Einführung in Approximative Algorithmen und Parametrisierte Komplexität

Tobias Lieber

10. Dezember 2010

Grundlegendes

Approximationsalgorithmen

Parametrisierte Komplexität

Definition (Die Klassen P und NP)

Eine Sprache $L \subseteq \Sigma^*$ ist in P (NP) enthalten, wenn es eine (nicht)deterministische Turingmaschine M gibt, deren Ausgabe, nach polynomiell vielen Schritten in der Größe der Eingabe x, durch

$$M(x) = \begin{cases} 1 & x \in L \\ 0 & x \notin L \end{cases}$$

gegeben ist.

Definition (Many-One-Reduktionen)

Eine Sprache $A \subseteq \Sigma^*$ ist (polynomiell many-one-)reduzierbar auf $B \subseteq \Sigma^*$ wenn es eine in Polynomialzeit berechenbare Funktion $f : \Sigma^* \mapsto \Sigma^*$ gibt, so dass $x \in A \iff f(x) \in B$ gilt. Wir schreiben $A \leq_m B$.

Definition (NP-Vollständig)

Eine Sprache L ist NP-Vollständig, wenn $L \in \mathrm{NP}$ und für alle Sprachen $B \in \mathrm{NP}$ gilt: $B \leq_m L$.

Definition (Das Problem VertexCover)

Eine Menge $X \subseteq V$ ist ein VertexCover von einem Graphen $G = (V, E)$, wenn es für jede Kante $e \in E$ einen Knoten $v \in X$ gibt so dass $v \in e$ gilt. Die Sprache VertexCover enthält alle Eingaben $G = (V, E)$ und $k \in \mathbb{N}_0$, die ein VertexCover der Größe k haben.

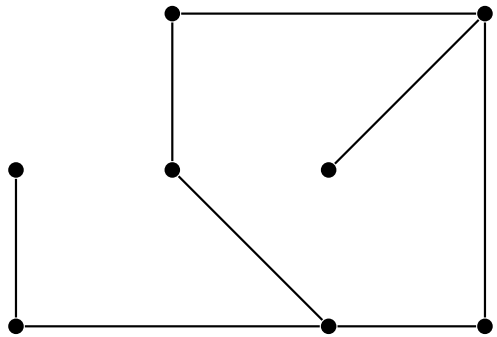

Abbildung: Beispiel: Ein VertexCover

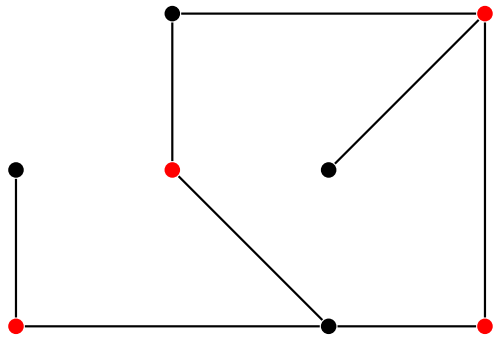

Abbildung: Beispiel: Ein VertexCover

Das (Entscheidungs-)Problem VertexCover ist NP-Vollständig (Reduktion von IndependentSet).

Algorithm 2.1: Approximationsalgorithmus für VertexCover

$C \leftarrow \emptyset$;
while $\exists e = (u, v) \in E$ **do**
    $C \leftarrow C \cup \{u, v\}$;
    entferne in G alle Kanten, die inzident zu u und v sind;
return C;

Algorithm 2.2: Approximationsalgorithmus für VertexCover

$C \leftarrow \emptyset$;
while $\exists e = (u, v) \in E$ **do**
    $C \leftarrow C \cup \{u, v\}$;
    entferne in G alle Kanten, die inzident zu u und v sind;
return C;

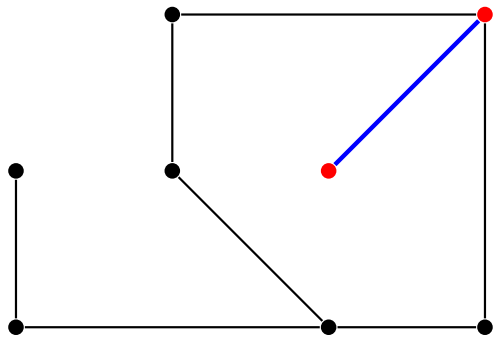

Abbildung: Veranschaulichung des Algorithmus

Algorithm 2.3: Approximationsalgorithmus für VertexCover

$C \leftarrow \emptyset$;
while $\exists e = (u, v) \in E$ **do**
    $C \leftarrow C \cup \{u, v\}$;
    entferne in G alle Kanten, die inzident zu u und v sind;
return C;

