Note Title

College 11: Complexiteit

Optimaliseringsalgoritmen → snelle "constructieve" algoritmen (bv kortste paden, opspannende bomen, sommige inwendige punt alg. voor LP)

→ alg. die langzaam zijn "in the worst case" (bv Branch-and-bound, dynamisch programmering)

Benaderingsalgoritmen: snel, maar garanderen niet de optimale oplossing

Sommige problemen: optimaal en snel
Andere —"— : optimaal en traag
of
benadering en snel

} onvermijdelijk ?

↑ complexiteit

Probleem is <u>"well solved" (gemakkelijk)</u> als er een polynomiale algoritme voor bestaat.

<u>Optimaliseringsprobleem:</u> Gegeven een instantie, bepaal de oplossing met minimale (maximale) waarde

<u>Beslissingsprobleem:</u> Gegeven instantie en getal z, is er een oplossing met waarde $\leq z$ (of $\geq z$)

Optimaliseringsprobleem polynomiaal oplosbaar
$\Rightarrow$ beslissingsprobleem polyn. oplosbaar

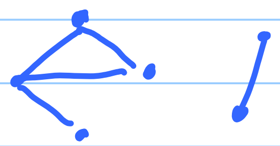
Als het beslissingsprobleem polyn. oplosbaar dan kunnen we ook de optimale doelfunctie-waarde bepalen in polyn. tijd als $\log(\bar{z})$ polyn begrensd is in de grootte v/d input. Hoe?
Pas binair zoeken toe over de waarde van z.

Voorbeeld: Stel we weten dat $z^* \in [0, \bar{z}]$, $\bar{z}$ is polyn. begr. in de grootte v/d input,
bv.: $z^* \in [0, 32]$ $(z^* = 11)$

Bestaat er een opl. met $z > 16$? Nee $\Rightarrow$ $z \in [0, 16]$
" " " " " $z > 8$? Ja $\Rightarrow$ $z \in [9, 16]$
" " " " " $z > 12$? Nee $\Rightarrow$ $z \in [9, 12]$
" " " " " $z > 10$? Ja $\Rightarrow$ $z \in [11, 12]$
" " " " " $z > 11$? Nee $\Rightarrow$ $z^* = 11$

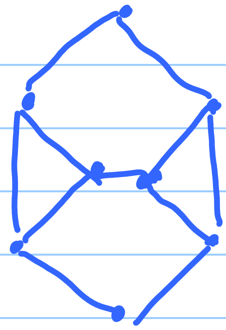
Interval halveert in elke stap $\Rightarrow$ we hebben $\log_2(\bar{z})$ stappen nodig.

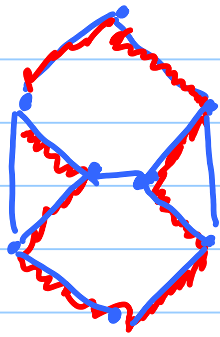
Probleem: Is de graaf G samenhangend?
Ja of Nee?



Snelle algoritme: bepaal kortste pad van en punt naar alle andere punten

Probleem: Is de graaf G Hamiltoniaans?
(Bestaat er een TSP-tour in G?)



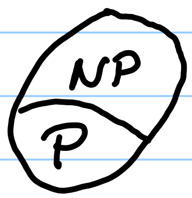


Ja of Nee ?

Instantie + certificaat (de TSP tour zelf) $\xrightarrow[\text{snel}]{}$ Ja klopt

P = verzameling beslissingsproblemen die in polynomiale tijd oplosbaar zijn

NP = verz. beslissingspr. die in polyn. tijd verifieerbaar zijn
= " " Π: $\forall$ ja-instantie I, $\exists$ certificaat $c(I)$ en polynomiale alg. om geldigheid van $c(I)$ na te gaan



$P \overset{?}{=} NP$

P = NP is heel onwaarschijnlijk want P bevat alle makkelijke problemen en NP bevat veel moeilijke problemen die wij niet "begrijpen" en die wij (nog) niet op kunnen lossen.

