



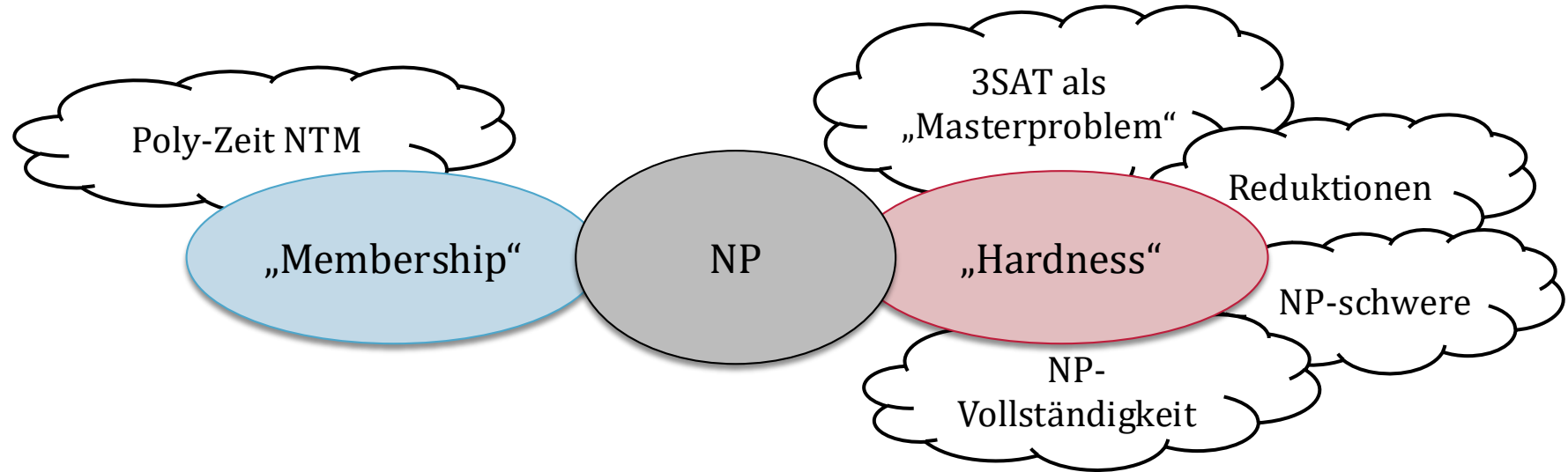
Technische
Universität
Braunschweig



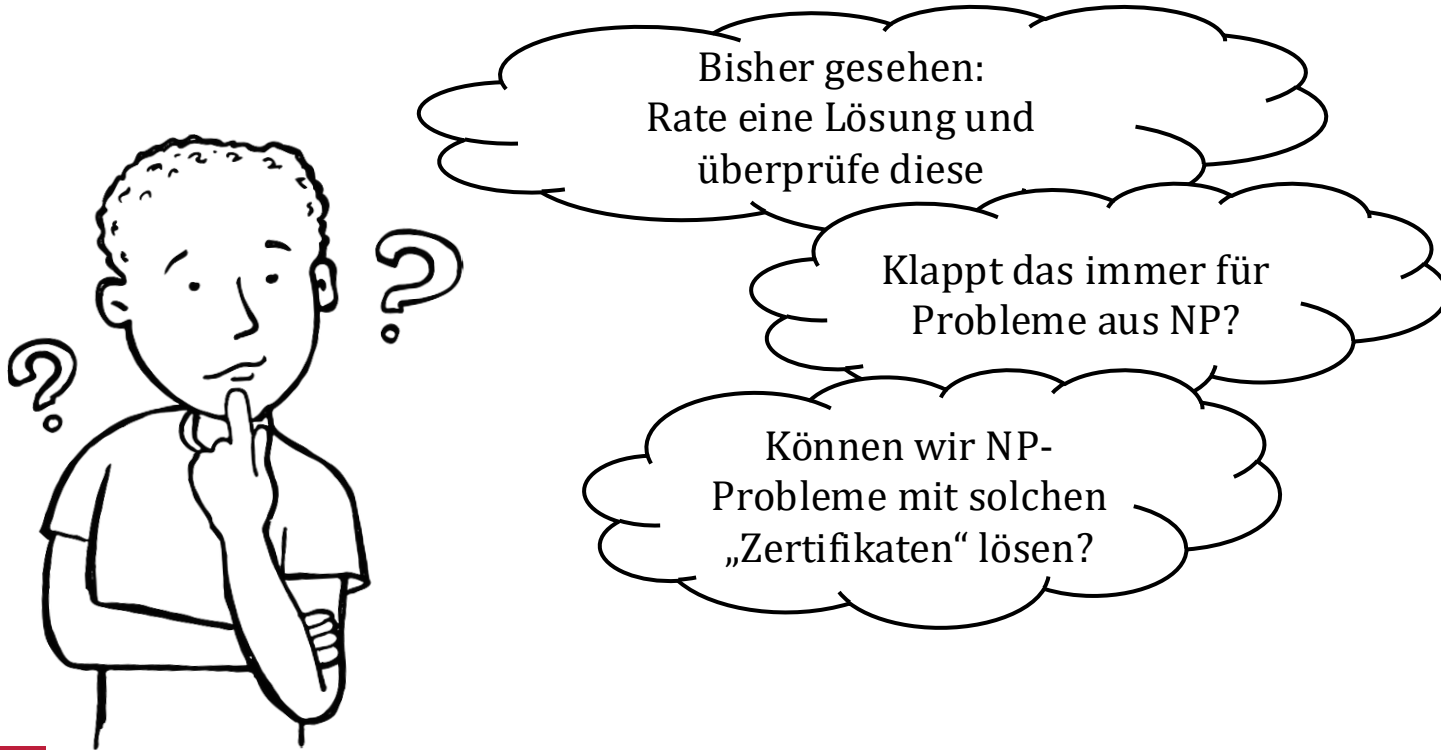
Theoretische Informatik 2

Arne Schmidt

Letztes Mal



Idee NTM-Konstruktion



Kapitel 11.3 – Zertifikate

Zertifikate

Definition 11.17

Ein **Verifizierer** ist eine deterministische, totale TM $\mathcal{V}$ mit zwei Eingabebändern: einem über Σ und einem Zertifikatsband über $\{0,1\}$.

$x \in \Sigma^*$ ist in $\mathcal{L}(\mathcal{V})$, wenn es ein **Zertifikat** $y \in \{0,1\}^*$ gibt, sodass $\mathcal{V}$ die Eingabe (x, y) akzeptiert.

$\mathcal{V}$ ist ein **Polynomialzeit-Verifizierer**, wenn dieser auf (x, y) in $O(|x| + |y|)^k$ Schritten hält und $|y| \leq |x|^c$ mit $k, c \in \mathbb{N}$.

Theorem 11.18

Eine Sprache $\mathcal{L}$ ist genau dann in NP, wenn es einen Polynomialzeit-Verifizierer $\mathcal{V}$ mit $\mathcal{L}(\mathcal{V}) = \mathcal{L}$ gibt.

Beweisskizze

Theorem 11.18

Eine Sprache $\mathcal{L}$ ist genau dann in NP, wenn es einen Polynomialzeit-Verifizierer $\mathcal{V}$ mit $\mathcal{L}(\mathcal{V}) = \mathcal{L}$ gibt.

„ $\Rightarrow$ “:

Für eine gegebene NTM M , die $\mathcal{L}$ in Zeit $O(n^k)$ entscheidet, konstruiere einen Polynomialzeit-Verifizierer $\mathcal{V}$.

Dieser erwartet ein Zertifikat der Länge n^k .