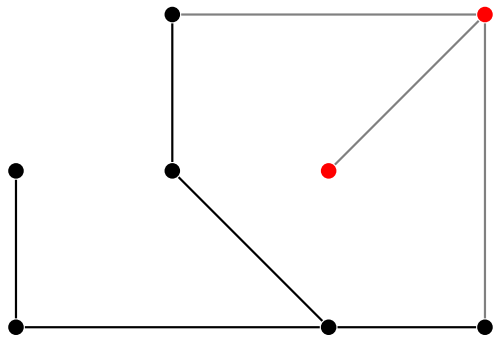

Abbildung: Veranschaulichung des Algorithmus

Algorithm 2.4: Approximationsalgorithmus für VertexCover

$C \leftarrow \emptyset$;
while $\exists e = (u, v) \in E$ **do**
    $C \leftarrow C \cup \{u, v\}$;
    entferne in G alle Kanten, die inzident zu u und v sind;
return C;

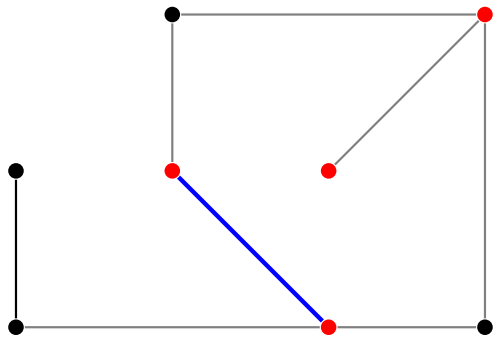

Abbildung: Veranschaulichung des Algorithmus

Definition (Approximationsgüte)

Die Approximationsgüte ρ von einem Algorithmus A ist definiert durch:

$$\rho_A = \max_{I \subseteq L} \left\{ \frac{A(I)}{opt_I}, \frac{opt_I}{A(I)} \right\}.$$

Approximationsgüte des VertexCover-Algorithmus

- Jede Kante muss von mindestens einem Knoten abgedeckt werden.
- Eine optimale Lösung muss auf alle Fälle die ausgewählten Kanten abdecken.

Approximationsgüte des VertexCover-Algorithmus

- Jede Kante muss von mindestens einem Knoten abgedeckt werden.
- Eine optimale Lösung muss auf alle Fälle die ausgewählten Kanten abdecken.
- Also muss die Optimallösung mindestens so viele Knoten enthalten, wie Kanten ausgewählt wurden.
- Die Approximationsgüte ist also 2.

Definition (Approximationsklassen)

Name	Güte	Zeit
	$f(n)$	$n^{\mathcal{O}(1)}$
APX	const.	$n^{\mathcal{O}(1)}$
PTAS	$1+\epsilon$	$n^{f(\frac{1}{\epsilon})}$
EPTAS	$1+\epsilon$	$f(\frac{1}{\epsilon})n^{\mathcal{O}(1)}$
FPTAS	$1+\epsilon$	$(n+\frac{1}{\epsilon})^{\mathcal{O}(1)}$

Definition (Die Klasse FPT)

Sei $L \subseteq \Sigma^* \times \mathbb{N}$ eine Sprache und $f : \mathbb{N} \mapsto \mathbb{N}$.
Wir schreiben $L \in \mathsf{FPT}$, wenn alle $(x, k) \in L$ in Zeit $f(k)|x|^{\mathcal{O}(1)}$ von einer DTM erkannt werden.

Algorithm 3.1: Hat G ein VertexCover der Größe k

$T_1 \longleftarrow \{\emptyset\}$;
for $i{=}1$ **to** k **do**
    forall the $X \in T_i$ **do**
        Wähle eine Kante $\{u, v\} \in G\,[V \setminus X]$;
        $T_{i+1} \longleftarrow (T_i \setminus X) \cup \{X \cup \{u\}, X \cup \{v\}\}$;
        if $G\,[V \setminus (X \cup \{u\})] = \emptyset$ **then return** $X \cup \{u\}$;
        if $G\,[V \setminus (X \cup \{v\})] = \emptyset$ **then return** $X \cup \{v\}$

Definition

Sei $L \subseteq \Sigma^* \times \mathbb{N}$ eine Sprache und $d, f, g, h : \mathbb{N} \mapsto \mathbb{N}$.

