Title

Computer Science 604

Advanced Algorithms

Lecture 8a: Dealing with NP-completeness/

Approximation

David Juedes

School of EECS

juedes@cs.ohiou.edu

TSP-OPT

Recall from last time, we defined two version of TSP.

Decision Version(TSP-D): Given a set of vertices V, a distance function $d : V \times V \longrightarrow \mathbb{Z}^+$, and a distance L, determine if there is a hamiltonian tour of V (a permutation π of V) such that the length of the tour ($\sum_{i=1}^{|V|-1} d(\pi_i, \pi_{i+1}) + d(\pi_{|V|}, \pi_1)$) is $\leq L$.

Optimization Version(TSP-OPT): Given a set of vertices V, a distance function $d : V \times V \longrightarrow \mathbb{Z}^+$, and a distance L, produce a hamiltonian tour of V (a permutation π of V) such that the length of the tour ($\sum_{i=1}^{|V|-1} d(\pi_i, \pi_{i+1}) + d(\pi_{|V|}, \pi_1)$) is minimized.

It is relatively easy to prove that TSP-D is NP-complete.

TSP-D

I claim (without proof) that

Hamiltonian Cycle (HC): Given a graph $G = (V, E)$, determine whether there exists a Hamiltonian cycle in G, i.e., whether there exists a permutation π of V such that for each i, $\{\pi_i, \pi_{i+1}\} \in E$, and $\{\pi_{|V|}, \pi_1\} \in E$.

is NP-complete.

We will next prove that TSP-D is NP-complete via a reduction from HC.

HC $\leq_m^p$ TSP-D

Theorem 6.1: TSP-D is NP-complete

Proof: Given a graph $G = (V, E)$, we build an instance $f(G) = \langle V', d, L \rangle$ of TSP-D, where $V' = V$,

$$d(u, v) = \begin{cases} 1 & if\ \{u, v\} \in E \\ 2 & \text{otherwise.} \end{cases}$$

and $L = |V|$.

It is relatively straightforward to see that G has a Hamiltonian cycle if and only if the instance $\langle V', d, L \rangle$ of TSP-D has a solution (permutation) whose total distance is $= |V|$. Notice that it is easy to see that a permutation whose total distance is $|V|$ corresponds to a Hamiltonian cycle in G. Likewise, a Hamiltonian cycle in G corresponds to a length $|V|$ solution in $\langle V', d, L \rangle$. Hence, $G \in HC$ iff $f(G) \in TSP - D$.

Cont'd

Since it is easy to see that TSP-D is NP, it follows that TSP-D is NP-complete. $\square$

Dealing with NP-completeness

As we saw in HW #2, TSP-OPT is computable in polynomial-time if and only if TSP-D is computable in polynomial-time. Since TSP-D is NP-complete, this means that if TSP-OPT has a polynomial-time algorithms, then P=NP.

Actually, the news about TSP-OPT is much worse......

Dealing with NP-completeness: Approximation

One common approach to dealing with NP-completeness is to use an efficient (p-time) algorithm to produce a solution that is "close" to the optimum.

Briefly, we write $OPT_{TSP}(I)$ for the value of the optimal solution for TSP-OPT on input I. Likewise, assume that A is a polynomial-time algorithm that produces a solution (permutation of V) to TSP-OPT, but not necessarily the optimum solution. Let $A(I)$ be the value (total length of the tour) of the solution produced by A on input I. Then, the *approximation ratio* of A on input I is

$$r_A(I) = \frac{A(I)}{OPT_{TSP}(I)} \geq 1.$$

Bad News about TSP

Theorem 6.2 Unless $P = NP$, for every polynomial-time algorithm A for TSP-OPT and every constant c, there exists an I such that

$$r_A(I) \geq c.$$

Proof: Assume that there exists a constant c and an algorithm A for TSP such that $r_A(I) \leq c$ for every I. We will show that this algorithm allows us to place HC in P.

Given an instance $G = (V, E)$ of HC, construct an instance $\langle V' = V, d, L \rangle$ of TSP, where

$$d(u, v) = \begin{cases} 1 & \text{if } \{u, v\} \in E \\ |V| * c + 1 & \text{otherwise} \end{cases}$$

Proof, cont'd

It is clear that $OPT(I) = |V|$ if $G \in HC$ and $OPT(I) \geq |V| * c + 1$ if $G \notin HC$.

Now, use the algorithm A for TSP on the instance $\langle V', d \rangle$. Since A approximates $TSP - OPT$ to the ratio c, we have that $\frac{A(I)}{OPT_{TSP}(I)} \leq c$. So, if $G \in HC$, then the optimal solution to the TSP instance has length $|V|$. It follows that $\frac{A(I)}{OPT_{TSP}(I)} \leq c = \frac{A(I)}{|V|} \leq c$ and hence $A(I) \leq |V| * c$. Similarly, if $G \notin HC$, then $|V| * c + 1 \geq OPT_{TSP}(I) \geq A(I)$. Hence, to determine whether $G \in HC$, it suffices to test whether $A(I) \leq |V| * c$. This solves HC in polynomial-time. So, $P = NP$.

It follows that no such algorithm exists for TSP unless P=NP.
$\square$

Commentary on Gap Producing Reduction

The implicit reduction f that we gave from HC to TSP in the proof of Theorem 6.2 produces a "gap" in the value of the optimal solution that depended on whether or not G had a Hamiltonian cycle. When G has a Hamiltonian cycle, then $OPT(f(G)) = |V|$. When G did not have a Hamiltonian cycle, then $OPT(f(G)) \geq |V| * c + 1$. This gap producing reduction was exploited to show that no constant ration approximation algorithm for TSP-OPT exists unless P=NP.

Notice that the reduction from HC to TSP-D in Theorem 6.1 *did not* produce a significant gap.

An aside: A related problem that is NP-complete, Hamiltonian Path

Hamiltonian-Path(HP): Given a graph $G = (V, E)$, and two vertices u and v, determine if there is a path from u to v in G that visits every vertex in G exactly once.

Let's prove that HP is NP-complete by a reduction from HC.