$\mathcal{V}$ verhält sich wie M , eliminiert über das Zertifikat aber den Nicht-Determinismus.

„ $\Leftarrow$ “:

Für einen gegebenen Polynomialzeit-Verifizierer $\mathcal{V}$, dessen Zertifikate Länge $\leq n^k$ besitzen, konstruiere eine NTM M .

M rät ein binäres Zertifikat und verhält sich dann wie $\mathcal{V}$.

Beispiel 11.19: SAT

$$(x_1 \vee x_2 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3) \wedge (x_2 \vee x_3 \vee \bar{x}_4)$$

Wie sieht ein Verifizierer aus?

- Zertifikate haben Länge n , bspw. 0110
- Jede Variable x_i hat damit den Wert an Stelle i des Zertifikats.
- Überprüfe in jeder Klausel, ob ein erfüllendes Literal existiert.

Beispiel 11.20: Independent Set

Problem Independent Set

Gegeben: Graph $G = (V, E)$ und Zahl $k \in \mathbb{N}$

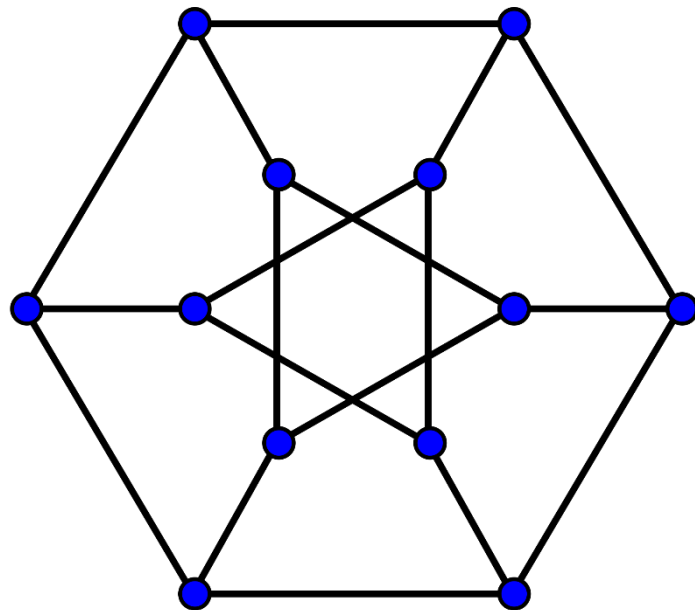
Frage: Existiert $I \subseteq V$ mit $|I| = k$ und für jede Kante $\{u, v\} \in E$ gilt $u \notin I$ oder $v \notin I$?

Hat der Graph ein Independent Set der Größe 4?

Hat der Graph ein Independent Set der Größe 5?

Wie sieht ein Zertifikat aus?

Wie überprüfe ich dies?



Beispiel 11.21: Subset Sum

Problem Subset Sum

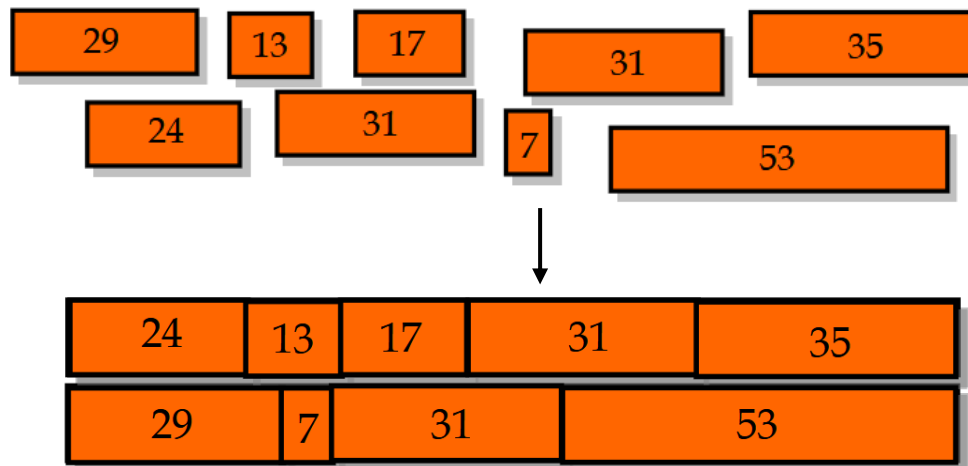
Gegeben: Menge von Zahlen $N = \{n_1, \dots, n_k\} \subset \mathbb{N}$ und eine Zahl $S \in \mathbb{N}$

