I dag

Løsning af $\mathcal{NP}$-hårde optimeringsproblemer

- Repetition: branch-and-bound
- Flere begreber
- Konkret eksempel: TSP
- Lagrange relaxering
- Parallel branch-and-bound

Opsummering

Løsning af $\mathcal{NP}$-hårde optimeringsproblemer

- Optimeringsproblemer $\leftrightarrow$ afgørlighedsproblemer
- $\mathcal{NP}$-hårdhed for optimeringsproblemer
- Løs til optimalitet i eksponentiel tid
- Find tilnærmet løsning i polynomielt tid
- Optimeringsproblemer kan ikke altid verificeres i polynomiel tid

Bemærk

- $\mathcal{NP}$-hårdhed siger ikke meget om faktisk sværhed af real-life instanser
- Intet bevis af sværhed for specifikke instanser

Branch-and-bound

- Systematisk gennemsøgning af alle mulige løsninger
- Dele af gennemsøgningen undgås grundet globale betragtninger
- Der findes instanser hvor et eksponentielt antal delløsninger skal undersøges (ellers er $\mathcal{NP} = \mathcal{P}$)

Opsummering

Minimeringsproblem:

$$\min_{x \in S} f(x)$$

- $f(x)$ objektfunktion
- S løsningsmængde
- hidtil bedste løsning z (incumbent)
- forgrening (branch)
  - dichotomisk eller polytomisk opdeling
  - bredde-først, dybde-først, bedste-først søgning
- grænseværdi ℓ_i
  - overholder $\ell_i \leq f(x)$ for alle $x \in S_i$.
  - delløsning S_i forkastes (fathoming) hvis $\ell_i \geq z$.
  - ivrig eller doven evaluering af grænseværdi.
  - grænseværdi $\ell_i \geq \ell_j$ når i er afkom af j.
- startløsning
  - skal være så god som mulig: heuristikker
  - hvis $z = z^*$ så skal kun kritiske knuder besøges

Reduktion

- Formål: at mindske instansens størrelse uden at påvirke optimal løsning
- Reduktion kan forbedre løsningstider væsentligt
- Kædeeffect i reduktion
- Kan fortolkes som tentativ branching

For binær beslutning (svarende til binær variabel x_i)

- Udregn ℓ med $x_i = 0$. Hvis $\ell > z$ så ved at $x_i = 1$
- Udregn ℓ med $x_i = 1$. Hvis $\ell > z$ så ved at $x_i = 0$

(minimeringsproblem)

Hvor mange knuder skal besøges

- Lad z være nuværende løsning
- Lad z^* være optimal løsning
- *Kritiske delproblemer* S_i

$$\ell_i < z^*$$

Hvis z^* kendt fra starten besøges samme antal knuder uanset søgestrategi (bredde-først, dybde-først, bedste-først).

Hvis z^* først findes undervejs i forløbet risikerer vi også at besøge knuder med

$$\ell_i < z \quad \text{men} \quad \ell_i \geq z^*$$

Bedste-først søgestrategi vil kun besøge kritiske delproblemer.

Indsigt

Vigtigste valg i branch-and-bound algoritme

- grænseværdi funktion
- opspaltning i delproblemer
- god startløsning

Alle kritiske delproblemer skal besøges for at bevise optimalitet

Sekundære valg:

- Ivrig eller doven evaluering af grænseværdi
- Søgestrategi (bedste-først, dybde-først, bredde-først)

Hvis z^* kendes fra starten skal samme antal knuder besøges uanset søgestrategi.

Grænseværdi beregning

Modification af objektfunktion

- Problemafhængigt

Relaxering

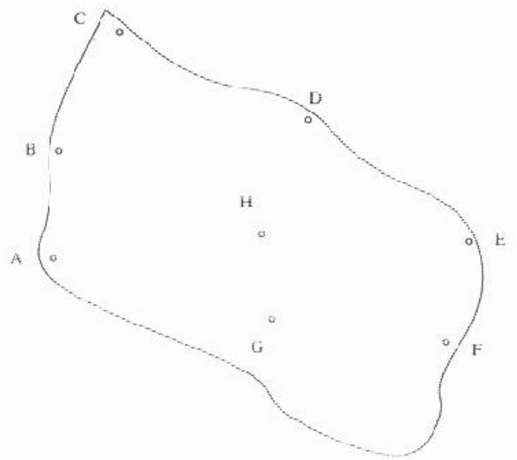
- Udvid mængden af lovlige løsninger
- Tilstræb nemt (polynomielt) resterende problem

