E 21496 F

DBW

DIE BETRIEBSWIRTSCHAFT

Fachzeitschrift herausgegeben von
Klaus Chmielewicz, Bochum, Adolf G. Coenenberg, Augsburg
Richard Köhler, Köln, Heribert Meffert, Münster
Gerhard Reber, Linz und Norbert Szyperski, Köln

2/88 APRIL/MAI

Aus dem Inhalt

C. E. Poeschel Verlag Stuttgart

Stephan Zelewski

Komplexitätstheorie – ihre Anwendung auf OR-Probleme, insbesondere auf die lineare Optimierung und die Petrinetz-Theorie

Karmarkar-/Khachiyan-Algorithmus; Komplexitätstheorie; Lineare Optimierung; NP-vollständige Probleme; Petrinetze

Im Rahmen der Komplexitätstheorie wird versucht, die Komplexität von Problemen durch den Ressourcenverzehr zu messen, der durch die Problemlösung im schlechtest denkmöglichen Fall oder durchschnittlich verursacht wird. Als wesentliche Analysekonzepte der Komplexitätstheorie werden Entscheidungsprobleme und Turing-Automaten eingeführt. Auf ihrer Grundlage lassen sich Komplexitätsklassen von Problemen bilden, aus denen wiederum Empfehlungen hinsichtlich erfolgversprechender Lösungsalgorithmen abgeleitet werden können. Einen Schwerpunkt bildet die Klasse der NP-vollständigen Probleme, deren Lösung besonders aufwendig ist. Auf neuere Erkenntnisse, die diese Problemklasse intern differenzieren und über sie hinausführen, wird näher eingegangen. Die Bedeutung komplexitätstheoretischer Analysen wird aus der Perspektive des Operations Research an einem Beispiel aus der Petrinetz-Theorie und aktuellen Entwicklungen auf dem Gebiet der linearen Optimierung (Khachiyan- und Karmakar-Algorithmus) verdeutlicht. Einschränkungen solcher Analysen werden anhand mehrfacher Validitätsprobleme aufgezeigt.

Kontaktadresse: Dr. Stephan Zelewski, Seminar für Allgemeine Betriebswirtschaftslehre, Industriebetriebslehre und Produktionswirtschaft (»Industrieseminar«), Universität Köln, Albertus-Magnus-Platz, 5000 Köln 41.
Köln 1987. Umfang 86 Seiten. Fotokopie DM 20,— (zu beziehen bei der Kontaktadresse); Bestell-Nr. DBW-Depot-88-2-5.

Arbeitsbericht Nr. 14

Komplexitätstheorie
- ihre Anwendung auf OR-Probleme, insbesondere auf die lineare Optimierung und die Petrinetz-Theorie -

von

Dr. Stephan Zelewski

3. Auflage des Arbeitsberichts 14/1986

2. Aufl. 1986 unter dem Titel
"Komplexitätstheorie - ihr Beitrag zur
Klassifizierung und Beurteilung von
Problemen des Operations Research -"

Köln 1986/87

Abstract

Im Rahmen der Komplexitätstheorie wird versucht, die Komplexität von Problemen durch den Ressourcenverzehr zu messen, der durch die Problemlösung im schlechtest denkmöglichen Fall oder durchschnittlich verursacht wird. Als wesentliche Analysekonzepte der Komplexitätstheorie werden Entscheidungsprobleme und Turing-Automaten eingeführt. Auf ihrer Grundlage lassen sich Komplexitätsklassen von Problemen bilden, aus denen wiederum Empfehlungen hinsichtlich erfolgversprechender Lösungsalgorithmen abgeleitet werden können. Einen Schwerpunkt bildet die Klasse der NP-vollständigen Probleme, deren Lösung besonders aufwendig ist. Auf neuere Erkenntnisse, die diese Problemklasse intern differenzieren und über sie hinausführen, wird näher eingegangen. Die Bedeutung komplexitätstheoretischer Analysen wird aus der Perspektive des Operations Research an einem Beispiel aus der Petrinetz-Theorie und aktuellen Entwicklungen auf dem Gebiet der linearen Optimierung (Khachiyan- und Karmakar-Algorithmus) verdeutlicht. Einschränkungen solcher Analysen werden anhand mehrfacher Validitätsprobleme aufgezeigt.

Inhaltsübersicht

1 Einführung in den aufwandsbezogenen Komplexitätsbegriff

Die Komplexität von Problemen kann - ausgehend von einer gemeinsamen systemtheoretischen Betrachtungsweise[1)] - auf zwei grundsätzlich verschiedenen Ebenen untersucht werden. Auf der ersten Ebene wird die Struktur von Systemen analysiert, auf der zweiten dagegen ein spezieller Aspekt des Systemverhaltens.

Der strukturorientierte Ansatz geht davon aus, daß Probleme, die es in der Realität zu lösen gilt (Realprobleme), auf Modelle abgebildet werden. Diese Modelle werden formalsprachlich als Systeme beschrieben (Ideal- oder Formalprobleme). Die Komplexität eines modellierten Problems läßt sich durch Größen messen, die auf die Struktur des Systems Bezug nehmen[2)]. Sie stellen Indikatoren[3)] zur Operationalisierung eines intuitiven Komplexitätsverständnisses[4)] dar, das Komplexität mit

1) Vgl. zu abweichenden, hier nicht weiter berührten Komplexitätskonzepten z.B. Savage (1976), S. 9ff.

2) Vgl. Luhmann (1980), Sp. 1064.

3) Beispielsweise dienen als Indikatoren der Modellkomplexität: die Anzahl der elementaren Modellkomponenten - z.B. die Konstanten und Variablen eines OR-Modells - (Elemente-Varietät), die Anzahl und Art der Attribute, die diesen Komponenten jeweils zukommen (Attribute-Variatät), die Anzahl und Art der Relationen, die zwischen diesen Komponenten definiert sind (Konnektivität), die Anzahl und Art von übergeordneten Komponenten- (z.B. Variablen mit charakteristischen Definitionsbereichen) und Relationengruppen (z.B. Relationen unterschiedlicher Stelligkeit oder Funktionen verschiedenen Grades) oder die Anzahl und Art möglicher Systemzustände (Variabilität). Neben die Deskriptoren "Anzahl" und "Art" können auch die Unschärfe, Unvollständigkeit oder Widersprüchlichkeit der vorgenannten Systemaspekte als Komplexitätsindikatoren treten. Vgl. hierzu Ulrich (1970), S. 116f.; Beensen (1971), S. 11f.; Kawamura (1977), S. 347; Pfohl (1977), S. 254; Kirsch (1978), S. 142f.; Luhmann (1980), Sp. 1065; Szyperski (1983), S. 4f.; Ernst (1984), S. 16.

4) Vgl. zur Schwierigkeit, das intuitive Komplexitätsverständnis zu operationalisieren, Luhmann (1980), Sp. 1065, der so weit geht, Komplexität als einen Reflexionsbegriff zu bezeichnen. In einem solchen Begriff werde "... reflektiert (oder mangels Reflexion unbestimmt gelassen), daß der ausgearbeitete Begriff selbst zu komplex (!, Anmk. des Verf.) wird für forschungsmäßige Verwendung."

quantitativer und qualitativer Vielfalt[5] assoziiert. Dieser vielfaltsbezogene, strukturorientierte Komplexitätsbegriff herrscht in Beiträgen vor, die sich mit der konstruktiven Gestaltung von Systemen und Modellen befassen[6].

Das verhaltensorientierte Konzept setzt die Abbildung von Realproblemen auf Modelle voraus. Es bezieht den Komplexitätsbegriff jedoch auf ein erweitertes Systemkonzept, das sowohl das problemabbildende Modell als auch die Klasse der Algorithmen umfaßt, die zur Lösung des Modells angewendet werden können. Gegenstand der Komplexitätsanalyse ist das Systemverhalten beim Versuch der Modellösung. Die Komplexität eines modellierten Problems wird durch den Ressourcenaufwand gemessen, der zum Auffinden der gesuchten Lösungen oder zum Nachweis ihrer Nichtexistenz erforderlich ist[7]. Mit diesem Komplexitätsbegriff wird die intuitive Auffassung operationalisiert, die Komplexität eines Problems sei mit der Schwierigkeit seiner Lösung verknüpft[8].

