CS466: Design and Analysis of Algorithms
Assignment 1
Spring 2016
Instructor: Mark Petrick

MingHao Lu
20465562

May 11, 2016

Question 1

Given the k-polynomial algorithm and an instance of the Hamiltonian cycle problem(a graph), construct a second, weighted graph such that:

1. the vertices of this graph is the same as the original graph
2. the graph is complete
3. for every edge e in the new graph, let $w(e)$ be 1 if e exist in the original graph, let $w(e)$ be $k|V|+1$ otherwise

We can now demonstrate a TSP tour with length $|V|$ exists iff there is a Hamiltonian cycle in the original graph. We first observe, given the same set of vertices, a Hamiltonian cycle is a valid TSP tour and vice versa(provided the edges in the tour/cycle exists).

- $Hamiltonian \Rightarrow TSP$: Simply consider the tour in the new graph that correspond to the Hamiltonian cycle in the original graph. By definition of Hamiltonian cycle, this tour has $|V|$ edges. Since each edge of this tour exist in the Hamiltonain cycle, it has weight 1, therefore this entire TSP tour in the new graph has length $|V|$.
- $TSP \Rightarrow Hamiltonian$: Suppose there is a TSP tour of length $|V|$. By definition of a TSP tour it must have $|V|$ edges. Since every edge has weight 1 or $|V|+1$ and the length of the tour is $|V|$, every edge must have length of 1. Thus every edge in this tour exist in the original graph, and the correponding cycle in the old graph is a valid Hamiltonian cycle.

To find a Hamiltonian cycle, run the k-polynomial time TSP tour algorithm on the new graph. We can show a tour of length $\leq k|V|$ will be found iff there is an tour of length $|V|$.

- tour of length $|V| \Rightarrow$ find length $\leq k|V|$: The algorithm is guaranteed to find a tour of length less or equal to k times the optimal length. If a tour of length $|V|$ exists, then by definition of optmial tour, $length(optimaltour) \leq |V|$, which means the algorithm is guaranteed to output a tour of length $\leq k * length(optimaltour) \leq k * |V|$.
- find length $\leq k|V| \Rightarrow$ tour of length $|V|$: If the algorithm find a TSP tour of length $\leq k|V|$, since each edge has weight either 1 or $k|V|+1$, every edge in the tour must be of weight 1. Since by definition a TSP tour must have $|V|$ edges, this imples this tour must be of length $|V|$.

We know by the observations above, the algorithm output a tour of length $\leq k|V|$ if and only if there exist a TSP tour of length $|V|$, and also a TSP tour of length $|V|$ exists if and only if a Hamiltonian cycle exists in the original graph. Thus we conclude, given a Hamiltonian cycle problem instance, simply construct the corresponding weighted graph and run the k-polynomial algorithm on the constructed graph. If the algorithm output a TSP tour of length $\leq k|V|$, we output yes, else we output no. Since this is a polynomial time solution and the Hamiltonian cycle problem is NPC, we conclude P=NP.

Question 2

let $M = max\{w(e) : e \in E\}$. Given a TSP problem instance(graph), construct a new graph where the edges and vertices are the same, and the new weight function $w'(e) = w(e) + M$.
First we observe this graph is metric. Let a,b,c be three edges that form a triangle, WLOG

$$\begin{aligned}
w(c) &\leq M && \text{by definition of M}\\
\Rightarrow w(c) + M &\leq M + M\\
\Rightarrow w'(c) &\leq M + M\\
\Rightarrow w'(c) &\leq w(a) + M + w(b) + M && \text{since we only deal with non-negative weight functions in this course}\\
\Rightarrow w'(c) &\leq w'(a) + w'(b)
\end{aligned}$$

Then we observe the optimal tour is preserved. Given a cycle, it's trivial to observe the cycle's new length will be its old length plus the extra weight M for each edge(sum to $M|V|$). The optimal tour is preserved since the length of every tour under the new weight function is increased by the same amount.

