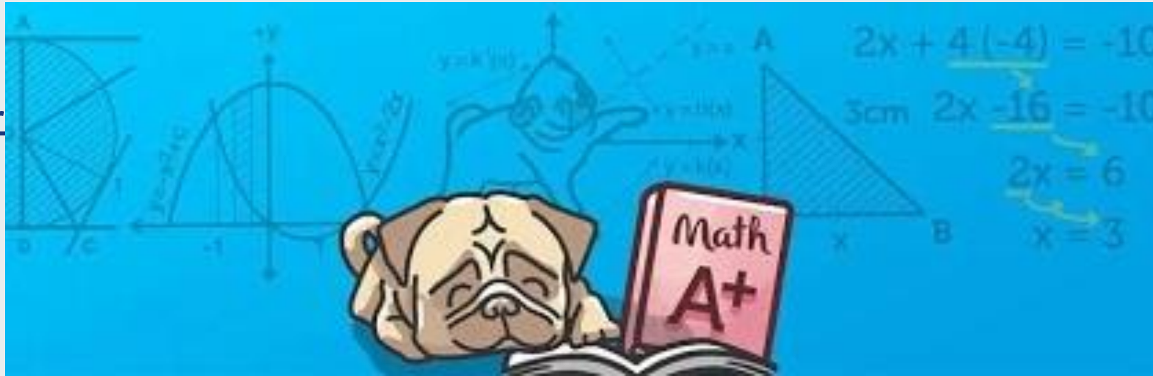
Algoritmi Avansați 2021
c-3
Hamiltonian Cycle Problem, TSP, bonus: Christofides’ algorithm

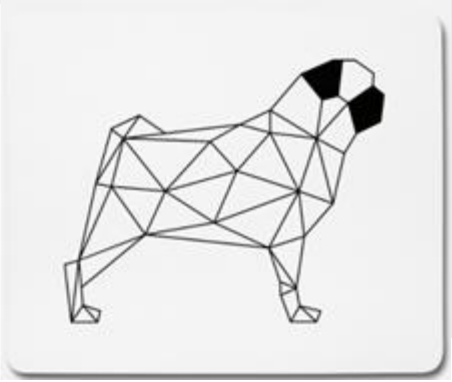
Lect. Dr. Ștefan Popescu

Email: stefan.popescu@fmi.unibuc.r

Grup Teams:



Ciclu Hamiltomian (HC-Problem)



Fie *G=(V,E)* un graf neorientat.

Numim *ciclu hamiltonian* un ciclu în *G* cu proprietatea că fiecare nod apare exact o singură dată.

HC-Problem este problema de decizie dacă într-un graf oarecare există sau nu un astfel de ciclu.

<u>HC-Problem este NP-Completa</u>

Traveling Salesman Problem (TSP)

Fie G un graf complet cu ponderi>0 pe muchii.

Evident G este graf hamiltonian, dar se pune problema găsirii ciclului hamiltonian de cost total minim.

Costul unui ciclu este suma costurilor muchiilor din componența sa.

Traveling Salesman Problem (TSP)

Fie G un graf complet cu ponderi>0 pe muchii.

Evident G este graf hamiltonian, dar se pune problema găsirii ciclului hamiltonian de cost total minim.

Costul unui ciclu este suma costurilor muchiilor din componența sa.

TSP:
”Un vânzător ambulant vrea să își promoveze produsele în *n* locații. El dorește să treacă prin toate localitățile o singură dată, la final ajungând în localitatea de unde a plecat. Pentru a lucra cât mai eficient, vânzătorul dorește sa minimizeze costul total al deplasării”

Traveling Salesman Problem (TSP)

Fie G un graf complet cu ponderi>0 pe muchii.

Evident G este graf hamiltonian, dar se pune problema găsirii ciclului hamiltonian de cost total minim.

Costul unui ciclu este suma costurilor muchiilor din componența sa.

TSP:
”Un vânzător ambulant vrea să își promoveze produsele în *n* locații. El dorește să treacă prin toate localitățile o singură dată, la final ajungând în localitatea de unde a plecat. Pentru a lucra cât mai eficient, vânzătorul dorește sa minimizeze costul total al deplasării”

TSP este o problema **NP-hard**. **Găsirea unui algoritm aproximativ este necesara!**

Traveling Salesman Problem (TSP)

TSP:
”Un vânzător ambulant vrea să își promoveze produsele în *n* locații. El dorește să treacă prin toate localitățile o singură dată, la final ajungând în localitatea de unde a plecat. Pentru a lucra cât mai eficient, vânzătorul dorește sa minimizeze costul total al deplasării”

TSP este o problema **NP-hard**. **Găsirea unui algoritm aproximativ este necesara!**

După cum vom vedea, nu dispunem de un astfel de algoritm.

