Approximation Algorithm

Prof. Palash Dey

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Week – 03

Lecture 15

Lecture 15 : 1.5-Approximation Algorithm for Metric TSP

welcome. So, in the last lecture we have studied the metric TSP problem and we have seen an easy two factor approximation algorithm for that problem. So, in today's lecture we will improve the approximation guarantee to 1.5 factor approximation algorithm and that is the popular Christofides algorithm for metric TSP. So, today's topic is Christofides algorithm for metric TSP metric travelling salesman problem.

So, this we are calling algorithm 2. here also the high level idea remains the same that we want to compute a small in total weights an Eulerian subgraph of the given graph, but ah we do not want to really ah really double all the edges of minimum spanning tree. So, the idea remains the same that we want to compute a low cost Eulerian subgraph of G . And here also we begin with computing a minimum spanning tree .

We compute ah minimum spanning tree T of the input graph using any greedy algorithm or in particular any algorithm which runs in polynomial time for example, Primm's algorithm. or Kruskal's algorithm ok. Now, the problem why T is not Eulerian is that T may have odd degree vertices. Now, T the minimum spanning treaty is not Eulerian because it has vertices of odd degree. for example, T has at least 2 vertices of degree 1.

Now, let O be the set of vertices whose degree in T is and odd number. This is important the degree in t the degree of every vertex in the input graph is $n-1$ because the input graph is a complete graph. Now, here is an easy fact from graph theory is that the number of vertices of odd degree is always even is an even number . And very easy to prove also ah you can pause this video and try to prove it ah and let me now give you a proof sketch . It is simple counting argument So, using handshaking lemma you see that the sum of the degrees of the is twice the number of edges.

You see for every edge contributes between i and j contributes 1 to the degree of i and to the degree of j. So, that is why the sum of the degrees is twice the number of edges. Now, here you split this sum as summation i in n, i in O, O is the set of vertices having odd degree degree i plus summation i in n i not in O degree i is twice E twice cardinality E from where we get that i in O degree i is twice cardinality E minus sum i in n minus O. Now, you see this is an even number, this is also an even number because each of the degrees is even if you add any number of even numbers you get an even number.

And if you subtract one even number from another even number this is an even number. on the other hand each of this summand is an odd number. So, in this equation the right hand side is even. So, left hand side must also be even and hence that is only possible if the number of summands is an even number. So, from here we conclude that cardinality O is an even number.

So, I have even number of odd degree vertices. Now, somehow if I add few edges so that in the resulting graph this even the degree of this vertices in O becomes even and the degree of the vertices in $[n]\backslash O$ the other vertices remain even then I got an Eulerian graph because the degree will be even and because all the edges in the spanning tree are remaining here the graph remains connected. So, the idea is as follows. So, suppose here is the minimum spanning tree again let us this could be one example. Now, what are the odd degree vertices let us highlight.

So, this is an odd degree vertex, this is an odd degree vertex and this is an odd degree vertex 1, odd degree vertex. So, this green marked vertices are odd degree vertices the idea is add edges from G between odd degree vertices. So, that so, that the graph becomes Eulerian and this is possible because the number of vertices of odd degrees even. So, if I add edges between odd degree vertices then you see the resulting graph is a Eulerian graph. So, I got H how I take T and between odd degree vertices I add edges from G.

ok and we observe that T is not Eulerian, but H is Eulerian and what is the cost of H? This is cost of T. plus cost of newly added edges. Now, because I want to find an Eulerian tour and it is cost is c of h and the rest of the algorithm remains same. Once I have found an Eulerian subgraph, I compute the Eulerian tour and short circuit it and then I output the TSP and the same argument follows. the rest of the algorithm is.

So, ALG is like the two factor approximation algorithm will be less than equal to C of t plus this cost, cost of newly added edges because short circuiting can never increase the cost. Now, to minimize ALG I should minimize the cost of newly added edges. To minimize ALG, we should minimize the total cost of the newly added edges ok. And this these edges can see in graph theoretic terminology these are called matching. So, we compute a minimum cost matching in the induced graph on O.