Hvilke begrænsninger skal relaxeres

- tilbageværende problem er polynomielt løseligt (e.g. min spanning tree, assignment problem, linear programming)
- resterende problem er $\mathcal{NP}$-hårdt men gode teknikker findes (e.g. knapsack)
- begrænsninger som er svære at beskrive matematisk (e.g. cutting)
- begrænsninger som er for omfattende at beskrive explicit (e.g. subtour elimination in TSP)

Konkret Problem: Symetrisk Traveling Salesman Problem

Symmetrisk TSP med $G = (V, E)$, afstande (d_{ij})
$x_{ij} = 1$ hvis knude j efterfølger knude i



	A	B	C	D	E	F	G	H
A	0	11	24	25	30	29	15	15
B	11	0	13	20	32	37	17	17
C	24	13	0	16	30	39	29	22
D	25	20	16	0	15	23	18	12
E	30	32	30	15	0	9	23	15
F	29	37	39	23	9	0	14	21
G	15	17	29	18	23	14	0	7
H	15	17	22	12	15	21	7	0

$$
\begin{array}{lll}
\min & \sum_{p,q \in V} d_{pq} x_{pq} & \\
\text{hvor} & \sum_{p \in V} x_{pq} = 1 & \text{for } q \in V \\
& \sum_{q \in V} x_{pq} = 1 & \text{for } p \in V \\
& \sum_{p,q \in S} x_{pq} \leq |S| - 1 & \text{for } S \subset V \\
& x_{pq} \in \{0, 1\} & \text{for } p, q \in V
\end{array}
$$

Antal lovlige løsninger: $(n-1)!/2$ (for $n = 50$ fås: $3 \cdot 10^{62}$)
Antal begrænsninger

Traveling Salesman Problem

Det følgende eksempel angiver afstandende mellem 8 byer på Bornholm.

$i \backslash j$	A	B	C	D	E	F	G	H
A	0	11	24	25	30	29	15	15
B	11	0	13	20	32	37	17	17
C	24	13	0	16	30	39	29	22
D	25	20	16	0	15	23	18	12
E	30	32	30	15	0	9	23	15
F	29	37	39	23	9	0	14	21
G	15	17	29	18	23	14	0	7
H	15	17	22	12	15	21	7	0

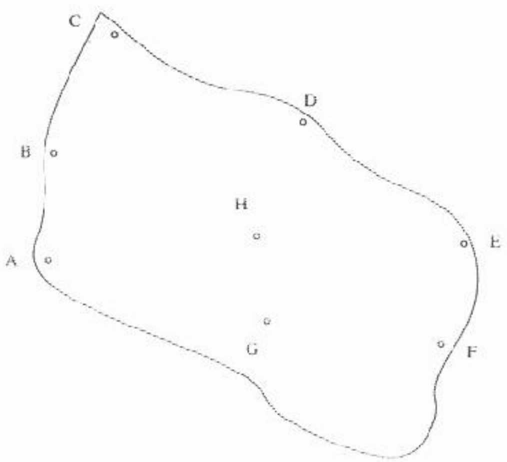
Den optimale løsning at besøge knuderne (byerne) i rækkefølgen: $A, B, C, D, E, F, G, H, A$, hvilket giver en samlet længde af Hamilton kredsen på $z^* = 100$.

Traveling Salesman Problem

- Opspaltning i delproblemer:

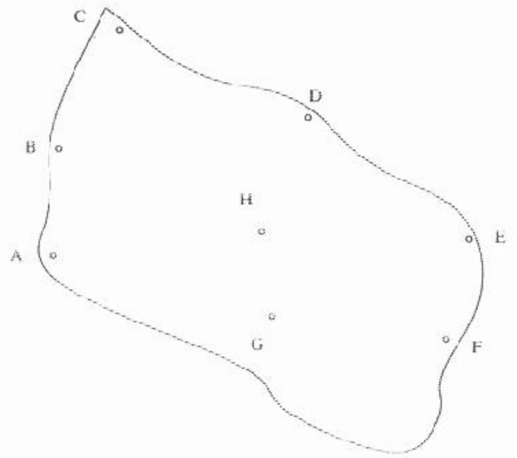
$$\begin{array}{ll} x_{ij} = 1 & \text{vælg en kant} \\ x_{ij} = 0 & \text{forbyd en kant} \end{array}$$

- Grænseværdi 1:
  - forkast ”deltur eliminering”.
  - (x_{ij}) matrix har egenskab at der netop er et 1-tal i hver række og søjle.
  - Tildelings-problem.
  - Ungarsk algoritme: $O(n^3)$.
- Grænseværdi 2:
  - 1-træ minimum udspændende træ.
  - Fjern en knude a fra en Hamilton-kreds.
  - Mindste udspændende træ på øvrige knuder (Kruskal, Prim).



