

Föreläsning 33 i ADK20

**Repetition av kursen
Kursens betygssystem
Återstående examination**

Viggo Kann

KTH

Cykelbudets problem

- Cykelbudfirman Cybud får varje dag en mängd brev som ska levereras till olika adresser i stan
- Cykelbudet Bud vill planera sin cykelrutt så att den tar så kort tid som möjligt
- Hur ska han göra?

Cykelbudets problem

Indata:

- $n - 1$ stycken platser (adresser) $p_1, \dots, p_{n-1}$ samt budfirmans bas p_0
- Tid att ta sig från p_i till p_j : $t(p_i, p_j)$

Lösning:

- Cykelrutt som utgår från p_0 och passerar alla platser $p_1, \dots, p_{n-1}$ så snabbt som möjligt
- d.v.s. permutation $\pi_1, \dots, \pi_{n-1}$ som minimerar:

$$t(p_0, p_{\pi_1}) + \sum_{i=1}^{n-2} t(p_{\pi_i}, p_{\pi_{i+1}}) + t(p_{\pi_{n-1}}, p_0)$$

Går det att lösa optimalt?

Nej, inte i polynomisk tid om inte $\mathcal{P} = \mathcal{NP}$ eftersom TSP är NP-fullständigt

Vad gör man då?

- 1 Hitta approximationsalgoritm
- 2 Hitta heuristik
- 3 Hitta exakt algoritm för små indata
- 4 Förenkla problemet genom att utnyttja egenskaper hos det praktiska problemet, ta reda på om det förenklade problemet är NP-svårt och börja om från 1

Kan TSP approximeras inom en konstant?

Nej! Vi har visat att $\text{TSP} \notin \text{APX}$

- Anta motsatsen, d.v.s. att TSP kan approximeras inom f
- Reduktion från hamiltonsk cykel:

function HAMILTONIANCYCLE($G = \langle V, E \rangle$)

$n \leftarrow |V|$

for $(v_i, v_j) \in E$ **do**

$t(p_i, p_j) \leftarrow 1$

$t(p_j, p_i) \leftarrow 1$

for $(v_i, v_j) \notin E$ **do**

$t(p_i, p_j) \leftarrow |V| \cdot f$

if TSPAPPROX($\{p_i\}, t$) $\leq |V| \cdot f$ **then**

return true

else return false

- Om TSPAPPROX kan approximera TSP inom faktorn f så avgör ovanstående algoritm ifall det finns en hamiltonsk cykel i G , vilket är NP-fullständigt. Motsägelse!

Algoritm som löser TSP exakt

Använd dynamisk programmering:

- Låt ℓ_i^S vara längden av kortaste stigen från p_0 till p_i som på vägen passerar p_j för alla $j \in S$
- $$\text{OPT} = \min_{i \in \{1, \dots, n-1\}} \left(\ell_i^{\{1, \dots, n-1\} - \{i\}} + t(p_i, p_0) \right)$$
- $\ell_i^\emptyset = t(p_0, p_i)$
- $$\ell_j^S = \min_{i \in S} \left(\ell_i^{S - \{i\}} + t(p_i, p_j) \right)$$
- Beräkna nu ℓ_j^S för större och större delmängder S av $\{1, \dots, n-1\}$
- Beräkna sedan OPT ur formeln ovan.

Tid: $\mathcal{O}(n^2 \cdot 2^n)$

Budfirman ligger på Manhattan!

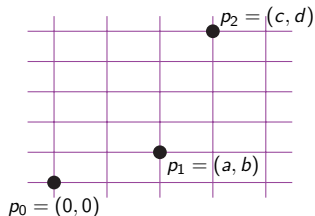
- $\Rightarrow p_i = (a_i, g_i)$ och $t(p_i, p_j) = |a_i - a_j| + |g_i - g_j|$
- Mycket enkel metrik! (Rektilinjär- eller Manhattanmetrik)

Fortfarande NP-fullständigt?

- Slå upp i litteraturen:
[Garey & Johnson: Computers and intractability] Problem ND23
" ... remains $\mathcal{NP}$ -complete if the distance is replaced by the L_1
"Rectilinear" metric [Garey, Graham, Johnson 76]"
- Attans!

Kan det förenklade problemet approximeras?

- Slå upp i Viggos lista med approximationsresultat
<https://www.people.kth.se/~viggo/problemlist/>
"TSP med triangelolikhet kan approximeras inom $\frac{3}{2}$ med Christofides algoritm"
- Kolla om triangelolikheten är uppfylld:



- Visa $t(p_0, p_2) \leq t(p_0, p_1) + t(p_1, p_2)$
- $|c| + |d| \leq |a| + |b| + |c - a| + |d - b|$
- $\left. \begin{array}{l} c \geq a \geq 0 \\ d \geq b \geq 0 \end{array} \right\} \quad c + d \leq a + b + c - a + d - b = c + d$
- OK!

Kan vi göra ännu bättre än Christofides?

- Christofides algoritm garanterar en lösning som är högst 50% längre än den optimala.
- På föreläsning 31 presenterades olika TSP-heuristiker.
- Låt oss köra både Christofides och ett antal heuristiker
- Välj den bästa lösningen av alla dessa algoritmer