It says that I have this graph G, the induced graph on a subset of vertices O is the subgraph of G where you delete all vertices outside O. So, in this graph you compute a minimum cost matching let us call it M. It turns out that computing a minimum cost matching can be done in polynomial time. We can compute ah minimum cost matching minimum cost perfect matching, because I want to match all the vertices in O a matching is called a perfect matching if all vertices are matched. So, this is important perfect matching.

in polynomial time . There this is a cornerstone result in combinatorial optimization there is there exist famous ah algorithms based on primal dual methods and other techniques using flows and so on ah polynomial time using primal dual method ok. So, now, H equal to T union M that means, this is that is vertex set remains same. and H set I add this edges ok. So, H is an Eulerian graph.

since it is a spanning connected subgraph of G and the degree of every vertex is an even number ok. So, let C be an Eulerian of H, we short circuit C to obtain a travelling salesman to C prime and we output it. So, what is ALG? ALG is nothing, but cost of C prime which is nothing, but costs cost is small c the which is less than equal to cost of the Eulerian tool which is capital C and this is nothing, but cost of T plus cost of matching M matching perfect matching M on O, O is the set of odd degree vertices in T. we have already argued in the last lecture that cost of T is less than equal to opt any travel any minimum spanning tree its weight is a is a lower bound on the minimum travelling salesman cost. So, this is opt plus C of m.

Now, how do we how do we connect C of M with opt and for that we prove this important claim cost of M is less than equal to $\frac{opt}{2}$. So, assuming this claim let us finish this analysis is less than equal to $opt+\frac{opt}{2}$ which is $\frac{3}{2}opt$ proof. for that let us consider any optimal travelling salesman tour. Let C_1 be any optimal travelling salesman tool of G. First what we do? We remove all the vertices which are outside O from C_1.

We remove all vertices in $V[G]\setminus O$ from C_1 and we short circuit. Let C_2 be the resulting travelling salesman tool . See observe that after short circuiting it is a travelling salesman tool of the induced graph on O with a resulting travelling salesman to of the induced graph G o ok. Now, we observed that ah this C_2 contains 2 perfect matchings on O. So, because O is a have even cardinality.

So, if you if you look at O this C_2 if you look at C_2 it is a travelling salesman tour on O.

So, it contains two perfect matchings if I take alternate this is one and another one is the other alternate edges . So, we observe that C_2 contains 2 perfect matchings on O, let us call them M_1 and M_2. Now, notice that we have added a minimum cost perfect matching on T. So, this M is a minimum cost perfect matching on O.

This is that means, cost of M is less than equal to an M_1 and M_2 are two perfect matchings on O. That means, $C(M) \leq C(M_1) + C(M_2)$. So, from here you can write C of M is less than equal to $\frac{C(M_1)+C(M_2)}{2}$, but $C(M_1)+C(M_2)$ is nothing, but the cost of the tour C_2. So, this is half cost of C_2. which is less than equal to cost of the optimal travelling salesman to C_1 because we have short circuited and here also we are using the metric property because of which short circuiting can never increase the total cost C of C 1 is opt.

So, this is half opt. which finishes the proof of the claim and hence it shows that this is our algorithm is a $\frac{3}{2}$ factor approximation algorithm. And this is the best known for symmetric travelling metric travelling salesman problem. On the lower bound side we know that we know this following theorem that let alpha be or there exist an alpha there exists real number alpha greater than 1 such if there is a polynomial time alpha factor approximation algorithm for metric TSP, then P equal to NP. In particular assuming P not equal to NP, there is no petas for metric TSP. However, it is possible to make further assumption for example, if we assume that the vertices are points in a Euclidean space and the distance is the Euclidean distance between two vertices then there exist a petas for the TSP problem ok. So, let us stop here. Thank you.