1. Die Sprache L ist in FPT, wenn alle $(x, k) \in L$ in Zeit $d(k)|x|^{\mathcal{O}(1)}$ von einer DTM erkannt werden.
2. Die Sprache L ist in FPT, wenn alle $(x, k) \in L$ in Zeit $f(k) + g(k)|x + k|^{\mathcal{O}(1)}$ von einer DTM erkannt werden.
3. Die Sprache L ist in FPT, wenn alle $(x, k) \in L$ in Zeit $h(k) + |x|^{\mathcal{O}(1)}$ von einer DTM erkannt werden.

Beweis.

□

Definition

Sei $L \subseteq \Sigma^* \times \mathbb{N}$ eine Sprache und $d, f, g, h : \mathbb{N} \mapsto \mathbb{N}$.

1. Die Sprache L ist in FPT, wenn alle $(x, k) \in L$ in Zeit $d(k)|x|^{\mathcal{O}(1)}$ von einer DTM erkannt werden.
2. Die Sprache L ist in FPT, wenn alle $(x, k) \in L$ in Zeit $f(k) + g(k)|x + k|^{\mathcal{O}(1)}$ von einer DTM erkannt werden.
3.

Beweis.

2) $\Rightarrow$ 1) Annahme: $p(x) = x^d$. Wenn $a, b \geq 1$ gilt, folgt aus $a + b \leq a(b + 1)$:

$$g(k) + f(k)(n + k)^d \leq (g(k) + f(k)(k + 1)^d)(n + 1)^d$$

□

Definition

Sei $L \subseteq \Sigma^* \times \mathbb{N}$ eine Sprache und $d, f, g, h : \mathbb{N} \mapsto \mathbb{N}$.

1. Die Sprache L ist in FPT, wenn alle $(x, k) \in L$ in Zeit $d(k)|x|^{\mathcal{O}(1)}$ von einer DTM erkannt werden.
2.
3. Die Sprache L ist in FPT, wenn alle $(x, k) \in L$ in Zeit $h(k) + |x|^{\mathcal{O}(1)}$ von einer DTM erkannt werden.

Beweis.

1) $\Rightarrow$ 3)

$$f(k)p(n) \leq f(k)^2 + p(n)^2$$

□

Definition

Sei $L \subseteq \Sigma^* \times \mathbb{N}$ eine Sprache und $d, f, g, h : \mathbb{N} \mapsto \mathbb{N}$.

1.
2. Die Sprache L ist in FPT, wenn alle $(x, k) \in L$ in Zeit $f(k) + g(k)|x + k|^{\mathcal{O}(1)}$ von einer DTM erkannt werden.
3. Die Sprache L ist in FPT, wenn alle $(x, k) \in L$ in Zeit $h(k) + |x|^{\mathcal{O}(1)}$ von einer DTM erkannt werden.

Beweis.

3) $\Rightarrow$ 2) klar $\square$

Definition (Die Klasse XP)

Sei $L \subseteq \Sigma^* \times \mathbb{N}$ eine Sprache und $f : \mathbb{N} \mapsto \mathbb{N}$.
Wenn alle $(x, k) \in L$ von einer DTM in $|x|^{f(k)}$ Schritten erkannt werden, dann ist $L \in \mathsf{XP}$.

Definition (Die Klasse XP)

Sei $L \subseteq \Sigma^* \times \mathbb{N}$ eine Sprache und $f : \mathbb{N} \mapsto \mathbb{N}$.
Wenn alle $(x, k) \in L$ von einer DTM in $|x|^{f(k)}$ Schritten erkannt werden, dann ist $L \in \mathsf{XP}$.

Beispiel für XP: IndependentSet ist in XP. Aber Färbbarkeit von Graphen ist nicht in XP ausser $\mathsf{P} = \mathsf{NP}$ (3-Färbbarkeit ist NP-Vollständig).

Lemma

$L \in EPTAS \Rightarrow L \in FPT$.

Beweis.

- Wir wollen entscheiden ob eine Eingabe X mit Parameter k eine Lösung hat.
- Wir führen ein EPTAS A mit der Eingabe X und $\epsilon = \frac{1}{k+1}$.
- Erhalten wir eine Lösung Y, die ein Beweis für (X, k) ist, so akzeptieren wir.
- Sonst ist Y höchstens ein Beweis für $(X, k+1)$. Somit:

$$opt_X = \frac{A(X)}{\rho_L} \geq \frac{k+1}{1+\frac{1}{k+1}} = \frac{(k+1)^2}{k+2} > k$$

Wir können also mit Sicherheit ablehnen.

□

Vielen Dank für Ihre Aufmerksamkeit