Frage: Existiert $N' \subseteq N$, sodass $\sum_{n \in N'} n = S$?

Ist diese Instanz mit $S = 120$ lösbar?

Wie sieht ein Zertifikat aus?

Wie überprüfe ich dies?



Beispiel 11.22: Factorization

Problem Factorization

Gegeben: Zwei Zahlen $n, b \in \mathbb{N}$

Frage: Gibt es eine Zahl m mit $1 < m \leq b$, die n teilt?

Ist diese Instanz eine Ja-Instanz?

$$n = 15239, b = 10$$

Wie sieht ein Zertifikat aus?

Wie überprüfe ich dies?

Zertifikat ist eine Zahl m mit $1 < m \leq b$.

Die Berechnung $n \bmod m$ kann in Polyzeit berechnet werden.

Beispiel 11.22: Factorization

Problem Factorization

Gegeben: Zwei Zahlen $n, b \in \mathbb{N}$

Frage: Gibt es keine Zahl m mit $1 < m \leq b$, die n teilt?

Ist diese Instanz eine Ja-Instanz?

$$n = 15349, b = 124$$

Wie sieht ein Zertifikat aus?

Wie überprüfe ich dies?

Das wirkt deutlich schwieriger!

Beispiel 11.22: Factorization

Problem Factorization

Gegeben: Zwei Zahlen $n, b \in \mathbb{N}$

Frage: Gibt es keine Zahl m mit $1 < m \leq b$, die n teilt?

Zunächst:

Jede Zahl $n \in \mathbb{N}$ hat eine eindeutige Primfaktorzerlegung

$$p_1^{k_1} \cdot \dots \cdot p_\ell^{k_\ell} = \prod_{i=1}^{\ell} p_i^{k_i}$$

Man kann zeigen:

- Für jede Primfaktorzerlegung kann ein polynomiell großes Zertifikat definiert werden.
- Die Überprüfung des Zertifikats kann in polynomieller Zeit geschehen:
 - Prüfe jeden Faktor, ob dieser eine Primzahl ist
 - Prüfe, ob $\prod_{i=1}^{\ell} p_i^{k_i} = n$
 - Prüfe, ob der kleinste Faktor $> b$ ist.

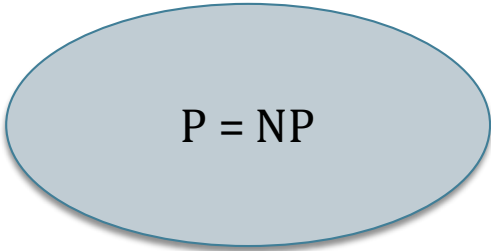


Factorization liegt damit
in $NP \cap coNP$

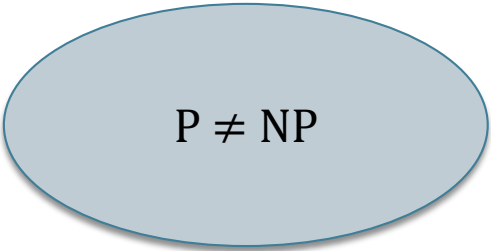
Man vermutet

- $NP \neq coNP$
 - andernfalls würde die polynomielle Hierarchie zusammenfallen
 - Damit: Factorization ist nicht (co-)NP-schwer
- Factorization $\notin P$
 - Factorization $\in NP \setminus P$

P vs NP


$$P = NP$$

Viele Optimierungsprobleme in der Praxis
ließen sich „effizient“ lösen!
(Aber mit welcher Laufzeit?!)

