Polynomialzeit-Approximationsschema

27.01.2012

Elisabeth Sommerauer, Nicholas Höllermeier

Inhalt

1. NP-Vollständigkeit
   - Was ist NP-Vollständigkeit?
   - Die Klassen P und NP
   - Entscheidungsproblem vs. Optimierungsproblem
2. Polynomiale Zeit
   - Allgemeines
   - Polynomiale Laufzeit
   - Superpolynomiale Laufzeit

Inhalt

3. Approximationsschema
   - Optimierungsprobleme
   - Güte von Approximationsalgorithmen
     - Maximierungsproblem
     - Minimierungsproblem
   - Polynomialzeit Approximationsschema
   - Das Problem des Handlungsreisenden

1. NP-Vollständigkeit

Was ist NP-Vollständigkeit?

Problem lösbar in:

- Polynomiale Zeit

  behandelbar oder einfach

- Superpolynomiale Zeit

  unbehandelbar oder hart

Was ist NP-Vollständigkeit?

NP-Vollständige Probleme: Status unbekannt

- es wurde kein Algorithmus mit polynomialer Laufzeit entdeckt
- kein Beweis, dass kein Algorithmus mit polynomialer Laufzeit existieren kann

Die Klassen P und NP

Probleme der Klasse P:

- sind in polynomialer Zeit lösbar

$$O\left(n^k\right)$$

- k ist konstant, n ist die Eingabegröße

Die Klassen P und NP

Probleme der Klasse NP:

- sind in polynomialer Zeit verifizierbar:

  Lösung kann in polynomialer Zeit
  (n = Eingabegröße des Problems)
  überprüft werden

- Beispiel: Überprüfung gefundener Nullstellen eines Polynoms

Entscheidungsprobleme vs. Optimierungsprobleme

NP-Vollständigkeit ist auf Entscheidungsprobleme anwendbar

- Klasse P und NP bestehen aus Entscheidungsproblemen
- Lösung: Ja oder Nein

Entscheidungsprobleme vs. Optimierungsprobleme

Viele interessante Probleme sind Optimierungsprobleme

- jede zulässige Lösung hat bestimmten Wert
- optimales Ergebnis soll ermittelt werden:

  minimaler oder maximaler Wert

Entscheidungsprobleme vs. Optimierungsprobleme

- Einem Optimierungsproblem kann ein entsprechendes Entscheidungsproblem zugeordnet werden

- dem zu optimierendem Wert wird eine Schranke auferlegt

Entscheidungsprobleme vs. Optimierungsprobleme

Beispiel: Weg zwischen 2 Knoten eines Graphen

- Optimierungsproblem

  Was ist der kürzeste Weg?

- Entscheidungsproblem

  Existiert ein Weg der max. Länge x?

2.Polynomiale Zeit

Allgemeines zur polynomialen Zeit

In polynomialer Zeit lösbare Probleme werden im Allgemeinen als handhabbar betrachtet, aber:

$$O\left(n^{100}\right)$$

- ist unhandlich
- kommt jedoch in der Praxis äußerst selten vor

Allgemeines zur polynomialen Zeit

Die Erfahrung hat gezeigt:

- Laufzeit des derzeit besten Algorithmus

$$O\left(n^{100}\right)$$

- hohe Wahrscheinlichkeit, dass bald ein besserer Algorithmus gefunden wird

Allgemeines zur polynomialen Zeit

Probleme der Klasse P besitzen angenehme Abgeschlossenheitseigenschaften:

- Polynome sind abgeschlossen bezüglich Addition, Multiplikation und Komposition
- Probleme können zerlegt und wieder zusammengeführt werden

Allgemeines zur polynomialen Zeit

Algorithmus mit polynomialer Laufzeit

- als Eingabe eines anderen Algorithmus:

  Zusammensetzung ebenfalls polynomial

- konstant oft aufgerufene Unterroutine mit polynomialer Laufzeit:

  Gesamtalgorithmus ebenfalls polynomial

Polynomiale Laufzeit

n	20	100	1000
$5*n$	100	500	5.000
$n*\log(n)$	86	665	9.966
n^2	400	10.000	1.000.000
n^3	8.000	1.000.000	1.000.000.000

Superpolynomiale Laufzeit

n	20	100	1000
2^n	1.048.576	31 Stellen	302 Stellen
$n!$	19 Stellen	161 Stellen	unvorstellbar
n^n	27 Stellen	201 Stellen	unvorstellbar

Laufzeitbeispiele

Bei 10^9 Rechenoperationen pro Sekunde:

- 1000^3 etwa 1 Sekunde
- $20!$ etwa 80 Jahre
- 2^{100} etwa 3000 mal so lang wie das Universum alt ist

3.Approximationsschema

Approximationsalgorithmen

- Approximation im Sinne der Analysis
  - Satz von Weierstrass(1815-1897)

    Sei f eine stetige Funktion auf [a,b] .

