Homework 18

Robbie McKinstry, Jack McQuown, Cyrus Ramavarapu

12 October 2016

Problem 15:

Problem 16:

The dominating set problem can be shown to be **NP-Hard** via a reduction to the vertex cover problem. In the vertex cover problem, the goal is to find a set of vertices of given cardinality k that touches every edge.
The dominating set problem, on the otherhand, produces the minimum set of vertices, $\mathcal{S}$, from the input graph $\mathcal{H}$ such that every element not in the dominating set $\mathcal{S}$ is adjacent to an element in the dominating set $\mathcal{S}$.

The strategy to show vertex cover can be solved if there is a polynomial time algorithm for the dominating set problem will be local reduction. Since in the vertex cover problem there exists a minimum number of vertices that must be included so that every edge is touched, this number of vertices can be related to minimum number of vertices in the dominating set. This can be accomplished by forcing the dominating set problem to select one vertex for every edge. This can be done by looking at every edge and creating a triangle by adding a new vertex and then adding an edge between the new vertex and the two original vertices.

Although this returns the minimum dominating set, this value will also represent the minimum vertex cover. Although the input to the vertex cover problem can be different than this minimum value, the decision for this problem can be solved by a comparison to check if the input value is greater than or less than the result from the dominating set problem. If it is greater than, the vertex cover problem will return true; otherwise, it will return false.

Algorithmically, this process is represented by the following code.

Function: *Vertex Cover*
Input: *Graph G, int k*
`/*` V_G `is the number of vertices in graph G */`
for *edge e in* $\mathcal{G}$ **do**
  $\mathcal{G}' = add\_triangle(\mathcal{G},\ e)$
$\mathcal{S} = dominating\_set(\mathcal{G}')$
if $|\mathcal{S}| \leq k$ **then**
  Return: 1
else
  Return: 0

A runtime analysis of this code shows that the triangles can be made in polynomial time since the process only iterates over all the vertices. Furthermore, the necessary and sufficient condition of the transformation is satisfied by the existence of a triangle for every edge which forces the dominating set problem to pick at least one vertex in each triangle.