

# Theoretische Grundlagen der Informatik

Vorlesung am 15. Dezember 2016

INSTITUT FÜR THEORETISCHE INFORMATIK



Zu einem polynomialen Approximationsalgorithmus  $\mathcal{A}$  sei

$$\mathcal{R}_{\mathcal{A}}^{\infty} := \inf \left\{ r \geq 1 \mid \begin{array}{l} \text{es gibt ein } N_0 > 0, \text{ so dass } \mathcal{R}_{\mathcal{A}}(I) \leq r \\ \text{für alle } I \text{ mit } \text{OPT}(I) \geq N_0 \end{array} \right\}$$

$$\mathcal{R}_{\mathcal{A}}^{\infty} := \inf \left\{ r \geq 1 \mid \begin{array}{l} \text{es gibt ein } N_0 > 0, \text{ so dass } \mathcal{R}_{\mathcal{A}}(I) \leq r \\ \text{für alle } I \text{ mit } \text{OPT}(I) \geq N_0 \end{array} \right\}$$

## Problem COLOR (Optimalwertfassung)

**Gegeben:** Graph  $G = (V, E)$

**Frage:** Wieviele Farben benötigt man um  $V$  zu färben, so dass je zwei adjazente Knoten verschiedene Farben besitzen?

## Satz:

Falls  $\mathcal{P} \neq \mathcal{NP}$ , dann existiert kein relativer Approximationsalgorithmus  $\mathcal{A}$  für COLOR mit  $\mathcal{R}_{\mathcal{A}}^{\infty} < \frac{4}{3}$ .

$$\mathcal{R}_{\mathcal{A}}^{\infty} := \inf \left\{ r \geq 1 \mid \begin{array}{l} \text{es gibt ein } N_0 > 0, \text{ so dass } \mathcal{R}_{\mathcal{A}}(I) \leq r \\ \text{für alle } I \text{ mit } \text{OPT}(I) \geq N_0 \end{array} \right\}$$

## Satz:

Falls  $\mathcal{P} \neq \mathcal{NP}$ , dann existiert kein relativer Approximationsalgorithmus  $\mathcal{A}$  für COLOR mit  $\mathcal{R}_{\mathcal{A}}^{\infty} < \frac{4}{3}$ .

## Beweis:

- Angenommen es gibt einen relativen Approximationsalgorithmus  $\mathcal{A}$  für COLOR mit  $\mathcal{R}_{\mathcal{A}}^{\infty} < \frac{4}{3}$ .
- Wir benutzen  $\mathcal{A}$  um 3COLOR zu lösen.
- Dies ist ein Widerspruch zu  $\mathcal{P} \neq \mathcal{NP}$

Zu zwei Graphen

$$G_1 = (V_1, E_1) \text{ und } G_2 = (V_2, E_2)$$

sei

$$G := (V, E) := G_1[G_2]$$

definiert durch

$$V := V_1 \times V_2$$

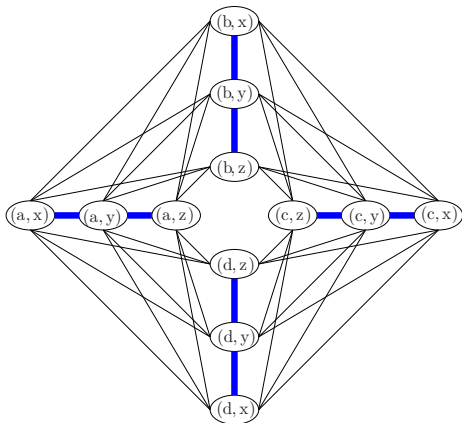
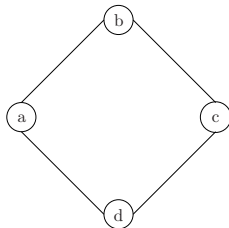
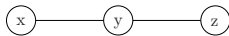
und

$$E := \left\{ \left\{ (u_1, u_2), (v_1, v_2) \right\} \mid \begin{array}{l} \text{entweder } \{u_1, v_1\} \in E_1, \text{ oder} \\ u_1 = v_1 \text{ und } \{u_2, v_2\} \in E_2 \end{array} \right\}$$

## Anschaulich

- Jeder Knoten aus  $G_1$  wird durch eine Kopie von  $G_2$  ersetzt
- Jede Kante aus  $E_1$  durch einen vollständig bipartiten Graphen zwischen den entsprechenden Kopien.

