Kapitel 5

Komplexitätstheoretische Zwischenbetrachtungen: Klassen & eine Hierarchie

In den vorhergehenden Kapiteln sind wir einmal quer durch das Gebiet der Approximationsalgorithmen gelaufen. Wir haben die wesentlichen Begriffe für dieses interessante Gebiet eingeführt und anhand von Beispielen mit Leben erfüllt. Dabei haben wir festgestellt, daß wir verschiedene Optimierungsprobleme verschieden „gut" lösen können, und auch gesehen, daß bei der Lösung bestimmter Probleme bestimmte Qualitätskriterien gar nicht erreicht werden können.

Wir wollen den ersten Teil dieses Buches damit abschließen, daß wir im komplexitätstheoretischen Sinn etwas Ordnung in die untersuchten Probleme bringen.

Wir haben inzwischen sehr oft von den Klassen P und NP gesprochen. Die analogen Klassen für Optimierungsprobleme und eine sogar feinere Unterteilung werden jetzt eingeführt.

Wir sagen im folgenden – benutzt haben wir diese Formulierung bereits öfters, aber nun brauchen wir sie genau – abkürzend, daß etwas in Polynomzeit in $|I|$ berechnet werden kann, wenn es ein Polynom q gibt, so daß *für alle* gemeinten I die Laufzeit höchstens $q(|I|)$ ist. Wichtig ist die Reihenfolge der implizierten Quantoren!

5.1 Definition:
Ein Optimierungsproblem Π liegt in der Sprachklasse NPO, wenn gilt:

(a) (1) Instanzen I können in Polynomzeit in $|I|$ als solche erkannt werden, d. h. die Frage „$I \in \mathcal{D}$?" kann in Polynomzeit in $|I|$ entschieden werden. Wir lassen also keine „unsinnigen" Eingaben zu.

(2) Es gibt ein Polynom p, so daß bei Instanz I gilt:

(a) Für jede zulässige Lösung $\sigma \in S(I)$ ist $|\sigma| \leq p(|I|)$;
(b) Für jede Zeichenfolge y mit $|y| \leq p(|I|)$ kann in Polynomzeit in $|I|$ entschieden werden, ob $y \in S(I)$ ist.

(3) Bei Instanz I kann zu jeder zulässigen Lösung $\sigma \in S(I)$ der Wert $f(\sigma)$ in Polynomzeit in $|I|$ berechnet werden.

oder

(b) Es gibt ein Optimierungsproblem $\Pi' \in$ NPO mit $\Pi \leq_p \Pi'$,

Diese Definition kommt wie unsere Einführung der Klasse NP auf Seite 6 ohne nichtdetermi-
Ü. 5.1 nistische Turing-Maschinen aus. Punkt (a2) übernimmt hier die Aufgabe des Zertifikats. Zur Rolle von (b) siehe Übung 5.1(b). Alle bislang vorgestellen Optimierungsprobleme liegen in NPO.

Wir haben schon oft den Zusammenhang zwischen Optimierungs- und Entscheidungsproblemen betont. Wir definieren das jetzt formal.

5.2 Definition:
Sei $\Pi = (\mathcal{D}, S, f, \mathsf{ziel})$ ein Optimierungsproblem gemäß Definition 1.2. Dann ist das *zugehörige Entscheidungsproblem* die Menge

$$\Pi_{\text{ent}} = \begin{cases} \{\langle I,k\rangle \mid I \in \mathcal{D},\ k \in \mathbb{N}, \exists \sigma \in S(I) : f(\sigma) \leq k\} & \text{falls } \mathsf{ziel} = \min \\ \{\langle I,k\rangle \mid I \in \mathcal{D},\ k \in \mathbb{N}, \exists \sigma \in S(I) : f(\sigma) \geq k\} & \text{sonst.} \end{cases}$$

Jedes Optimierungsproblem Π läßt sich also in ein Entscheidungsproblem Π_{ent} verwandeln, indem wir zu jeder Instanz I eine Zahl $k \in \mathbb{N}$ hinzufügen und fragen, ob es eine zulässige Lösung σ gibt mit $f(\sigma) \leq k$, falls Π ein Minimierungsproblem ist, und $f(\sigma) \geq k$, falls Π ein
Ü. 5.3 Maximierungsproblem ist.