Der aufwandsbezogene, verhaltensorientierte Komplexitätsbegriff liegt der Komplexitätstheorie[9] zugrun-

---

5) Vgl. annähernd Simon (1962), S. 468.
6) Vgl. z.B. Ulrich (1970), S. 116f.; Pfohl (1977), S. 254; Kirsch (1978), S. 142f.
7) Vgl. Parker (1982a), S. 4.
8) Der Prozeß der Problembewältigung, der entweder die gesuchten (Modell-)Lösungen generiert oder deren Nichtexistenz aufzeigt, wird fortan vereinfachend als Problemlösung angesprochen.
9) Vgl. als Einführungen in die Komplexitätstheorie Cook (1971), S. 151ff.; Karp (1972), S. 85ff.; Karp (1975b), S. 45ff.; Lenstra (1977), S. 343ff.; Paul (1978), 181ff.; Garey (1979), S. 6ff.; Bachem (1980), S. 813ff.; Brucker (1981), S. 26ff.; Parker (1982a), S. 3ff.; Parker (1982b), S. 83ff.; Cook (1983), S. 401ff.; Karp (1986), S. 98ff., insbesondere S. 100f.; sowie die annotierten Bibliographien im Sammelband O'hEigeartaigh (1985) und - als Überblick über aktuelle Forschungsschwerpunkte - die Beiträge in dem Sammelwerk Selman (1986).

de[10], die seit Beginn der sechziger Jahre als selbständige Erkenntnisrichtung etabliert wurde[11]. Die nachfolgenden Ausführungen erstrecken sich nur noch auf dieses spezielle Komplexitätsverständnis. Diese Einschränkung erfolgt aus der Perspektive einer gestaltungsorientierten Ausrichtung des Operations Research (OR)[12]. Sie verfolgt als primäre Intention bei der Auseinandersetzung mit realen Problemen nicht deren Abbildung, sonderen deren Lösung[13]. Der strukturorientierte Komplexitätsbegriff genügt dieser Sichtweise nicht, weil oftmals kein klarer Zusammenhang zwischen der Vielfalt einer Modellstruktur und dem Aufwand für die Modellösung besteht. Als ein extremes Beispiel sei das Vierfarbenproblem angeführt, das auf einfache Weise modelliert werden kann, dessen Lösung aber erst Mitte der siebziger Jahre mit erheblichem Computereinsatz -

---

10) Es könnte eingewendet werden, die Komplexitätstheorie befasse sich nicht mit der Komplexität von Problemen, sondern mit der von Algorithmen. Ein solches Argument ließe sich auf die nachfolgenden Ausführungen zur algorithmischen Problemlösung und auf die Anmerkungen - vgl. Abschnitt 2.5 - zur Validitätsproblematik der Komplexitätstheorie stützen. Dieser Einwand ist jedoch nicht stichhaltig, da in der einschlägigen Literatur explizit die Problemkomplexität als Erkenntnisziel herausgestellt wird; vgl. z.B. Rinnooy Kan (1976), S. 139 ("... we usually examine the complexity of the problem ..."); Parker (1982a), S. 4 ("... complexity theory seeks to classify problems in terms ... of the computational resources required to solve the problems") u. S. 7 ("... complexity theory seeks to classify problems, not algorithms").

11) Vgl. Cook (1983), S. 401; Hopcroft (1984), S. 47; Karp (1986), S. 101.

12) Vgl. als Einführungen in die Komplexitätstheorie, die speziell Interessenten aus dem Bereich des Operations Research gewidmet sind, Bachem (1980), S. 812ff., und Lenstra (1982), S. 201ff. Daß das Komplexitätsverständnis der Komplexitätstheorie zunehmende Beachtung findet, wird auch aus den Beiträgen von Simon (1976), S. 282ff., und Goldberg (1984), S. 43 u. 46ff., deutlich, die komplexitätstheoretische Erkenntnisse aus der Sicht der Künstlichen Intelligenz-Forschung reflektieren.

13) Hiermit wird nicht die große Bedeutung bestritten, welche der Abbildung von Realproblemen auf Idealprobleme durch den Prozeß der Modellkonstruktion zukommt. Dieser Prozeß wird hier aber nur als Hilfsmittel, als Vorstufe zur intendierten Problemlösung verstanden.

1.200 Stunden Operationszeit auf drei Großrechnern - gelang[14).

2 Konzeptionelle Grundlagen der Komplexitätstheorie

2.1 Rückführung der Komplexitätsbetrachtungen auf Entscheidungsprobleme

Ein Entscheidungsproblem[15) ist ein Problem, das zu seiner Lösung die Auswahl zwischen zwei kontradiktorischen Alternativen erfordert. Es wird dem Entscheidungsträger eine Frage unterbreitet, die dieser nur durch ein "Ja" oder "Nein" zu beantworten vermag.

Trotz seiner einfachen Struktur reicht das Konzept der Entscheidungsprobleme aus, um - mittelbar - auch alle Optimierungsprobleme des Operations Research zu umfassen. Wenn ein Optimierungsproblem als Satisfizierungsproblem gegeben ist, wird im korrespondierenden Entscheidungsproblem nach der Existenz einer zulässigen Lösung für das ursprüngliche Satisfizierungsproblem gefragt.

Falls ein Extremierungsproblem vorliegt[16), ist zunächst - unter Vernachlässigung der zu maximierenden oder zu minimierenden Zielfunktion - das Entscheidungsproblem zu lösen, ob für das verbleibende Rumpfproblem mindestens eine zulässige Lösung existiert. Wenn keine solche Lösung existiert, ist das ursprüngliche Optimierungsproblem durch Nachweis seiner Unlösbarkeit bereits bewältigt. Für den Fall, daß eine zulässige Lösung an-

---

14) Vgl. Appel (1977), S. 108ff.; Stockmeyer (1979), S. 90.

15) Vgl. zum Begriff des Entscheidungsproblems Davis (1958), S. 69; Ecker (1977), S. 242; Paul (1978), S. 56; Garey (1978), S. 501; Garey (1979), S. 18; Bachem (1980), S. 817f.; Parker (1982a), S. 8.

16) Vgl. zum folgenden Brucker (1975), S. 4; Ullman (1976), S. 140f.; Ecker (1977), S. 248ff.; Lenstra (1977), S. 344; Lenstra (1978), S. 23; Garey (1979), S. 19 u. 115ff.; Lenstra (1979), S. 124; Bachem (1980), S. 818f.; Lenstra (1982), S. 202.

gegeben werden kann[17], wird der Zielfunktionswert für diese Lösung ermittelt und als Restriktion dem ursprünglichen Rumpfproblem hinzugefügt. Nach dem Konzept der parametrischen Programmierung wird der Zielfunktionswert dieser ergänzten Restriktion sukzessiv erhöht oder erniedrigt, je nachdem ob das zugrundeliegende Extremierungsziel zum Maximierungs- bzw. Minimierungstyp zählt. Für jedes derart modifizierte Rumpfproblem wird das Entscheidungsproblem untersucht, ob mindestens eine zulässige Lösung existiert. Die Konstruktion parametrisch modifizierter Rumpfprobleme und die Beantwortung der zugehörigen Entscheidungsprobleme bricht ab, sobald nachgewiesen wird, daß keine zulässigen Lösungen existieren. Die zuletzt erzeugten zulässigen Lösungen mit bestmöglicher Erfüllung der zugrundeliegenden Zielfunktion stellen dann die gesuchten optimalen Lösungen dar.

17) Es wird unterstellt, daß der Nachweis der Existenz zulässiger Lösungen konstruktiv erfolgt, so daß aus dem Nachweisverfahren die konkrete Gestalt einer solchen Lösung direkt abgelesen oder zumindest indirekt rekonstruiert werden kann. Daß diese Prämisse für Optimierungsmodelle des Operations Research in der Regel erfüllt ist, geht aus den Ausführungen von Valiant (1978), S. 330, in Verbindung mit den Anmerkungen im Abschnitt 3.1 dieser Arbeit hervor.