Traveling Salesman Problem (TSP)

TSP:
”Un vânzător ambulant vrea să își promoveze produsele în *n* locații. El dorește să treacă prin toate localitățile o singură dată, la final ajungând în localitatea de unde a plecat. Pentru a lucra cât mai eficient, vânzătorul dorește sa minimizeze costul total al deplasării”

Teorema 1.

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, decât dacă **P=NP**.

Demo: Vom arată că există un asemenea algoritm aproximativ, dacă și numai dacă putem rezolva problema HC în timp polinomial.

Seria 23 & Seria 24

Traveling Salesman Problem (TSP)

În ciuda pesimismului oferit de rezultatul anterior, putem fi optimiști. :-)

Pug-ul nostru comis-voiajor se deplasează într-un spațiu euclidian. Deci se respectă întotdeauna regula triunghiului!

Traveling Salesman Problem (TSP)

În ciuda pesimismului oferit de rezultatul anterior, putem fi optimiști. :-)

Pug-ul nostru comis-voiajor se deplasează într-un spațiu euclidian. Deci se respectă întotdeauna regula triunghiului!

Regula triunghiului (recap): Pentru orice triunghi cu lungimea laturilor $L_1 \geq L_2 \geq L_3$, avem $L_3 + L_2 \geq L_1$

Traveling Salesman Problem (TSP)

În ciuda pesimismului oferit de rezultatul anterior, putem fi optimiști. :-)

Pug-ul nostru comis-voiajor se deplasează într-un spațiu euclidian. Deci se respectă întotdeauna regula triunghiului!

Regula triunghiului (recap): Pentru orice triunghi cu lungimea laturilor $L_1 \geq L_2 \geq L_3$, avem $L_3 + L_2 \geq L_1$

Pentru un graf complet ponderat, care respectă regula triunghiului, există algoritmi aproximativi pentru rezolvarea TSP!

Traveling Salesman Problem (TSP)

În ciuda pesimismului oferit de rezultatul anterior, putem fi optimiști. :-)

Pug-ul nostru comis-voiajor se deplasează într-un spațiu euclidian. Deci se respectă întotdeauna regula triunghiului!

Regula triunghiului (recap): Pentru orice triunghi cu lungimea laturilor $L_1 \geq L_2 \geq L_3$, avem $L_3 + L_2 \geq L_1$

Pentru un graf complet ponderat, care respectă regula triunghiului, există algoritmi aproximativi pentru rezolvarea TSP!!!

Traveling Salesman Problem (TSP)

Regula triunghiului pe grafuri ne spune că pentru oricare 3 noduri interconectate *u,v,w* avem:

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

Altfel spus, odată ce am traversat nodurile *u,v,w* - în această ordine, este mai eficient ca să ne întoarcem în *u* direct din *w* decât via *v*.

Observație 2:

Fie G un graf complet, ponderat, care respectă regula triunghiului. Și fie $v_1, v_2, v_3, \ldots, v_k$ un lanț în graful G. Atunci avem $len((v_1,v_k)) \leq len(v_1, v_2, v_3, \ldots, v_k)$

Demo: Seria 23 & Seria 24 Hint: Inducție

Traveling Salesman Problem (TSP) algoritm 2-aproximativ:

Arbore parțial de cost minim - algoritmi și timpi de lucru

Asemănare dintre MST și TSP?

Traveling Salesman Problem (TSP) algoritm 2-aproximativ:

Arbore parțial de cost minim - algoritmi și timpi de lucru

Asemănare dintre MST și TSP?

Ambele caută un traseu de cost total minim care sa cuprindă toate nodurile

Traveling Salesman Problem (TSP) algoritm 2-aproximativ:

Arbore parțial de cost minim - algoritmi și timpi de lucru

Diferențe dintre MST și TSP?

Traveling Salesman Problem (TSP) algoritm 2-aproximativ:

Arbore parțial de cost minim - algoritmi și timpi de lucru

Diferențe dintre MST și TSP?