Π_1 reduceert naar Π_2 ($\Pi_1 \propto \Pi_2$) als $\exists$ functie $\varphi : \Pi_1 \to \Pi_2$, polynoom p: $\forall$ instantie I van Π_1 :

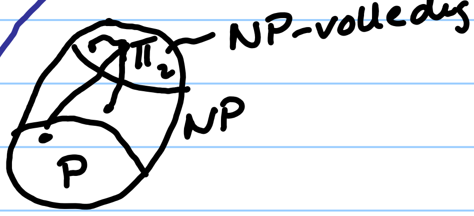
(i) I ja-instantie van Π_1 $\Leftrightarrow$ $\varphi(I)$ is ja-instantie van Π_2

(2) De grootte van $\varphi(I) \leq p$(grootte van I)

(Π_1 is "speciaal geval" van Π_2)

<u>Def</u>: Π_2 is <u>NP-volledig</u> als (a) $\Pi_2 \in NP$

(b) $\forall \Pi_1 \in NP : \Pi_1 \propto \Pi_2$



Stel $\Pi_1, \Pi_2 \in NP$ $\Pi_1 \propto \Pi_2$

(i) Als $\Pi_2 \in P$ dan ook $\Pi_1 \in P$

(ii) Als Π_1 NP-volledig dan ook Π_2 NP-volledig

(iii) Als Π_2 NP-volledig $\Rightarrow$ Je weet niets over Π_1

NP-volledig: waarschijnlijk niet polynomiaal.
Als Π_2 NP-voll. is, dan is het onwaarschijnlijk dat $\Pi_2 \in P$ want dan zou gelden $\forall \Pi_1 \in NP : \Pi_1 \propto \Pi_2 \in P \Rightarrow$ ook $\Pi_1 \in P$, dus geldt $P = NP$ en dat is onwaarschijnlijk.

Optimaliseringsprobleem heet <u>NP-lastig</u> als bijbehorende beslissingsprobleem NP-volledig is.

Hoe bewijs je dat een probleem NP-volledig is?

(a) vaak duidelijk

(b) kan moeilijk zijn
- maar alleen de eerste keer (Cook's stelling (1971))
- daarna "makkelijk" vanwege transitiviteit

SAT (SATISFIABILITY): Gegeven Boolse uitdrukking in "Conjunctive Normal Form" (CNF), is er een waardetoewijzing aan de variabelen waardoor de uitdrukking waar is?

Voorbeeld: x_1, x_2, x_3, x_4 Boolse variabelen

$\bar{x}_j$ = "not x_j"

Uitdrukking in CNF:

$$\underbrace{(x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4) \wedge \underbrace{(\bar{x}_3 \vee x_4)}_{\text{disjunctie}} \wedge (x_3 \vee x_1) \wedge (x_2)}_{\text{conjunctie van disjuncties}}$$

"or" "and"

Cook's stelling (1971):

* SAT $\in$ NP
* $\forall\, \Pi \in$ NP : $\Pi \propto$ SAT

Reducties:

IP (integer programming) is NP-volledig

IP: Gegeven is een geheeltallig $m \times n$ matrix A en een geheeltallig m-vector b. Bestaat er een geheelt. n-vector $x \geq 0$ zdd $Ax = b$?

Hulpstelling: Als $X = \{x \geq 0, x \in \mathbb{Z}^n : Ax = b\} \neq \emptyset$ dan bestaat er een vector $x \in X$ zdd de lengte van de (binaire) codering van x polyn. begr. is in de lengte v/d input (zie "Example 15.8").

IP ∈ NP: Hulpstelling impliceert IP ∈ NP.

SAT ∝ IP: Gegeven een instantie van SAT, construeer de volgende instantie van IP:

Laat $y_j = \begin{cases} 1 & \text{als } x_j \text{ "waar" is} \\ 0 & \text{anders} \end{cases}$

Voor elke "clause" C_i eisen we dat er tenminste een variabele is met waarde 1

$$\Rightarrow \quad \forall \text{ Clause } C_i : \sum_{\{j : x_j \in C_i\}} y_j + \sum_{\{j : \bar{x}_j \in C_i\}} (1 - y_j) \geq 1$$

Voeg daarna excessvariabelen toe en eis dat deze variabelen gelijk zijn aan nul.

Deze transformatie is polynomiaal.

