work

	1	2	3	4	5	
Never	○	○	○	○	○	Very Often

Screenshot of an answer:

QA Result Visualization

Currently, there is little ability to visualize the results of quantum annealing.

This project would take the real results from an annealer like DWave's machine, over many anneals, and attempt to visualized those results. Possibilities exist for looking at an overview, an aggregation of all the anneals, correlations of qubits, etc.

This may assist in investigating errors, or gaining a better understanding of these processes.

I often work with the DWave results:

	1	2	3	4	5	
Never	○	○	○	◉	○	Very Often

A tool that allows me to better visualize quantum annealing results would be valuable to my work

	1	2	3	4	5	
Not at all valuable	○	○	○	◉	○	Very valuable

What would you use a tool like this for?

get intuition about the structure of the near optimal solution

What specific features would you like this tool to have? Be as imaginative/descriptive as possible.

since dwave will return bunch of near optimal solution, I like to see the frequency of time a variable is -1 or 1

APPENDIX 2

Sample of notes from 1QBit unstructured interview with:
“Logical heatmap very useful as well

Length of chain, lots of metrics
Pattern of chains is more important

For example, many chains connected to one end points of a chain.
One side pulls to 1, one to -1

Research on embedding is a lot of intuition”