- unul este un arbore, altul este un ciclu
- una este P iar alta este NP hard!

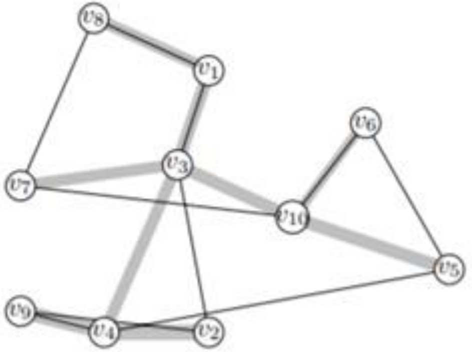
Traveling Salesman Problem (TSP) algoritm 2-aproximativ:

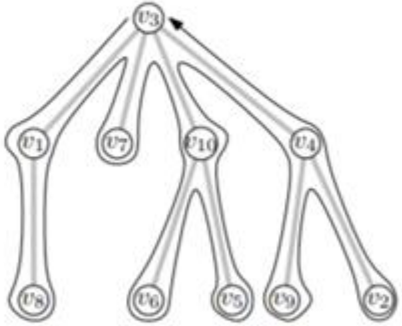
Lema 3:

Fie OPT costul soluției optime pentru TSP, iar MST - ponderea totală a unui Arbore parțial de cost minim pe baza aceluiași graf. Avem relația

OPT≥MST

Demo: [Seria 23](#) & [Seria 24](#)





Traveling Salesman Problem (TSP) algoritm 2-aproximativ:

ApproxTSP(G)

1: Calculam arborele partial de cost minim T pentru graful G.

2: Alegem un nod u ∈ T pe post de radacina.

3: Γ=Ø.

4: Parcurgere (u, Γ)

5:concatenam nodul u la finalul lui Γ pentru a inchide un ciclu .

6: return Γ

Traveling Salesman Problem (TSP) algoritm 2-aproximativ:

Parcurgere(u, Γ)

1: Concatenam pe u la Γ.

2: pentru fiecare v, fiu al lui u:

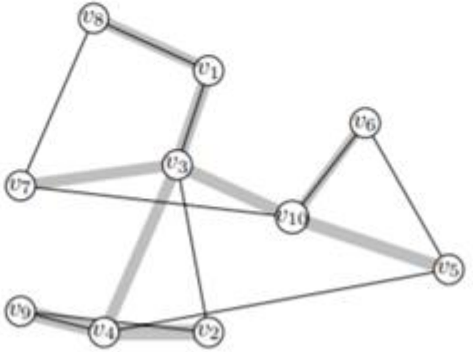
3: Parcurgere(v, Γ)

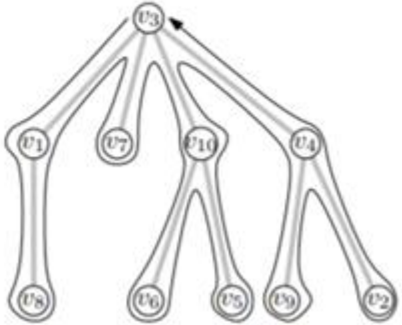
Traveling Salesman Problem (TSP) algoritm 2-aproximativ:

Teorema 4:

Algoritmul descris anterior este un algoritm 2-aroximativ pentru TSP

Demo: [Seria 23](#) & [Seria 24](#)





Traveling Salesman Problem (TSP)

Se poate oare mai bine?

Traveling Salesman Problem (TSP)

Se poate oare mai bine?

DA!

Traveling Salesman Problem (TSP) BONUS!

Se poate oare mai bine?

Algoritmul lui Christofides!

Traveling Salesman Problem (TSP) BONUS!

ChristofidesTSP(G)

1: Calculam T, un APCM in G

2: Fie $V* \subset V$ multimea de varfuri de grad impar din T . (va exista mereu un numar par de varfuri de grad impar)

3: Fie graful $G* = (V*, E*)$ - graful complet indus de $V*$.

4: Calculam M - cuplajul perfect de pondere totala minima pentru $G*$

5: reunim multimile M si T ,

6: deoarece toate nodurile au grad par, putem evidentia un ciclu Eulerian Γ in multigraful indus de M U T

7: Pentru fiecare varf din Γ, eliminam toate “dublurile” sale, reducand costul total.

8: return Γ

Next time: