Approximation Algorithm

Prof. Palash Dey

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Week – 03

Lecture 13

Lecture 13 : Inapproximability of the Traveling Salesman problem

Welcome. So, in the last class we have seen scheduling jobs in multiple identical machines and we have seen that simple greedy algorithm gives us a $\frac{4}{3}$ approximation algorithm. So, today we will study a very popular algorithm very well studied problem which is called the travelling salesman problem. So, that is the topic of today's lecture. So, we have travelling salesman problem TSP in short. So, what is this problem? So, let us assume that there are n cities suppose there are in cities where a salesman wants to visit.

the distance between cities i and j is c_{ij}. So, for all pair of cities we have the distance given that is the input. The goal is to find shortest travelling salesman tour. A travelling salesman tour is a permutation of 1 to n say $\sigma(1),\ldots,\sigma(n)$ depicting the order or sequence of the cities that the salesman will visit.

that is the salesman starts at city $\sigma(1)$, then he goes to $\sigma(2)$, city $\sigma(2)$, then $\sigma(3)$ and so on and at the end goes to $\sigma(n)$ and then returns to $\sigma(1)$. So, it must be a cycle. The cost of the travelling salesman tour is. So, how does the tour look like? He starts with city sigma and then he goes to city $\sigma(2)$, then he goes to city $\sigma(3)$ and so on. And at the end he goes to city $\sigma(n)$ and then returns to city $\sigma(1)$.

So, the cost is $C_{\sigma(n)\sigma(1)} + \sum_{i=1}^{n-1} C_{\sigma(i)\sigma(i+1)}$. And the goal is to minimize this cost find a travelling salesman tour which minimizes this cost. So, there are two versions of travelling salesman problem. in the symmetric version between any two cities i and j the distance from i to j is same as distance from j to i. That means, the underlying graph is weighted, but undirected.

for all $i, j \in [n]\, i \neq j, C_{ij} = C_{ji}$. This is the symmetric version and we will focus on the symmetric version now. We will focus on symmetric version now ok. In the asymmetric version we may have or we are allowed to have cities i, j such that the distance from i to j is not equal to the distance from j to i. So, this travelling salesman problem can be shown to be NP complete again let me remind you we have defined the problem as an optimization problem.

So, that we can design approximation algorithm. So, when I say that this problem is NP complete it should be understood that the decision version of this problem is NP complete. So, take it as a homework to show very easy. The travelling salesman problem. is NP complete .

For example, we can reduce the Hamiltonian circuit problem to the travelling salesman problem . So, we can reduce the Hamiltonian circuit problem to TSP. So, what is the Hamiltonian circuit problem? It is the input is a graph unweighted graph and the question is does there exist a Hamiltonian circuit in this graph or not. A cycle is called a Hamiltonian circuit if it visits all edge all the vertices exactly in the Hamiltonian circuit problem, the input is an undirected graph and we need to compute if the graph has a Hamiltonian circuit. circuit of a graph is a cycle that visits every node of the graph.

exactly once ok. Now, you pause here and try to prove this theorem before proceeding further because I am now going to explain the reduction, but this really an easy reduction, but this has interesting corollaries. So, that is why I am going to explain, but for your benefit you please pause the video here and try to prove this theorem that the travelling salesman problem is NP complete by reducing from Hamiltonian circuit problem ok. Hamiltonian circuit many to one reduces to travelling salesman problem in polynomial So, let G be an arbitrary instance of Hamiltonian circuit ok. So, from this we construct an instance of travelling salesman, we construct $G^{'}$ as follows.

Their vertex set is same as G is the vertex set of $G^{'}$ this is same as V of G. $G^{'}$ is a complete graph that is what is needed what is a requirement for travelling salesman problem. $G^{'}$ is a complete graph and the cost between the vertices of u and v in $G^{'}$. is 1 if this edge u v is present in the graph G, otherwise it is 2. Now, it is easy to see that G has a Hamiltonian circuit.

if and only if $G^{'}$ has a travelling salesman tour. of cost at most n . why let us let us see the see the proof sketch. If G has a Hamiltonian circuit that Hamiltonian circuit serves as a travelling salesman tour of $G^{'}$ of total cost n. On the other hand if G does not have a Hamiltonian circuit then any travelling salesman tour of G prime must use an edge which is present only in G prime, but not in G whose weight greater than 1 and because the cost of every edge is at least 1 and any travelling salesman tour must use exactly n edges because if G does not have a Hamiltonian circuit then the minimum cost of any travelling salesman tour will be strictly greater than n at least $n+1$.

So, this proves the NP completeness. Now, we can slightly modify this proof to show inapproximability. For example, if I replace this 2 with n, then what we get is if the following result. If G has a Hamiltonian circuit then as usual, then $G^{'}$ has a travelling salesman tour of cost n. if that is one, the other is more important if G does not have any Hamiltonian circuit then every travelling salesman to of $G^{'}$ has cost at least when this C u v was 2 for non edges then it was at least $n+1$ because it must use at least an edge which is present only in $G^{'}$, but not in G at least n plus Now, when I make increase the cost of non edges to n, then and the algorithm must use any travelling salesman tour must use at least one edge of G prime which is not present in G, then the total cost of that tour will increases to or let us make it $n+1$, it increases to $2n$.

So, this shows that there is no better than two factor approximation algorithm. Not only that we can apply this boosting trick, we can increase this further to say any number say $10n+1$, then here we will get $10n$. this will show that there is no approximation algorithm for travelling salesman problem whose approximation guarantee is better than 10, because then we will use this algorithm along with this reduction to have a polynomial time algorithm for Hamiltonian circuit. circuit, that algorithm will success if the if G has a Hamiltonian circuit and the approximation ratio of that algorithm is less than 10, then that algorithm is forced to output the Hamiltonian circuit. And not only that how far we can stretch for example, if we can change this to say 2^n+1, then this also change to 2^n and so on.

So, what for so, what we essentially have essentially proved is this theorem. how large this function can grow up recall this is a polynomial time reduction. So, we need to write the cost of this number in polynomial time that means, this should have polynomially many bits. And if we can write that function then that shows that there is if I can if there is a computable function $\alpha(n)$ which we can write in polynomial in n time, then I can plug in that alpha n here may be $\alpha(n)+1$ and here I get $\alpha(n)$ that will show sorry alpha that will show that there is no better than $\alpha(n)$ factor approximation algorithm. So, what we have essentially proved is let $\alpha(n)$ be any number whose binary representation can be written in polynomial time.

So, let $\alpha(n)$ be any such number it could be $2^n, 10^n, 100^n$ or anything any c^n or any number c^{n^2} and so on. Then there is no $\alpha(n)$ approximation algorithm for the travelling salesman problem. unless P = NP. Why? Because we can use such an approximation algorithm to design a polynomial time algorithm for the Hamiltonian circuit problem which is an NP complete problem. So, in the next lecture we will see a very natural restriction on this travelling salesman problem to take care of to bypass this impossibility result and we will design a constant factor approximation algorithm for this travelling salesman problem under that very natural assumption ok.

Let us stop here. Thank you.