

- I. Grundläggande metoder för algoritmkonstruktion.
- II. NP-fullständighet.
- III. Inexakta metoder.**

Fö. 10: Approximationsalgoritmer.

Fö. 11: Randomiserade algoritmer.

Fö. 12: Approximation + Randomisering.

Fö. 13: Övningsuppgift 3.

Fö. 14. Exempeltentamen.

Fö. 15. Frågestund inför tentamen. (Lämplig tid?)

Tentamen. 2020-03-25, 14-19.

Approximationsalgoritmer.

- 1 Grunder.
- 2 Nodhölje 1.
- 3 Nodhölje 2.
- 4 Handelsresandeproblemet.

Betrakta ett optimeringsproblem X .

Betrakta ett optimeringsproblem X .

Vi säger att X kan *approximeras* inom $c > 1$ om följande gäller.
Det finns en polynomisk algoritm A som givet en instans I av X genererar en lösning S sådan att

$$\frac{OPT(I)}{c} \leq m(S) \leq c \cdot OPT(I).$$

Betrakta ett optimeringsproblem X .

Vi säger att X kan *approximeras* inom $c > 1$ om följande gäller. Det finns en polynomisk algoritm A som givet en instans I av X genererar en lösning S sådan att

$$\frac{OPT(I)}{c} \leq m(S) \leq c \cdot OPT(I).$$

$m(S)$ är *måttet* på lösningen. Exempel: antal noder i Nodhölje och Oberoende Mängd, antal satisfierande klausuler i MaxSAT etc.

Definitionen blir något enklare för "vanliga" minimerings- och maximeringsproblem.

Minimering (ingen lösning kan vara mindre än $OPT(I)$):

$$OPT(I) \leq m(S) \leq c \cdot OPT(I).$$

Definitionen blir något enklare för "vanliga" minimerings- och maximeringsproblem.

Minimering (ingen lösning kan vara mindre än $OPT(I)$):

$$OPT(I) \leq m(S) \leq c \cdot OPT(I).$$

Maximering (ingen lösning kan vara större än $OPT(I)$):

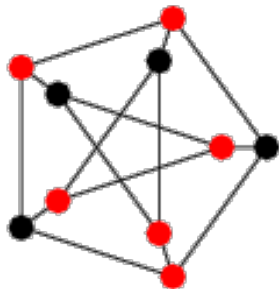
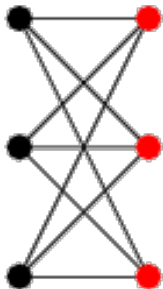
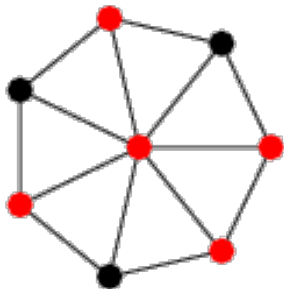
$$\frac{OPT(I)}{c} \leq m(S) \leq OPT(I).$$

Exempel 1: Nodhölje 1

Nodhölje 1

Ett *nodhölje* i en oriktad graf $G = (V, E)$ är en delmängd $V' \subseteq V$ sådan att varje båge i E är ansluten till minst en nod i V' .

Nodhölje 1



Nodhölje 1

Problem. Nodhölje.

Indata. Oriktad graf $G = (V, E)$.

Utdata. Nodhölje $V' \subseteq V$ med minimal storlek.

Nodhölje 1

Problem. Nodhölje.

Indata. Oriktad graf $G = (V, E)$.

Utdata. Nodhölje $V' \subseteq V$ med minimal storlek.

Polynomisk algoritm finns (troligen) inte eftersom problemet är **NP**-fullständigt. (Fö.7)

Nodhölje 1

Problem. Nodhölje.

Indata. Oriktad graf $G = (V, E)$.

Utdata. Nodhölje $V' \subseteq V$ med minimal storlek.

Polynomisk algoritm finns (troligen) inte eftersom problemet är **NP**-fullständigt. (Fö.7)

Polynomisk algoritm existerar för träd (Fö. 2)

Nodhölje 1

Problem. Nodhölje.

Indata. Oriktad graf $G = (V, E)$.

Utdata. Nodhölje $V' \subseteq V$ med minimal storlek.

