Nikola Kapamadzin

NP Completeness of Hamiltonian Circuits and Paths

February 24, 2015

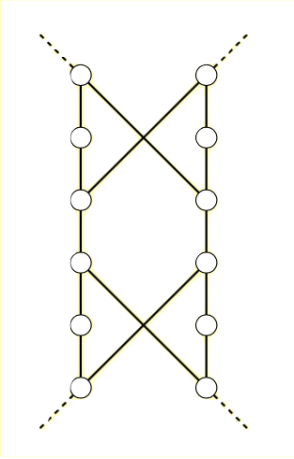
Here is a brief run-through of the NP Complete problems we have studied so far. We began by showing the circuit satisfiability problem (or SAT) is NP Complete. Then we reduced SAT to 3SAT, proving 3SAT is NP Complete. Next we reduced the vertex cover problem, graph coloring, and minesweeper to 3SAT, showing the all of these problems are NP Complete. We then took a jump and reduced The World's Hardest Game to Hamiltonian Paths, however we have not yet shown that Hamiltonian Paths are NP Complete. Now we will look at a proof that Hamiltonian circuits can be reduced to the vertex cover problem, and then that Hamiltonian Paths can be reduced to Hamiltonian Circuits. This will complete our logic bringing us to the conclusion that The World's Hardest Game is NP Complete.

Hamiltonian Circuits

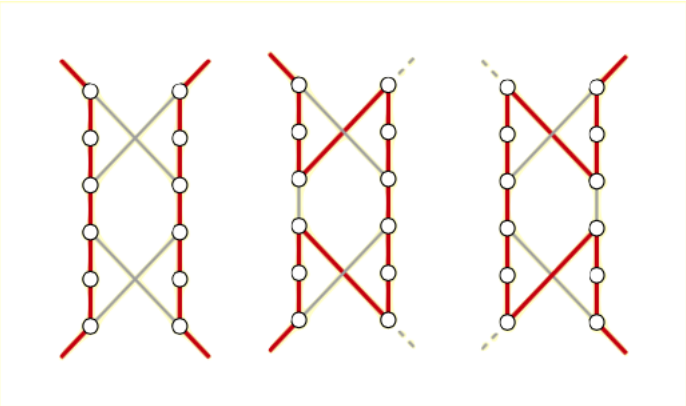
A graph G has a Hamiltonian Circuit if there exists a cycle that goes through every vertex in G. We want to show that there is a way to reduce the vertex cover a graph with a vertex cover, to a graph with a hamiltonian circuit. To do this we will construct a graph G', so G has a vertex cover of size k if and only if G' has a hamiltonian circuit. First to show hamiltonian circuits are in NP. This is obvious because we simply need to traverse the given edges and make sure every vertex is run through, and that we start and end at the same points, which can be done in polynomial time. Now to show hamiltonian circuits are NP Hard.

Hamiltonian Circuits are in NP Hard

The details of this proof may not seem intuitive as we begin, so we will run through an example while constructing a hamiltonian circuit from a vertex cover, and then discuss why this will always work. We begin by creating gadgets that look like the following

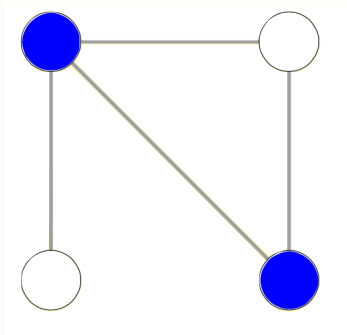


You can enter and exit each gadget through the dotted lines at the ends. We see that there are exactly three different ways to pass through every vertex in the gadget. The red lines represent the possible paths.

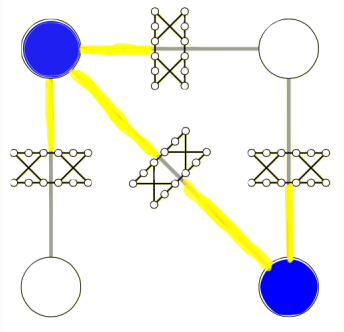


For this first case, we move through one side of the gadget, go somewhere else in the graph, and then come back through the other side. In the other two cases we run through all of the gadget at once. For every edge in our vertex cover graph, where the vertices in the vertex cover are blue, we place a gadget.

For Graph G



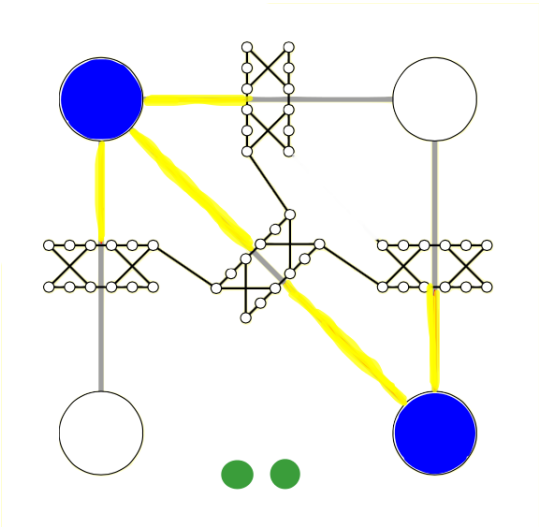
We place the gadgets as such, and since every edge contains at least one vertex in the vertex cover, we will color the edge yellow on the side of each gadget where there is a vertex in the vertex cover.