	A	B	C	D	E	F	G	H
A	0	11	24	25	30	29	15	15
B	11	0	13	20	32	37	17	17
C	24	13	0	16	30	39	29	22
D	25	20	16	0	15	23	18	12
E	30	32	30	15	0	9	23	15
F	29	37	39	23	9	0	14	21
G	15	17	29	18	23	14	0	7
H	15	17	22	12	15	21	7	0

Traveling Salesman Problem



	A	B	C	D	E	F	G	H
A	0	11	24	25	30	29	15	15
B	11	0	13	20	32	37	17	17
C	24	13	0	16	30	39	29	22
D	25	20	16	0	15	23	18	12
E	30	32	30	15	0	9	23	15
F	29	37	39	23	9	0	14	21
G	15	17	29	18	23	14	0	7
H	15	17	22	12	15	21	7	0

Bedre grænseværdi ved transformation

- Omdefinerer afstands-matrix.
- Ønsker at mindste udspændende træ ”kommer tættere på”en Hamilton-kreds.

$$\sum_{p\in V,\ p\neq q} x_{pq} = 2 \quad \text{for } q \in V$$

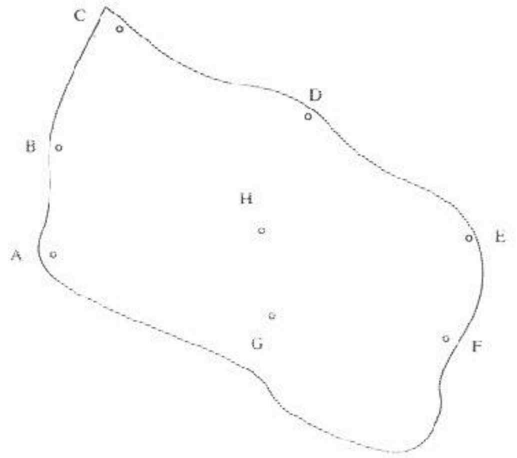
- Begrænsning overtrådt med

$$\pi_q = \sum_{p\in V,\ p\neq q} x_{pq} - 2$$

- Sæt $d'_{pq} \leftarrow d_{pq} + k(\pi_p + \pi_q)$
- Lovlig grænseværdi da en Hamilton-kreds samme længde

Traveling Salesman Problem

Udregner værdier af π_q



	A	B	C	D	E	F	G	H
A	0	11	24	25	30	29	15	15
B	11	0	13	20	32	37	17	17
C	24	13	0	16	30	39	29	22
D	25	20	16	0	15	23	18	12
E	30	32	30	15	0	9	23	15
F	29	37	39	23	9	0	14	21
G	15	17	29	18	23	14	0	7
H	15	17	22	12	15	21	7	0

Traveling Salesman Problem

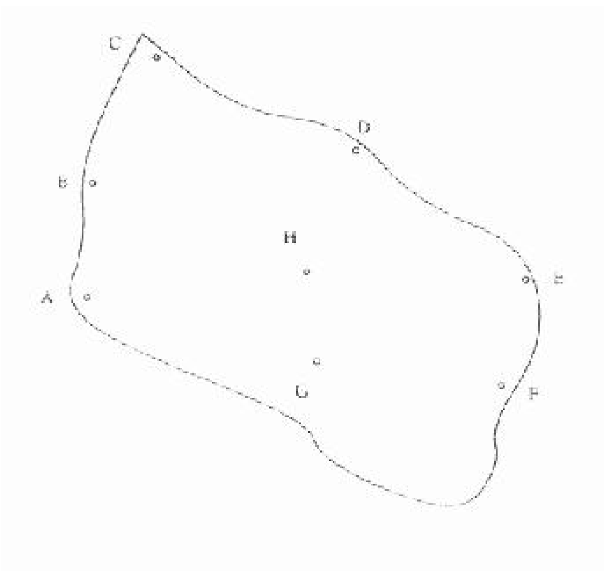
Original formulering

$i \backslash j$	A	B	C	D	E	F	G	H
A	0	11	24	25	30	29	15	15
B	11	0	13	20	32	37	17	17
C	24	13	0	16	30	39	29	22
D	25	20	16	0	15	23	18	12
E	30	32	30	15	0	9	23	15
F	29	37	39	23	9	0	14	21
G	15	17	29	18	23	14	0	7
H	15	17	22	12	15	21	7	0

