Approximationsalgorithmen für NP-harte Optimierungsprobleme

Prof. Dr. Berthold Vöcking
Lehrstuhl Informatik 1
Algorithmen und Komplexität
RWTH Aachen

4. Januar 2011

Was tun mit NP-harten Problemen?

Viele praxisrelevante Optmierungsprobleme sind NP-hart, z.B.

- Bin Packing Problem (BPP)
- Knapsack Problem (KP)
- Traveling Salesperson Problem (TSP)

Approximationsalgorithmen sind ein theoretisch fundierter und praktikabler Ansatz für den Umgang mit diesen Problemen.

Def: Approximationsalgorithmus

Sei Π ein Optimierungsproblem. Für eine Instanz I von Π bezeichnen wir den optimalen Zielfunktionswert mit $opt(I)$.

- Ein α-*Approximationsalgorithmus*, $\alpha > 1$, für ein Minimierungsproblem Π berechnet für jede Instanz I von Π eine zulässige Lösung mit Zielfunktionswert höchstens $\alpha \cdot opt(I)$.
- Ein α-*Approximationsalgorithmus*, $\alpha < 1$, für ein Maximierungsproblem Π berechnet für jede Instanz I von Π eine zulässige Lösung mit Zielfunktionswert mindestens $\alpha \cdot opt(I)$.

α wird auch als *Approximationsfaktor* oder *Approximationsgüte* bezeichnet.

BPP – Approximationsalgorithmus

Zur Erinnerung:

Beim Bin Packing Problem (BPP) suchen wir eine Verteilung von N Objekten mit Gewichten $w_1, \ldots, w_N$ auf eine möglichst kleine Anzahl von Behältern mit Gewichtskapazität jeweils b.

Satz

Für BPP gibt es einen effizienten 2-Approximationsalgorithmus.

Der Begriff *effizienter Algorithmus* wird als Synonym für *Algorithmus mit polynomiell beschränkter Laufzeit* verwendet.

BPP – Approximationsalgorithmus

Algorithmus FIRST FIT:

Initial seien alle Kisten *ungeöffnet*.

Betrachte die Objekte in beliebiger Reihenfolge:

- Wenn das gerade betrachtete Objekt in keine der geöffneten Kisten eingefügt werden kann, ohne die Kapazität zu überschreiten, dann *öffne* eine neue Kiste für dieses Objekt.
- Ansonsten füge das Objekt in die erste bereits geöffnete Kiste ein, in die es passt.

Analyse des Approximationsfaktors:

Wieviele geöffnete Kisten gibt es, die weniger als halb gefüllt sind?

- FIRST FIT öffnet nur dann eine neue Kiste und fügt ein Objekt mit Gewicht kleiner $b/2$ in diese Kiste ein, wenn keine geöffnete Kiste mit Gewicht höchstens $b/2$ existiert.
- Es folgt, dass es zu keinem Zeitpunkt zwei geöffnete Kisten gibt, die Gewicht weniger als $b/2$ enthalten.

Wenn am Ende k Kisten geöffnet sind, gibt es somit mindestens $k-1$ Kisten, die mindestens halb voll sind.

Sei $W = \sum_{i=1^N} w_i$. Es folgt $W \geq \frac{k-1}{2} b$.

Aus der trivialen unteren Schranke $opt \geq W/b$ folgt nun $\frac{k-1}{2} \leq opt$ und somit $k \leq 2opt$.

$\square$

Satz

Für BPP gibt es keinen effizienten α-Approximationsalgorithmus mit $\alpha < \frac{3}{2}$; es sei denn $\mathsf{P} = \mathsf{NP}$*.*

Beweis:

Wir zeigen, dass aus einem effizienten α-Approximationsalgorithmus mit $\alpha < \frac{3}{2}$ für BPP ein effizienter Algorithmus für PARTITION abgeleitet werden kann.

Das NP-vollständige PARTITON-Problem wäre dann in P und daraus würde $\mathsf{P} = \mathsf{NP}$ folgen.

BPP – Untere Schranke für den Approximationsfaktor

Analog zum Konzept der polnomiellen Reduktion, erzeugen wir aus einer Eingabe $a_1, \ldots, a_N$ für PARTITION eine Eingabe für BPP, indem wir $w_1 = a_1, \ldots, w_N = a_N$ und $b = (\sum_{i=1}^{N} a_i)/2$ setzen.

Es gilt:

- Falls es sich um eine JA-Instanz von Partition handelt, gibt es eine Lösung für die BPP-Instanz mit $k = 2$ Kisten.
- Im Falle einer JA-Instanz würde ein α-Approximations-algorithmus für BPP somit eine Aufteilung auf zwei Kisten finden, weil ... eine Lösung mit drei oder mehr Kisten um mindestens den Faktor $\frac{3}{2} > \alpha$ vom Optimum abweichen würde.
- Falls es sich um eine NEIN-Instanz von Partition handelt, gibt es hingegen keine BPP-Lösung mit zwei Kisten.

Durch Aufruf des Approximationsalgorithmus für BPP kann man somit die JA- und NEIN-Instanzen von PARTITION unterscheiden.

$\square$

Nichtapproximierbarkeit vom allgemeinen TSP

Zur Erinnerung:

Beim Traveling Salesperson Problem (TSP) ist ein vollständiger Graph $G = (V, E)$ mit Kantenkosten $c(u, v) \in \mathbb{N}$ für $u, v \in V$ mit $c(u, v) = c(v, u)$ gegeben.

