Teorema 1, slide 7

Nu există nicio valoare c pentru care sa existe un algoritm în timp polinomial și care să ofere o soluție cu un factor de aproximare c pentru TSP (in forma generala), decât dacă P=NP.

Justificare:
presupunem ca exista un algoritm aproximativ astfel incat daca exista S - costul optim al unui traseu inchis pt TSP, atunci algoritmul meu va oferi un traseu inchis de cost c*S

Fie G - un graf simplu neponderat. Problema determinarii unui HC in G este NPC.

Construim G' pe baza lui G dupa cum urmeaza:

V(G')=V(G)=n

toate muchiile din G apar si in G' cu costul 1.

completam cu muchii pana cand G' va fi graf complet. Fiecare muchie care este completata pe langa cele din G va avea costul c*n

Daca G avea un ciclu hamiltonian?

Algortmul aproximativ pt G' va oferi un traseu de cost cel mult c*n (obtinut in timp polinomial)

Daca G nu are un ciclu hamiltonian?

Inseamna ca cel mai bun traseu inchis din G va contine cel mult n-1 muchii de cost 1 si o muchie de cost c*n. Deci cel mai bun traseu va fi (n-1)+c*n. Iar algoritmul aproximativ va oferi un rezultat >c*n (in timp polinomial)

pe G' il obtin in timp poliomial. Algoritmul ruleaza in timp polinomial, pe baza algortmului pot vedea daca G este hamiltonian sau nu.

Contradictie - HC se poate rezolva in timp polinomial

Lema 2 slide 12

Fie G un graf complet, ponderat, care respectă regula triunghiului. Și fie v1, v2, v3, …., vk un lanț în graful G. Atunci avem len((v1,vk))≤len(v1, v2, v3, …., vk)

Justificare: inductie

presupunem ca len((v1,v_{k-1}))≤len(v1, v2, v3, …., v_{k-1})

din regula triunghiului avem ca

len((v1,vk))≤len((v1,v_{k-1}))+len((v_{k-1},v_k))≤len(v1, v2, v3, …., v_{k-1})+len((v_{k-1},v_k))

=len(v1, v2, v3, …., vk) qed.

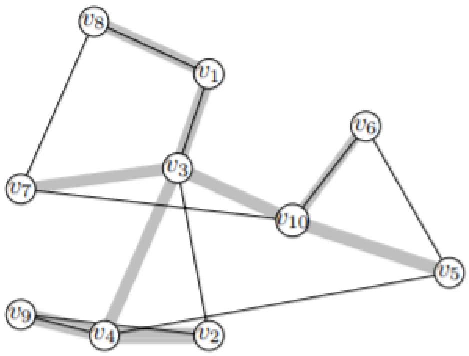
Lema 3 slide 17:
OPT>= MST

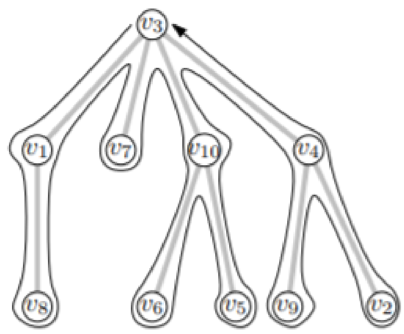
Justificare:

presupunem ca MST>OPT; Dar, din ciclul hamiltonian putem sa scoatem orice muchie si obinem un lant care trece prin toate nodurile (deci este arbore partial) cu un cost total <MST. XXX

Teorema 4 slide 20:

Algoritmul descris in slideurile 18-19 este un algoritm 2-aproximativ pentru TSP.





Observam ca ciclul nostru contine atat muchii/lanturi din MST - parcurse o singura data (ex: v3-v1-v8) dar si muchii care nu sunt in MST (ex: v8-v7), dar costul unei muchii de forma (xy) - care nu este in MST - va fi mai mic decat costul unicului lant din MST care uneste x de y (ex costul lui v8-v7 va fi < costul lantului v8-v1-v3-v7) conform Lemei 2

ALG va contine muchii care sunt in MST si muchii care nu sunt in MST dar au costul < decat lanturile echivalente din MST

ALG<=2*MST<=**2*OPT**