Now we can demonstrate the transformation can't be used with the 2x metric TSP algorithm to prove P=NP. If we run the 2x algorithm on the transformed graph, we will receive a tour. This is the ONLY information we gain after running the algorithm. In the first question, we demonstrated if we can create a k-approx for the general TSP, we can prove P=NP. To utilize the first question, we need to somehow find a k-approx tour for the original graph. The only information we have is a tour that is less or equal in length to two times the optimal length of the transformed graph. Let L be the length of the optimal tour under the original weight function, F be the length of the found tour under the original weight function, L', F' be their counterparts under the new weight function.

$$\begin{aligned} F' &\leq 2 * L' && \text{since F is founded by running 2-approx} \\ \Leftrightarrow F + M|V| &\leq 2 * (L + M|V|) && \text{since new length is old length plus } M|V| \\ \Leftrightarrow F &\leq 2 * L + M|V| \end{aligned}$$

So ALL we know about the tour under the old weight function is the length is bounded by $2 * L + M|V|$. But M is a property of the input graph, and is not bounded by a constant factor of L(consider any graph where the optimal tour does not include the maximum edge, the weight of that edge can then be arbitrarily increased without affecting L). Thus the found tour under the old weights is not bounded by a constant factor of L. Since this tour is all the information we have to work with, we can't produce a k-approx tour in the original graph. This is why we can not utilize question 1 to prove P=NP.

Question 3

Suppose the optimal tour has a cross. Let (v_i, a_1) and (a_k, v_j) be two of the edges that cross over. WLOG, represent the optimal tour as $v_1, v_2, \ldots, v_i, a_1, a_2, \ldots, a_{k-1}, a_k, v_j, \ldots, v_n, v_1$. Consider the cycle $v_1, v_2, \ldots, v_i, a_k, a_{k-1}, \ldots a_2, a_1, v_j, \ldots, v_n, v_1$ (I switched $a_1, ..., a_k$ to $a_k, ..., a_1$). This is obviously still a Hamiltonian tour(it contains every vertice once, except the first and last vertices is the same). I claim the optimal tour is greater than this tour.

Let z be the intersection point between the edges (v_i, a_1) and (a_k, v_j). Observe that $(v_i, a_k), (v_i, z), (z, a_k)$ form a triangle. By triangle inequality, $|(v_i, a_k)| \leq |(v_i, z)| + |(z, a_k)|$. The equal case only happens if (v_i, z) joint with (z, a_k) is the line (v_i, a_k), which we dont consider. (on piazza we confirmed no three points are colinear, for (v_i, z) joint with (z, a_k) to be the same line as (v_i, a_k) requires z to be between (v_i, a_k), but since z is defined to be between (v_i, a_1), this imples v_i, a_k, a_1 to be colinear, contradiction). So $|(v_i, a_k)| < |(v_i, z)| + |(z, a_k)|$ By similar argument, $|(a_1, v_j)| < |(a_1, z)| + |(z, v_j)|$

One final observations, since v_i, z, a_1 are colinear, $|(v_i, z) + (z, a_1)| = |(v_i, z)| + |(z, a_1)|$. Similarly, $|(a_k, z) + (z, v_j)| = |(a_k, z)| + |(z, v_j)|$.

$$\begin{aligned} |(v_i, a_1)| + |(a_k, v_j)| &= |(v_i, z) + (z, a1)| + |(a_k, z) + (z, v_j)| \\ &= |(v_i, z)| + |(z, a1)| + |(a_k, z)| + |(z, v_j)| && \text{what we just proved} \\ &> |(v_i, a_k)| + |(a1, v_j)| && \text{what we proved earlier} \end{aligned}$$

Our new tour contains the exact same edges as the optimal tour except the two. We have shown for those two edges, the optimal edges combined is greater than our edges combined. Therefor the optimal tour has a greater length than our tour. Contradiction. Since the only thing we assumed about this optimal tour is that it has two edges crossing, an actual optimal tour must have no edges crossing.