# Approximierbarkeit von COLOR



$$\mathcal{R}_{\mathcal{A}}^{\infty} := \inf \left\{ r \geq 1 \mid \begin{array}{l} \text{es gibt ein } N_0 > 0, \text{ so dass } \mathcal{R}_{\mathcal{A}}(I) \leq r \\ \text{für alle } I \text{ mit } \text{OPT}(I) \geq N_0 \end{array} \right\}$$

- Angenommen es gibt einen relativen Approximationsalgorithmus  $\mathcal{A}$  für COLOR mit  $\mathcal{R}_{\mathcal{A}}^{\infty} < \frac{4}{3}$ .
- Dann existiert ein  $N \in \mathbb{N}$  so, dass  $\mathcal{A}(G) < \frac{4}{3} \text{OPT}(G)$  für alle Graphen  $G$  mit  $\text{OPT}(G) \geq N$ .

- Dann existiert ein  $N \in \mathbb{N}$  so, dass  $\mathcal{A}(G) < \frac{4}{3} \text{OPT}(G)$  für alle Graphen  $G$  mit  $\text{OPT}(G) \geq N$ .
- Sei also  $G = (V, E)$  ein beliebiges Beispiel für 3COLOR.
- Dann definiere  $G^* := K_N[G]$ , wobei  $K_N$  der vollständige Graph über  $N$  Knoten ist.
- Dann gilt:  $\text{OPT}(G^*) = N \cdot \text{OPT}(G) \geq N$ .

## Fallunterscheidung:

- Falls  $G$  dreifärbbar ist, gilt:

$$\mathcal{A}(G^*) < \frac{4}{3} \text{OPT}(G^*) = \frac{4}{3} \cdot N \cdot \text{OPT}(G) \leq \frac{4}{3} \cdot N \cdot 3 = 4N.$$

- Andererseits, falls  $G$  nicht dreifärbbar ist, gilt

$$\mathcal{A}(G^*) \geq \text{OPT}(G^*) = N \cdot \text{OPT}(G) \geq 4N.$$

**Fazit:**  $G$  ist dreifärbbar genau dann, wenn  $\mathcal{A}(G^*) < 4N$ .



- Die Größe von  $G^*$  ist polynomial in der Größe von  $G$ .
- Also kann  $G^*$  in polynomialer Zeit konstruiert werden.
- Damit ist die Anwendung von  $\mathcal{A}$  auf  $G^*$  polynomial in der Größe von  $G$ .
- Also haben wir einen polynomialen Algorithmus zur Lösung von 3COLOR konstruiert.
- Dies ist ein Widerspruch zur Annahme, dass  $\mathcal{P} \neq \mathcal{NP}$ .

## TSP-Optimalwertproblem mit Dreiecksungleichung

**Gegeben:** Graph  $G = (V, E)$  vollständig und gewichtet mit Gewichtsfunktion  $c: E \rightarrow \mathbb{Q}$ .  
Es gilt  $c(u, w) \leq c(u, v) + c(v, w)$  für alle  $u, v, w \in V$

**Frage:** Wie lange ist eine optimale Tour zu  $G$  bezüglich  $c$ ?

### Satz:

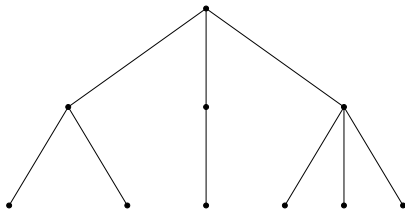
Für das TSP-Optimalwertproblem mit Dreiecksungleichung existiert ein Approximationsalgorithmus  $\mathcal{A}$  mit  $\mathcal{R}_{\mathcal{A}} \leq 2$  für alle Instanzen  $I$ .

## Beweis.

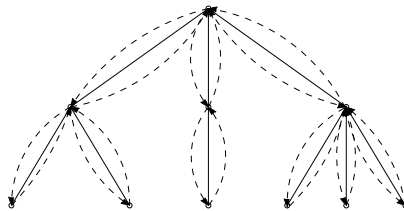
- Sei  $(G = (V, E), c)$  eine Instanz des TSP-Optimalwertproblems mit Dreiecksungleichung.

