Solving the Bridges of Königsberg problem using the D-Wave's quantum annealing processor

D-Wave Systems

June 6, 2016

Let $G = (V, E)$ be an undirected multigraph (a graph in which more than one edge may appear between the same two vertices). An *eulerian path* is a walk in G that uses every edge in E exactly once. A graph has a eulerian path if and only if at most two vertices have odd degree, so there is a very fast way of determining whether or not G has a eulerian. However, determining the number of eulerian paths in G is a harder problem.[1] This note will show how to map this problem to D-Wave's quantum annealing hardware, in a way that potentially allows the number of paths to be counted.

In order to solve the eulerian path problem using D-Wave's hardware, we must encode it in a *quadratic unconstrained binary optimization* (QUBO). That is, the D-Wave hardware solves problems of the form

$$\min_{x_1, x_2, \ldots, x_n} \sum_{1 \le i \le j \le n} q_{ij} x_i x_j,$$

where each x_i is a $\{0, 1\}$ variable, the coefficients q_{ij} are real numbers.[2]

[1] #P-hard.

[2] Actually the D-Wave hardware solves an Ising model, which uses $\{\pm 1\}$ variables instead of $\{0, 1\}$ variables. But it is easy to transform from QUBOs to Ising models and back.

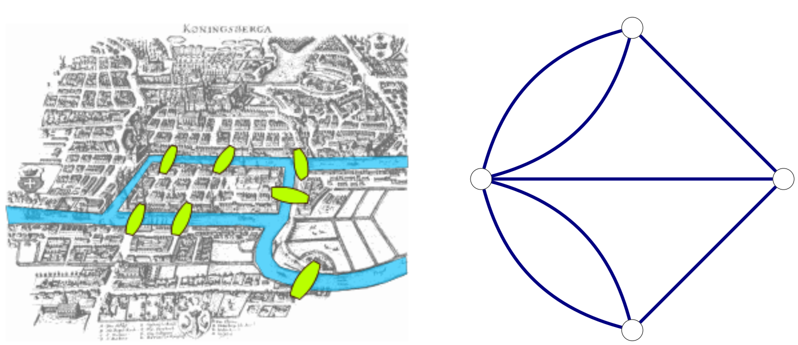

Figure 1: Representing the bridges of Königsberg as a multigraph.

To model the problem using binary variables, we represent a eulerian path an ordered list of arcs (directed edges). First, turn $G = (V, E)$ into a digraph $D = (V, A)$: replace every edge $e = \{u, v\}$ by a pair of arcs $a = (u, v)$ and its reverse $a' = (v, u)$. Then, for every arc $a \in A$ and index $i \in \{1, \ldots, |E|\}$, define a variable $x_{a,i}$ such that $x_{a,i} = 1$ if arc a is the i-th arc in the ordering, and 0 otherwise. To represent a eulerian path, the variables must satisfy the following constraints:

1. $\sum_{a \in A} x_{a,i} = 1$, for all i. (Exactly one arc is selected to be the i-th one in the ordering.)
2. $\sum_{i=1}^{|E|} x_{a,i} + x_{a',i} = 1$, for all a. (The arc a or its reverse a' occurs exactly once in the ordering.)
3. if arcs $a = (u, v)$ and $b = (y, z)$ do not satisfy $v = y$, then either $x_{a,i} = 0$ or $x_{b,i+1} = 0$, for all a, b and $i < |E|$. (If b does not leave the same vertex that a enters, then b cannot follow a in the ordering.)

For each constraint, we will write down a quadratic expression that is minimized when the constraint is satisfied. For constraint 1, the QUBO

$$\left(\sum_a x_{a,i} - 1 \right)^2$$

is 0 if the constraint is satisfied and greater than 0 otherwise. The second constraint is similar. Constraint 3 is represented by the QUBO

$$x_{a,i} x_{b,i+1}$$

for a and b with b not leaving the vertex that a enters. Combining all constraints, we have

$$\begin{aligned} Q(\mathbf{x}) = & \sum_{i=1}^{|E|} \left(\sum_{a \in A} x_{a,i} - 1 \right)^2 \\ & + \sum_{a \in A} \left(\sum_{i=1}^{|E|} x_{a,i} + \sum_{i=1}^{|E|} x_{a',i} - 1 \right)^2 \\ & + \sum_{\substack{a=(u,v), \\ b=(y,z): \\ v \neq y}} \left(\sum_{i<|E|} x_{a,i} x_{b,i+1} \right). \end{aligned}$$

For the bridges of Königsberg, there are 7 edges, so this QUBO has 98 variables, and $Q(\mathbf{x})$ will be 0 if $\mathbf{x}$ represents a eulerian path, and greater than 0 otherwise.[3]

[3] We can simplify the problem slightly by noticing that any eulerian path can be reversed to give another eulerian path. Therefore we remove one of the arcs, say a, and assume without loss of generality that its reversal a' is used. This gives us a QUBO with 91 variables instead of 98.