Infolge der Möglichkeit, jedes Optimierungs- in ein lösungsäquivalentes Entscheidungsproblem zu transformieren[18], wird fortan nur noch auf Entscheidungspro-

18) Es ist allerdings die Besonderheit zu beachten, daß sich die Klassifizierung der Problemkomplexität nicht immer invariant gegenüber der Transformation von einem Optimierungs- in ein Entscheidungsproblem verhält. Eine mögliche Invarianzverletzung wird von Lenstra (1979), S. 124, und Clausen (1986), S. 33, für die Klasse der NP-vollständigen Probleme aufgezeigt (Näheres zu solchen Problemen auf S. 35ff.). Die Eigenschaft der NP-Vollständigkeit kann nur für Entscheidungsprobleme nachgewiesen werden, da nur für deren Analyse gesichert ist, daß das Konzept der nondeterministischen Turing-Automaten den Lösungsaufwand korrekt wiedergibt. Das Optimierungsproblem, das in ein solches NP-vollständiges Entscheidungsproblem transformiert wurde, muß dagegen nicht NP-vollständig sein, noch nicht einmal in der - weiter definierten - Klasse NP liegen (abweichender Ansicht ist French (1982), S. 146). Es ist noch ungelöst, ob zur (polynomial beschränkten) Bewältigung eines Optimierungsproblems das Konzept der nondeterministischen Turing-Automaten ausreicht. Daher werden Optimierungsprobleme, deren zugehörigen Entscheidungsprobleme NP-vollständig sind, nur als NP-hart bezeichnet. Dies bedeutet, daß solche Optimierungsprobleme mindestens so schwierig zu lösen sind wie alle anderen Probleme aus der Klasse NP, vielleicht aber auch noch komplexer ausfallen. Vgl. zu diesem Verständnis NP-harter Optimierungsprobleme Lenstra (1979), S. 124; Garey (1979), S. 114ff.; French (1982), S. 148f.; Lenstra (1982), S. 202; Parker (1982b), S. 84, und die Ausführungen zu NP-harten Problemen auf S. 34f. Dieser Sachverhalt läßt sich dahingehend verallgemeinern, daß aus der nachgewiesenen Komplexität eines Entscheidungsproblems nur gefolgert werden kann, daß das assoziierte Optimierungsproblem mindestens ebenso komplex sein muß, u.U. aber noch komplexer sein kann; vgl. Garey (1979), S. 19.

bleme eingegangen[19]. Der Transformationsaufwand kann hinsichtlich der nachfolgenden komplexitätstheoretischen Analysen vernachlässigt werden[20]. Da Algorithmen zur Lösung von Entscheidungsproblemen nicht für Unikate definiert sind, sondern sich auf die Lösung von Problemklassen erstrecken, ist der Begriff des Entscheidungsproblems stets als eine abstrakte Bezeichnung für eine Klasse von ähnlichen konkreten Entscheidungsproblemen zu verstehen, die Ausprägungen des abstrakten Problems darstellen.

---

19) Trotz der Vorbehalte, die in der voranstehenden Fußnote geäußert wurden, kann für eine große Anzahl von Optimierungsproblemen, deren zugehörigen Entscheidungsprobleme NP-vollständig sind, gezeigt werden, daß sie sich auf Probleme aus der Klasse NP reduzieren lassen (vgl. zum Reduzierungsbegriff S. 34f.). Dann können diese Optimierungsprobleme nicht komplexer als NP-vollständige Probleme sein und werden als NP-leichte Probleme bezeichnet. Da solche Optimierungsprobleme - wie oben ausgeführt - zugleich NP-hart, also mindestens so komplex wie ihre assoziierten Entscheidungsprobleme sind, müssen sie notwendig genau so komplex wie NP-vollständige Probleme sein. Diese sowohl NP-harten als auch NP-leichten, als NP-äquivalent bezeichneten Optimierungsprobleme besitzen also die gleiche Komplexität wie die zugrundeliegenden NP-vollständigen Entscheidungsprobleme. Unter dieser - oftmals erfüllten - Voraussetzung können trotz der o.a. Vorbehalte Optimierungsprobleme ohne Verzerrung der Klassifizierung ihrer Komplexität durch Entscheidungsprobleme ersetzt werden. Vgl. zu den voranstehenden Darlegungen Garey (1979), S. 117.; vgl. ansatzweise auch Garey (1978), S. 505; Parker (1982b), S. 84f.

20) Wenn ein Entscheidungsproblem gelöst werden kann, auf das ein Optimierungsproblem zurückgeführt wurde, so ist es möglich, das Lösungsverfahren so zu transformieren, daß das zugehörige Optimierungsproblem ebenfalls gelöst wird. Es ließ sich nachweisen, daß der Lösungsaufwand des Optimierungsproblems nur polynomial beschränkt größer als der für das zugrundeliegende Entscheidungsproblem ist; vgl. Ecker (1977), S. 249. Wie unten näher ausgeführt wird, gelten seitens der Komplexitätstheorie Probleme als gleich schwierig, wenn sie durch polynomial beschränkte Transformationen aufeinander abgebildet werden können. Also wirkt sich der Transformationsaufwand auf das klassifizierende Komplexitätsurteil nicht aus.

Ein Entscheidungsproblem gilt als entscheidbar, wenn mindestens ein Algorithmus existiert, der es gestattet, alle konkreten Problemausprägungen zu lösen. Eine Problemausprägung ist gelöst, wenn bewiesen werden konnte, daß entweder die Anwort "Ja" oder "Nein" für die zur Entscheidung unterbreitete Fragestellung gültig ist. Für unentscheidbare Entscheidungsprobleme ist dagegen bekannt, daß es mindestens eine konkrete Problemausprägung gibt, bezüglich derer nicht bewiesen werden kann, daß die Antwort "Ja" oder "Nein" gültig ist[21)].

Die - prima facie paradoxe - Beschäftigung mit unentscheidbaren Entscheidungsproblemen ist keineswegs müßig. Denn es konnte aufgezeigt werden, daß tatsächlich gehaltvolle Entscheidungsprobleme formuliert werden können, die sich nicht entscheiden lassen. Hierzu zählen insbesondere[22)] die Gödel- und die Church-/ Turing-Variante der Unentscheidbarkeit des Gültigkeitsproblems für den Kalkül der Prädikatenlogik (1. Ord-

21) Vgl. Gödel (1931), S. 174; Bachem (1980), S. 829.

22) Vgl. zu weiteren unentscheidbaren Entscheidungsproblemen (oder Theorien, innerhalb derer unentscheidbare Entscheidungsprobleme formuliert werden können) Kleene (1952), S. 382ff.; Hermes (1952), S. 188f.; Rogers (1967), S. 24ff., 32ff. u. 95f.; Garey (1979), S. 12; Bachem (1980), S. 840f. Tiefere Einsichten in die Theorie unentscheidbarer Probleme vermittelt Putnam (1973), S. 71ff.
Die voranstehend und in den folgenden Fußnoten angeführten Quellen erstrecken sich nur auf Probleme, die in einem formalsprachlichen, d.h. rein syntaktisch definierten Entscheidungskalkül unentscheidbar sind. Der formalsprachliche Ausdrucksreichtum ist bereits groß genug - wie ursprünglich von Gödel aufgezeigt - um solche Entscheidungsprobleme konstruieren zu können. Darüber hinaus zeigt Weizenbaum (1982), S. 99ff., daß sich in einfacher Weise Probleme in semantisch interpretierenden Systemen formulieren lassen, die mit formalsprachlichen Instrumenten (den u.a. Turing-Automaten) nicht entschieden werden können. Die gesamten komplexitätstheoretischen Ausführungen dieser Ausarbeitung beschränken sich jedoch auf formalsprachliche Kalküle.

nung)[23], die Unentscheidbarkeit des 10. Problems aus dem Katalog fundamentaler mathematischer Probleme von