Betrachte folgenden Algorithmus:

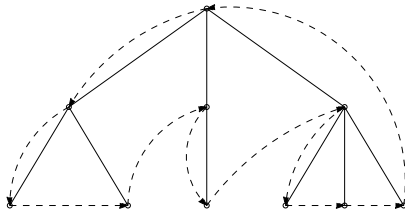
- Berechne einen *MST* (Minimum Spanning Tree) von  $G$ .
- Wähle einen beliebigen Knoten  $w$  als Wurzel.
- Durchlaufe den *MST* in einer Tiefensuche mit Startpunkt  $w$
- **Ergebnis:** Tour  $T$  mit Start- und Endpunkt  $w$ , die jede Kante zweimal durchläuft.
- Konstruiere aus  $T$  eine Tour  $T'$ , indem bereits besuchte Knoten übersprungen werden und die Tour beim nächsten unbesuchten Knoten fortgesetzt wird.



(a) MST eines Graphen



(b) Tiefensuch-Tour durch den MST



(c) TSP-Tour als abgekürzte Tiefensuch-Tour

- Bezeichne  $c(G')$  die Summe der Kantengewichte in Subgraph  $G'$

Es gilt

$$c(T') \leq c(T) = 2 \cdot c(MST) .$$

Eine TSP-Tour kann als ein aufspannender Baum plus eine zusätzliche Kante betrachtet werden. Also gilt

$$c(MST) \leq c(OPT) .$$

Insgesamt erhält man

$$c(T') \leq c(T) = 2 \cdot c(MST) \leq 2 \cdot c(OPT) ,$$

also

$$\mathcal{R}_{\mathcal{A}} = \frac{c(T')}{c(OPT)} \leq 2 .$$

Ein (polynomiales) **Approximationsschema (PAS)** für ein Optimierungsproblem  $\Pi$  ist eine Familie von Algorithmen  $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$ , so dass für alle  $\varepsilon > 0$

- $\mathcal{R}_{\mathcal{A}_\varepsilon} \leq 1 + \varepsilon$  ist (d.h.  $\mathcal{A}_\varepsilon$  ist ein  $\varepsilon$ -approximierender Algorithmus).
- $\mathcal{A}_\varepsilon$  polynomial in der Größe des Inputs ist.

Ein Approximationsschema  $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$  heißt **vollpolynomial (FPAS)** falls seine Laufzeit zudem polynomial in  $\frac{1}{\varepsilon}$  ist.

Bezeichne  $\langle I \rangle$  die Kodierungslänge der Eingabe-Instanz  $I$ .

**Satz:**

Sei  $\Pi$  ein  $\mathcal{NP}$ -schweres Optimierungsproblem mit

- $\text{OPT}(I) \in \mathbb{N}$  für alle  $I \in D_{\Pi}$ , und
- es existiert ein Polynom  $q$  mit  $\text{OPT}(I) < q(\langle I \rangle)$  für alle  $I \in D_{\Pi}$ .

Falls  $\mathcal{P} \neq \mathcal{NP}$ , so gibt es kein FPAS  $\{\mathcal{A}_{\varepsilon} \mid \varepsilon > 0\}$  für  $\Pi$ .

## Satz:

Sei  $\Pi$  ein  $\mathcal{NP}$ -schweres Optimierungsproblem mit

- $\text{OPT}(I) \in \mathbb{N}$  für alle  $I \in D_{\Pi}$ , und
- es existiert ein Polynom  $q$  mit  $\text{OPT}(I) < q(\langle I \rangle)$  für alle  $I \in D_{\Pi}$ .

Falls  $\mathcal{P} \neq \mathcal{NP}$ , so gibt es kein FPAS  $\{\mathcal{A}_{\varepsilon} \mid \varepsilon > 0\}$  für  $\Pi$ .

## Beweis:

- O.B.d.A. sei  $\Pi$  ein Maximierungsproblem.
- Sei  $\{\mathcal{A}_{\varepsilon} \mid \varepsilon > 0\}$  ein FPAS für  $\Pi$
- Zu  $I \in D_{\Pi}$  sei  $\varepsilon_0 := \frac{1}{q(\langle I \rangle)}$
- Dann ist  $\mathcal{A}_{\varepsilon_0}$  polynomial in  $\langle I \rangle$  und in  $\frac{1}{\varepsilon_0} = q(\langle I \rangle)$



## Satz:

Sei  $\Pi$  ein  $\mathcal{NP}$ -schweres Optimierungsproblem mit

- $\text{OPT}(I) \in \mathbb{N}$  für alle  $I \in D_{\Pi}$ , und
- es existiert ein Polynom  $q$  mit  $\text{OPT}(I) < q(\langle I \rangle)$  für alle  $I \in D_{\Pi}$ .

