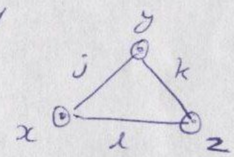
ASSIGNMENT: HW13

NAME: PRASANTH BHAGAVATULA

CWID: A20355611

① Consider the general travelling salesman problem. Let 'm' be the maximum cost between any two cities of the problem. Now, for each and every edge, we add ~~an~~ 'm'. This makes the problem to satisfy the triangle inequality. Below shows us ~~how~~ the proof for the property

Let 'j', 'k', 'l' be the cost between three cities x, y, z in the original problem. Now, ~~let~~ this may or may not satisfy the triangle inequality property. Now, we are adding 'm' to each and every edge.



$$\therefore \cancel{a}\ l + m \le l + m + j + k$$

$$\le m + m + j + k \quad (\because \text{'m' is the max value})$$

$$\le (j + m) + (k + m)$$

Hence the new problem satisfies the triangle inequality.

This transformation takes only polynomial time, because we just need to identify maximum value 'm' and increase every edge by that value.

Now we prove that, both the instances have the same set of optimal tours. Let 'O' be an optimal tour in original problem and 'C' the the cost of the tour. Now, in the ~~new~~ transformed instance, the ~~optimal~~ corresponding tour is O' and the cost is $C' = C + mn$, where 'n' is the number of ~~edges~~ vertices.

Now, if we assume that there is an optimal toure in the transformed instance O'', with cost $C'' < C' < C + mn$, ~~then~~ and so, there should be a corresponding tour in the original problem for O'' which will cost $C'' - mn$ $< (C + mn) - mn < C$. But, this contradicts that 'C' is the ~~cost~~ optimal cost. ~~Hen~~

Similarly if O is an optimal ~~in~~ tour in the transformed instance with cost C, then the corresponding tour in original instance will be O' and has cost $C' = C - mn$; If we assume that, there is an optimal tour in original instance O'' with cost $C'' < C' = C - mn$, then the corresponding tour of O'' in transformed instance will cost

$$C'' + mn < C - mn + mn < C$$

Hence this contradicts the statement that C is the cost of optimal tour in transformed instance.

Hence both the instances will have the same set of optimal tours.

Now, ~~these~~ this polynomial-time transformation does not contradict the theorem because, the transformation that is used in the proof of theorem does not obey the triangular inequality rule. Let, u, v, w be three vertices or cities in the graph. ~~Now~~ and let (u, v) and $(v, w) \in E$

$\therefore$

$$c(u,v) = c(v,w) = 1 \text{ and } c(u,w) = \rho|V|+1$$

By triangular inequality

$$c(u,w) \leq c(u,v) + c(v,w)$$

$$\Rightarrow \rho|V|+1 \leq 1+1 \Rightarrow \rho|V| \leq 1$$

which is not the case

Hence, the new transformed instance does not contradict the theorem.

ASSIGNMENT : HW13

NAME: PRASANTH BHAGAVATULA

CWID: A20355611

(2) Given inequality is

$$OPT \geq 2 \sum_{i=\lceil n/2 \rceil+1}^{n} l_i$$

For all even numbers $n = 2k$, if we substitute the same in the equation, we get

$$OPT \geq 2 \sum_{i=\lceil 2k/2 \rceil+1}^{2k} l_i$$

$$\Rightarrow OPT \geq 2 \sum_{i=k+1}^{2k} l_i$$

which is the inequality (2) and hence the given inequality is true for all even numbers of 'n'.

For odd numbers of 'n', we will consider the shortest edge connecting, let's say, c_i ~~and~~ and c_j and we now introduce a new city c_k such that $l_{ik} + l_{jk} = l_{ij} = l_n$. Hence, even with the introduction of new city, the triangular inequality still holds good and let $n+1 = 2m$

Now substituting in the equation, we get

$$OPT \geq 2 \sum_{i=\lceil \frac{n+1}{2} \rceil +1}^{n+1} l_i \qquad = 2 \sum_{i=m+1}^{2m} l_i$$

$$\Rightarrow OPT \geq 2 \sum_{i=m+1}^{n+1} l_i$$

$$\Rightarrow OPT \geq 2 \sum_{i=m+1}^{n-1} l_i + l_n + l_{n+1}$$

As per the new graph, l_n is nothing but l_{ik} and l_{n+1} is nothing but l_{kj} and we know that $l_{ij} + l_{kj} = l_{ij} = l_n$ in the original graph. Hence this inequality holds true even in case of 'n' is odd and so

$$OPT \geq 2 \sum_{i=\lceil n/2 \rceil +1}^{n} l_i$$

ASSIGNMENT: HW13
NAME: PRASANTH BHAGAVATULA
CWID: A20355611

(3) (a) If 'C' is a symmetric cost matrix that satisfies the triangular inequality, then it is same as the closest insertion problem with the only difference that we get the cost between two cities as input instead of the distance. Similar, to the distances following the triangular inequality and the costs also following the triangular inequality, instead of finding the minimum distance city which is not on tour, we find city 'k' such that

$$C_{ik} + C_{jk} - C_{ij} \text{ is minimum.}$$

where C_{ij} represents cost of travel between city 'i' and city 'j' and here i, j belong to tour and city k' does not belong to tour. Going by the proof of closest insertion algorithm

$$C_{ik} + C_{jk} - C_{ij} \leq 2c'$$

where c' is the cost of an edge only the optimal tour. Hence summing up all the inequalities for every iteration we get

$$|\text{cost of cheapest insertion tour}| \leq 2\,|\text{cost of optimal tour}|$$

$$\Rightarrow \frac{|\text{cost of cheapest insertion tour}|}{|\text{cost of optimal tour}|} \leq 2$$

(b) Initially, we need to ~~constr compute the minimum cost of it find~~ ~~ci~~ compute $C_{ik} + C_{kj} - C_{ij}$ for all cities k not on tour and (i, j) are only on the tour. Now, ~~we~~ for each and every iteration, we insert the city where $C_{ik} + C_{kj} - C_{ij}$ value is minimum. Now, since city k is included in the tour, we need to re-compute the ~~cost~~ value $C_{ik'} + C_{k'k} - C_{ik}$ and $C_{jk'} + C_{k'k} - C_{kj}$ for all k' ~~in~~ in "not on tour" and ~~compar~~ take the minimum value as that set of minimum values will be used to add the next ~~set~~ of city into tour.

~~The computat~~ The number of cities in the "not on tour" reduces for each and every iteration and the number of iterations is proportional to the number of cities.

Hence the number of operation are $O(n^2)$

(c) This is similar to the closest insertion ~~proble~~ algorithm producing a tour of length which is almost twice of the optimal tour. Similar to the closest insertion algorithm, the cheapest insertion algorithm also satisfies

$$\frac{|\text{length of cheapest insertion tour}|}{|\text{length of optimal tour}|} \leq = 2 \cdot \left(1 - \frac{1}{n}\right)$$