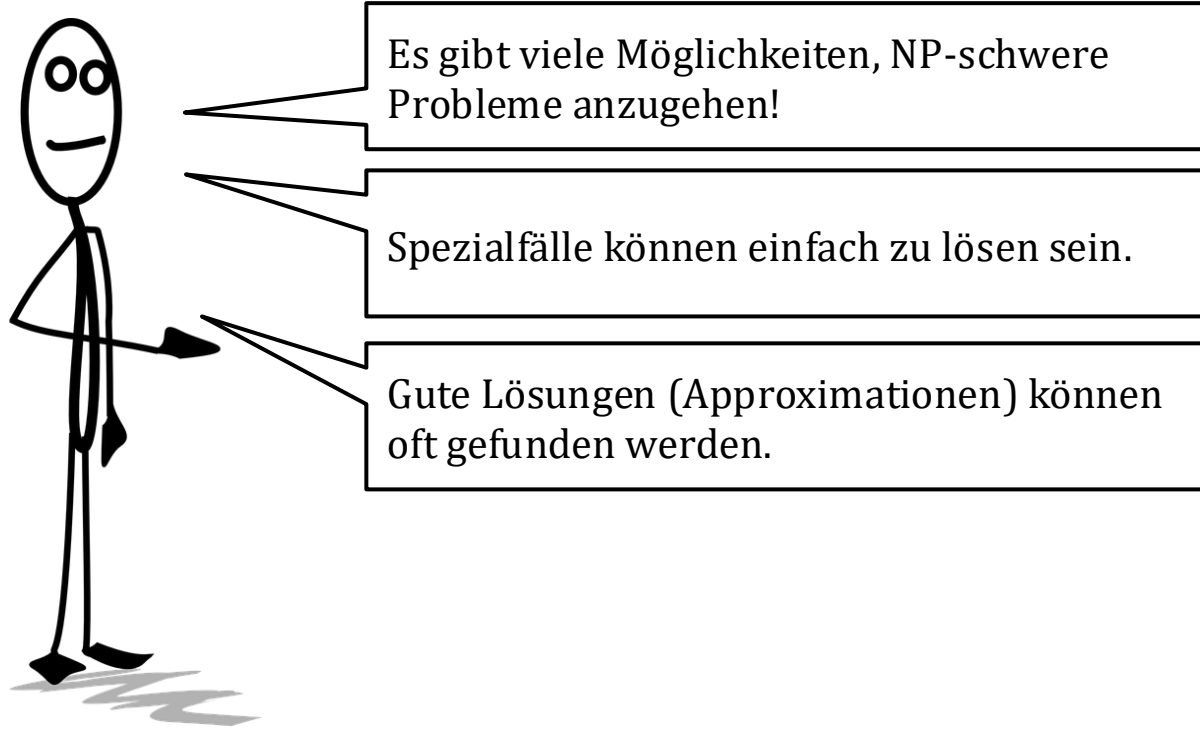

$$P \neq NP$$

Verschlüsselungsverfahren bleiben sicher.
Sie basieren oft auf Primfaktorzerlegungen.

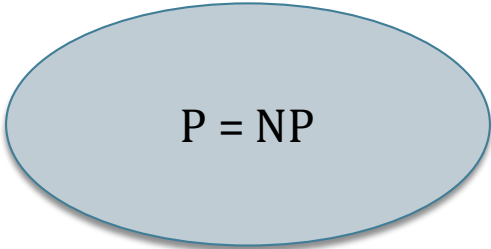
Wichtig:

$P \neq NP$ sagt nur aus, dass nicht alle Instanzen eines Problems effizient gelöst werden können!