Falls  $\mathcal{P} \neq \mathcal{NP}$ , so gibt es kein FPAS  $\{\mathcal{A}_{\varepsilon} \mid \varepsilon > 0\}$  für  $\Pi$ .

Es gilt: 
$$\text{OPT}(I) \leq (1 + \varepsilon_0) \mathcal{A}_{\varepsilon_0}(I) \text{ und}$$

Also auch 
$$\text{OPT}(I) < q(\langle I \rangle) = \frac{1}{\varepsilon_0}$$

$$\text{OPT}(I) - \mathcal{A}_{\varepsilon_0}(I) \leq \varepsilon_0 \cdot \mathcal{A}_{\varepsilon_0}(I) \leq \varepsilon_0 \cdot \text{OPT}(I) < 1$$

- Da  $\text{OPT}(I) \in \mathbb{N}$ , ist  $\text{OPT}(I) = \mathcal{A}_{\varepsilon_0}(I)$
- Widerspruch zur Annahme, dass  $\mathcal{P} \neq \mathcal{NP}$ .

## Problem KNAPSACK

**Gegeben:** Eine endliche Menge  $M$ ,  
eine Gewichtsfunktion  $w : M \rightarrow \mathbb{N}_0$ ,  
eine Kostenfunktion  $c : M \rightarrow \mathbb{N}_0$ ,  $W \in \mathbb{N}$ .

**Aufgabe:** Gib eine Teilmenge  $M'$  von  $M$  an, so dass  
 $\sum_{i \in M'} w_i \leq W$  und  $\sum_{i \in M'} c_i$  maximal ist.

# Ein pseudopolynomialer, optimaler Algorithmus für KNAPSACK

Bezeichne, für  $r \in \mathbb{N}_0$

$$w_r^j := \min_{M' \subseteq \{1, \dots, j\}} \left\{ \sum_{i \in M'} w_i \mid \sum_{i \in M'} c_i = r \right\}$$

## ■ Initialisierung

Für  $1 \leq j \leq n$  setze  $w_0^j := 0$  ansonsten setze  $c := \sum_{i=1}^n c_i$

## ■ Berechnung

Solange  $w_r^j \leq W$  berechne für  $2 \leq j \leq n$  und  $1 \leq r \leq c$  den Wert

$$w_r^j = \min \left\{ w_{r-c_j}^{j-1} + w_r^j, w_r^{j-1} \right\} .$$

## ■ Ausgabe

$$c^* := \max_{1 \leq i \leq n} \left\{ r \mid w_r^i \leq W \right\}$$

und die entsprechende Menge  $M' \subseteq M$  mit  $c^* = \sum_{i \in M'} c_i$ .

# Ein pseudopolynomialer, optimaler Algorithmus für KNAPSACK

Bezeichne, für  $r \in \mathbb{N}_0$

$$w_r^j := \min_{M' \subseteq \{1, \dots, j\}} \left\{ \sum_{i \in M'} w_i \mid \sum_{i \in M'} c_i = r \right\}$$

## ■ Initialisierung

Für  $1 \leq j \leq n$  setze  $w_0^j := 0$  ansonsten setze  $c := \sum_{i=1}^n c_i$

## ■ Berechnung

Solange  $w_r^j \leq W$  berechne für  $2 \leq j \leq n$  und  $1 \leq r \leq c$  den Wert

$$w_r^j = \min \left\{ w_{r-c_j}^{j-1} + w^j, w_r^{j-1} \right\}.$$

## ■ Ausgabe

$$c^* := \max_{1 \leq i \leq n} \left\{ r \mid w_r^i \leq W \right\}$$

und die entsprechende Menge  $M' \subseteq M$  mit  $c^* = \sum_{i \in M'} c_i$ .

**Laufzeit:** in  $\mathcal{O}(n \cdot c)$ . **Lösung:** optimal.

⇒ Optimaler pseudopolynomialer Algorithmus.