Polynomisk algoritm finns (troligen) inte eftersom problemet är **NP**-fullständigt. (Fö.7)

Polynomisk algoritm existerar för träd (Fö. 2)

Nodhölje kan approximeras inom 2.

Nodhölje 1

algorithm $N((V, E))$

$C := \emptyset, E' := \emptyset, A := \emptyset$

Nodhölje 1

algorithm $N((V, E))$

$C := \emptyset, E' := \emptyset, A := \emptyset$

while $E' \neq \emptyset$

 välj (u, v) godtyckligt i E'

$A := A \cup \{(u, v)\}$

$C := C \cup \{u, v\}$

 tag bort varje båge i E' som innehåller u och/eller v

returnera C

Nodhölje 1

Observationer.

Algoritmen returnerar alltid en lösning.

Nodhölje 1

Observationer.

Algoritmen returnerar alltid en lösning.

Algoritmen går i polynomisk tid.

Nodhölje 1

Observationer.

Algoritmen returnerar alltid en lösning.

Algoritmen går i polynomisk tid.

Approximerar algoritmen optimum inom 2?

Nodhölje 1

Låt C^* vara ett optimalt nodhölje till G .

Nodhölje 1

Låt C^* vara ett optimalt nodhölje till G .

Inga bågar i A har gemensamma noder så $|C| = 2|A|$.

Nodhölje 1

Låt C^* vara ett optimalt nodhölje till G .

Inga bågar i A har gemensamma noder så $|C| = 2|A|$.

Varje nodhölje, exempelvis C^* , måste inkludera minst en nod från varje båge i A .

Nodhölje 1

Låt C^* vara ett optimalt nodhölje till G .

Inga bågar i A har gemensamma noder så $|C| = 2|A|$.

Varje nodhölje, exempelvis C^* , måste inkludera minst en nod från varje båge i A .

Eftersom två bågar i A aldrig delar några noder gäller $|A| \leq |C^*|$.

Nodhölje 1

Låt C^* vara ett optimalt nodhölje till G .

Inga bågar i A har gemensamma noder så $|C| = 2|A|$.

Varje nodhölje, exempelvis C^* , måste inkludera minst en nod från varje båge i A .

Eftersom två bågar i A aldrig delar några noder gäller $|A| \leq |C^*|$.

$2|A| \leq 2|C^*|$.

Nodhölje 1

Låt C^* vara ett optimalt nodhölje till G .

Inga bågar i A har gemensamma noder så $|C| = 2|A|$.

Varje nodhölje, exempelvis C^* , måste inkludera minst en nod från varje båge i A .

Eftersom två bågar i A aldrig delar några noder gäller $|A| \leq |C^*|$.

$$2|A| \leq 2|C^*|.$$

$$|C| = 2|A| \leq 2|C^*|.$$

Nodhölje 1

Kom ihåg att C^* är en optimal lösning och C är en lösning så
 $|C^*| \leq |C|$.

Nodhölje 1

Kom ihåg att C^* är en optimal lösning och C är en lösning så $|C^*| \leq |C|$.

Slutsats: $|C^*| \leq |C| \leq 2|C^*|$.

Exempel 2: Nodhölje 2

Standardmetodik.

Standardmetodik.

1. Omformulering till "matematiskt" optimeringsproblem.

Standardmetodik.

1. Omformulering till "matematiskt" optimeringsproblem.
2. Relaxation.

Standardmetodik.

1. Omformulering till "matematiskt" optimeringsproblem.
2. Relaxation.
3. Lös relaxerat problem.

Standardmetodik.

1. Omformulering till "matematiskt" optimeringsproblem.
2. Relaxation.
3. Lös relaxerat problem.
4. "Avrunda" lösning till relaxerat problem.

1. Omformulering.

Nodhölje 2

1. Omformulering.

Låt $G = (V, E)$ vara en instans av Nodhölje med $V = \{v_1, \dots, v_n\}$.

