

Approximations-algoritmer

- Motivation
- Definitioner
- Approximations-algoritme for knudeoverdækning
- Approximations-algoritme for TSP med trekantsulighed
- Negativt resultat om generel TSP
- Approximations-algoritme for SET-COVERING

- Fuldt polynomiel-tids approximations skema (FPTAS) for SUBSET-SUM

1

Løsningsmetoder for \mathcal{NP} -hårde opt.problemer

Opdelingskriterier

- løsningskvalitet: optimal/ikke-optimal
- beregningstid: polynomiel/ikke-polynomiel

Ikke-optimale metoder:

- løsningskvalitet:
 - ingen garanti kan gives
 - garanti kan gives, men det kan ikke gøres vilkårligt godt
 - garanti kan gives, og det kan gøres vilkårligt godt
- beregningstid:
 - ingen garanti kan gives
 - polynomiel i inddata
 - polynomiel i inddata og præcision

2

Eksempel: knapsack problem

Heuristik for Knapsack Problemet

$$\begin{aligned} \max \quad & \sum_{j=1}^n p_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^n w_j x_j \leq c \\ & x_j \in \{0, 1\} \end{aligned}$$

- sorter efter aftagende effektivitet p_j/w_j
- fyld grådigt 1, 2, ... så længe plads

Vilkårligt dårlig løsning:

$$\begin{aligned} p_1 = 1, w_1 = 1 \\ p_2 = M, w_2 = M + 1 \\ c = M + 1 \end{aligned}$$

Vi har:

- heuristisk løsning: $C = 1$
- optimal løsning: $C^* = M$
- forhold:

$$\frac{C^*}{C} = M$$

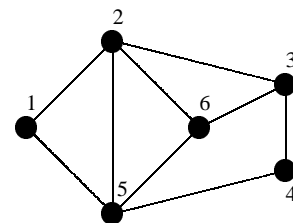
3

Eksempel: knudeoverdækning

En knudeoverdækning af en ikke orienteret graf $G = (V, E)$ er en delmængde af $V' \subseteq V$ så

$$(u, v) \in E \Rightarrow u \in V' \text{ eller } v \in V' \text{ (eller begge)}$$

Størrelsen af en knudeoverdækning er $|V'|$.
Find den mindste overdækning i grafen.



Approximations algoritme:

- Vælg tilfældig kant $(u, v) \in E$.
- Lad $V' \leftarrow V' \cup \{u\} \cup \{v\}$
- Fjern kanter fra E som er incidente med u eller v

Approximations algoritmen finder en knudeoverdækning V' af størrelse C som højst er dobbelt så stor som den optimale knudeoverdækning V^* af størrelse C^* .

4

Approximations-algoritmer

Ikke-eksakte løsningsmetoder, der giver garanti for hvor tæt på optimum man kommer.

- C — algoritmens løsningsværdi
- C^* — problemets optimale løsningsværdi

Minimeringsproblem ($C \geq C^*$)

Mål: gør $\frac{C}{C^*}$ så lille som muligt

Maximeringsproblem ($C \leq C^*$)

Mål: gør $\frac{C^*}{C}$ så lille som muligt

Generelt krav

Gør

$$\max\left(\frac{C}{C^*}, \frac{C^*}{C}\right)$$

så lille som muligt.

$$\max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \leq \rho(n)$$

hvor $\rho(n)$ er "ratio bound". Bemærk $\rho(n) \geq 1$

5

Approximations-algoritmer

Man kan også måle relativ fejl

$$\frac{|C - C^*|}{C^*} \leq \varepsilon(n)$$

Minimering

$$\frac{|C - C^*|}{C^*} = \frac{C}{C^*} - 1$$

så

$$\frac{C}{C^*} \leq \rho(n) \Leftrightarrow \frac{C}{C^*} - 1 \leq \rho(n) - 1$$

Altså: $\varepsilon(n) = \rho(n) - 1$

Maximering

$$\frac{|C - C^*|}{C^*} = 1 - \frac{C}{C^*}$$

så

$$\frac{C^*}{C} \leq \rho(n) \Leftrightarrow \frac{C}{C^*} \geq \frac{1}{\rho(n)} \Leftrightarrow$$

$$1 - \frac{C}{C^*} \leq 1 - \frac{1}{\rho(n)} = \frac{\rho(n) - 1}{\rho(n)}$$

Altså: $\varepsilon(n) = \frac{\rho(n) - 1}{\rho(n)}$ (dermed: $\varepsilon(n) \leq \rho(n) - 1$)

6

Approximations-algoritmer

approximations skema (smuk)

- Input: instans, $\varepsilon > 0$
- Approximations-algoritme træffer valg på basis af n og ε
- Output: Løsning som ikke afviger mere end ε i relativ fejl.

