CMSC 641, Design and Analysis of Algorithms, Spring 2010

Sandipan Dey,
Homework Assignment - 8

30/30

April 13, 2010

Hamiltonian Path Problem is NP-Complete

V

10

Hamiltonian Path $\in$ NP

Given a (yes) certificate, i.e., a graph $G(V, E)$ with a sequence of vertices $P = \{v_1, v_2, \ldots v_n\}$, we need to show that we can verify whether P is a valid Hamiltionian path in polynomial time.

We need to verify the following:

- P is a path in $G(V, E) \Rightarrow$
  1. Each vertex on P is from $V(G)$, i.e., $\forall v_i \in P,\ v_i \in V(G)$, a $\theta(n)$ check.
  2. P does not contain any duplicate vertices (contains each vertex exactly once), i.e., $i \neq j \Leftrightarrow v_i \neq v_j,\ \forall v_i, v_j \in P$, a $\theta(n^2)$ check in the worst case.
  3. 2 consecutive vertices on P are connected by an edge in G, i.e., $(v_i, v_{i+1}) \in E(G),\ \forall v_i, v_{i+1} \in P$, again a $\theta(n + |E|)$ check in the worst case.
- P is a Hamiltonian path (covers all the vertices in $V(G)$), i.e., $|V(G)| = n$, a $\theta(1)$ operation.

Hence, whether P is a valid Hamiltionian path in G can be verified in polynomial time $\Rightarrow$ Hamiltonian Path $\in$ NP.

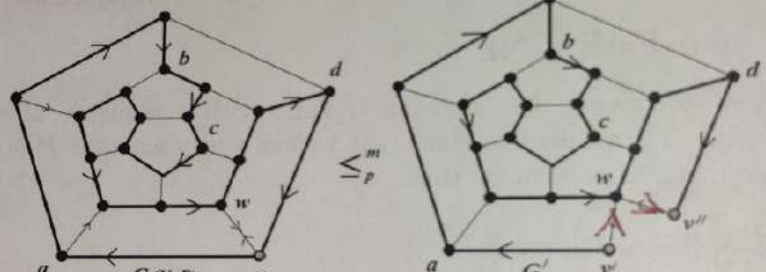
Hamiltonian Path is NP-hard

Let's reduce Hamiltonian Path from well known NP-hard problem Hamiltonian Cycle, i.e., we shall show Hamiltoinan Path $\leq_p^m$ $HAMCYCLE$.

Construction

Given instance of Hamiltonian Cycle on an undirected graph $G(V, E)$, construct another directed graph $G'(V', E')$.

- First convert the undirected graph G to a directed graph by adding directed edges $u \to v$ and $v \to u$ in place of undirected edge (u, v).
- Choose an arbitrary node $v \in V(G)$ and split it into two nodes to v', v'' get graph $G'(V', E')$ as shown in the figure, with $V' = V - \{v\} \cup \{v', v''\}$.
- Replace all incoming edges (w, v) to v in G by (w, v'') in G'.
- Similarly, replace all outgoing edges (v, w) from v in G by (v', w) in G'.
- Any Hamiltonian Path on G' must start at v' and end at v''.



pay attention

Hamiltonian Cycle
v-a-b-c-w-d-v

$V' = (V - \{v\}) \cup \{v', v''\}$
$E' = (E - (v, w) - (w, v)) \cup (v', w) \cup (w, v'')$
Hamiltonian Path
v'-a-b-c-w-d-v''

Proof: G' has a Hamiltonian Path iff G has a Hamiltonian Cycle

- $\Rightarrow$ If G'' has a Hamiltonian Path, then the same ordering of nodes, after gluing v' and v'' back together is a Hamiltonian cycle in G.
- $\Leftarrow$ If G has a Hamiltonian Cycle, then the same ordering of nodes is a Hamiltonian path of G' after splitting up v into v' and v''.

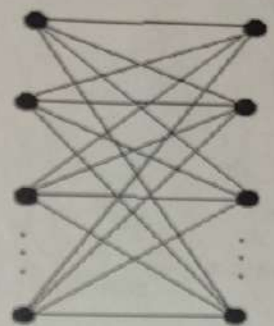
The Reduction is a polynomial time reduction

We need to only split the node into two nodes which is a constant time operation and accordingly change all the edges to this node, which can be at most $\theta(|V|)$, hence polynomial.

Bad Graph for APPROX-VERTEX-COVER

10

Complete bipartite graph $K_{n,n}$, shown in the following figure, is an example graph for which the APPROX-VERTEX-COVER will always yield suboptimal solution.



$K_{n,n}$

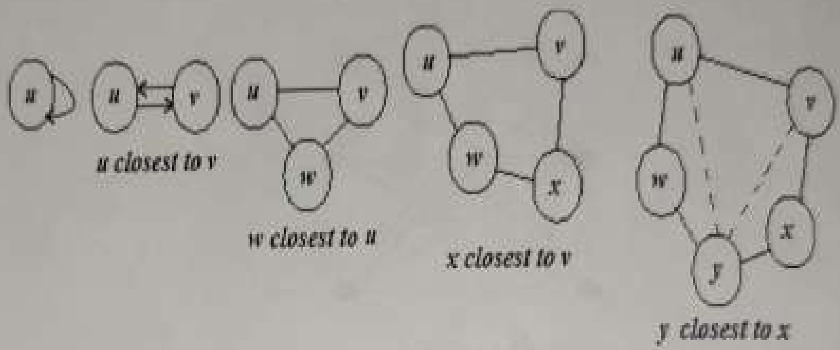
- Any side of the bipartition gives a complete vertex cover, hence $|C^*| = n$.
- Since there will be total n matching for $K_{n,n}$, no matter whiever order the edges are picked up by the APPROX-VERTEX-COVER algorithm, there will be exactly n edges picked up (each edge removes one matching, there are n of them). Hence, $|C| = 2|A| = 2n$.
- With complete bipartite graph, the $2-factor$ approximation algorithm always picks twice $\left(\frac{|C|}{|C^*|} = 2\right)$ as much edges when compared to the optimal vertex cover, achieving the upper bound.

Closest-Point Heuristic for TSP

10

- There are two disjoint sets V_C (vertices on the cycle) and $V - V_C$ (vertices not on the the cycle) throughout the run of the algorithm, intially $V_C = \{v\}$ (any arbitrarily chosen vertex), finally $V_C = V$.
- There is a CUT and always the lightest weight repecting the CUT is chosen to augment the cycle.
- Hence we shall have a minimum spanning tree included in the cycle when the algorithm terminates.
- Since each iteration adds one edge to the cycle and finally the cycle has exactly n edges inside which there must be an MST T with $n-1$ edges and one more edge e. Hence, cost of the cycle $= c(C) = c(T) + c(e)$.
- But by triangle inequality, $c(e) \leq c(T) \Rightarrow c(C) \leq 2.c(T)$.

- Also, optimal cycle cost $c(H^*) \geq c(T)$ (since if an edge is removed from the optimum cycle it becomes a spanning tree, having cost more than or equal to the MST T).
- Hence, $c(C) \leq 2.c(T) \leq 2c(H^*) \Rightarrow \frac{c(C)}{c(H^*)} \leq 2$.



w-u-v-x-y MST

c(MST) = c(w-u-v-x-y)

$c(w,y) \leq c(w,u) + c(u,y) \leq c(w,u) + c(u,v) + c(v,y) \leq c(w,u) + c(u,v) + c(v,x) + c(x,y) = c(MST)$

$c(C) = c(MST) + c(w,y) \leq 2.\, c(MST)$

TSP Closest-Point Heuristic