Nodhölje 2

1. Omformulering.

Låt $G = (V, E)$ vara en instans av Nodhölje med $V = \{v_1, \dots, v_n\}$.

$$\min v_1 + \dots + v_n$$

Nodhölje 2

1. Omformulering.

Låt $G = (V, E)$ vara en instans av Nodhölje med $V = \{v_1, \dots, v_n\}$.

$$\min v_1 + \dots + v_n$$

$$v_i + v_j \geq 1 \text{ för alla } (v_i, v_j) \in E$$

Nodhölje 2

1. Omformulering.

Låt $G = (V, E)$ vara en instans av Nodhölje med $V = \{v_1, \dots, v_n\}$.

$$\min v_1 + \dots + v_n$$

$$v_i + v_j \geq 1 \text{ för alla } (v_i, v_j) \in E$$

$$v_i \in \{0, 1\}, 1 \leq i \leq n$$

Nodhölje 2

1. Omformulering.

Låt $G = (V, E)$ vara en instans av Nodhölje med $V = \{v_1, \dots, v_n\}$.

$$\min v_1 + \dots + v_n$$

$$v_i + v_j \geq 1 \text{ för alla } (v_i, v_j) \in E$$

$$v_i \in \{0, 1\}, 1 \leq i \leq n$$

Optimum för detta optimeringsproblem är storleken på det minsta nodhöljet i G .

Nodhölje 2

Hitta lösning till denna typ av optimeringsproblem är (minst) lika svårt som Nodhölje (dvs. **NP**-svårt).

Nodhölje 2

Hitta lösning till denna typ av optimeringsproblem är (minst) lika svårt som Nodhölje (dvs. **NP**-svårt).

Hur förenkla?

2. Relaxation.

2. Relaxation.

$$\min v_1 + \cdots + v_n$$

$$v_i + v_j \geq 1 \text{ för alla } (v_i, v_j) \in E$$

$$\boxed{v_i \in \{0, 1\}}, 1 \leq i \leq n$$

2. Relaxation.

$$\min v_1 + \cdots + v_n$$

$$v_i + v_j \geq 1 \text{ för alla } (v_i, v_j) \in E$$

$$v_i \in \{0, 1\}, 1 \leq i \leq n$$

$\Downarrow$

$$0 \leq v_i \leq 1, 1 \leq i \leq n$$

3. Lösning.

3. Lösning.

Alla villkor är nu linjära. Instansen kan lösas i polynomisk tid med (exempelvis) Karmarkars algoritm.

Varning: Simplexalgoritmen är *inte* polynomisk.

4. Avrundning.

4. Avrundning.

Antag att $(v'_1, \dots, v'_n)$ är en optimal lösning till det relaxerade problemet.

4. Avrundning.

Antag att $(v'_1, \dots, v'_n)$ är en optimal lösning till det relaxerade problemet.

Kom ihåg att $0 \leq v'_i \leq 1$ gäller.

4. Avrundning.

Antag att $(v'_1, \dots, v'_n)$ är en optimal lösning till det relaxerade problemet.

Kom ihåg att $0 \leq v'_i \leq 1$ gäller.

Sätt $v_i^* = 1$ om $v'_i \geq \frac{1}{2}$ och låt $v_i^* = 0$ annars.

Nodhölje 2

$$\sum_{i=1}^n v_i^* \leq \sum_{i=1}^n 2v_i' = 2 \sum_{i=1}^n v_i' \leq 2 \cdot OPT$$

Nodhölje 2

$$\sum_{i=1}^n v_i^* \leq \sum_{i=1}^n 2v'_i = 2 \sum_{i=1}^n v'_i \leq 2 \cdot OPT$$

Notera först att $\sum_{i=1}^n v'_i$ är det optimala värdet på det relaxerade problemet.

Nodhölje 2

$$\sum_{i=1}^n v_i^* \leq \sum_{i=1}^n 2v'_i = 2 \sum_{i=1}^n v'_i \leq 2 \cdot OPT$$

Notera först att $\sum_{i=1}^n v'_i$ är det optimala värdet på det relaxerade problemet.

Första olikheten gäller därför att en variabel maximalt kan bli dubbelt så stor av avrundningen.