23) Vgl. Gödel (1931), S. 173ff., insbesondere S. 193ff., der den Nachweis der Unentscheidbarkeit erstmals erbrachte (S. 187ff.), allerdings nicht direkt für die Prädikatenlogik, sondern für ein allgemeiner konzipiertes, als formal widerspruchsfrei vorausgesetztes System, das strukturell reichhaltig genug ist, um prädikatenlogische und zahlentheoretische Aspekte zu inkorporieren; Church (1936b), S. 40f., und Church (1936b), S. 101f., der unmittelbar die Unentscheidbarkeit der Prädikatenlogik 1. Stufe bewies; Turing (1937a), S. 259ff.; Turing (1937b), S. 544ff.; Myhill (1952), S. 180ff.; Davis (1958), S. 127ff.; Lucas (1961), S. 112ff.; Nagel (1964), S. 69ff., insbesondere S. 82ff.; Stegmüller (1973), S. 5ff., insbesondere S. 20ff. u. 27 (in bezug auf Gödel), sowie S. 44f. u. 55ff. (hinsichtlich Church); Boolos (1974), S. 115ff. u. 173ff.; Hermes (1978), S. 165ff.; Hopcroft (1984), S. 34 u. 44.
Die Unentscheidbarkeit des Gültigkeitsproblems im Sinne von Gödel besagt, es existiert mindestens eine prädikatenlogische Formel (1. Ordnung), für die mit keinem Algorithmus bewiesen ("entschieden") werden kann, daß sie hinsichtlich eines - als gültig vorausgesetzten - Axiomensystems entweder gültig oder aber (durch Beweis der Gültigkeit der Negat-Formel) ungültig ist. Die Unentscheidbarkeit des Gültigkeitsproblems im Sinne von Church und Turing bedeutet dagegen, daß kein Algorithmus existiert, mit dessen Hilfe sich die Gültigkeit jeder prädikatenlogischen Formel hinsichtlich eines Axiomensystems beweisen läßt. Die Gödel-Variante bezieht sich auf eine partikulare Formel, deren Gültigkeitsstatus unentscheidbar bleibt. Die Church-/ Turing-Variante zielt dagegen auf einen generellen Algorithmus für Gültigkeitsbeweise ab. Vgl. zu dieser Differenzierung Turing (1937a), S. 259; Myhill (1952), S. 181.

Hilbert[24] und die Unentscheidbarkeit des Halteproblems für Turing-Automaten[25]. Auf Turing-Automaten wird später näher eingegangen. Aus der Unentscheidbarkeit des 10. Problems von Hilbert folgt die - für das Operations Research fundamentale - Konsequenz, daß kein Algorithmus existiert, mit dem alle ganzzahligen Optimierungsprobleme gelöst werden können[26].

24) Vgl. Hilbert (1900), S. 276; Matijasevic (1970), S. 354ff., der erstmals den Unentscheidbarkeitsbeweis erbrachte; Post (1944), S. 288; Davis (1958), S. 102ff.; Davis (1973a), S. 233f. u. 262 i.V.m. S. 237ff. ; Davis (1973b), S. 85, 87 u. 90f.; Cohors-Fresenborg (1977), S. 47ff.; Bachem (1980), S. 841. Die Unentscheidbarkeit des 10. Problems von Hilbert bedeutet die Unmöglichkeit, einen Algorithmus zu konstruieren, mit dem für jede diophantische Gleichung mit ganzzahligen Koeffizienten - präziser: für die Nullstellen eines Polynomials mit ganzzahligen Koeffizienten - mindestens eine ganzzahlige Lösung existiert (oder nicht). Diese Unentscheidbarkeit aus dem Bereich der Zahlentheorie bedeutet eine Verschärfung des o.a. Unentscheidbarkeits-Resultats für die Prädikatenlogik (vgl. Davis (1973a), S. 263): Zu jeder Axiomatisierung der Zahlentheorie existiert mindestens eine diophantische Gleichung, die keine ganzzahlige Lösung besitzt. Die Nichtexistenz dieser Lösung kann aber innerhalb der gegebenen Axiomatisierung nicht bewiesen werden (Unvollständigkeit jeder axiomatisierten Zahlentheorie).

25) Vgl. Turing (1937a), S. 246ff., insbesondere S. 248 u. 262, der als erster die Unentscheidbarkeit bewies; Burks (1966), S. 52f. u. 124f.; Rogers (1967), S. 24ff.; Brauer (1968), S. 49ff.; Savage (1976), S. 185ff.; Cohors-Fresenborg (1977), S. 44f.; Hermes (1978), S. 144ff.; Paul (1978), S. 59f.; Hopcroft (1984), S. 45.
Die Unentscheidbarkeit des Halteproblems besteht darin, daß es keinen Algorithmus geben kann, der es gestattet, für jede Informationseingabe in den Automaten zu entscheiden, ob dieser nach endlicher Zeit seine internen informationsverarbeitenden Operationen beenden wird (oder nicht).

26) Noch strenger läßt sich nachweisen, daß es nicht einmal Algorithmen geben kann, die es gestatten, alle ganzzahligen Optimierungsprobleme mit linearen Zielfunktionen und quadratischen Restriktionen zu lösen; vgl. Jeroslow (1973), S. 221ff., insbesondere S. 223.

Ein Entscheidungsproblem ist "gelöst"[27], wenn seine Entscheidbarkeit oder seine Unentscheidbarkeit bewiesen wurde. Falls keiner dieser beiden Nachweise erfolgt ist, gilt das Entscheidungsproblem als - vorläufig - ungelöst[28]. Zu den ungelösten Entscheidungs(meta)problemen der Komplexitätstheorie gehört die Frage, ob die Klassen der P- und der NP-komplexen Probleme identisch sind. Auf diese Problemklassen wird weiter unten eingegangen[29]. Ein weiteres Problem - das Erreichbarkeitsproblem der Petrinetz-Theorie - galt lange Zeit als ungelöst, konnte aber inzwischen durch den Nachweis seiner Entscheidbarkeit gelöst werden[30].

Eine erste Klassifizierung der Komplexität von Entscheidungsproblemen erfolgt durch die Unterscheidung zwischen gelösten und ungelösten Entscheidungsproblemen. Die Versuche zur Bewältigung der letztgenannten haben sich als derart schwierig erwiesen, daß sie bislang erfolglos blieben.

Auf einer zweiten Stufe kann innerhalb der weniger schwierigen Klasse der gelösten Entscheidungsprobleme differenziert werden, ob diese Probleme zur (Sub-)Klasse der entscheidbaren oder der unentscheidbaren Probleme gehören. Die unentscheidbaren Probleme sind insofern schwieriger als die entscheidbaren, als für die letztgenannten, nicht aber für die erstgenannten ein allgemeingültiger Lösungsalgorithmus existiert.

---

27) Es handelt sich hier um einen metasprachlichen Lösungsbegriff, da er nicht die Lösungen des Entscheidungsproblems auf der Objektebene betrifft. Vielmehr ersteckt er sich auf das Metaproblem der Entscheidbarkeit oder Unentscheidbarkeit des Objektproblems.

28) Bachem (1980), S. 813 u. 840f., vertritt ein abweichendes Verständnis des Lösungsbegriffs. Er identifiziert unlösbare mit unentscheidbaren Problemen. Zur Unterscheidung von dieser Position verwendet der Verf. anstelle des Attributs "unlösbar" das Partizip "ungelöst", zumal ein derzeit ungelöstes Problem der Komplexitätstheorie zukünftig - durch Nachweis seiner (Un-)Entscheidbarkeit - gelöst werden kann, also keineswegs grundsätzlich "unlösbar" ist.

29) Vgl. S. 29ff. u. 38ff.

30) Auf dieses Erreichbarkeitsproblem wird noch mehrfach zurückgekommen; vgl. S. 47, 49 u. Abschnitt 3.2.1.

Fortan werden nur noch entscheidbare Entscheidungsprobleme näher betrachtet. Unentscheidbare[31] und ungelöste Entscheidungsprobleme lassen sich zwar in dem o.a. Sinn - im Vergleich zu den gelösten, entscheidbaren Problemen - intuitiv als schwieriger qualifizieren. Doch kann auf die beiden ausgegrenzten Arten von Entscheidungsproblemen der aufwandsbezogene Komplexitätsbegriff nicht angewendet werden, weil für diese Probleme per definitionem keine (allgemeingültigen) Lösungsalgorithmen existieren, für deren Ausführung entsprechende Ressourcen einzusetzen wären.

