

Approximations-algoritmer

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

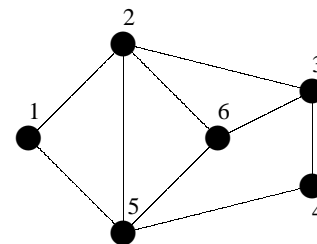
- heuristisk løsning: $C = 1$
- optimal løsning: $C^* = M$
- forhold:

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

3

Eksempel: knudeoverdækningEn 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 mulig.

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

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

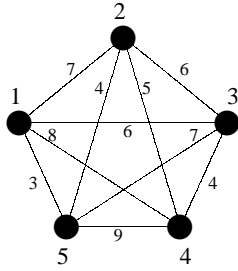
- **A** Approximations-algoritme
- **PTA** Polynomielt-Tids Approximations-algoritme
- **AS** Approximations Skema
- **PTAS** Polynomielt-Tids Approximations Skema
- **FPTAS** Fuldt Polynomielt-Tids Approximations Skema

- **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 $A \subseteq E$

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

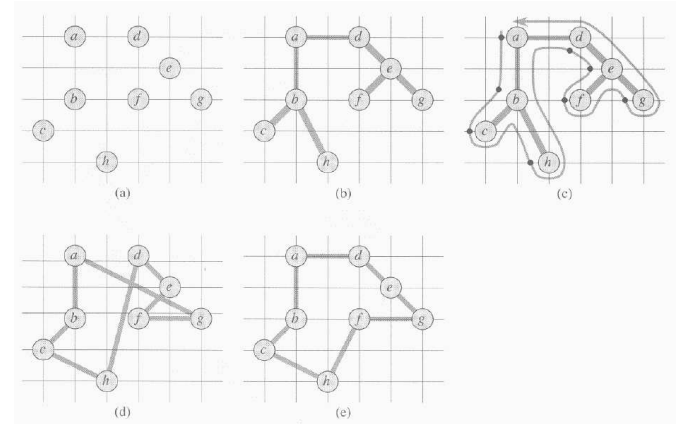
9

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

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

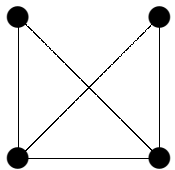
- Tildel kantvægte

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

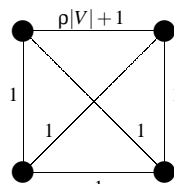
Løs TSP for (G', c) .

12

Approx. algoritme for TSP (generel)



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

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

13

Subset-sum Problem

Subset-sum problem (delmængde sum):

- Givet mængde af heltal $S = \{x_1, x_2, \dots, x_n\}$ samt t
- Find en delmængde $S' \subseteq S$ så

$$\sum_{j \in S'} x_j \leq t$$

er størst mulig

For eksempel

$$S = \{1, 4, 16, 64, 256, 1040, 1093, 1284, 1344\}$$

$$t = 3754$$

har løsning $S' = \{1, 16, 64, 256, 1040, 1093, 1284\}$.

Viser: Fuldt polynomiel-tids approximations skema

14

Ekspontiel algoritme

Dynamisk programmering-lignende algoritme.

Lister over tal som kan opnås med en delmængde af S

- Liste L af positive heltal, f.eks. $L = \langle 1, 2, 3, 5, 9 \rangle$
- x positivt heltal, f.eks. $x = 2$
- $L + x = \langle 3, 4, 5, 7, 11 \rangle$

Lad

$$P_i = \{\text{tal, der kan opnås som sum af } \{x_1, \dots, x_i\}\}$$

så gælder rekursionsligningen

$$P_i = P_{i-1} \cup (P_{i-1} + x_i)$$

Her kan foreningsmængden $L_1 \cup L_2$ udregnes i lineær tid da begge lister er sorterede i voksende orden (MERGE-LISTS).

Eksempel

Lad $S = \{4, 5, 7\}$, $t = 14$

$$P_1 = \{0, 4\}$$

$$P_2 = \{0, 4, 5, 9\}$$

$$P_3 = \{0, 4, 5, 7, 9, 11, 12, 19\}$$

15

Ekspontiel algoritme

Algoritme

EXACT-SUBSET-SUM(S, t)

```

1   $n \leftarrow |S|$ 
2   $L_0 \leftarrow \langle 0 \rangle$ 
3  for  $i \leftarrow 1$  to  $n$  do
4       $L_i \leftarrow \text{MERGE-LISTS}(L_{i-1}, L_{i-1} + x_i)$ 
5      remove from  $L_i$  every element that is greater than  $t$ 
6  return the largest element in  $L_n$ 

```

Total køretid: nt .