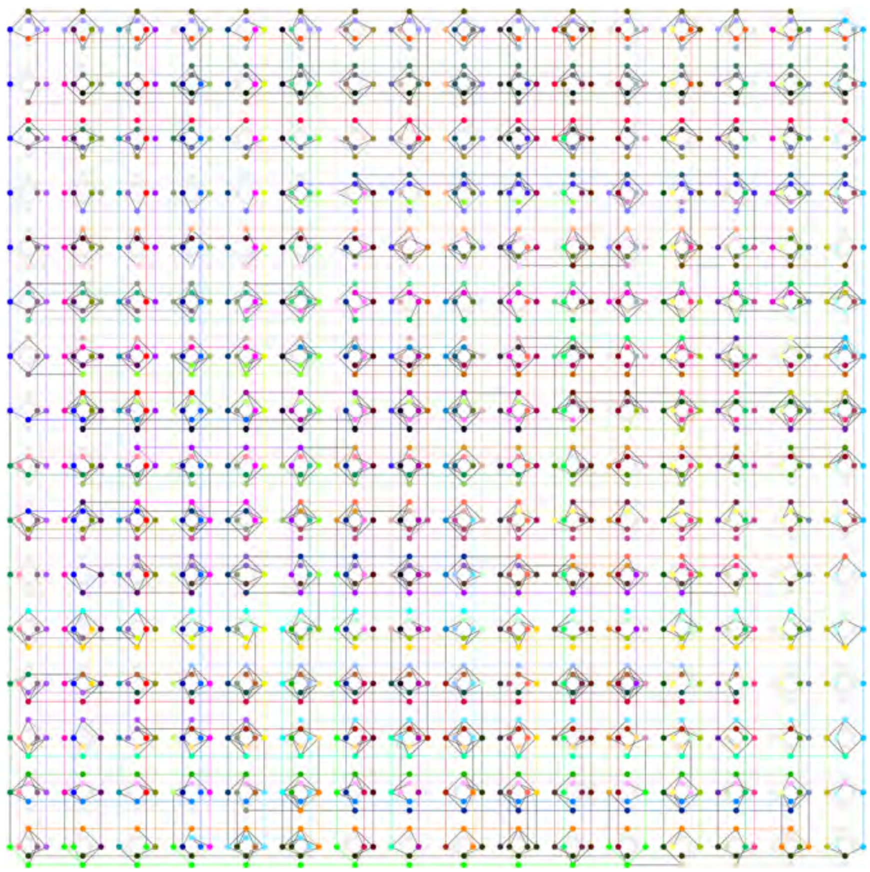

Figure 2: Representing the bridges of Königsberg problem on the D-Wave hardware. Each colour represents a different variable in the QUBO.

One final step is necessary before our problem can be solved by the D-Wave hardware. The D-Wave 2X allows for only a restricted set of quadratic terms, so the quadratic terms in our QUBO must match the quadratic terms in the hardware. To make this happen, we let several qubits in the hardware represent the same variable in our QUBO. We can force two qubits q_1 and q_2 to take the same value by including a quadratic term

$$(q_1 - q_2)^2$$

in our optimization.[4] Figure 2 shows one way to map all the variables from the bridges of Königsberg problem onto D-Wave's hardware .

[4]This process is called *minor-embedding*. Determining which qubits should represent which variables is itself a constraint satisfaction problem that can solved with heuristic algorithms.

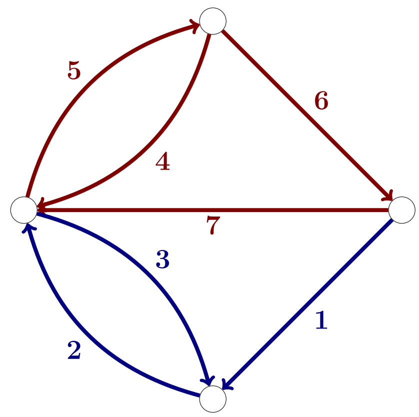

Figure 3: An answer to the bridges of Königsberg problem returned by the D-Wave hardware. After walking along arc 3 we cannot continue along arc 4, so this is not a eulerian path.

When we run the problem on the D-Wave hardware, we get an answer $\mathbf{x}^*$ such that $Q(\mathbf{x}^*) = 1$; these variables represent the arc ordering shown in Figure 3. Since the value of Q is greater than 0, this answer does not represent an eulerian path: after walking on the third arc we cannot continue along the fourth one.

Running the problem many many times, we find many answers with $Q(\mathbf{x}^*) = 1$, but none with $Q(\mathbf{x}^*) = 0$. This is evidence that no eulerian path exists. If we were 100% certain that the D-Wave hardware had found the optimal answer for Q, this would be a proof that no eulerian path exists. However because quantum annealing acts probabilistically, it is only evidence. On other hand, for graphs in which an eulerian path does exist, by running the problem many times the D-Wave hardware has the potential to find all optimal solutions and count the number of distinct eulerian paths.