Was tun bei NP-schweren Optimierungsproblemen



Wie zeigt man $P \neq NP$?

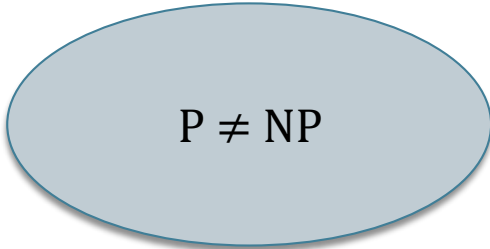

$$P = NP$$

Konstruktiv:

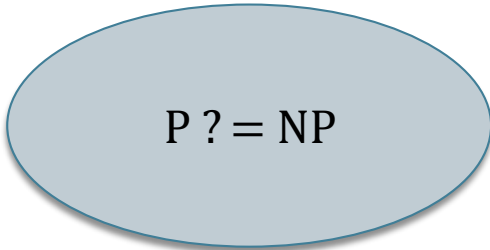
Angabe eines poly-Zeit-Algorithmus für ein NP-vollständiges Problem.

Nicht-konstruktiv:

Formaler Beweis, dass man die Probleme effizient lösen kann, man kennt aber den Algorithmus nicht.


$$P \neq NP$$

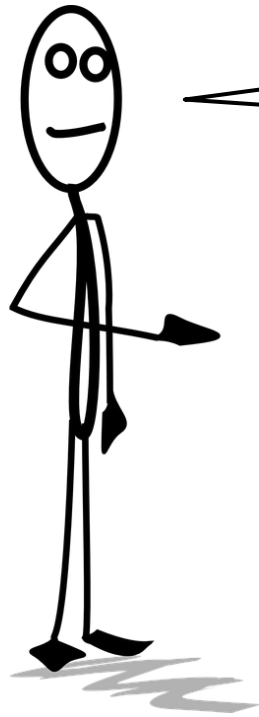
Formaler Beweis


$$P ? = NP$$

Nicht lösbar.

Zeige logische Unabhängigkeit.
(Beispiel: Kontinuumshypothese)

Was glaubt man?



Man geht davon aus, dass $P \neq NP$.

Als kleines Indiz zur Überlegung:
Es gibt tausende Probleme in P und tausende NP-schwere Probleme.

Für keiner dieser über 10^6 Kombinationen wurde eine Reduktion gefunden!

Kapitel 11.4 – coNP-Vollständige Probleme

Liste von coNP-Vollständigen Problemen

Problem UNSAT

Gegeben: Boolesche Formel F in CNF

Frage: Existiert keine F erfüllende Belegung?

$$(x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_1) \wedge (\neg x_1 \vee x_2)$$

Problem TAUTOLOGY

Gegeben: Boolesche Formel F in DNF

Frage: Erfüllt jede Variablenbelegung F ?

$$(\neg x_1 \wedge \neg x_2) \vee (x_1 \wedge x_2) \vee (\neg x_1) \vee (x_1 \wedge \neg x_2)$$

