Übersicht

Absolute Approximation

Sei P Optimierungsproblem, $x \in I_P$ und $y \in S_P(x)$.

Definition

Absoluter Fehler: $D(x, y) := \left| m^*(x) - m(x, y) \right|$

Ein Approximations-Algorithmus heißt absolut,
wenn $D(x, A(x)) \leq k$ ist, für alle $x \in I_P$ und k konstant.

Beispiel: Minimum Planar Graph Coloring

Satz

Falls $P \neq NP$ ist, gibt es keinen absoluten Approximations-Algorithmus für Maximum Knapsack.

Relative Approximation

Definition

Optimierungsproblem P, $x \in I_P$ und $y \in S_P(x)$.

Performance ratio: $$R(x,y) := \max\left(\frac{m(x,y)}{m^*(x)}, \frac{m^*(x)}{m(x,y)}\right)$$

Relativer Fehler: $$E(x,y) := \frac{|m^*(x) - m(x,y)|}{\max(m^*(x), m(x,y))} = 1 - \frac{1}{R(x,y)}$$

Für $r \geq 1$ ist ein r-Approximations-Algorithmus
ein Algorithmus A, der für jedes $x \in I_P$ ein $A(x) \in S_P(x)$
mit $R(x, A(x)) \leq r$ liefert.

P heißt r-approximierbar, falls es einen
Polynomialzeit-r-Approximations-Algorithmus für P gibt.

Die Klasse APX

APX ist die Klasse aller Optimierungsprobleme $P \in \mathrm{NPO}$, die r-approximierbar für ein $r \geq 1$ sind.

$$\mathrm{PO} \subseteq \mathrm{APX} \subseteq \mathrm{NPO}$$

APX enthält NP-schwere Probleme, z.B. Maximum Knapsack, also

$$\mathrm{P} \neq \mathrm{NP} \implies \mathrm{PO} \subsetneq \mathrm{APX}$$

Maximum SAT

Maximum SAT

Instanz:	CNF $D_1 \wedge \ldots \wedge D_m$ in Variablen $x_1, \ldots, x_n$
Lösung:	Bewertung $\alpha : \{x_1, \ldots, x_n\} \to \{0, 1\}$
Maß:	Anzahl erfüllter Klauseln $\#\{i; \alpha \models D_i\}$

Maximum SAT $\in$ APX.

Traveling Salespersons revisited

Satz

MINIMUM TRAVELING SALESPERSON *ist nicht in APX, außer wenn* $P = NP$.

Also gilt:

$$\mathsf{P} \neq \mathsf{NP} \implies \mathsf{PO} \subsetneq \mathsf{APX} \subsetneq \mathsf{NPO}$$

Euler-Tour

Multigraph: kann parallele Kanten haben

Euler-Tour: geschlossener Weg, der jede Kante genau einmal durchläuft.

Satz (Euler)

Es gibt Euler-Tour in G,
gdw. $\deg v$ *gerade ist für alle Knoten* v.

Euler-Tour in G kan in polynomieller Zeit berechnet werden.

TSP-Tour aus Euler-Tour

Sei $V = \{c_1, \ldots, c_n\}$ mit $D \in \mathbb{R}^{n \times n}$ Instanz von Metric TSP.

Sei $G = (V, E)$ ein Euler'scher Graph auf V.

Lemma

Aus Euler-Tour in G extrahiert man TSP-Tour π mit:

$$m(\pi) \leq \sum_{\{c_i, c_j\} \in E} d(i, j)$$

Ein 2-Approximations-Algorithmus

- Berechne minimalen Spannbaum T
- verdoppele jede Kante in T
- berechne Euler-Tour in diesem Graphen
- extrahiere daraus eine TSP-Tour

Satz

Für den Wert $m_A(x)$ der so berechneten Tour gilt:

$$m_A(x) \leq 2m^*(x)$$

Korollar

Minimum Metric Traveling Salesperson ist in APX.