Ü. 5.2 Der nachstehende Satz folgt direkt aus Punkt (a2) der Definition.

5.3 Satz:
Ist $\Pi \in$ NPO, *dann ist* $\Pi_{\text{ent}} \in$ NP.

Die zu P analoge Klasse bekommen wir direkt dadurch, daß wir fordern, eine optimale Lösung in Polynomzeit berechnen zu können.

5.4 Definition:
Das Optimierungsproblem Π liegt in der Sprachklasse PO, wenn $\Pi \in$ NPO ist und zu jeder

Instanz I in polynomieller Zeit in $|I|$ eine optimale Lösung berechnet werden kann.

Die Berechnung kürzester Wege, die Bestimmung minimaler Spannbäume (MST), die Berechnung maximaler Flüsse (MAXFLOW) und die Lösung linearer Optimierungsprobleme seien als nichttriviale Beispiele für Sprachen aus PO angeführt.

Aus der Definition folgt PO $\subseteq$ NPO direkt. Aber ebenso, wie für P und NP, ist unbekannt, ob PO $\subsetneq$ NPO ist. Die zweite Frage ist natürlich mit der ersten eng verwandt, wie folgender Satz zeigt.

5.5 Satz:
PO $=$ NPO $\Rightarrow$ P $=$ NP.

Beweis:
Wir wissen, daß CLIQUE $\in$ NPO ist. Aus PO $=$ NPO folgt, daß es einen Polynomzeit-Algorithmus gibt, der zu jedem Eingabegraphen einen vollständigen Teilgraphen maximaler Größe berechnet. Wir können also das zugehörige, NP-vollständige Entscheidungsproblem CLIQUE$_{\text{ent}}$ in Polynomzeit lösen. Aus der NP-Vollständigkeit von CLIQUE$_{\text{ent}}$ folgt damit P $=$ NP. □

In unseren Untersuchungen haben wir aber inzwischen schon sehr viel mehr herausgefunden. So haben wir gesehen, daß das TSP kein polynomielles Approximationsverfahren mit konstanter relativer Güte zuläßt (Satz 3.16), dagegen das ΔTSP mit Christofides' Algorithmus der relativen Güte $3/2$ sehr wohl (Satz 3.8). Insofern erlaubt die relative Güte eine feinere Strukturierung von NPO.

5.6 Definition:
Sei F eine Menge von Funktionen. Ein Optimierungsproblem $\Pi \in$ NPO liegt in der Sprachklasse F-APX, wenn es einen polynomiellen Approximationsalgorithmus A gibt mit relativer Güte $\rho_A(n) \in F$.

Die für uns wichtigste Klasse in diesem Zusammenhang ist die, wenn F die Menge der konstanten Funktionen ist. Diese Klasse wird dann auch einfach nur APX genannt. Dann haben wir: ΔTSP $\in$ APX und TSP $\notin$ APX.

Mit log-APX bezeichnet man die Klasse der Probleme, die durch Algorithmen mit relativer Güte $\rho(n) = O(\log n)$ approximiert werden können. In diese Klasse fällt das Problem SETCOVER, das wir aus Übung 2.12 kennen. Einen entsprechenden Approximationsalgorithmus für SETCOVER werden wir in Abschnitt 7.5.3 kennenlernen.

Die Approximationsschemata ergeben eine weitere Unterteilung von APX.

5.7 Definition:
Ein Optimierungsproblem $\Pi \in$ NPO liegt in der Sprachklasse PAS bzw. FPAS, wenn es für Π ein PAS bzw. ein FPAS gibt.