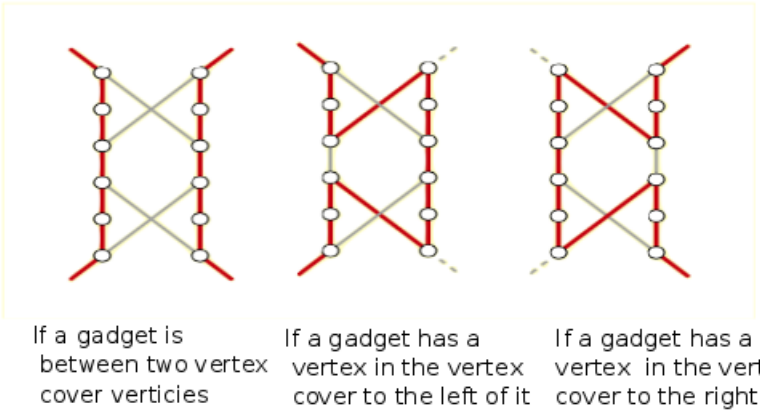
Now create new (green) *copy vertices* for every vertex in the vertex cover of G. Label these *copy vertices* $\{v_1, v_2 ... v_k\}$. These *copy vertices* are part of G' as are the gadgets. We make connections between gadgets and other gadgets, and between gadgets and the *copy vertices*. We make a connection between two gadgets if all of the following conditions are met

- Both gadgets share a vertex, v_1, in the vertex cover relative to the edge they correspond to.
- There exist two gadgets that contain the same vertex, v_1, in the vertex cover that only have one connection to other gadgets.
- A maximum of 2 gadgets can be connected to a different gadget, where the connections happen at opposite ends of the gadget.

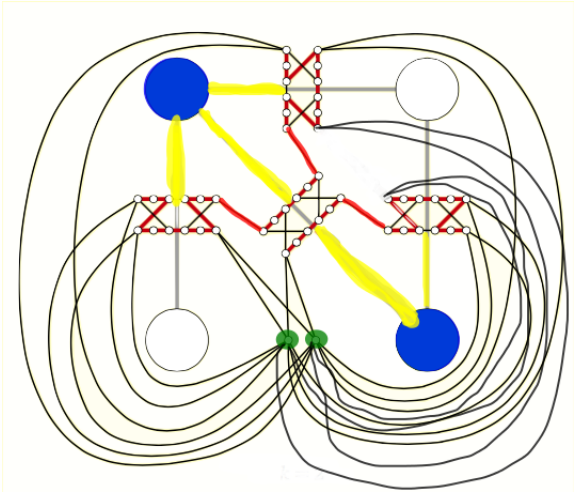
We generally leave the two gadgets that are furtherest away, yet connected to the same v_1 to not be connected, however it does not matter. Here is a way we can connect the gadgets



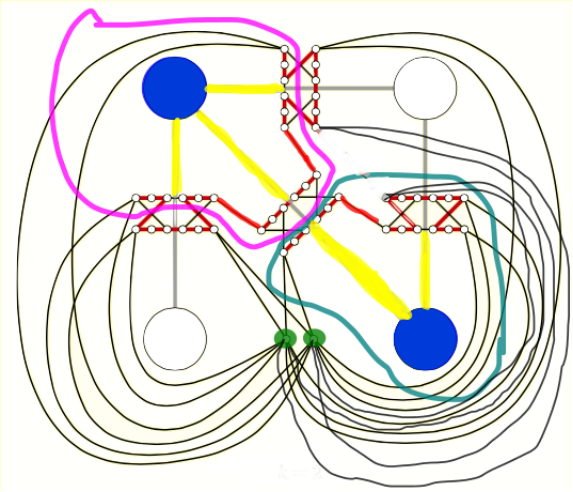
Within the gadgets, we make the following connections that correspond to which side a vertex in a vertex cover is in relation to the gadget. The side of the gadget facing a vertex cover vertex must be entered and exited in (side where we have 6 verticies in a line).