Nodhölje 2

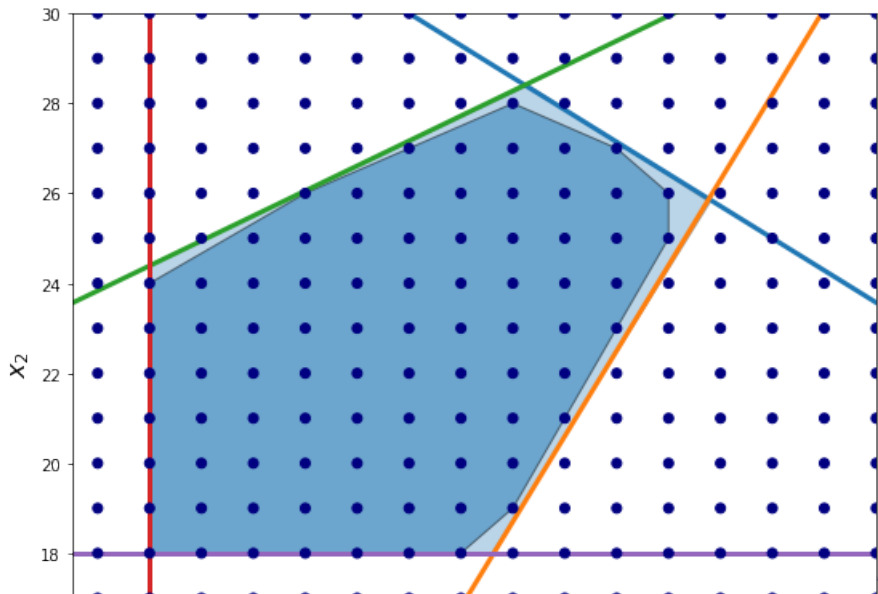
$$\sum_{i=1}^n v_i^* \leq \sum_{i=1}^n 2v'_i = 2 \sum_{i=1}^n v'_i \leq 2 \cdot OPT$$

Notera först att $\sum_{i=1}^n v'_i$ är det optimala värdet på det relaxerade problemet.

Första olikheten gäller därför att en variabel maximalt kan bli dubbelt så stor av avrundningen.

Sista olikheten gäller på grund av relaxation: Vi har fler tänkbara punkter med villkoret $0 \leq v_i \leq 1$ än med $v_i \in \{0, 1\}$. Eftersom vi har ett minimeringsproblem så kan optimum bli lägre i det relaxerade problemet än i det ursprungliga problemet.

Nodhölje 2



Nodhölje 2

Vi har visat att algoritmen approximerar Nodhölje inom 2.

Minimering vs. maximering.

Vi vet att Nodhölje kan approximeras inom 2.

Minimering vs. maximering.

Vi vet att Nodhölje kan approximeras inom 2.

Vi vet att om $G = (V, E)$ har ett nodhölje av storlek k så innehåller G en oberoende mängd av storlek $|V| - k$.

Minimering vs. maximering.

Vi vet att Nodhölje kan approximeras inom 2.

Vi vet att om $G = (V, E)$ har ett nodhölje av storlek k så innehåller G en oberoende mängd av storlek $|V| - k$.

Ger detta att vi kan approximera Oberoende Mängd på ett bra sätt?

Minimering vs. maximering.

Vi vet att Nodhölje kan approximeras inom 2.

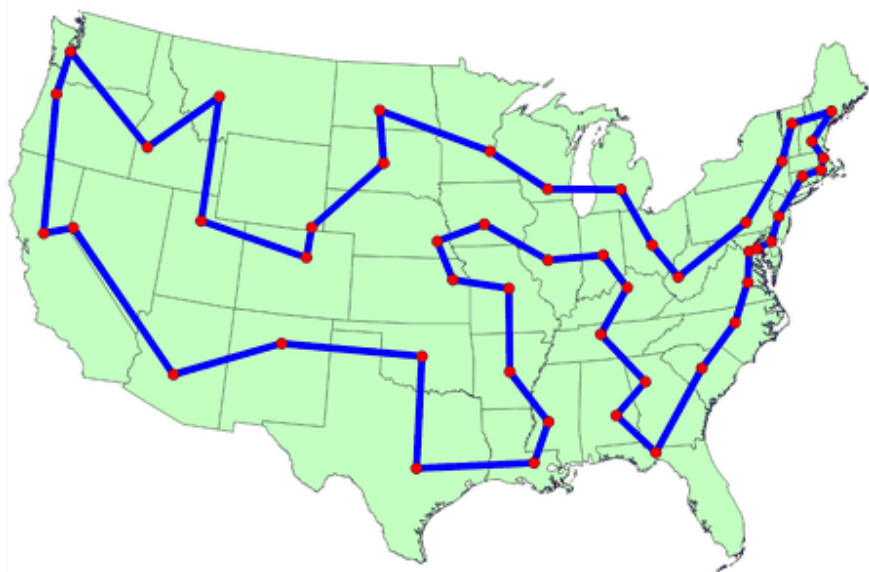
Vi vet att om $G = (V, E)$ har ett nodhölje av storlek k så innehåller G en oberoende mängd av storlek $|V| - k$.