- Bezeichne  $\mathcal{A}$  obigen pseudopolynomialen Algorithmus mit Laufzeit  $\mathcal{O}(n \cdot c)$  für KNAPSACK.
- Sei  $k$  beliebig aber fest.
- **Betrachte das skalierte Problem  $\Pi_k$  zu mit  $c'_i := \lfloor \frac{c_i}{k} \rfloor$  für alle  $i \in M$ .**
- Dann liefert  $\mathcal{A}$  für jedes  $I_k \in \Pi_k$  eine Menge  $M' \subseteq M$  mit  $\sum_{i \in M'} c'_i = \text{OPT}(I_k)$ .
- Setze nun  $c_{\max} := \max_{i \in M} c_i$ .
- **Zu  $\varepsilon > 0$  sei  $\mathcal{A}_\varepsilon$  Algorithmus  $\mathcal{A}$  angewendet auf  $I_k$ , wobei**

$$k := \frac{c_{\max}}{\left(\frac{1}{\varepsilon} + 1\right) \cdot n}$$

**Satz:**

$\mathcal{R}_{\mathcal{A}_\varepsilon}(I) \leq 1 + \varepsilon$  für alle  $I \in D_{\text{II}}$  und die Laufzeit von  $\mathcal{A}_\varepsilon$  ist in  $\mathcal{O}(n^3 \cdot \frac{1}{\varepsilon})$  für alle  $\varepsilon > 0$ , d.h.  $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$  ist ein FPAS für KNAPSACK.

**Satz:**

$\mathcal{R}_{\mathcal{A}_\varepsilon}(I) \leq 1 + \varepsilon$  für alle  $I \in D_{\Pi}$  und die Laufzeit von  $\mathcal{A}_\varepsilon$  ist in  $\mathcal{O}(n^3 \cdot \frac{1}{\varepsilon})$  für alle  $\varepsilon > 0$ , d.h.  $\{\mathcal{A}_\varepsilon \mid \varepsilon > 0\}$  ist ein FPAS für KNAPSACK.

**Beweis:**

Die Laufzeit von  $\mathcal{A}_\varepsilon$  ist in  $\mathcal{O}(n \cdot \sum_{i=1}^n c'_i)$  und

$$\sum_{i=1}^n c'_i < \sum_{i=1}^n \frac{c_i}{k} \leq n \cdot \frac{c_{\max}}{k} = \left(\frac{1}{\varepsilon} + 1\right) n^2.$$

Also ist die Laufzeit von  $\mathcal{A}_\varepsilon$  in  $\mathcal{O}(n^3 \cdot \frac{1}{\varepsilon})$  für alle  $\varepsilon > 0$ .

Für die Abschätzung von  $\mathcal{R}_{\mathcal{A}_\varepsilon}$  betrachte  $M'$  mit  $\text{OPT}(I) = \sum_{i \in M'} c_i$ . Es gilt

$$\text{OPT}(I_k) \geq \sum_{i \in M'} \left\lfloor \frac{c_i}{k} \right\rfloor \geq \sum_{i \in M'} \left( \frac{c_i}{k} - 1 \right).$$

Also ist

$$\text{OPT}(I) - k \cdot \text{OPT}(I_k) \leq k \cdot n.$$

Da  $\frac{1}{k} \mathcal{A}_\varepsilon(I) \geq \text{OPT}(I_k)$  ist, folgt

$$\text{OPT}(I) - \mathcal{A}_\varepsilon(I) \leq k \cdot n$$

und wegen  $\text{OPT}(I) \geq c_{\max}$  (wir setzen wieder o.B.d.A.  $W \geq w_i$  für alle  $i \in M$  voraus) folgt

$$\begin{aligned} \mathcal{R}_{\mathcal{A}_\varepsilon}(I) &= \frac{\text{OPT}(I)}{\mathcal{A}_\varepsilon(I)} \leq \frac{\mathcal{A}_\varepsilon(I) + kn}{\mathcal{A}_\varepsilon(I)} = 1 + \frac{kn}{\mathcal{A}_\varepsilon(I)} \leq 1 + \frac{kn}{\text{OPT}(I) - kn} \\ &\leq 1 + \frac{kn}{c_{\max} - kn} = 1 + \frac{1}{\frac{1}{\varepsilon} + 1 - 1} = 1 + \varepsilon. \end{aligned}$$



# Ein allgemeineres Resultat

Mit einem ähnlichen Beweis kann man zeigen:

## Satz:

Sei  $\Pi$  ein Optimierungsproblem für das gilt:

- $\text{OPT}(I) \in \mathbb{N}$  für alle  $I \in D_{\Pi}$
- es existiert ein Polynom  $q$  mit  $\text{OPT}(I) \leq q(\langle I \rangle + \max \#(I))$   
( $\max \#(I)$  ist die größte in  $I$  vorkommende Zahl)

Falls  $\Pi$  ein FPAS hat, so hat es einen pseudopolynomialen optimalen Algorithmus.