2.2 Turing-Automaten als allgemeine Instrumente für die Problemlösung

Dem Nachweis der Entscheidbarkeit von Problemen und der Ermittlung des Aufwands, der zu ihrer Bewältigung erforderlich ist, wird seitens der Komplexitätstheorie in der Regel das Konzept der (sequentiellen) Turing-

31) Vgl. zur betriebswirtschaftlichen Bedeutung unentscheidbarer Probleme - mit Schwerpunkt auf Fragestellungen des Operations Research - Bachem (1980), S. 841; Zelewski (1986), S. 937ff.

Automaten[32] zugrundegelegt[33]. Auf neuere Analysen, welche die Problemkomplexität auf der Basis von nichtsequentiellen (nebenläufigen, parallelen) Automatenarchitekturen untersuchen[34], wird hier nicht eingegan-

---

32) Vgl. Turing (1937a), S. 231ff. u. 241ff.; Hermes (1937), S. 114ff., insbesondere S. 118ff.; Kleene (1952), S. 356ff.; Hermes (1952), S. 184ff.; Hermes (1954), S. 49f.; Peter (1957), S. 202ff.; Wang (1957), S. 85ff.; Davis (1958), S. 3ff.; Shepherdson (1963), S. 233ff.; Hartmanis (1965), S. 286ff.; Fischer (1965), S. 570ff.; Putnam (1966), S. 140ff.; Rogers (1967), S. 13ff.; Brauer (1968), S. 12ff.; Putnam (1973), S. 69ff.; Böhling (1974), S. 8ff.; Herschel (1974), S. 67ff.; Boolos (1974), S. 19ff.; Ottmann (1975), S. 5ff.; Baur (1976), S. 11ff.; Savage (1976), S. 174ff.; Schnorr (1976), S. 96f.; Mehlhorn (1977), S. 190ff.; Zervos (1977), S. 285ff.; Hermes (1978), S. 18ff., 33ff. u. 203ff.; Paul (1978), S. 25ff. u. 62ff.; Garey (1979), S. 23ff.; Brucker (1981), S. 148ff. u. 153ff.; Weizenbaum (1982), S. 80ff., insbesondere S. 88ff.; Hopcroft (1984), S. 34ff.
Äquivalente oder spezialisierte Automaten-Konzepte werden beschrieben von: Post (1936), S. 103ff.; Wang (1957), S. 63ff.; Kaphengst (1958), S. 366ff.; Asser (1959), S. 346ff.; Yamada (1962), S. 754f.; Shepherdson (1963), S. 219ff. u. 244ff.; Schnorr (1974), S. 28f. u. 46ff.; Cohors-Fresenborg (1977), S. 9ff.; Paul (1978), S. 86ff. u. 94ff.; Priese (1979), S. 508ff.; Heinemann (1980), S. 62ff. Alle Automaten-Konzepte der Komplexitätstheorie sind zu Turing-Automaten in der Weise äquivalent (oder deren Spezialfälle), daß sich die ersten auf die zweiten durch polynomial beschränkte Transformationen zurückführen lassen.

33) Vgl. Karp (1986), S. 103.

34) Vgl. Cook (1983), S. 405f.; Li (1984), S. 212ff.; Pan (1986), S. 127ff.

gen. Trotz ihrer strukturellen Einfachheit[35] nehmen Turing-Automaten eine zentrale interdisziplinäre Posi-

35) Ein (deterministischer, sequentieller) Turing-Automat läßt sich grob als ein symbolisch-informationsverarbeitender, speicherprogrammierter Automat beschreiben, der im wesentlichen aus einem internen Speicher und einem Schreib-/Lese-Kopf besteht, mit dem er auf ein Band als externen Speicher zugreift. Das Band ist in linear aneinandergereihte Felder, in denen jeweils eine Informationseinheit als Symbol codiert werden kann, eingeteilt und beidseitig unbegrenzt, um von der Fiktion eines - zumindest potentiell - unbeschränkten Speicherplatzes ausgehen zu können (Ausklammerung von Aspekten der Raumkomplexität). Der interne Speicher nimmt die Gestalt einer endlichen zweidimensionalen Matrix ("Maschinentafel") an, die das Verarbeitungsprogramm repräsentiert (interner Speicherplatz für das Ablegen von Zwischenergebnissen existiert nicht). In jedem Matrixfeld wird einem Informationsinput (1. Matrixdimension) und einem Automatenzustand (2. Matrixdimension mit zwei ausgezeichneten Zuständen: dem Start- und dem Zielzustand) ein Tripel zugeordnet, das einen Informationsoutput (einschließlich eines "nil"-Elements für den Fall der Outputunterlassung), den nachfolgenden Automatenzustand und die Bewegung des Schreib-/Lese-Kopfes bestimmt. Dieser Kopf kann auf dem Band um ein Feld nach rechts oder links bewegt oder an seiner alten Position belassen werden.
Die Operationen des Automaten beginnen damit, daß er in seinem Startzustand auf ein Feld des Bandes angesetzt wird. Das dort befindliche Symbol wird vom Automatenkopf als Informationsinput eingelesen und gemäß der Verarbeitungs-Matrix ausgewertet. Gegebenenfalls wird eine Information als Output auf das Feld unter dem Schreib-/Lese-Kopf geschrieben, die neue Position des Kopfes ermittelt und das dort vorgefundene Symbol als neuer Input eingelesen. Dieser Verarbeitungszyklus wird so lange wiederholt, bis der automatenspezifische Zielzustand erzeugt wird und der Automat anhält. Oder der Automat setzt seine Operationen unendlich fort. Das Konzept der Struktur und Funktionsweise von Turing-Automaten wird ergänzt durch die Spezifizierung eines endlichen Alphabets zulässiger Symbole, mit denen die Informationen auf dem Band codiert werden. Im einfachsten Fall wird das Dualalphabet aus der Null und der Eins vorausgesetzt.
Verfeinerungen dieses Konzepts sind - wie in den Quellen der Fußnote 32) ausgeführt - in vielfacher Weise möglich, z.B. durch die Zulässigkeit mehrerer Bänder, durch deren ein- oder zweiseitige Begrenzung, durch spezielle Annahmen über die Anzahl oder Eigenart der Automatenzustände, durch größere Zugriffsbereiche oder Schrittweiten der Verschiebung des Schreib-/Lese-Kopfes oder durch differenziertere Automatenalphabete (z.B. Einführung eines "Leer"-Symbols, dessen Output das Löschen einer Informationseinheit auf dem Band ermöglicht).

tion ein. Denn stellen ein Bindeglied zwischen Komplexitätstheorie, numerischer und symbolischer Informationsverarbeitung[36], Metamathematik[37] und formaler Logik dar[38].

Der aus der Sicht des Operations Research wichtigste Zusammenhang wird durch die These von Church gestiftet, daß die Konzepte der - im endlichen Sinne - berechenbaren Funktionen einerseits und der (allgemein-[39])/

36) Trotz ihrer unterschiedlichen Struktur und Funktionsweise läßt sich zeigen, daß alle realen Computer-Architekturen auf das Konzept der Turing-Automaten zurückgeführt werden können; vgl. Kleene (1952), S. 377; Yamada (1962), S. 753f.; Savage (1976), S. 182; Paul (1978), S. 94ff. i.V.m. S. 86ff.; Weizenbaum (1982), S. 93; Hopcroft (1984), S. 39.

37) Die Metamathematik erstreckt sich auf die Analyse mathematischer Grundstrukuren, die vor allem im Kontext der Zahlentheorie erfolgt; vgl. z.B. Kleene (1952) und Stegmüller (1973) (passim).

38) Vgl. ansatzweise Rogers (1967), S. 19.