Ons voorbeeld:

$(x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4) \wedge (\bar{x}_3 \vee x_4) \wedge (x_3 \vee x_1) \wedge (x_2)$

$$\begin{aligned}
y_1 + y_2 + (1-y_3) \quad & \geq 1 \\
(1-y_1) + (1-y_2) + y_4 \quad & \geq 1 \\
(1-y_3) + y_4 \quad & \geq 1 \\
y_1 + y_3 \quad & \geq 1 \\
y_2 \quad & \geq 1
\end{aligned}$$

Als SAT ja-instantie : $\exists$ tenminste een "term" (y_j of $(1-y_j)$) met waarde 1 in elke voorwaarde van IP

Als IP ja : Elke clause is waar

CLIQUE is NP-volledig :

CLIQUE : Gegeven een ongerichte graaf $G=(V,E)$ en een geheel getal k, heeft G een clique (volledige deelgraaf) van grootte $\geq k$?

SAT $\propto$ CLIQUE : Gegeven een instantie van SAT, construeer de volgende instantie van CLIQUE.

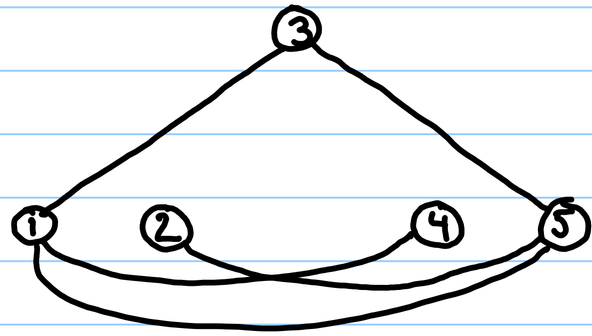
$V = \{(x,i) : x \text{ is een variabele in clause } C_i\}$

$E = \{\{(x,i),(y,j)\} : x \neq \bar{y},\ i \neq j\}$

$k = s$ (# of clauses)

Voorbeeld: $(x_1) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (x_3)$

Vertices
1: $(x_1, 1)$
2: $(\bar{x}_1, 2)$
3: $(x_2, 2)$
4: $(\bar{x}_3, 2)$
5: $(x_3, 3)$



Edges: $\{1,3\}$, $\{1,4\}$, $\{1,5\}$
$\{2,5\}$
$\{3,5\}$

Als SAT Ja-inst. ⇒ tenminste een var. in elke clause is TRUE. Neem een variabele uit elke clause die true is. Die var. zijn in G aan elkaar verbonden en omdat $k=s$ is er een clique van grootte k.

Als CLIQUE Ja-inst. ∃ clique van grootte k. De punten "zitten" in verschillende clauses door constructie en kunnen onafhankelijk van elkaar waarde TRUE krijgen.

INDEPENDENT SET is NP-volledig:

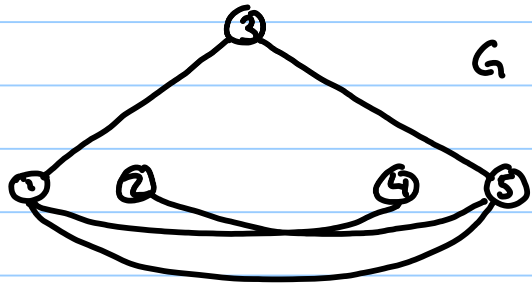
INDEPENDENT SET: Gegeven een ongerichte graaf $G' = (V', E')$ en een geheel getal k', heeft G' een independent set van grootte k'?

verz. punten in G' die niet aan elkaar grenzen

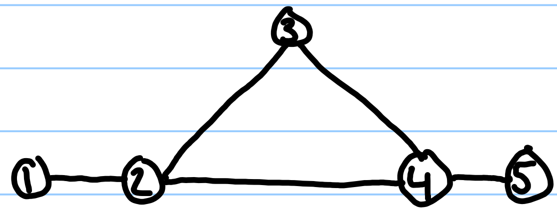
CLIQUE ≤ INDEPENDENT SET: Gegeven een instantie van CLIQUE, construeer de volgende instantie van INDEPENDENT SET

$$\left.\begin{array}{l} V' = V \\ E' = \{ \{i,j\} : i \neq j,\ \{i,j\} \notin E \} \end{array}\right\} \text{"complementary graph" van } G$$

Voorbeeld: CLIQUE



INDEP. SET



Stelling Stel, G' is "complementary graph" van G. Een verzameling punten W in G' is "independent" $\Leftrightarrow$ W een clique in G.