In Übung 3.11 haben wir gesehen, daß es kein polynomielles Approximationsverfahren für
Ü. 5.4 BINPACKING der Güte $\rho < \frac{3}{2}$ gibt, während es ziemlich einfach ist, für BINPACKING ein Verfahren der relativen Güte 2 anzugeben. D. h. BINPACKING $\in$ APX, aber BINPACKING $\notin$ PAS.

Da ΔTSP stark NP-vollständig ist, wissen wir, daß ΔTSP $\notin$ FPAS ist, während RUCKSACK $\in$ FPAS ist. Wir haben am Ende von Kapitel 4 erwähnt, daß es für das Euklidische TSP ein PAS gibt, aber kein FPAS gibt.

5.8 Satz:

$$\begin{array}{ccccccccc}
\text{PO} & \subseteq & \text{FPAS} & \subseteq & \text{PAS} & \subseteq & \text{APX} & \subseteq & \text{NPO} \\
\cup\!\! \shortmid & & \cup\!\! \shortmid & & \cup\!\! \shortmid & & \cup\!\! \shortmid & & \cup\!\! \shortmid \\
\text{MST} & & \text{RUCKSACK} & & \text{EuklidTSP} & & \text{BINPACKING} & & \text{TSP} \\
\text{MAXFLOW} & & & & & & \Delta\text{TSP} & &
\end{array}$$

Ist P $\neq$ NP, *sind alle Inklusionen echt und die genannten Probleme nicht in den Klassen nach links enthalten.*

In Kapitel 2 haben wir die absolute Güte κ eingeführt und gesehen, daß das Kantenfärbungsproblem mit $\kappa = 1$ approximiert werden kann, während RUCKSACK nicht mit konstanter absoluter Güte gelöst werden kann. Haben wir, da RUCKSACK $\in$ FPAS ist, eine Unterteilung von FPAS? Leider nicht, wie folgende Überlegung zeigt.

Bezeichne ABS die Klasse der Optimierungsprobleme, die mit konstanter absoluter Güte ap-
Ü. 5.5 proximiert werden können. Eine einfache Überlegung erbringt, daß ABS $\subseteq$ APX ist.

Die symmetrische Differenz von PAS und ABS ist nicht leer, d. h. es gibt Probleme in ABS, die nicht in PAS liegen, und umgekehrt: (a) Das Kantenfärbungsproblem liegt in ABS (Satz 2.11), aber ist es ist nicht in PAS (Satz 4.8). (b) RUCKSACK ist in FPAS (Satz 4.7) und damit auch in PAS, aber es liegt nicht in ABS (Satz 2.12). Eine Problemklasse, die PAS $\cup$ ABS enthält, aber
Ü. 5.6 unterhalb von APX liegt, wird in den Übungen untersucht.

Damit haben wir insgesamt die in Abbildung 5.1 dargestellte Hierarchie bewiesen.

In Tabelle 5.1 stellen wir die in Teil I herausgefundenen Gütegarantien und unteren Schranken zusammen (Achtung: Dies sind nicht immer die besten zur Zeit bekannten Resultate überhaupt).

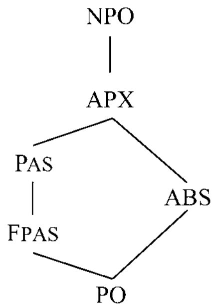

Abbildung 5.1: Hierarchie, falls P $\neq$ NP.