Modificeret afstandsmatrix

$i \backslash j$	A	B	C	D	E	F	G	H
A	0	12	23	26	28	27	17	16
B	12	0	13	22	31	36	20	19
C	23	13	0	16	27	36	30	22
D	26	22	16	0	14	22	21	14
E	28	31	27	14	0	5	23	14
F	27	36	36	22	5	0	14	20
G	17	20	30	21	23	14	0	10
H	16	19	22	14	14	20	10	0

π_j	0	1	-1	1	-2	-2	2	1



Branching strategi

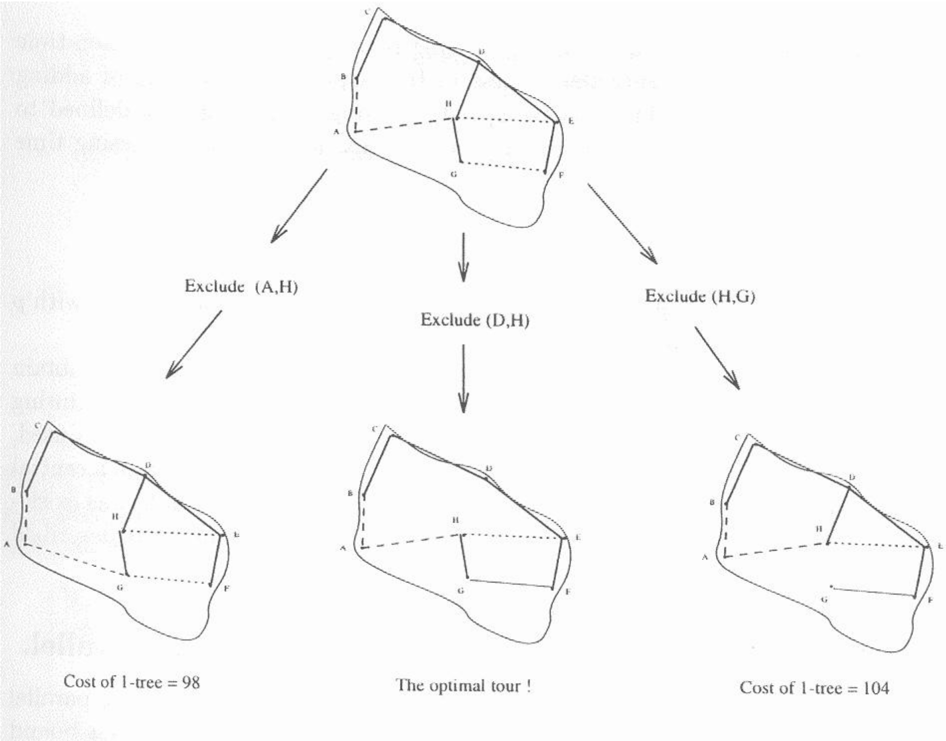


Figure 9: Branching from a 1-tree in a B&B algorithm for the symmetric TSP.

Forskellige relaxeringer

- Lineær relaxering
- Sletning af begrænsninger
- Lagrange relaxering
- Surrogate relaxering

Relaxeringer bruges ofte i kombination.

Lagrange relaxering

Minimeringsproblem

$$\begin{aligned} \text{min} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{hvor} \quad & \sum_{j=1}^{n} a_j x_j = b \\ & \text{“andre begrænsninger”} \end{aligned} \tag{1}$$

For λ reelt tal, så er Lagrange relaxerede problem

$$\begin{aligned} \text{min} \quad & \sum_{j=1}^{n} c_j x_j + \lambda\left(\sum_{j=1}^{n} a_j x_j - b\right) \\ \text{hvor} \quad & \text{“andre begrænsninger”} \end{aligned} \tag{2}$$

Sætning

Optimal løsning til (2) er nedre grænseværdi på (1).