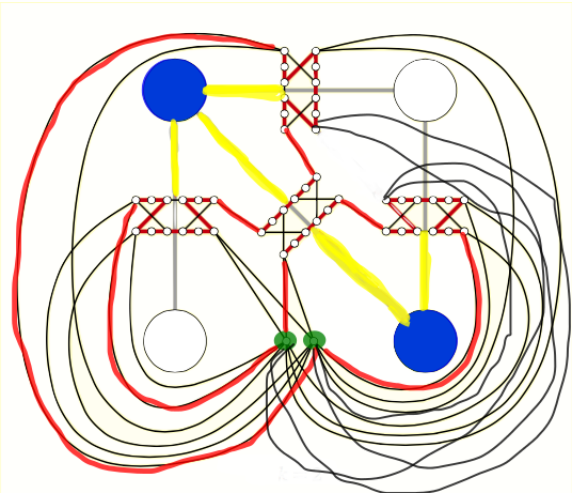
We make a connection for both sides of the gadget (here sides are where the original edge breaks up the gadget, so the two connections represent one for in, and another for out), that either goes to other gadgets or the *copy vertices* that were the copys of the vertex cover. Every gadget must be entered and exited on the side neighboring the red edge, which exists for every edge. If an endpoint of a gadget does not make a connection to another gadget, then it connects to two of the *copy vertices*. This gives us the following



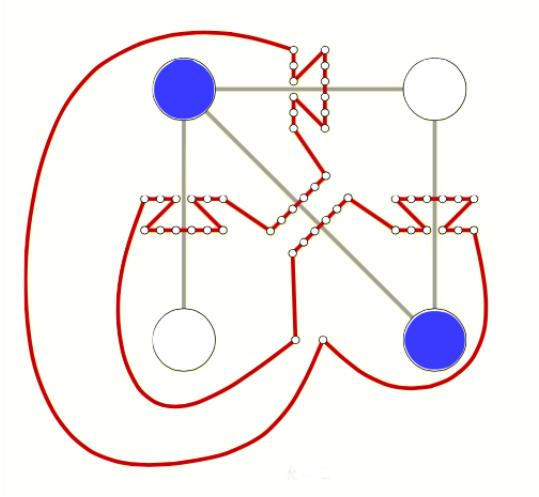
As you can see, a vertex in the original vertex cover always has a certain amount of sides of gadgets that belong directly to it. You can think of each of the vertices in the vertex cover as breaking up the graph into regions, because we surround each vertex in the vertex cover with a red path that comes from all the edges the vertex is contained in. Since each gadget can contain a maximum of 2 vertices in it's corresponding edge and a minimum of 1, and there exists paths in the gadget that correspond to both of those conditions, we will always make this kind of surrounding of the vertices in the vertex cover.



Now let's connect gadgets to the *copy vertices* in red for our actual path. All of the enterances and exits for our gadgets are inside regions corresponding to a vertex in the vertex cover, and there are exactly two endpoints of gadgets that are not connected, in red, to anything. We connect both of these endpoint vertices that aren't connected in red, to distinct *copy vertices* such that our first connection is to v_1, and our second connection is to v_2. Since we know there is another connection to be made from each vertex, now connect v_2 to an endpoint of the next region. If we had more vertices we would connect the second endpoint of this new region to v_3, but since we have only two *copy vertices*, we connect back to v_1. This ensures every *copy vertex* is connected to exactly two gadgets (that come from different regions), and don't make smaller cycles within the graph. This gives us the following construction



Now drawn without the rest of the graph



As you can see, this makes a hamiltonian circuit! Now why does this construction always create a hamiltonian circuit? As we saw earlier, we enclose each vertex in the vertex cover with our paths from the gadgets with respect to the edges they correspond to. Now we connect each of these unconnected endpoints in the gadgets to distinct *copy verticies*, which further enclosed the vertices in the vertex cover. Each of these *copy vertices* connects to exactly one other unconnected endpoint of a gadget from a different region (as these are the only connections that can be made from the *copy vertices*). However there is one more unconnected endpoint of a gadget in this new region that now connects to the next *copy vertex*. This continues until we reach the point where the only *copy vertex* that a regions endpoint can connect to is the very first *copy vertex* we connected to. We always have this situation because there is one region for each vertex in the vertex cover, and one *copy vertex* for each vertex in the vertex cover. Once we make this last connection, we finally completely enclose the cycle, which by construction goes through all of our points.

NP Complete

We have shown that with a clever construction, any graph with a vertex cover, can be used to make a graph with a Hamiltonian Circuit. Since creating such a graph can be done under polynomial time, simply replace edges with gadgets and make proper connections, we have a reduction from Vertex Coverings to Hamiltonian Circuits. This means that finding whether a graph has a Hamiltonian Circuit or not is NP Hard. Since this is also in NP, we have an NP Complete Problem

Hamiltonian Paths

We have shown that finding Hamiltonian Circuits in a graph is NP Complete, however in the proof of The Worlds Hardest Game, we used Hamiltonian Paths. The reduction here is quite simple to show Hamiltonian Paths are also NP Complete. We know that the decision problem of whether a Hamiltonian Path exists in a graph is

in NP, because given a set of edges, we can check in polynomial time whether we've gone through all the vertices. Given any Hamiltonian circuit it is already a path as it goes through each vertex. Any Hamiltonian Path can be made into a Hamiltonian Circuit through a polynomial time reduction by simply adding one edge between the first and last point in the path. Therefore we have a reduction, which means that Hamiltonian Paths are in NP Hard, and therefore in NP Complete.

Sources: Norbert Zeh, Computer Science Professor at Dalhousie University.
NP Completeness, https://web.cs.dal.ca/ nzeh/Teaching/3110/Notes/np-new.pdf