Ekspontiel i input størrelsen $n \log t$.

F.eks: $t = 2^n$, køretid $n2^n = O(2^n)$, input n^2 .

Idé til forbedring

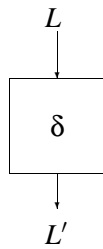
- fjern elementer i P_i som ligger "tæt" på hinanden
- behold det mindste af to tal tæt på hinanden

TRIM liste L

16

Trimning

trim parameter δ med $0 < \delta < 1$



- Fjern så mange elementer tæt på hinanden som muligt
- For ethvert fjernet element $y \in L$ eksisterer $z \in L'$ med $z \leq y$ så

$$\frac{y}{1 + \delta} \leq z$$

$$\left(y \leq (1 + \delta)z \quad \frac{y}{z} \leq (1 + \delta) \right)$$

Eksempel: $\delta = 0.1$

$L = \langle 10, 11, 12, 15, 20, 21, 22, 23, 24, 29 \rangle$

$L' = \langle 10, 12, 15, 20, 23, 29 \rangle$

17

Trimning

Algoritme:

```

TRIM(L, delta)
1 m ← |L|
2 L' ← ⟨y_1⟩
3 last ← y_1
3 for i ← 2 to m do
4   if y_i > last · (1 + delta) then
5     append y_i onto the end of L'
5     last ← y_i
6 return L'
  
```

Køretid: lineær.

18

Approximations algoritme

Givet instans (S, t) , tilladt fejl ϵ .

- vælg trimningsfaktor $\delta = \frac{\epsilon}{2n}$

APPROX-SUBSET-SUM(S, t, ϵ)

```

1 n ← |S|
2 L_0 ← ⟨0⟩
3 for i ← 1 to n do
4   L_i ← MERGE-LISTS(L_{i-1}, L_{i-1} + x_i)
5   L_i ← TRIM(L_i, epsilon/2n)
6   remove from L_i every element that is greater than t
7 return z^A given by the largest element in L_n
  
```

(Bogen kalder returnerede værdi z^* , hvilket ikke harmonerer med at z^* angiver optimal løsning)

Sætning

Fuldt polynomiel-tids approximations skema

- 1 Finder en lovlig løsning z^A
- 2 Relative fejl er mindre end ϵ

$$\frac{|z^* - z^A|}{z^*} \leq \epsilon$$

- 3 Algoritmen kører i polynomiel tid i n og $1/\epsilon$

Vi kan antage $\epsilon < 1$, da 2-approximation nem (overvej!)

19

Eksempel

Instans: $S = \{104, 102, 201, 101\}$, $t = 308$, $\epsilon = 0.40$
Vælger $\delta = \epsilon/2n = 0.40/8 = 0.05$.

Forløb:

line 2: $L_0 = \langle 0 \rangle$

line 4: $L_1 = \langle 0, 104 \rangle$

line 5: $L_1 = \langle 0, 104 \rangle$

line 6: $L_1 = \langle 0, 104 \rangle$

line 4: $L_2 = \langle 0, 102, 104, 206 \rangle$

line 5: $L_2 = \langle 0, 102, 206 \rangle$

line 6: $L_2 = \langle 0, 102, 206 \rangle$

line 4: $L_3 = \langle 0, 102, 201, 206, 303, 407 \rangle$

line 5: $L_3 = \langle 0, 102, 201, 303, 407 \rangle$

line 6: $L_3 = \langle 0, 102, 201, 303 \rangle$

line 4: $L_4 = \langle 0, 101, 102, 201, 203, 302, 303, 404 \rangle$

line 5: $L_4 = \langle 0, 101, 201, 302, 404 \rangle$

line 6: $L_4 = \langle 0, 101, 201, 302 \rangle$

Approximativ løsning: $z^A = 302$

Optimal løsning: $z^* = 307$, (faktisk afvigelse 2%)

20

Finder en lovlig løsning

- Kun lovlig summer i L , alle $\leq t$
- Naturligvis lovlig sum i sidste iteration

Relativ fejl mindre end ε

For ethvert fjernet element $y \in P$ eksisterer $z \in L$ så

$$\frac{y}{1+\delta} \leq z \leq y$$

Ved induktion i i kan det vises at efter i iterationer gælder:

For ethvert fjernet element $y \in P_i$ eksisterer $z \in L_i$ så

$$\frac{y}{(1+\delta)^i} \leq z \leq y$$

Gælder specielt for $z^* \in P_n$, dvs. der findes $z \in L_n$ så

$$\frac{z^*}{(1+\delta)^n} \leq z \leq z^*$$

Må også gælde for z^A som er den største værdi i L_n så

$$\frac{z^*}{(1+\delta)^n} \leq z^A$$

eller

$$\frac{z^*}{z^A} \leq (1+\delta)^n$$

Vil vise at $(1+\delta)^n \leq 1+\varepsilon$ når $\delta = \varepsilon/2n$

Relativ fejl mindre end ε

Betragter $f(n, \varepsilon) = (1 + \frac{\varepsilon}{2n})^n$.