Matchings in gewichteten Graphen

Matching $M \subseteq E$ in $G = (V, E)$ ist perfekt, wenn $|M| = |V|/2$ ist.

Minimum Weight Perfect Matching

Instanz:	Graph $G = (V, E)$, Kantengewicht $w : E \to \mathbb{N}$
Lösung:	perfektes Matching $M \subseteq E$
Maß:	$\sum_{e \in M} w(e)$

Satz

Minimum Weight Perfect Matching *ist in PO.*

Christofides' Algorithmus

- Berechne einen Minimalen Spannbaum T
- sei U die Menge der Knoten ungeraden Grades in T
- berechne perfektes Matching M minimalen Gewichtes auf U
- berechne Euler-Tour in $T \cup M$
- extrahiere daraus eine TSP-Tour

Satz

Für den Wert $m_C(x)$ der von Christofides' Algorithmus berechneten Tour gilt:

$$m_C(x) \leq \frac{3}{2} m^*(x)$$

Approximationsschemata

Definition: Ein Polynomialzeit-Approximationsschema für P ist ein Algorithmus A mit

- Eingabe: $x \in I_P$ und $1 < r \in \mathbb{Q}$, Ausgabe: $A(x, r) \in S_P(x)$
- Performance ratio $R(x, A(x, r)) \leq r$
- Laufzeit $|x|^{O(1)}$

PTAS ist die Klasse aller Optimierungsprobleme in NPO, für die es ein Polynomialzeit-Approximationsschema gibt.

$$\text{PO} \subseteq \text{PTAS} \subseteq \text{APX} \subseteq \text{NPO}$$

PTAS enthält NP-schwere Probleme (z.B. Maximum Knapsack):

$$\text{P} \neq \text{NP} \implies \text{PO} \subsetneq \text{PTAS}$$

Beispiel: ein Approximationsschema

Minimum Partition

Instanz:	Menge $X = \{x_1, \ldots, x_n\}$, für $x_i \in X$ Gewicht $a_i \in \mathbb{N}$
Lösung:	Partition $X = Y_1 \uplus Y_2$
Maß:	$\max\left(\sum_{x_i \in Y_1} a_i \, , \sum_{x_i \in Y_2} a_i \right)$

Approximationsschema:

sortiere X so dass $a_1 \geq a_2 \geq \ldots \geq a_n$
$k := \lceil (2-r)/(r-1) \rceil$
finde eine optimale Partition $Y_1 \uplus Y_2 = \{x_1, \ldots, x_k\}$
for $j := k+1$ to n
    if $\sum Y_1 \leq \sum Y_2$
        then $Y_1 := Y_1 \cup \{x_j\}$
        else $Y_2 := Y_2 \cup \{x_j\}$

Ein Problem in PTAS

Maximum Integer Knapsack

Instanz: Menge $X = \{x_1, \ldots, x_n\}$,
für jedes $x_i \in X$: Gewicht $a_i \in \mathbb{N}$
Wert $p_i \in \mathbb{N}$
Kapazität $b \in \mathbb{N}$

Lösung: Funktion $c : X \to \mathbb{N}$ mit $\sum_{x_i \in Y} c(x_i) a_i \leq b$

Maß: Gesamtwert $\sum_{x_i \in Y} c(x_i) p_i$

Approximationsschema für Integer Knapsack

sortiere X so dass $p_1 \geq p_2 \geq \ldots \geq p_n$

$c := \vec{0}$

$\delta := \lceil \frac{1}{(r-1)} \rceil$

for all f with $\sum f(x_i) \leq \delta$ and $\sum f(x_i)a_i \leq b$

$k := \max_i f(x_i) > 0$

$b_k := b - \sum f(x_i)a_i$

sei $x_m \in \{x_k, \ldots, x_n\}$ mit $\frac{p_m}{a_m}$ maximal

$f(x_m) := f(x_m) + \lfloor \frac{b_k}{a_m} \rfloor$

if $\sum f(x_i)p_i \geq \sum c(x_i)p_i$ then

$c := f$

APX versus PTAS

Lückentechnik: Sei Q NP-vollständig und $P \in \text{NPO}$. Wenn es Funktionen $f : I_Q \to I_P$ und $c : I_Q \to \mathbb{N}$ in FP gibt mit

