

## Second Level Algorithms

Prof. Palash Dey

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Week – 06

Lecture 30

Welcome to the 30-th lecture of the second-level algorithm course. In the last lecture, we have seen some generalizations of the classical maximum S-T flow problem, which can be reduced to the classical maximum S-T flow problem and thus can be computed in polynomial time. In this lecture, we will see some more generalizations of this S-T flow problem. Okay, so let us begin. In the last class, we have seen three generalizations.

The fourth generalization is demand on edges. Here, every edge  $e$  has a demand  $d_e$  along with capacity  $c_e$ . The requirement is that the flow through the edge  $e$ , which is  $f_e$ , should be at least the demand and, of course, at most the capacity.

Again, this version can also be reduced in polynomial time to the classical maximum S-T flow problem and thus can be computed in polynomial time. The fifth one, very important, is defining cost on the edges. Mean cost, max flow. So here, every edge  $E$  has a cost  $d_e$  per unit of flow. So, if you pass 10 units of flow along an edge  $e$ , the total cost you incur is  $10 d_e$ .

The total cost of a flow is the sum of the costs incurred in all the edges. The goal is to compute a minimum-cost maximum S-T flow. We observe that the maximum S-T flow may not be unique.

There could be multiple ways to send the maximum amount of flow from S to T, but the cost of all those maximum S-T flows may not be the same. So, the goal here is to find a minimum-cost flow among all maximum S-T flows. This also can be computed in polynomial time, but the algorithm is substantially more involved, although it uses the ideas from basic maximum S-T flow. This problem also can be solved. in polynomial time using some ideas of maximum S-T flow and various other things; various new ideas are also needed. So, the algorithm is substantially more non-trivial than the max-flow problem, but the important point is this problem also can be solved in polynomial time.

There is another version of minimum-cost max flow for which it is unlikely to have a polynomial-time algorithm; that is called 0-1 minimum-cost maximum flow. So, what is 0-1 minimum-cost maximum flow?

Here, also, every edge  $e$  has a cost  $d_e$ . We incur it irrespective of the amount of flow. Passing through the edge  $e$  given, so let us call the amount of flow  $f_e$ , given  $f_e$  greater than 0. So, if you do not pass any flow along that edge, that means  $f_e$  is 0, then you do not incur any cost. But if you pass any non-zero amount of flow along the edge, then the cost is  $d_e$ . It is not proportional to the amount of flow passing through the edge.

In particular, The cost is not proportional to the amount of Passing through it. This problem is NP-complete. We will see what NP-complete means in the later part of the course, but for now, it means that it is unlikely to have

A polynomial-time algorithm for it. So, these are the various generalizations of the classical maximum flow problem. Many of them admit polynomial-time algorithms, some do not, and it is unlikely that they admit a polynomial-time algorithm. We will formalize this statement when we discuss NP-completeness in the later part of the course. Next, we will see some interesting structural results that can be proved using the maximum s-t flow problem and the algorithm we have seen. These are some classical results in graph theory. The first one is the max-flow min-cut theorem.

Says that the value of a maximum s-t flow is the same as the capacity of a minimum s-t cut. An s-t cut is a vertex partition, say  $A$  and  $V \setminus A$ , such that the vertex  $s$  belongs to  $A$  and the vertex  $t$  belongs to  $V \setminus A$ . The capacity of this

is the sum of the capacities of all edges going from  $A$  to  $V \setminus A$ . Here is the set of vertices, here is the cut,  $A$ ,  $V \setminus A$ , here is the source, Here is the sink. Now consider the edges going from the  $s$  side to the  $t$  side and add the capacities of all the edges. That is the capacity of this s-t cut.

The max-flow min-cut theorem says that the value of the maximum s-t flow is the capacity of the minimum s-t cut. So, minimum over all possible s-t cuts. There is, what is an s-t cut? It is just a bipartition of the vertices so that one part has  $s$ , the other part has  $t$ . So, if there are  $n$  vertices, then there are  $2^n - 2$  s-t cuts, and all of those cuts may have different capacity values.

So, look at the minimum capacity S-T cut. That capacity value is the same as the maximum S-T flow. Proof. Let  $f$  be a maximum S-T flow, and  $A, V \setminus A$  be an S-T cut, be any S-T cut. So, we will show that the value of  $f$  is at most the capacity of this cut.

So, what is the value of  $f$ ? So, for that, here is this cut  $A, V \setminus A$ .  $A$  has  $S$  and does not have  $T$ . So again, the trick is to consider the sum of the flow values—the net flow values—of all the vertices in  $A$ . So, consider the following: What is  $S$ ?  $S$  is, for all the vertices  $v \in A$ , flow out of  $V$  minus flow into  $V$ .

Now, for all vertices in  $A$ , this quantity is 0 except  $S$ . So,  $S$  is the value of  $f$ . Okay, now again, we will consider—we will compute this sum  $S$  from the edges' point of view. There are four kinds of edges. The edges whose both endpoints are in  $A$ , edges whose both endpoints are in  $V \setminus A$ , edges which start from  $A$  and go to  $V \setminus A$ , and edges that go from  $V \setminus A$  to  $A$ . Let us call them type 1, type 2, type 3, and type 4.

We will see what are the contribution of each type of edges in this sum  $S$ . The net contribution of the edges of type 1 to  $s$  is 0 because every such edge the flow value appears once positively and once negatively in the sum  $s$ . The edges of type 2 do not appear in the sum  $s$  so their contribution to  $s$  is 0. edges of type 3 they appear only positively so total contribution of the edges of type 3 is the capacity of the cut it is the it is at most the capacity of the cut which is the sum of the flow values so let me write the contribution of the edges of type 3 to  $S$  is  $\sum_{uv \in E, u \in A, v \in V \setminus A} f_{uv}$ . On the other hand, edges of type 4 contributes negatively to  $S$ . the contribution of the edges of type 4 to  $S$  is minus sum over all edges So,  $S$  is sum of flow values of all edges leaving  $A$  minus sum of all flow values entering  $A$ . So, this is greater than equal to if I ignore the second term because each flow value is greater than equal to 0 sorry not greater than equal to less than equal to because I am removing the ignoring some negative term so this is less than equal to. the total flow going out of  $A$ . But the flow value is at most the capacity values so this is less than equal to and this is the capacity of the cut. So,  $S$  is at most the capacity of the cut .

On the other hand, we have seen  $S$  is the value of the flow  $f$ . The value of  $f$  is less than or equal to the capacity of the S-T cut  $A, V \setminus A$ , because this is an arbitrary cut—an arbitrary S-T cut that we have picked for every arbitrary S-T cut. This inequality holds: the value of  $f$  is less than or equal to the capacity of the S-T cut. So, we conclude that since it was any S-T cut, The value of  $f$  is less than or equal to the minimum over all S-T cuts,

The capacity of the S-T cut. So, we have shown that max flow is less than or equal to min cut—min ST cut. Max S-T flow is less than or equal to min S-T cut. In the next lecture, we will show that max S-T flow is greater than or equal to minimum S-T cut, thereby proving equality. So, let us stop here. Thank you.