    Dann gibt es zu jedem $\varepsilon > 0$ ein Polynom P_ε mit:

$$\max_{x \in [a,b]} \left| f(x) - P_\varepsilon(x) \right| < \varepsilon$$

Approximationsalgorithmen

- Für schwere Optimierungsprobleme
- Annäherung an polynomiellen Algorithmus durch Abschwächung der Anforderung

Optimierungsproblem Π

$$\Pi = \left(D, S, f, ziel \right)$$

D … Menge der Eingaben

$S\left(I\right)$ für $I \in D$ … Menge der zur Eingabe I zulässigen Lösungen

$f : S\left(I\right) \rightarrow \mathbb{N}$ … Bewertunsfunktion

$$ziel \in \left\{ max, min \right\}$$

- Algorithmus A ist ein Approximationsalgorithmus, wenn A für jede (Problem)Instanz I, eine gültige Lösung liefert.

Gütegarantie von Approximationsalgorithmen

- absolute Gütegarantie für Maximierungs- und Minimierungsprobleme

A hat absolute Gütegarantie k, $(k > 0)$,
wenn für jede Instanz I gilt:

$$|OPT(I) - A(I)| \leq k$$

Maximierungsproblem

- relative Gütegrantie k,wenn für jede Instanz gilt:

  $$A(I) \geqslant k \cdot OPT(I)$$

  ist ein *k*-Approximationsalgorithmus.
- relativer Abweichung ε

  $$\frac{OPT(I) - A(I)}{OPT(I)} \leqslant \varepsilon \Leftrightarrow A(I) \geqslant (1-\varepsilon) OPT(I)$$

  ist ein (1- ε) - Approximationsalgorithmus.

Minimierungsproblem

- relative Gütegarantie k ,wenn für jede Instanz gilt:

  $$A(I) \leqslant k \cdot OPT(I)$$

  ist ein *k*-Approximationsalgorithmus.
- relativer Abweichung ε

  $$\frac{A(I) - OPT(I)}{OPT(I)} \leqslant \varepsilon \Leftrightarrow A(I) \leqslant (1+\varepsilon) OPT(I)$$

  ist ein (1- ε) - Approximationsalgorithmus.

Ein scharfer Algorithmus? ;)

- Man bezeichnet einen Algorithmus als scharf wenn:
  - die Ungleichungen der Gütegarantie, für eine Probleminstanz, als Gleichheit erfüllt sein.
  - es eine Folge von Instanzen gibt,sodass die Gleichheit im Grenzwert gilt.
- Algorithmen ohne Güte bezeichnet man als Heuristiken (Daumenregel).

Polynomiales Approximationsschema

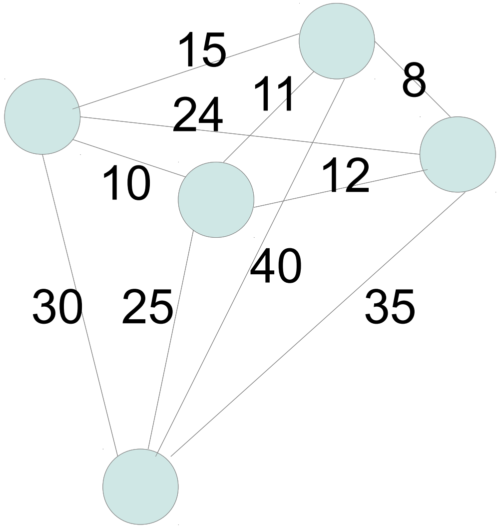
- Ein Approximationsschema A ist ein ...
  - Polynomiales Approxitmationsschema (PTAS), wenn die Laufzeit von A polynomial in der Länge der Instanz ist.
  - Voll-Polynomiales Approximationsschema (FTPAS), wenn die Laufzeit von A polynomial in der Länge der Instanz I und in $1/\varepsilon$ ist .

Das Problem des Handlungsreisenden

- Der Handlungsreisende besucht seine Geschäftspartner in verschiedenen Städten und holt jeweils einen Gegenstand(Ware)ab.
- Gesucht wird der kürzeste Weg dieser Rundreise.
- z.B.: Postbotenroutenplanung, Navigationssystem, Logistik, Mikrochips ...

Das Problem des Handlungsreisenden

- Ziel: möglichst kurzen Weg zurück legen und wieder zum Ausgangsort zurückkehren.
- Bekannt: Entfernung zwischen den Städten



Das Problem des Handlungsreisenden

- Konzept: Weglänge aller möglichen Wege vergleichen und den kürzesten Weg wählen
- Problem:
  - Anzahl der verschiedenen Wege bei n Städten:

$$(n-1)!$$

- Lösung: Heuristiken anwenden
  - nicht exakt kürzeste Route, nur Annäherung

Das Problem des Handlungsreisenden

- Bewiesen: Es handelt sich um eine Menge von Problemen mit exponentiellen Verhalten
- Klasse NP
- Ist bei kleiner Probleminstanz lösbar
- Es konnte noch nicht bewiesen werden, ob es ein Verfahren gibt, das in der Größe der Eingabe nicht exponentiell aufwendig wird

Quellen

- Theoretische Informatik (4.Auflage,2011); Juraj Hromkovič
- Algorithmische Konzepte der Informatik (2001); Juraj Hromkovič
- Approximationsalgorithmen (2006), Rolf Wanka
- Algorithmen – Eine Einführung (3. Auflage, 2009); Th. Cormen, Ch. Leiserson, R. Rivest, C. Stein

Vielen Dank für
eure Aufmerksamkeit