39) Vgl. zu allgemein-rekursiven Funktionen Kleene (1952), S. 270ff.; Peter (1957), S. 30ff. u. 172ff.; Stegmüller (1973), S. 48ff.; Davis (1973a), S. 257ff.; Hermes (1978), S. 114ff., insbesondere S. 118ff.

partiell-[40]) rekursiven Funktionen anderseits äquiva-

40) Die hier vorgestellten metamathematischen Erkenntnisse wurden ursprünglich auf die Klasse der allgemein-rekursiven Funktionen bezogen. In der Folgezeit wurden sie jedoch mehrfach für die Klasse der partiell-rekursiven Funktionen reformuliert, weil mitunter Bedenken aufkamen, die allgemein-rekursiven Funktionen seien nicht wohl-definiert oder könnten nicht immer von Turing-Automaten berechnet werden. Vgl. hierzu Shepherdson (1963), S. 217; Brauer (1968), S. 6. Auf jeden Fall muß auf partiell-rekursive Funktionen übergegangen werden, wenn die Definitionsbereiche der zu berechnenden Funktionen rekursiv-aufzählbare, aber nicht-rekursive Mengen darstellen; vgl. Savage (1976), S. 182 i.V.m. S. 187. Vgl. zur Differenzierung zwischen rekursiven und rekursiv-aufzählbaren Mengen (Prädikaten) Post (1944), S. 285 u. 290ff.; Rogers (1967), S. 57ff., 95 u. 161ff.; Cohors-Fresenborg (1977), S. 45ff. Vgl. zu partiell-rekursiven Funktionen Kleene (1952), S. 325ff.; Peter (1957), S. 175 u. 186; Davis (1958), S. 41ff.; Kaphengst (1958), S.374; Shepherdson (1963), S. 221f.; Rogers (1967), S. 18f.; Savage (1976), S. 187ff.; Hermes (1978), S. 119f.
Allgemein- und partiell-rekursive Funktionen unterscheiden sich nur dadurch, daß erste für alle möglichen Argumente aus ihren Vorbereichen definiert sind, während bei zweiten die Definitions- echte Teilmengen der Vorbereiche darstellen. Allgemein-rekursive Funktionen sind daher ein Spezialfall der partiell-rekursiven Funktionen; vgl. Peter (1957), S. 186. Weil die Differenzierung zwischen allgemeiner und partieller Rekursivität für die nachfolgenden Erörterungen nicht erheblich ist, wird nur noch von rekursiven Funktionen gesprochen.

lent (koextensiv) sind[41]. Diese These kann zwar nicht im strengen Sinne verifiziert werden[42], weil zwischen dem unscharfen, zunächst nur intuitiv gedeuteten[43] Begriff der Berechenbarkeit und dem präzise definierten Konzept der rekursiven Funktionen ein kategorialer Unterschied besteht.

41) Vgl. Church (1936a), S. 356ff., der von effektiver Berechenbarkeit (effectively calculable) spricht; Post (1944), S. 285; Kleene (1952), S. 300 u. 318f.; Putnam (1966), S. 162; Brauer (1968), S. 5; Stegmüller (1973), S. 46f.; Hermes (1978), S. 31. Nicht auf die allgemein-, sondern auf die partiell-rekursiven Funktionen wird die Church-These bezogen von Kleene (1952), S. 332; Shepherdson (1963), S. 217; Schnorr (1972), S. 54; Cohors-Fresenborg (1977), S. 44; Bachem (1980), S. 815.
Von Peter (1957), S. 202ff., und Kleene (1952), S. 317ff., wird eine Reihe von Plausibilitätsargumenten erörtert, welche die Church-These stützen; vgl. hierzu auch die Bestätigung der Kontraposition der Church-These in Fußnote 46) durch die Rado-Funktion. Allerdings wird die Angemessenheit der Church-These durch das Argument bestritten, daß nicht-allgemein-rekursive Funktionen existieren, die sich dennoch berechnen lassen; vgl. Kalmar (1955), S. 93ff.; Peter (1957), S. 223ff. (Dieser Einwand wäre angesichts der o.a. Substitution von allgemein- durch partiell-rekursive Funktionen erneut zu überprüfen, doch wird dieser Randaspekt hier nicht weiterverfolgt.) Dem widerspricht Brauer (1968), S. 6, mit der Feststellung, daß bisher jede tatsächlich berechenbare Funktion als allgemein-rekursiv nachgewiesen werden konnte. Von der großen Mehrheit der Experten auf dem Gebiet der Metamathematik wird die Church-These anerkannt; vgl. Brauer (1968), S. 5f.; Stegmüller (1973), S. 71; Schnorr (1974), S. 20; Paul (1978), S. 54. Hiervon wird fortan ausgegangen.
Die Church-These behauptet des weiteren die Äquivalenz bezüglich eines dritten Konzepts, des der Funktionen aus dem λ-Kalkül. Hierauf wird jedoch nicht weiter eingegangen. Näheres zu diesem Kalkül und zu seiner Äquivalenz mit rekursiven Funktionen sowie mit Funktionen, die von Turing-Automaten berechnet werden können, bei Church (1936a), S. 346ff.; Kleene (1936), S. 342ff.; Turing (1937c), S. 153ff. u. 160ff.; Hermes (1978), S. 207ff.

42) Vgl. Kleene (1952), S. 317; Peter (1957), S. 217; Brauer (1968), S. 5; Stegmüller (1973), S. 46; Schnorr (1974), S. 54; Paul (1978), S. 53 i.V.m. S. 8.
Kleene (1952), S. 295, definiert einen präzisen Berechenbarkeitsbegriff im Kontext der Zahlentheorie und kann auf dieser Grundlage beweisen (S. 295f.), daß eine Funktion in diesem Sinne genau dann berechenbar ist, wenn sie zur Klasse der rekursiven Funktionen gehört.

43) Vgl. Turing (1937a), S. 249ff.; Peter (1957), S. 5; Stegmüller (1973), S. 46; Paul (1978), S. 8.

Doch läßt sich die Church-These in einem abgeschwächten - und zugleich erweiterten - Sinne als operationale Ausdeutung des Begriffs der Berechenbarkeit verstehen. Die Erweiterung der Church-These erfolgte durch die Arbeiten von Turing und Post, die - voneinander unabhängig - das Konzept der Turing-Automaten entwickelten[44]. Es wurde nachgewiesen, daß die Klasse der Funktionen, die von Turing-Automaten berechnet werden können[45], mit der Klasse der rekursiven Funktionen übereinstimmt[46]. Da gemäß der Church-These bereits alle berechenbaren mit den rekursiven Funktionen identifiziert worden waren, folgt als erweiterte Church-Post-Turing-These, daß sich die Werte aller - im intuitiven

---

44) Vgl. Kleene (1952), S. 356; Brauer (1968), S. 6; Hermes (1978), S. 31.

45) Vgl. zur Berechenbarkeit von Funktionen durch Turing-Automaten Turing (1937a), S. 230 u. 254f.; Hermes (1937), S. 120f.; Kleene (1952), S. 360ff. u. 377ff.; Hermes (1954), S. 50f.; Wang (1957), S. 68f.; Yamada (1962), S. 755ff.; Brauer (1968), S. 40ff.; Savage (1976), S. 181f.; Hermes (1978), S. 38f. u. 96ff.

46) Vgl. Kleene (1952), S. 360 u. 373ff.; Peter (1957), S. 211ff.; Wang (1957), S. 66f., 69ff. u. 86; Shepherdson (1963), S. 217f. u. 234f.; Stegmüller (1973), S. 47; Putnam (1973), S. 71; Boolos (1974), S. 89ff.; Paul (1978), S. 35ff.
Es gilt jedoch nicht die schwächere Aussage, daß alle Funktionen, die durch Turing-Automaten dargestellt (statt: "berechnet") werden können, auch rekursiv sind. Vielmehr existieren Funktionen, die nicht rekursiv, aber dennoch durch Turing-Automaten definiert sind. Hierzu zählt z.B. die Rado-Funktion R(n), für die gezeigt werden kann, daß sie mit wachsendem n so schnell ansteigt, daß sie grundsätzlich nicht für jedes beliebige (große) n durch Turing-Automaten berechnet werden kann; vgl. Rado (1962), S. 880ff.; Brauer (1968), S. 45ff.; Ludewig (1983), S. 4 u. 8; Dewdney (1984), S. 9 u. 12; Hopcroft (1984), S. 45ff.; siehe auch die Anmerkungen zum assoziierten Problem des "fleißigen Bibers" auf S. 47. Die Nichtberechenbarkeit der nicht-rekursiven Rado-Funktion bestätigt die Kontraposition der Church-These, daß nicht-rekursive Funktionen in keiner endlichen Weise berechnet werden können; vgl. zu dieser Kontraposition Peter (1957), S. 201.