Formel (3.13) side 53 siger at

$$\lim_{n \rightarrow \infty} \left(1 + \frac{x}{n}\right)^n = e^x$$

og da $\frac{d}{dn} f(n, \varepsilon) > 0$ er f voksende.

$$\left(1 + \frac{\varepsilon}{2n}\right)^n \leq e^{\varepsilon/2}$$

Formel (3.12) side 53 siger $e^x \leq 1 + x + x^2$ for $|x| < 1$ så

$$\left(1 + \frac{\varepsilon}{2n}\right)^n \leq 1 + \varepsilon/2 + (\varepsilon/2)^2$$

og da $\varepsilon < 1$ får vi:

$$\left(1 + \frac{\varepsilon}{2n}\right)^n \leq 1 + \varepsilon$$

Kombineret med formlen forrige side fås

$$\frac{z^*}{z^A} \leq 1 + \varepsilon$$

og dermed

$$\frac{|z^* - z^A|}{z^*} = 1 - \frac{z^A}{z^*} \leq 1 - \frac{1}{1+\varepsilon} = \frac{1+\varepsilon-1}{1+\varepsilon} \leq \varepsilon$$

Kører i polynomiell tid i n og $1/\varepsilon$

TRIM sletter et element $z \geq z'$ hvis

$$\frac{z}{z'} \leq (1+\delta)$$

Så på ethvert tidspunkt vil elementerne i L_i overholde

$$z > (1+\delta)z' = \alpha z' \quad \alpha = (1+\delta) \text{ afstands-faktor}$$

Den tættest mulige liste ser ud som

$$0, 1, \alpha, \alpha^2, \alpha^3, \dots, \alpha^k \approx t$$

Antal elementer er $k+2$, hvor k findes som:

$$\alpha^k = t \Leftrightarrow k \ln \alpha = \ln t \Leftrightarrow k = \frac{\ln t}{\ln \alpha} = \frac{\ln t}{\ln(1+\delta)}$$

Formel (3.16) side 54 siger at for $\delta > -1$ gælder

$$\frac{\delta}{1+\delta} \leq \ln(1+\delta) \Leftrightarrow \frac{1}{\ln(1+\delta)} \leq \frac{1+\delta}{\delta}$$

så

$$k \leq \frac{(1+\delta) \ln t}{\delta} = \frac{(1+\varepsilon/2n) \ln t}{\varepsilon/2n}$$

Da $\varepsilon < 1$ gælder at $(1+\varepsilon/2n) \leq 2$ så

$$k \leq \frac{4n \ln t}{\varepsilon}$$

Køretid: $O(n(k+2)) = O(\frac{1}{\varepsilon} n^2 \ln t)$

Indsigt

Approximabilitet og reduktion af problemer

- Da SUBSET-SUM er $\mathcal{N}(\mathcal{P})$ -fuldstændigt gælder f.eks.

$$\text{TSP} \leq_p \text{SUBSET-SUM}$$

- Vi kan approximere SUBSET-SUM vilkårligt godt i polynomiell tid
- Gælder tilsvarende resultat for TSP?

Reduktion gælder kun for afgørlighedsproblemer

Såfremt vi vil løse SUBSET-SUM-DECISION skal $\varepsilon = 0$

Branch-and-bound som Approximations Skema

Minimeringsproblem: (givet $\epsilon > 0$)

- For ethvert underproblem (knode) lad ℓ være en nedre grænseværdi
- Lad z være hidtil bedste løsning
- Forkast underproblem hvis $\ell(1 + \epsilon) \geq z$
- Eksempel: $\epsilon = 0.1$, $\ell = 11$, $z = 12$. Forkast knode

Approximations skema. Relativ afvigelse $< \epsilon$.

Køretid: ???

Branch-and-bound i Polynomielt tid

Undersøg kun et polynomielt antal knuder $M = n^k$

- Best-first søgning
- Stop efter M knuder

Køretid polynomielt.

Løsningskvalitet: ???