Tabelle 5.1: In Teil I *bewiesene* Gütequalitäten für Probleme

Problem	relative Güte		absolute Güte	
	$\leq$	$>$	$\leq$	$>$
RUCKSACK	FPAS		$O(2^{\vert I \vert})$	jede Konst.
SETCOVER				jede Konst.
Kantenfärbung	$4/3$	$4/3-\varepsilon$	1	0
Knotenfärbung	$O(\vert V \vert)$	$4/3-\varepsilon$	$O(\vert V \vert)$	
Knotenfärbung planar	$3/2$	$4/3-\varepsilon$	3	0
TSP	$O(2^{\vert I \vert})$	jede Konst.		
ΔTSP	$3/2$	kein FPAS		
BINPACKING	2	$3/2-\varepsilon$		
CLIQUE	$O(\vert V \vert)$	jede Konst.		

5.1 Literatur zu Kapitel 5

[ACG+99] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation – Combinatorial Optimization Problems and Their Approximability Properties*. Springer, Berlin, 1999.

[Mor98] B. M. Moret. *The Theory of Computation*. Addison-Wesley, Reading, MA, 1998.

Übungen zu Kapitel 5

Aufgabe 5.1
Zeige, daß die folgenden Probleme in NPO sind.

(a) TSP

(b) BESTERNICHTOPTRUCKSACK, wobei eine Instanz des Rucksackproblems gegeben ist und nach einer besten nichtoptimalen Rucksackfüllung gesucht wird.

Aufgabe 5.2
Beweise Satz 5.3, d. h. zeige: Ist $\Pi \in$ NPO, dann ist $\Pi_{\mathrm{ent}} \in$ NP.

Aufgabe 5.3
Sei CLIQUEGIGANTE das Maximierungsproblem, in dem alles identisch zu CLIQUE ist bis auf die Wertefunktion. Bei einer Instanz G sind die zulässigen Lösungen vollständige Teilgraphen H von G. Der Wert von H ist

$$f(H) = 2^{2^{|\langle G\rangle|}} + |V(H)| \quad .$$

In welcher Sprachklasse liegt CLIQUEGIGANTE$_{\mathrm{ent}}$?

Zur Erinnerung: $V(H)$ ist die Knotenmenge von H. (vgl. Übung 3.10).

Aufgabe 5.4
Zeige, daß der folgende Algorithmus NEXTFIT das Optimierungsproblem BINPACKING (vgl. Aufgabe 3.11 auf S. 60) mit relativer Güte 2 löst.

ALGORITHMUS NEXTFIT

$k := 1$; $B_1 := \emptyset$;
for $i := 1$ **to** n **do**
  if $s(a_i) + s(B_k) > 1$ **then** $k := k+1$; $B_k := \emptyset$ **fi**;
  $B_k := B_k \cup \{a_i\}$
done;
gib $(B_1, \ldots, B_k)$ aus.

Hinweis: Begründe, warum (a) $\mathrm{OPT}(I) \geq \lceil s(M) \rceil$ und (b) $\mathrm{NEXTFIT}(I) \leq 2\lceil s(M) \rceil$ ist.

Aufgabe 5.5
Zeige, daß ABS $\subseteq$ APX ist.

Aufgabe 5.6
Sei Π ein kombinatorisches Optimierungsproblem. Ein *asymptotisches polynomielles Approximationsschema* (APAS) ist ein Approximationsalgorithmus A für Π, der als Eingabe eine Instanz I und ein $\varepsilon \in]0,1[$ bekommt und für den es eine Konstante k gibt, so daß $\rho_A(I,\varepsilon) \leq 1+\varepsilon+\frac{k}{\mathrm{OPT}(I)}$ für alle I und die Laufzeit ein Polynom in $|I|$ ist.
Die Klasse PAS^∞ ist die Menge aller Optimierungsprobleme, für die es ein APAS gibt.

(a) Zeige: $\mathrm{ABS} \subseteq \mathrm{PAS}^\infty \subseteq \mathrm{APX}$.

(b) Zeige: $\mathrm{PAS}^\infty \subsetneq \mathrm{APX}$, falls $\mathrm{P} \neq \mathrm{NP}$.

(c) Zeichne die um PAS^∞ erweiterte Hierarchie entsprechend Abbildung 5.1.