Sinne - berechenbaren Funktionen durch Turing-Automaten ermitteln lassen[47].

Alle Algorithmen können konzeptionell auf die Berechnung von Funktionswerten zurückgeführt werden[48]. Da das intuitive Verständnis der Berechenbarkeit von Funktionen durch die Einsatzmöglichkeit von Turing-Automaten operationalisiert wurde, lassen sich Algorithmen grundsätzlich als Turing-Automaten darstellen[49]. Die Turing-Automaten vermögen wiederum alle re-

---

47) Vgl. Kleene (1952), S. 321, 356 u. 376; Rogers (1967), S. 20; Brauer (1968), S. 6; Putnam (1973), S. 69; Stegmüller (1973), S. 47; Paul (1978), S. 53; Peterson (1981), S. 201f., der sich auf berechenbare Systeme anstelle von Funktionen bezieht; Hopcroft (1984), S. 39 u. 44. Kleene (1952), S. 381, und Weizenbaum (1982), S. 94f., weiten den Berechnungsbezug der o.a. These auf die Bearbeitung aller Probleme aus, die sich durch Ausdrücke aus einem (formalsprachlichen) endlichen Alphabet formulieren lassen. Vgl. zur allgemeinen Akzeptanz der Church-Post-Turing-These z.B. Rogers (1967), S. 20. Von manchen der voranstehend angeführten Autoren wird diese erweiterte Church-Post-Turing-These auch nur als Church- oder nur als Turing-These bezeichnet.

48) Vgl. Church (1936a), S. 351 u. 356; Rogers (1967), S. 18ff., mit der starren Verknüpfung der Attribute "berechenbar" und "algorithmisch", u. 26f.; Schnorr (1974), S. 19; Paul (1978), S. 8; Hermes (1978), S. 9; Bachem (1980), S. 815f.
Hier und nachfolgend werden Algorithmen im sehr weit gefaßten Sinne von Markow als Vorschriften zur systematischen Veränderung von Symbolfolgen verstanden; vgl. zum Konzept der Markow-Algorithmen Asser (1959), S. 355f.; Brauer (1968), S. 7f.; Cohors-Fresenborg (1977), S. 84; Hermes (1978), S. 245f. Die o.a. These der Äquivalenz von Algorithmen und Funktionsberechnungen läßt sich unter dieser Voraussetzung dadurch streng beweisen, daß die Klassen der (partiell-)rekursiven Funktionen und der Funktionen, die durch Markow-Algorithmen berechnet werden können, koextensiv sind; vgl. Peter (1957), S. 222; Brauer (1968), S. 7f.; Bachem (1980), S. 815. Auf die Äquivalenz von Algorithmen und rekursiven Funktionen verweist auch Kaphengst (1958), S. 376.

49) Vgl. Asser (1959), S. 356ff.; Brauer (1968), S.7f.; Savage (1976), S. 76, 174 u. 181 (der von Prozeduren spricht, die - neben endlichen Algorithmen - auch nicht-terminierende Prozesse umfassen); Hermes (1978), S. 18f.
Wenn ein Entscheidungsproblem durch einen Turing-Automaten nicht entschieden werden kann, dann existiert auch keine andere (uniforme) Prozedur, durch die das Problem entscheiden werden könnte; vgl. Putnam (1973), S. 70f.

kursiven Funktionen zu berechen. Das Konzept dieser Funktionen ist so mächtig, daß sich mit ihm alle Lösungsalgorithmen für Optimierungsprobleme des Operations Research formulieren lassen[50]. Daher ist es möglich, komplexitätstheoretische Analysen von Fragestellungen des Operations Research immer auf die Betrachtung von Turing-Automaten zurückzuführen.

Ein Entscheidungsproblem kann durch einen Turing-Automaten genau dann entschieden werden, wenn es mit endlichem Ressourceneinsatz möglich ist, aus der Eingabe der (endlichen) Beschreibung jeder Problemausprägung ein "Ja" oder "Nein" als Problemlösung abzuleiten[51]. Der Ressourceneinsatz erstreckt sich sowohl auf den erforderlichen Speicherplatz, der hier im wesentlichen[52] als externer Band-Speicher realisiert ist, als auch auf die Operationszeit des Automaten, die zwischen Problemeingabe und Lösungsausgabe vergeht.

---

50) Zu den rekursiven Funktionen rechnen nicht nur sämtliche algebraischen Funktionen, sondern z.B. auch die Funktionen, deren Abbildungsvorschriften durch Minimierungs- oder Maximierungsoperatoren ausgedrückt werden; vgl. Peter (1957), Tabelle zwischen S. 40 u. 41, Positionen Nr. 11 u. 12.

51) Im Spezialfall der Betrachtung von Funktionen ist das Problem zu untersuchen, ob ein Funktionswert für vorgegebene Werte der unabhängigen Variablen existiert. Da das Konzept der Turing-Automaten konstruktiv ist, kann aus der bejahenden Automatenantwort der gesuchte Funktionswert rekonstruiert - "berechnet" - werden.

52) Einen kleinen Anteil des verwendeten Speicherplatzes stellt die (interne) Automatenmatrix dar, die jedoch nicht zur Speicherung von Zwischenergebnissen genutzt werden kann.

Die nachfolgenden Ausführungen beziehen sich vornehmlich[53] auf die Operationszeit[54]. Der aufwandsbezogene Komplexitätsbegriff berücksichtigt also insofern nur den Teilaspekt der Zeitkomplexität[55]. Hinsichtlich der Ausklammerung des komplementären Aspekts der Raumkomplexität[56] wird unterstellt, daß stets ein zwar endliches, aber beliebig großes Angebot von Speicherplatz vorhanden ist. Die trade offs, die daraus resultieren, daß Algorithmen mit höherem (niedrigerem) Speicherplatzbedarf in der Regel zu niedrigeren (höheren) Operationszeiten führen, werden nicht weiter erörtert[57].

2.3 Analysekonzepte der Komplexitätstheorie

Die Klasse der entscheidbaren Entscheidungsprobleme läßt sich hinsichtlich ihrer (Zeit-)Komplexität mit der Hilfe der Komplexitätsfunktion O(n) in Sub-Klassen zerlegen.

Ausgangspunkt dieser Funktion ist die Überlegung, daß die Operationszeit, die zur Lösung eines Entscheidungsproblems aufgewendet werden muß, im Regelfall mit dem Umfang der Beschreibung der konkret zu lösenden Problemausprägung variiert. Dieser Problemumfang wird durch die Größe n gemessen. Je nach dem zugrundegelegten Meßkonzept, das jeweils eine bestimmte Codierung

53) Vgl. zu einer Ausnahme die Ausführungen auf S. 48f.
54) Vgl. zu dieser - komplexitätstheoretisch typischen - Einschränkung Parker (1982a), S. 4.
55) Vgl. zum Konzept der Zeitkomplexität Böhling (1974), S. 136ff.; Ecker (1977), S. 54f; Paul (1978), S. 91ff.; Garey (1979), S. 26f.; Brucker (1981), S. 151ff.; Parker (1982a), S. 5.
56) Vgl. zum Konzept der Raumkomplexität von Turing-Automaten Böhling (1974), S. 204ff.; Hopcroft (1974), S. 624; Cardoza (1976), S. 52; Jones (1977), S. 280; Paul (1978), S. 93; Garey (1979), S. 170ff.
57) Vgl. hierzu etwa Cook (1973), S. 29ff.; Cook (1983), S. 404.

für das zu lösende Problem impliziert[58], kann die Komplexitätsfunktion für dasselbe Entscheidungsproblem unterschiedliche Gestalt annehmen. Diese Codierung[59] bleibt jedoch - von wenigen Ausnahmen abgesehen[60] - für Komplexitätsurteile unerheblich[61]. Sofern nicht ausdrücklich anders vermerkt, wird unterstellt, daß die Größe n die Anzahl binärer Informationseinheiten (bits) mißt, die zur Codierung einer Problemausprägung für die Eingabe in einen Turing-Automaten erforderlich sind (binäre Codierung)[62].