Fortolkning

$$\begin{array}{ll} \min & \sum_{j=1}^{n} c_j x_j + \lambda\left(\sum_{j=1}^{n} a_j x_j - b\right) \\ \text{hvor} & \text{“andre begrænsninger”} \end{array}$$

Vi ønsker en så stor grænsværdi som muligt

- multiplikator λ er “straf” for ikke at overholde

$$\sum_{j=1}^{n} a_j x_j = b$$

- hvis $\lambda = 0$ slettes begrænsning.

Grafdeling

Graph Partitioning Problem

Givet graf $G = (V,E)$ og omkostnings funktion $c : E \to \mathbb{N}$. Del V i to *lige store dele* V_1 og V_2 så omkostningen af kanter (u,v) hvor $u \in V_1$ og $v \in V_2$ minimeres.

Lad $x_v = 1$ hvis $v \in V_1$ mens $x_v = 0$ hvis $v \in V_2$.
Kvadratisk minimeringsproblem:

$$\min \sum_{u,v \in V} c_{uv} x_u (1 - x_v)$$

hvor

$$\sum_{v \in V} x_v = |V|/2$$

Antal mulige løsninger:
hvis $2n = |V|$, binominal koefficient $C(2n,n)$
for $2n = 120$ fås: $9.6 \cdot 10^{34}$

Grafdeling

Grænseværdi beregning

$$\min \sum_{u\in V} \sum_{v\in V} c_{uv} x_u (1 - x_v)$$

hvor

$$\sum_{v\in V} x_v = |V|/2$$

Lagrange relaxer begrænsningen: $\lambda \in \mathbb{R}$

$$\min \sum_{u\in V} \sum_{v\in V} c_{uv} x_u (1 - x_v) - \lambda \left(\sum_{u\in V} x_u - |V|/2 \right)$$

I objektfunktionen

$$\min \sum_{u\in V} \sum_{v\in V} c_{uv} x_u (1 - x_v) - \sum_{u\in V} \lambda x_u + \lambda |V|/2$$

- første led: grafdeling uden begrænsninger på $|V_1|$ og $|V_2|$. Optimal løsning f.eks. $V_2 = V$.
- andet led: $\lambda > 0$ er "belønning" for hver knude der kommer med i V_1. $\lambda < 0$ er "straf" for hver knude der kommer med i V_1.
- tredie led: "straf" for positiv værdi af λ, "belønning" for negativ værdi af λ.

Relaxeret problem kan løses ved maksimal strømning.

Grafdeling

Branching regler:

- Dichotomisk forgræning: Tag en knude v som endnu ikke er tildelt V_1 eller V_2
- Tildel v til V_1
- Tildel v til V_2

Parallel løsning af Branch-and-bound

Hvorfor?

- Virksomhed *skal* løse et problem indenfor given tid

Ikke alle sekventielle algoritmer er velegnet til parallelisering

- Dijkstra: korteste veje
- Quicksort: sortering
- Branch-and-bound: løsning af optimeringsproblemer

Ideel model PRAM findes ikke $\rightarrow$ kommunikation er dyr

Parallel løsning af Branch-and-bound

Hvis alt går vel vil p processorer løse opgave p gange hurtigere.

- Tidstab på kommunikation
- En processor har ikke noget at lave
- To processorer laver det samme arbejde
- Overflødigt arbejde laves i forhold til sekventiel algoritme

Speed-up

$$S(p) = \frac{T(1)}{T(p)}$$

Parallel løsning af branch-and-bound

- Initiel fordeling af problemer på processorer: Breadth first search.
- *Load balancing*.
- Udveksling af hidtil bedste løsning.

Erfaringer

Gode råd

- Brug ikke parallel B&B på for nemme problemer
- Brug “løse” grænseværdier med stort søgetræ
- Brug relativt simpel protokol for ”Load balancing”
- Vær kritisk overfor resultater, hvis dårligt speed-up for små problemer.

Opsummering

Branch-and-bound

- Forgreningsregler (opdeling i underproblemer)
- Søgestrategi (bredde først, dybde først, bedste først)
- Initiel løsning
- Grænseværdier

Grænseværdier kan udledes ved

- modifikation af objektfunktion
- sletning af begrænsninger
- lineær relaxering (heltalsvariable gøres til reelle variable)
- Lagrange relaxering

Branch-and-bound kan paralleliseres med succes.