Ger detta att vi kan approximera Oberoende Mängd på ett bra sätt?

Nej. Oberoende Mängd kan inte approximeras inom $|V|^{1-\epsilon}$ för något $\epsilon > 0$ såvida inte **P** = **NP**.

Exempel 3: Handelsresandeproblemet

Handelsresandeproblemet



Handelsresandeproblemet

Problem. Handelsresandeproblemet (HRP)

Indata. En $n \times n$ symmetrisk avståndsmatrix.

Utdata. En handelsresandeväg av minimal längd.

Handelsresandeproblemet

Triangelolikhet: $\text{dist}(A \rightarrow B) \leq \text{dist}(A \rightarrow C \rightarrow B)$.

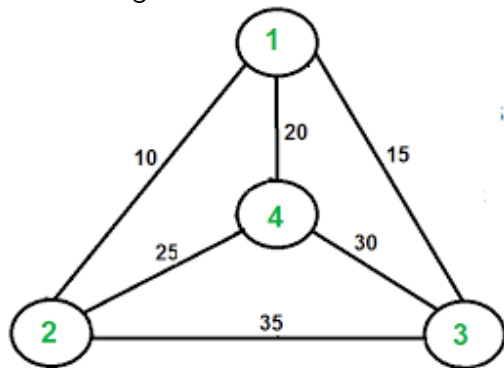
Handelsresandeproblemet

Triangelolikhet: $\text{dist}(A \rightarrow B) \leq \text{dist}(A \rightarrow C \rightarrow B)$.

Δ HRP är HRP begränsat till avståndsmatriser som uppfyller Δ -olikheten.

Handelsresandeproblemet

Notera att vi kan betrakta avståndsmatrisen som en graf med viktade bågar.



Handelsresandeproblemet

algorithm $H(G)$ där $G = (V, E, w)$

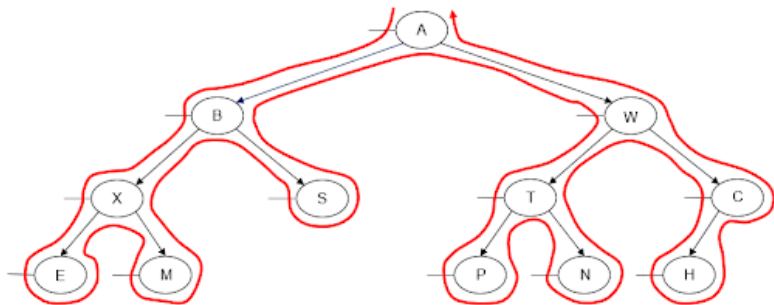
Hitta ett minsta uppspännande träd T till G .

Lista noderna i T i preorder utgående från en godtycklig nod v .

Lägg till v sist i listan.

Returnera listan.

Handelsresandeproblem



$(A, B, X, E, M, S, W, T, P, N, C, H, A)$

Handelsresandeproblemet

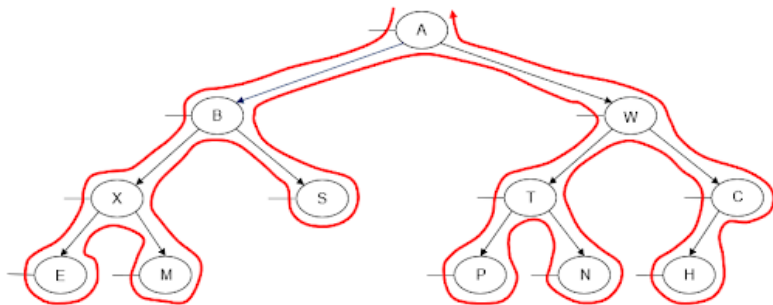
T är det uppspännande trädet.