Skema \rightarrow idet familie af algoritmer

polynomielt-tids approx. skema (smukkere)

- Approximations skema
 - Algoritme kører i polynomielt tid i størrelsen af n
- køretid f.eks. $2^{1/\varepsilon} n^3$

fuldt polynomielt-tids approx. skema (smukkeste)

- Approximations skema
 - Algoritmen kører i polynomielt tid målt i n og $1/\varepsilon$
- køretid f.eks. $(1/\varepsilon)^2 n^3$
(findes ej for stærkt \mathcal{NP} -hårde problemer)

7

Betegnelser

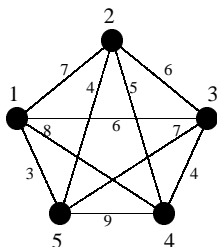
- **A** Approximations-algoritme
- **PTA** Polynomielt-Tids Approximations-algoritme
- **AS** Approximations Skema
- **PTAS** Polynomielt-Tids Approximations Skema
- **FPTAS** Fuldt Polynomielt-Tids Approximations Skema

- **Heuristik** Ingen garanti for løsningskvalitet

8

Traveling Salesman Problem

Givet graf $G = (V, E)$, omkostning $c(u, v)$ for hvert kant $(u, v) \in E$. Find billigste Hamilton-kreds.



Trekantsulighed: $u, v, w \in V$

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

(bemærk: komplet graf)

- Overholdt: geometriske problemer
- Ej overholdt: flypriser

Definition: pris af kantemængde $A \subseteq E$

$$c(A) = \sum_{(u,v) \in A} c(u, v)$$

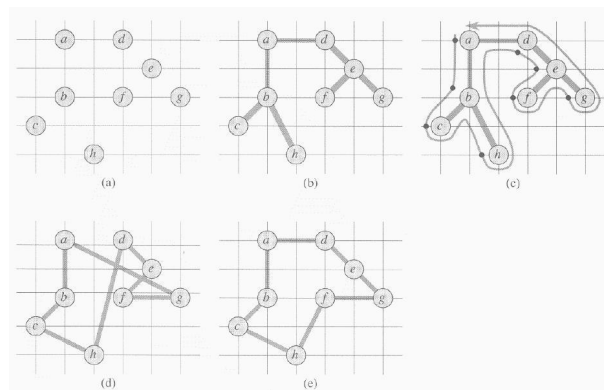
9

Approx. algoritme for TSP (trekantsulighed)

APPROX-TSP-TOUR(G, c)

- 1 select a vertex $r \in V[G]$ to be a "root" vertex
- 2 grow a minimum spanning tree T for G from root r using MST-PRIM(G, c, r)
- 3 let L be the list of vertices visited in a preorder tree walk of T
- 4 return the Hamiltonian cycle H that visits the vertices in the order L

Eksempel



10

Sætning

APPROX-TSP-TOUR er en approximations algoritme med "ratio-bound" $\rho = 2$ for TSP opfyldende trekantsulighed.

Dvs:

$$\frac{c(H)}{c(H^*)} \leq 2$$

Bevis

Mindste udspændende træ: T

$$c(T) \leq c(H^*)$$

"full walk" i grafen, besøger hver kant to gange

$$c(W) = 2c(T)$$

"walk" W ej Hamilton kreds \rightarrow sletter knuder.

Trekantsulighed sikrer

$$c(H) \leq c(W)$$

Totalt:

$$c(H) \leq c(W) = 2c(T) \leq 2c(H^*)$$

11

Approx. algoritme for TSP (generel)

Hvis $\mathcal{NP} \neq \mathcal{P}$ findes der ingen polynomieltids approximationsalgoritme for generelt TSP med "ratio bound" ρ

Bevis

Antag at fandtes polynomielt approximations algoritme A med "ratio bound" $\rho \geq 1$. Dvs finder tur C med

$$\frac{C}{C^*} \leq \rho$$

Vil vise at HAM-CYCLE kan løses i polynomieltid $\Rightarrow \mathcal{NP} = \mathcal{P}$

Givet instans af HAM-CYCLE defineret på $G = (V, E)$.

- Komplet graf: $G' = (V, E')$ hvor

$$E' = \{(u, v) : u, v \in V \text{ og } u \neq v\}$$

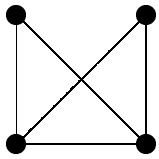
- Tildel kantvægte

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

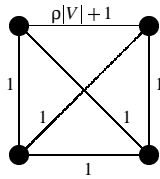
Løs TSP for (G', c) .

12

Approx. algoritme for TSP (general)



HAM-CYCLE



TSP

Afgør problem

- Hvis $C \leq \rho|V|$ så findes der en Hamilton kreds
- Hvis $C > \rho|V|$ så findes der ikke en Hamilton kreds

Vi har

- Hvis der findes en Hamilton kreds i G har TSP problemet optimal løsning $C^* = |V|$.
- Hvis der ikke findes en Hamilton kreds i G vil TSP problemet vælge mindst en "dyr" kant $c(u, v) = \rho|V| + 1$ så $C^* > \rho|V| + 1$
- Approximations algoritme A finder en løsning med pris C opfyldende