Gesucht ist eine Rundreise (ein *Hamiltonkreis*) mit kleinstmöglichen Kosten.

Satz

Sei $\alpha \geq 1$ beliebig gewählt. Für das TSP-Problem gibt es keinen α-Approximationsalgorithmus.

Beweis:

Wir zeigen, dass aus einem effizienten α-Approximations-algorithmus $\mathcal{A}$ für TSP ein effizienter Algorithmus $\mathcal{A}'$ für das Hamiltonkreisproblem (HC) abgeleitet werden könnte.

Algorithmus $\mathcal{A}'$:

- Sei $G' = (V, E')$ die Eingabe für HC.
- $\mathcal{A}'$ transformiert G' in einen gewichteten, vollständigen Graphen $G = (V, E)$, in dem nur die Kanten aus E' die Länge 1 haben, alle anderen Kanten in E erhalten die Länge αn.
- Dann ruft $\mathcal{A}'$ den Algorithmus $\mathcal{A}$ auf G auf.
- Falls $\mathcal{A}$ eine TSP-Tour der Länge höchstens αn findet, so gibt $\mathcal{A}'$ aus, dass G' einen Hamilton-Kreis enthält.
- Ansonsten meldet $\mathcal{A}'$, dass G' keinen Hamilton-Kreis enthält.

Korrektheit von $\mathcal{A}'$:

- Zuerst nehmen wir an, G' enthält einen Hamilton- Kreis. Jeder Hamilton-Kreis in G' entspricht einer TSP-Tour der Länge n in G. $\mathcal{A}$ findet somit eine Tour der Länge höchstens αn.
- Jetzt nehmen wir an, G' enthält keinen Hamilton-Kreis. Dann haben alle TSP-Touren in G die Länge mindestens $\alpha n + n - 1 > \alpha n$.

Damit kann $\mathcal{A}'$ mit Hilfe von $\mathcal{A}$ die JA- und NEIN-Instanzen von *HC* unterscheiden.

Aus der Existenz von $\mathcal{A}$ würde somit $\mathrm{P} = \mathrm{NP}$ folgen. $\square$

Metrisches TSP – Definition

Beim metrischen TSP ist ein vollständiger Graph $G = (V, E)$ mit Kantenkosten $c(u, v) \in \mathbb{N}$ für $u, v \in V$ gegeben, die für beliebige Knoten u, v, w die folgenden Bedingungen erfüllen:

- $c(u, u) = 0$
- $c(u, v) = c(v, u)$ (Symmetrie)
- $c(u, w) \leq c(u, v) + c(v, w)$ (Dreiecksungleichung)

Gesucht ist eine Rundreise (ein *Hamiltonkreis*) mit kleinstmöglichen Kosten.

Metrisches TSP – NP-Härte

Beobachtungen:

- Metrisches TSP ist ein Spezialfall des allgemeinen TSP.
- $\{1,2\}$-TSP erfüllt die Dreiecksungleichung und ist somit ein Spezialfall des metrischen TSP.
- Aus der NP-Härte von $\{1,2\}$-TSP folgt somit die NP-Härte vom metrischen TSP.

Satz

Für das metrische TSP gibt es einen effizienten 2-Approximations-algorithmus.

Beweis:

Algorithmus:

1. Finde einen minimalen Spannbaum T von G;
2. Verdopple die Kanten von T und erhalte dadurch den Euler-Graphen T';
3. Berechne eine Euler-Tour auf T';
4. Bereinige die Euler-Tour um wiederholt vorkommende Knoten.

Analyse des Approximationsfaktors:

- Aus einer TSP-Tour können wir einen Spannbaum erzeugen, indem wir eine Kante löschen.
- Also ist ein minimaler Spannbaum nicht teurer als die Länge einer minimalen TSP-Tour.
- Die Kosten der berechneten Euler-Tour entspricht den doppelten Kosten des minimalen Spannbaums, ist also höchstens zweimal so teuer wie die minimale TSP-Tour.
- Aufgrund der Dreiecksungleichung erhöht das Überspringen von mehrfach besuchten Knoten in Schritt 3 die Kosten nicht.

$\square$

Metrisches TSP – Bemerkungen

- Der beste bekannte Approximationsalgorithmus für metrisches TSP erreicht einen Approximationsfaktor von $\frac{3}{2}$.
- Dieser von Christofides im Jahr 1975 vogestellte Algorithmus wird in der Vorlesung Effiziente Algorithmen erklärt.
- Seit 1975 ist es ein offenes Problem diesen Approximationsfaktor zu schlagen!
- Im Jahr 2000 haben Papadimitriou und Vempala eine untere Schranke von $1 + 1/219$ für den Approximationsfaktor bewiesen.

Approximationsschema

Ein *Approximationsschema* ist ein Algorithmus, der es ermöglicht, für jedes vorgegebene $\epsilon > 0$ eine zulässige Lösung mit Approximationsfaktor $1 + \epsilon$ bzw. $1 - \epsilon$ zu berechnen.

Satz

Für das Rucksackproblem (KP) gibt es ein Approximationsschema.

Dieses Approximationsschema wird in der Vorlesung Effiziente Algorithmen vorgestellt.

Vorlesung Effiziente Algorithmen

Inhalt:

- Flüsse und Matchings
- Lineare Programmierung
- Approximationsalgorithmen
- Online-Algorithmen
- Randomisierte Algorithmen

Die Vorlesung wird im kommenden Sommersemester angeboten.