Handelsresandeproblemet

T är det uppspännande trädet.

S_0 : uppräkning av noder som "följer det uppspännande trädet".
Varje båge passeras två gånger.

Handelsresandeproblem



(A, B, X, E, X, M, X, B, S, B, A, W, T, P, T, N, T, W, G, H, G, W, A)

Handelsresandeproblemet

T är det uppspännande trädet.

S_0 : uppräknig av noder som "följer det uppspännande trädet".
Varje båge passeras två gånger.

(*) Notera att $w(S_0) = 2w(T)$.

Handelsresandeproblemet

T är det uppspännande trädet.

S_0 : uppräkning av noder som "följer det uppspännande trädet".
Varje båge passeras två gånger.

(*) Notera att $w(S_0) = 2w(T)$.

S : "filtrera" S_0 genom att bara ta med första förekomsten av varje nod i S_0 (utom startnoden).

Handelsresandeproblemet

T är det uppspännande trädet.

S_0 : uppräkning av noder som "följer det uppspännande trädet".
Varje båge passeras två gånger.

(*) Notera att $w(S_0) = 2w(T)$.

S : "filtrera" S_0 genom att bara ta med första förekomsten av varje nod i S_0 (utom startnoden).

(**) Notera att $w(S) \leq w(S_0)$ pga. $\triangle$ -olikheten.

Handelsresandeproblemet

T är det uppspännande trädet.

S_0 : uppräkning av noder som "följer det uppspännande trädet".
Varje båge passeras två gånger.

(*) Notera att $w(S_0) = 2w(T)$.

S : "filtrera" S_0 genom att bara ta med första förekomsten av varje nod i S_0 (utom startnoden).

(**) Notera att $w(S) \leq w(S_0)$ pga. $\triangle$ -olikheten.

Notera att S är svaret från algoritmen.

Handelsresandeproblemet

$$w(S) \leq^{**} w(S_0) =^* 2w(T) \leq 2 \cdot OPT$$

Handelsresandeproblemet

$$w(S) \leq^{**} w(S_0) =^* 2w(T) \leq 2 \cdot OPT$$

Varför gäller $w(T) \leq OPT$?

Handelsresandeproblemet

$$w(S) \leq^{**} w(S_0) =^* 2w(T) \leq 2 \cdot OPT$$

Varför gäller $w(T) \leq OPT$?

Vad får man om man tar bort en båge ur en handelsresandeväg?

Handelsresandeproblemet

$$w(S) \leq^{**} w(S_0) =^* 2w(T) \leq 2 \cdot OPT$$

Varför gäller $w(T) \leq OPT$?

Vad får man om man tar bort en båge ur en handelsresandeväg?

Ett uppspännande träd.

Handelsresandeproblemet

$$w(S) \leq^{**} w(S_0) =^* 2w(T) \leq 2 \cdot OPT$$

Varför gäller $w(T) \leq OPT$?

Vad får man om man tar bort en båge ur en handelsresandeväg?

Ett uppspännande träd.

Alltså är $w(T) \leq OPT$ eftersom T är ett minsta uppspännande träd.

Handelsresandeproblemet

$$w(S) \leq^{**} w(S_0) =^* 2w(T) \leq 2 \cdot OPT$$

Varför gäller $w(T) \leq OPT$?

Vad får man om man tar bort en båge ur en handelsresandeväg?

Ett uppspännande träd.

Alltså är $w(T) \leq OPT$ eftersom T är ett minsta uppspännande träd.

Slutsats: Algoritmen approximerar $\triangle HRP$ inom 2.

Standardmetodik.

1. Omformulering till "matematiskt" optimeringsproblem.
2. Relaxation.
3. Lös relaxerat problem.
4. "Avrunda" lösning till relaxerat problem.

Handelsresandeproblemet

algorithm $H(G)$ där $G = (V, E, w)$

Hitta ett minsta uppspännande träd T till G .

Lista noderna i T i preorder utgående från en godtycklig nod v .

Lägg till v sist i listan.

Returnera listan.