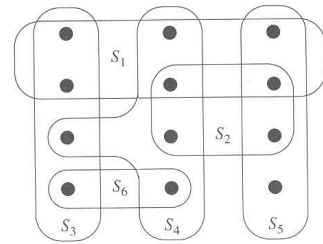
$$C \leq \rho C^*$$

Set Covering Problem

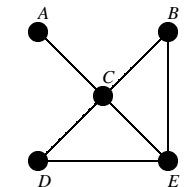
Givet en mængde X og en familie F af delmængder $S_i \subset X$, med $\cup_{i \in F} S_i = X$. Find en mindste delmængde $C \subset F$ så

$$X = \cup_{i \in C} S_i$$

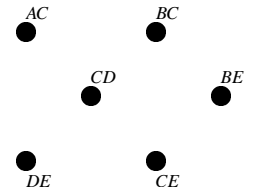
Vi siger at C overdækker X



Problemet er \mathcal{NP} -hårdt (reduktion fra VERTEX-COVER)



VERTEX-COVER



SET-COVER

Grådige algoritme

```

GREEDY-SET-COVER( $X, \mathcal{F}$ )
1  $U \leftarrow X$ 
2  $C \leftarrow \emptyset$ 
3 while  $U \neq \emptyset$  do
4   select an  $S \in \mathcal{F}$  that maximizes  $|S \cap U|$ 
5    $U \leftarrow U \setminus S$ 
6    $C \leftarrow C \cup \{S\}$ 
7 return  $C$ 
    
```

Sætning: Den grådige algoritme har "ratio-bound"

$$\frac{|C|}{|C^*|} \leq H(\max_{S \in \mathcal{F}} |S|)$$

hvor

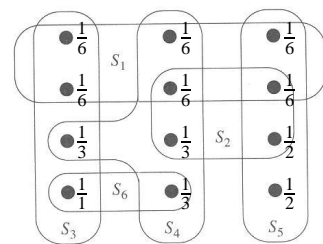
$$H(n) = \sum_{k=1}^n \frac{1}{k} \leq \ln n + 1$$

Bevis

- Antag S_i er i 'te mængde der tilføjes C i grådige algoritme.
- Lad c_x være prisen knyttet til $x \in S_i$.

$$c_x = \frac{1}{|S_i - (S_1 \cup S_2 \cup \dots \cup S_{i-1})|}$$

Set Covering Problem



Siden C^* også dækker X har vi

$$|C| = \sum_{x \in X} c_x \leq \sum_{S \in C^*} \sum_{x \in S} c_x$$

Vi vil om lidt vise at

$$\sum_{x \in S} c_x \leq H(|S|)$$

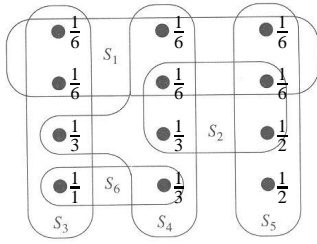
hvoraf vi får

$$\begin{aligned}
 |C| &\leq \sum_{S \in C^*} \sum_{x \in S} c_x \leq \sum_{S \in C^*} H(|S|) \\
 &\leq \sum_{S \in C^*} H(\max_{S \in \mathcal{F}} |S|) \leq |C^*| H(\max_{S \in \mathcal{F}} |S|)
 \end{aligned}$$

og dermed

$$\frac{|C|}{|C^*|} \leq H(\max_{S \in \mathcal{F}} |S|)$$

Set Covering Problem



Vil vise at

$$\sum_{x \in S} c_x \leq H(|S|)$$

Lad for enhver mængde $S \in F$

$$u_i = |S - (S_1 \cup S_2 \cup \dots \cup S_i)|$$

være antallet af ikke dækkede elementer i S efter det i 'te skridt i den grådige algoritme.

$$\sum_{x \in S} c_x = \sum_{i=1}^k (u_{i-1} - u_i) \cdot \frac{1}{|S_i - (S_1 \cup S_2 \cup \dots \cup S_{i-1})|}$$

Bemærk at

$$|S_i - (S_1 \cup S_2 \cup \dots \cup S_{i-1})| \geq |S - (S_1 \cup S_2 \cup \dots \cup S_{i-1})| = u_{i-1}$$

for enhver mængde S , så

$$\sum_{x \in S} c_x \leq \sum_{i=1}^k (u_{i-1} - u_i) \frac{1}{u_{i-1}}$$

Set Covering Problem

Der gælder at $H(n) = \sum_{j=1}^n \frac{1}{j}$ så for $b > a$

$$\begin{aligned} H(b) - H(a) &= \sum_{j=1}^b \frac{1}{j} - \sum_{j=1}^a \frac{1}{j} \\ &= \sum_{j=a+1}^b \frac{1}{j} \geq \sum_{j=a+1}^b \frac{1}{b} = (b-a) \frac{1}{b} \end{aligned}$$

Dermed får vi:

$$\begin{aligned} \sum_{x \in S} c_x &\leq \sum_{i=1}^k (u_{i-1} - u_i) \frac{1}{u_{i-1}} \\ &\leq \sum_{i=1}^k (H(u_{i-1}) - H(u_i)) \\ &= H(u_0) - H(u_k) \\ &= H(u_0) - H(0) \\ &= H(u_0) \\ &= H(|S|) \end{aligned}$$

da k er sidste index

da $H(0) = 0$