Exkurs – P, NP und Geometrie

Euklidisches TSP - Definition

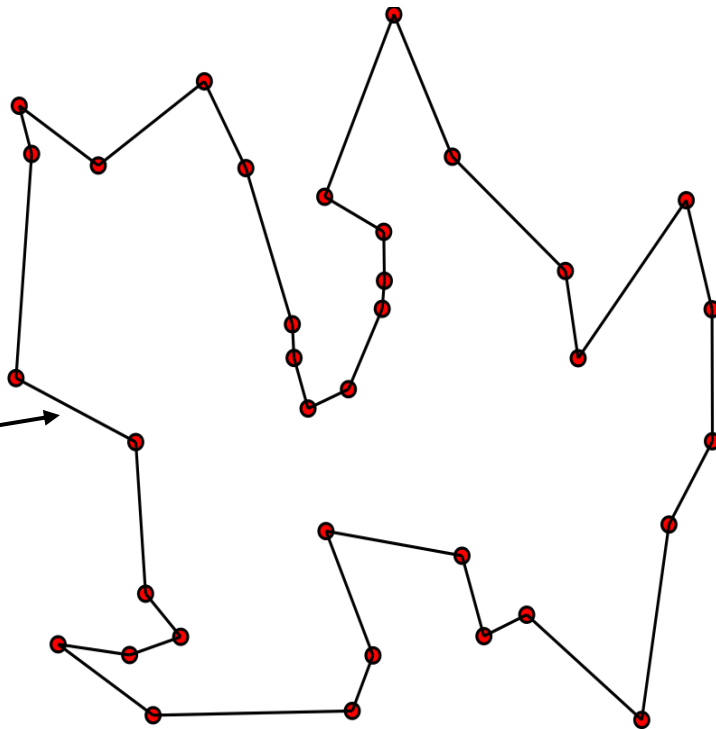
Gegeben: Punkte $p_1, \dots, p_n \in \mathbb{Z}^2$

Gesucht: Eine Permutation π , sodass

$$\sum_{i=1}^n \|p_{\pi(i)} p_{\pi(i+1)}\|$$

mit $\pi(n+1) = \pi(1)$ minimal ist.

Länge der Kante ist die
euklidische Distanz



THE EUCLIDEAN TRAVELING SALESMAN PROBLEM IS NP-COMPLETE*

Christos H. PAPADIMITRIOU

*Center for Research in Computing Technology, Harvard University, Cambridge, MA 02138,
U.S.A.*

Communicated by Richard Karp

Received August 1975

Revised July 1976

*: Das Entscheidungsproblem (gibt es eine Tour der Länge höchstens k) ist in NP, wenn die Distanzen gerundet werden.

Summe von Quadratwurzeln

Gegeben: n Zahlen $a_1, \dots, a_n \in \mathbb{N}$ und eine Zahl $k \in \mathbb{N}$

Frage: Ist

$$\sum_{i=1}^n \sqrt{a_i} \leq k?$$

The American Mathematical Monthly, Vol. 88, No. 10 (December **1981**)

6369*. *Proposed by Joseph O'Rourke, Johns Hopkins University.*

Let a_i and b_j be positive integers, $i, j = 1, 2$ ($1 \leq a_i, b_j \leq N$). Find the minimal positive value of $\sum a_i^{1/2} - \sum b_j^{1/2}$. Suggest or prove asymptotic results. Consider the corresponding problem with sums of three terms.

Offene Frage: Ist das Problem „Summe von Quadratwurzeln“ in NP?

Problem: Minimal Aufspannender Baum

Gegeben: Punkte $P = \{p_1, \dots, p_n\} \subset \mathbb{Z}^2$

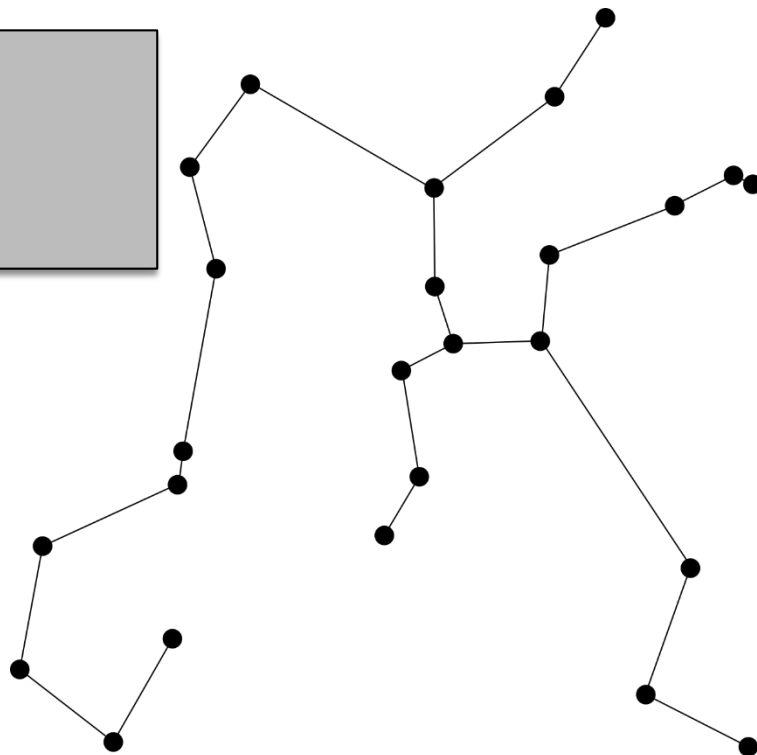
Gesucht: Segmente $s_1, \dots, s_{n-1} \in \binom{P}{2}$, sodass

