**CS5820 HW-8**

**Renhao Lu, NetID: rl839**

**2. Monotone SAT is a NP-complete**

1. Monotone SAT is in NP

Certificate: the values of

Check: if and the false number in is no less than , the answer is yes. Otherwise, answer is no. Assume has totally literals, checking whether is true takes and count whether false variable is less than takes . Because , the total checking time is is polynomial time.

2. Reduction of independent set problem to Monotone SAT problem:

(1) Convert input

Assume the input of independent set problem is integer and a graph , where and , . The independent set problem would be whether we can find an independent set and .

Then we create variables , each is corresponding to node . Then , where each clause is corresponding to an edge . Then , , and integer is the input of Monotone SAT problem. The monotone SAT problem would be whether we can find a solution of , so that is true and the number of false in is not less than .

(2) Convert output

The yes/no answer of monotone SAT would be the yes/no answer of independent set problem

(3) Running time

Running time for converting: Creating variables takes , and creating takes . So total converting time would be . Because , this is in polynomial time.

So if monotone SAT can be solved in polynomial time, then independent set can also be solved in polynomial time.

(4) Proof of correctness

We claim that the result of monotone SAT problem is equivalent to equivalent to independent set problem.

If monotone SAT has a solution, then we can create the independent set , and . We can claim that any two node in are not connected in . If and are connected by edge , then the clause would be false. Hence the is a solution of independent set problem.

If independent set problem has a solution, we assume the solution is , . Then for any that , we set is false; for other , we set them as true. Then the false number in is , and must be true, because no two false should be in the same clause. If two false are in one clause, it means two nodes in are connected, which is impossible because is an independent set. Hence is a solution of monotone SAT.

**3. Sparse Maximum Flow**

1. Sparse maximum flow is in NP

Certificate: The flow value of each edge in

Check: if there are at most non-zero edge in , then the answer is yes, otherwise is no. Running time: assume . In order to check there the flow is maximum flow, we need to first run a fast variant version of Ford-Fulkerson algorithm, which takes . Then we need to check whether the flow is a valid flow, takes to check flow conservation constrain and flow capacity constrain. Then we need to calculate the flow value, which is at most . Then we need to check whether non-zero flow edge is more than , which takes . So totally checking time is , and this is a polynomial time.

2. Reduction of vertex cover problem to sparse maximum flow problem

(1) Convert input

Given the input of vertex cover problem: undirected graph and integer (assume ). Create a directed graph and integer for input of sparse maximum flow problem. Here in , we split every node in input graph into two nodes and in , where is connect to the source with capacity of and is connect to the sink with capacity 1. For each edge from the vertex cover input , we create two edges and in with capacity 1. We also need to add edges for each with capacity 1.

In other words,

(2) Convert output

If each edge that has a non-zero flow, we pick the node in the vertex set for the solution of the vertex cover problem. If the sparse maximum flow problem has a solution, the vertex cover problem has a solution. Otherwise, the vertex cover problem doesn’t have a solution.

(3) Running time

Converting input: in new graph , creating takes , creating takes , creating takes . So totally . Because , is polynomial.

(4) Proof of correctness

We claim that the result of sparse maximum flow problem is equivalent to equivalent to vertex cover problem.

Claim 1: the maximum flow in is

Proof: Consider one cut , where , , then because there totally () edges with capacity 1. Then we know that the max flow .

Next, we consider a flow:

This is a valid flow and obviously the total flow value is . Hence the . As a summary, .

Claim 2: The number of edge with non-zero flow value is at most

Proof: In the solution of sparse maximum flow problem, the total flow value should be the maximum flow value, which is according to claim 1. Hence, each has a flow of 1. Meanwhile, for each at least one edge in should have a nonzero flow according to the flow conservation constraint. So far, the total nonzero edge . Hence the number of non-zero is at most .

If sparse maximum flow problem has a solution, we pick each node into the vertex set if edge has a non-zero flow value. According to claim 2, the size of . Because the maximum flow is , for each , there must be at least one non-zero flow from to ,. Hence the nodes in can cover all vertexes. Hence is a solution for vertex cover problem.

If vertex cover problem has a solution, we note this solution is a node set . Then we create a flow in according to . For all the node , we set the flow of as 0. Then we set the flow of all as 1. Because is a solution of vertex cover problem, for each edge , at least one node or should be in . If , we set and flow as 1. If , we set and flow as 1. If , we set and flow as 1. Then we set all according to flow conservation constrain and all other edges in with zero flow. The total flow is equal to the sum of all , which is , hence it is a maximum flow. All has exactly 1 inflow and 1 outflow, and we set flow of according to flow conservation constrain, so the flow is also a valid flow. Meanwhile, the nonzero flow edges including all , , that are described above. , and . So the total nonzero flow edge is . Hence the flow is a solution of sparse maximum flow problem.