$$m^*(f(x)) \leq c(x) \qquad \text{für } x \in Y_Q$$
$$m^*(f(x)) \geq c(x)(1+G) \qquad \text{für } x \in N_Q$$

für ein $G > 0$, dann ist P nicht r-approximierbar für $r < 1 + G$.

Satz

Falls $P \neq NP$, dann ist Minimum Bin Packing *nicht in PTAS.*

$$\text{P} \neq \text{NP} \implies \text{PO} \subsetneq \text{PTAS} \subsetneq \text{APX} \subsetneq \text{NPO}$$

Volle Approximationsschemata

Definition: Ein volles Approximationsschema für P ist ein Algorithmus A mit

- Eingabe: $x \in I_P$ und $1 < r \in \mathbb{Q}$, Ausgabe: $A(x, r) \in S_P(x)$
- Performance ratio $R(x, A(x, r)) \leq r$
- Laufzeit polynomial in x und $\frac{1}{r-1}$

FPTAS ist die Klasse aller Optimierungsprobleme in NPO, für die es ein volles Polynomialzeit-Approximationsschema gibt.

$$\mathsf{PO} \subseteq \mathsf{FPTAS} \subseteq \mathsf{PTAS} \subseteq \mathsf{APX} \subseteq \mathsf{NPO}$$

FPTAS enthält NP-schwere Probleme (z.B. Maximum Knapsack):

$$\mathsf{P} \neq \mathsf{NP} \implies \mathsf{PO} \subsetneq \mathsf{FPTAS}$$

FPTAS vs. PTAS

Definition: Ein Problem $P \in \text{NPO}$ ist beschränkt,
falls es ein Polynom $p()$ gibt mit:

für alle $x \in I_P$ und $y \in S_P(x)$ ist $m(x, y) \leq p(|x|)$.

Satz

Falls $P \neq NP$, ist kein NP-schweres, beschränktes Problem in FPTAS.

Es gibt NP-schwere, beschränkte Probleme in PTAS,
z.B. Maximum Planar Independent Set.

$$\text{P} \neq \text{NP} \implies \text{PO} \subsetneq \text{FPTAS} \subsetneq \text{PTAS} \subsetneq \text{APX} \subsetneq \text{NPO}$$

Pseudopolynomielle Probleme

Sei $P \in$ NPO mit Zahlen als Bestandteil der Instanzen,

für $x \in I_P$ sei $\max(x)$ das Maximum der Zahlen in x.

P heit pseudopolynomiell, wenn es Algorithmus A gibt, der P_C in Zeit $p(|x|, max(x))$ löst.

Beispiel: MAXIMUM KNAPSACK ist pseudopolynomiell.

Proposition

Sei $P \in$ FPTAS, so daß für $x \in I_P$ und $y \in S_P(x)$ gilt:

$$m(x, y) \leq p(|x|, max(x))$$

Dann ist P pseudopolynomiell.

Stark NP-schwere Probleme

Für $P \in \text{NPO}$ und Polynom p ist Problem $P\lceil_p$ definiert:

- $I_{P\lceil_p} := \{\, x \in I_p \,;\, max(x) \leq p(|x|) \,\}$
- $S_{P\lceil_p}(x) := S_P(x)$ für $x \in I_{P\lceil_p}$
- $m_{P\lceil_p}(x, y) := m(x, y)$ für $x \in I_{P\lceil_p}$ und $y \in S_{P\lceil_p}(x)$

P heißt stark NP-schwer, wenn $P\lceil_p$ NP-schwer ist.

Proposition

$\text{P} \neq \text{NP} \implies$

kein stark NP-schweres Problem ist pseudopolynomiell.