Approximation Algorithm

Prof. Palash Dey

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Week – 09

Lecture 44

Lecture 44 : Primal-dual Algorithm for Minimum Weighted Feedback Vertex Set

Welcome. So, in this class we will start seeing the Primal Dual Method for Designing Approximation Algorithm. In the beginning of this course we have seen how using Primal Dual Method we can get an f factor approximation algorithm for the set cover problem. So, in this class we will see the another problem which is called feedback vertex set and we see how using primal dual method we can get an approximation algorithm. So, the problem is feedback vertex set. So, what is the problem? Input is undirected and undirected age weighted graph $G=(V,E)$ and the weight function from $E \rightarrow R$ positive. So, this weights are for vertices not edges vertex weighted. Goal is to compute a subset S of V of minimum sum of weights. So, that the graph induced on $V \setminus S$ this is the graph where the vertex set is $V \setminus S$ and the edge set are the set of edges whose both end points are in $V \setminus S$. So, that the induced graph on $V \setminus S$ is acyclic. Equivalently the induced graph on $V \setminus S$ is a forest because it is a undirected graph equivalently for every cycle C of the graph G that cycle contains that cycle intersects with S in particular S hits every cycle C of the graph. These equivalences are immediate, but if it is not clear you must try to prove it. So, what is the first step of applying primal dual method? The first step is to write an integer linear programming formulation of the problem. and then we relax it to get a linear programming relaxation. So, let us write the ILP formulation.

We have a variable x_i for every vertex $i \in V$ which takes value 1 if we pick vertex i in our solution which is a feedback vertex set and 0 otherwise. So, the goal is to minimize the total weight of feedback vertex set $\sum_{i \in V} w_i x_i$ subject to here the equivalence condition is useful. writing as inequalities or constraints it turns out that the third one is useful it says that for every cycle in the graph G, I must pick at least one vertex. So, let us write down that constraint such that for all cycle C of G, think of cycle C as the set of vertices in the cycle then $i \in C$ x_i this should be greater than equal to 1 and x_i should take value either 0 or 1 for all $i \in V$. So, this is the ILP formulation to get the LP formulation we relax this integrality constraint with x_i.

to lie in between 0 and 1. And because we are minimizing $\sum_{i \in V} w_i x_i$ and because each weight w_i is positive, then we can remove this constraint that $x_i \leq 1$ because any optimal solution will always have $x_i \leq 1$. So, this is the LP formulation. Again here we see that we have exponentially many constraints because a graph can have exponentially many cycles. Since a graph can have an exponential number of cycles.

Give an example of such a graph take it as a homework. Give an example of a graph with exponentially many So, since a graph can have an exponential number of cycles, our LP relaxation may have depending on the input graph and exponential number of constraints. but this is perfectly fine because we are not going to solve the LP because in primal dual method we only use LPs to guide or to design a combinatorial algorithm. However, since we do not need to solve the LP in primal dual method this is fine. So, what we do we write the dual of the primal LP.

So, for each constraint we have a variable. So, for this constraint corresponding to C let us have a variable y_C and so, our dual LP maximize sum of dual variables $C \in \mathscr{C}$ is the set of cycles of G subject to for every vertex $i \in V$ we have sum over cycles which contain i as a vertex y_C this should be less than equal to w_i and for all cycle $C \in \mathscr{C}$ y_C should be greater than equal to 0. So, this is the dual LP. Now, let us briefly recall the primer dual method which we have overview for designing an f factor approximation algorithm for set cover. So, Overview of Primal Dual Method.

maintain ah dual feasible solution and ah partial primal because it is a partial solution it is infeasible. solution. So, let us not define the partial solution formally intuitively it is a partial solution even if you do not get what is partial solution for a problem it is fine it is not formally defined it is just the high level idea. Then iteratively increase one or more dual variables maintaining dual feasibility. and increasing the number of tight dual constraints.
Use tight dual constraints to make the partial primal infeasible solution more complete. continue till the primal solution the partial primal solution. becomes feasible feasible and thus complete. So, now, let us see this framework at work in the minimum feedback vertex set minimum weighted feedback vertex set problem. So, if I just write down this if I just apply this framework to this problem what we get is something like.

I need to start with a dual feasible solution. So, let us open the dual LP, here is the dual LP and you see setting y_C's to 0 for every C is a dual feasible solution. Here is another problem it seems we have is that there are exponentially many dual variables and because we are maintaining a dual feasible solution we need to maintain exponentially many variables it seems. but it is not problematic for this case because we will see that all the dual variables except polynomially many will be nonzero. So, we will be storing only nonzero dual variables which will always be polynomially many and hence although the number of dual variables is exponentially many.

we can store the a dual feasible solution in our algorithm succinctly with polynomially many numbers. So, start with dual feasible solution y_C equal to 0 for all cycle $C \in \notin$ ok. And what is the primal infeasible solution? The primal solution corresponds to vertices whether we are picking it or not. So, a natural partial primal solution is an empty state empty set of vertices. So, S is the set of vertices that we are that is the solution we are building iteratively.

So, S equal to empty set is the primal partial solution to begin with ok. Then because all w y's are positive you see all the dual constraints are not tight, none of the inequalities are equality all are strict inequalities. So, what a natural thing is what we did in set cover we take a dual variable and increase it till one of the dual constraint becomes tight. So, a natural idea is to find a cycle C in $G[V \setminus S]$ ok. And this cycle I need to kill I have to pick at least one vertex from that cycle.

So, I take a find a cycle and then increase the dual variable y_C till any dual constraint becomes tight again tight means the inequality becomes equality. So, in particular how by how much we can increase. each dual constraint which has y_C in the left hand side puts an upper bound on how much y_C can be increased. So, the and each constraint corresponds to a vertex.

So, for vertex in c. So, a vertex i allows y_C to increase by at most $w_i - \sum_{C' \in \notin, i \in C'} y_{C'}$. So, vertex i allows y_C to increase by this much amount and every vertex in C put some upper limit. So, the minimum maximum that we can increase y_C is the minimum of these quantities over all $i \in C$ let us call this ϵ. So, we increase y_C by ϵ and then at least one inequality or at least one inequality which involves y_C becomes tight, we pick at least one vertex corresponding to a tight inequality. We pick vertex i from the tight inequalities over $i \in C$ and put it in S ok.

And then we this way we make S more complete you know now in $G[V \setminus S]$ this particular cycle C is not there ok. And then we do a clean up step. So, we remove all degree 1 vertices because degree 1 vertices cannot participate in any cycle ok. So, in the next lecture we will see how this is a natural algorithm and how to analyze this algorithm and what approximation guarantee we can achieve from it ok. So, let us stop