- $\sum_{i=1}^{n-1} \|s_i\|_2$ minimal
- Segmente bilden keinen Kreis

Dieses Problem lässt sich einfach lösen:

- Gehe Segmente der Länge nach durch
- Falls das Hinzufügen des Segments keinen Kreis schließt, füge es hinzu.

Es existiert also ein Polynomialzeitalgorithmus für das Optimierungsproblem.



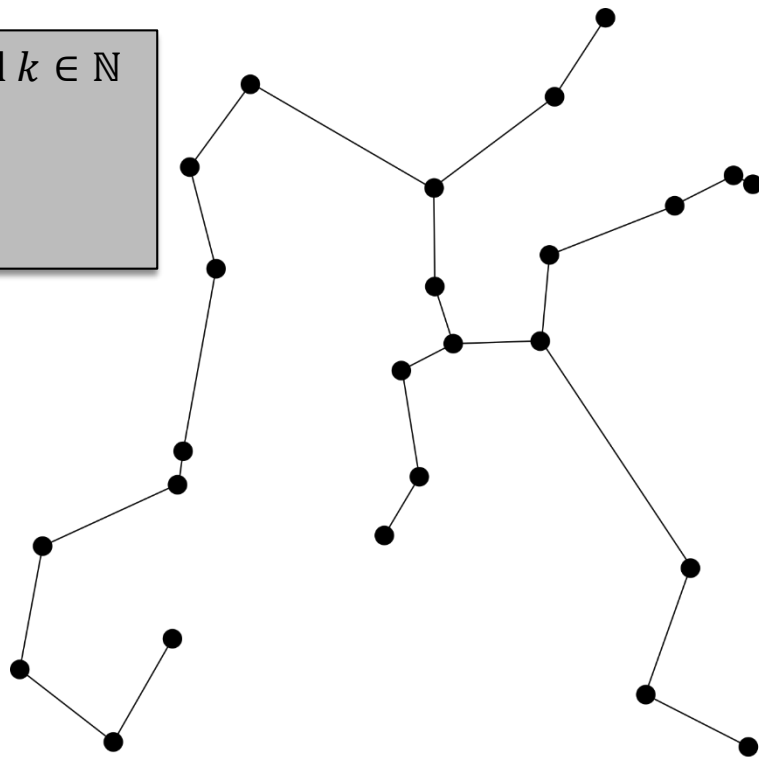
Problem: Minimal Aufspannender Baum

Gegeben: Punkte $P = \{p_1, \dots, p_n\} \subset \mathbb{Z}^2$ und eine Zahl $k \in \mathbb{N}$

Frage: Existieren Segmente $s_1, \dots, s_{n-1} \in \binom{P}{2}$, sodass

- $\sum_{i=1}^{n-1} \|s_i\|_2 \leq k$
- Segmente bilden keinen Kreis

Es ist nicht bekannt, ob dieses Entscheidungsproblem in NP liegt.



Nächste Woche