---

58) Dies resultiert aus dem - in Abschnitt 2.2 dargelegten - Sachverhalt, daß in der Komplexitätstheorie Problemlösungen stets auf die Operationen von (Turing-)Automaten zurückgeführt werden. Diese Automaten können auf ein Problem erst dann angesetzt werden, wenn ihnen eine automatenverständliche Problemcodierung eingegeben wird.

59) Vgl. zur Codierung von Problemen Garey (1978), S. 502f.; Garey (1979), S. 9f. u. 20ff.; Parker (1982a), S. 5f.

60) Vgl. hierzu die Ausführungen auf S. 42f. zur Variation der Komplexitätsresultate in Abhängigkeit von der binären oder unären Problemcodierung. Vgl. auch die Anmerkungen auf S. 67 zum Maß L für den Problemumfang.

61) Vgl. Ecker (1977), S. 56; Valiant (1978), S. 327. Die Irrelevanz der Codierungsweise folgt aus dem Umstand, daß seitens der Komplexitätstheorie Probleme als äquivalent (gleich komplex) behandelt werden, wenn sie durch polynomial beschränkte Algorithmen ineinander transformiert werden können. (Vgl. hierzu die Ausführungen auf S. 29ff. zu P-komplexen Problemen und auf S. 34f. zur Reduzierbarkeit von Problemen.) Die meisten Problemcodierungen lassen sich durch solche Algorithmen aufeinander abbilden; vgl. Garey (1979), S. 20 u. 23; Lenstra (1979), S. 124. Daher ändert die Codierung im allgemeinen nicht die Klassifizierung der Problemkomplexität. Dennoch spielt die Problemcodierung eine fundamentale Rolle in der Komplexitätstheorie. Durch die Codierung werden die Erkenntnisobjekte - die Probleme - mit den leistungsfähigen Analyseinstrumenten aus den Theorien der formalen Sprachen und der sprachverarbeitenden (-erkennenden) Automaten so verknüpft, daß gehaltvolle Komplexitätserkenntnisse gewonnen werden können. Vgl. zu diesem Codierungsaspekt Garey (1979), S. 19f.

62) Vgl. zur binären Codierungsweise Schnorr (1974), S. 134; Garey (1975), S. 410; Rabin (1976), S. 23; Schuster (1976), S. 38; Ullman (1976), S. 140f.; Valiant (1978), S. 327; Paul (1978), S. 29f. u. 34f.; Garey (1979), S. 21f.; Parker (1982a), S. 6; Bland (1981), S. 1079; Tardos (1986), S. 250.

Die Komplexitätsfunktion O(n) gibt die Anzahl des elementaren Operationen an, die von einem Turing-Automaten ausgeführt werden müssen, um für das eingegebene Problem des Umfangs n eine Lösung auszugeben oder die Nichtexistenz von (zulässigen) Lösungen festzustellen[63]. Die Komplexitätsfunktion abstrahiert von additiven und multiplikativen[64] Konstanten, weil sie nur die Größenordnung[65] des Lösungsaufwands angeben soll. Darüber hinaus erfolgt eine Begrenzung auf das asymptotische Verhalten[66] des Turing-Automaten für Probleme, deren Umfang beliebig groß werden kann. Formal gilt für die Komplexitätsfunktion O(n) mit p(n) als einer beliebigen nicht-negativen, in der Regel jedoch logarithmischen, polynomialen[67] oder exponentiellen Funktion des Problemumfangs n, mit r(n) als dem tatsächlichen Ressourceneinsatz, der zur Ermittlung der Problemlösung aufgewendet werden muß, und mit c als einer beliebigen, aber konstanten reellen Zahl[68]:

$$r(n) = O(p(n)) \quad \Leftrightarrow \quad \bigwedge_{n>0} \left(r(n) \leqslant c \cdot p(n) \right)$$

oder:

$$r(n) = O(p(n)) \quad \Leftrightarrow \quad \lim_{n \to \infty} \left(r(n) : p(n) \right) = c$$

63) Unter der - oben dargelegten - Rückführung auf Entscheidungsprobleme bedeutet dies, daß der Turing-Automat ein "Ja" bezüglich der Existenz zulässiger Lösungen, aus der konkrete Lösungen rekonstruiert werden können, bzw. ein "Nein" ausgibt.

64) Vgl. Hopcroft (1974), S. 620. Einige Autoren führen jedoch multiplikative Konstanten an, wie aus den komplexitätsbeschreibenden Polynomialen für den Simplex-, Khachiyan- und Karmarkar-Algorithmus im Abschnitt 3.2.2 hervorgeht.

65) Daher kann das Funktionssymbol "O" als Akronym für die (Größen-)"O"rdnung des Aufwands der betrachteten Algorithmen und Probleme betrachtet werden. Mitunter wird die Komplexitätsfunktion auch als Landau'sche Funktion bezeichnet.

66) Vgl. Hopcroft (1974), S. 620.

67) Auf Polynomiale wird unten - vgl. Fußnote 89) - näher eingegangen.

68) Vgl. - zur ersten Version - Garey (1979), S. 6; Clausen (1986), S. 31f., und - zur zweiten Version - French (1982), S. 140.

Die Festlegung der "elementaren" Operationen bereitet nur auf den ersten Blick Schwierigkeiten. Zwar läßt sich darüber streiten, auf welcher Abstraktionsebene die Automatenoperationen als nicht mehr zerlegbar (elementar) angesehen werden sollen[69]. Doch können unterschiedliche Ansätze von Elementaroperationen durch polynomial beschränkte Algorithmen aufeinander abgebildet werden[70]. Eine solche Transformation ist für die Einordnung von Problemen in Komplexitätsklassen irrelevant[71].

Die Analysen der Komplexitätstheorie lassen sich zunächst grob in die Betrachtung der jeweils schechtest möglichen ("worst case") und der - fiktiven - durchschnittlichen Problemausprägung ("average case") differenzieren.

Die worst case-Analyse stellt den Standardfall komplexitätstheoretischer Untersuchungen dar[72]. Bei ihr wird ein Entscheidungsproblem nur im Hinblick auf die Menge derjenigen Problemausprägungen betrachtet, die zum größtmöglichen Lösungsaufwand führen (schlechtest mögliche oder "pathologische"[73] Ausprägungen). Diese Betrachtung des schlechtesten Falls besitzt den Vorzug, daß sie mit einer rein theoretischen Problemanalyse auskommt. Daher konzentriert sich die Mehrzahl der kom-

69) Beispielsweise können als elementar betrachtet werden: die Addition und Multiplikation von Ganzzahlen, die zugrundeliegende Addition von Dualzahlen oder die abstraktere Operation eines Schrittes, der im Rahmen eines iterativen Algorithmus wiederholt auszuführen ist. Auch den Komplexitätsangaben zu den Algorithmen in Abschnitt 3.2.2 beruhen nicht immer auf den gleichen Konzepten elementarer Operationen, die jeweils den angeführten Quellen zu entnehmen sind.

70) Vgl. Valiant (1978), S. 327; Bachem (1980), S. 816.

71) Vgl. hierzu auch die analoge Argumentation in Fußnote 61) zur Irrelevanz der Problemcodierung.

72) Vgl. als Überblick über diese Hauptrichtung der Komplexitätstheorie die annotierten Bibliographien von Papadimitriou (1985), S. 39ff., und Kindervater (1985), S. 110ff.; vgl. auch Parker (1982a), S. 7f.

73) Bland (1981), S. 1041, spricht drastisch von den "perversesten" ("most perverse") Problemausprägungen.

plexitätstheoretischen Untersuchungen auf diese worst case-Analyse[74].

Die wesentliche Unzulänglichkeit der worst case-Analyse liegt in ihrer fehlenden Repräsentativität hinsichtlich des durchschnittlichen Lösungsaufwands für die praktische Problembewältigung[75]. Diese Analyseart schätzt nur die Größenordnung des Lösungsaufwands für den Grenzfall schlechtest möglicher Problemausprägungen ab. Für die Anwendung von Erkenntnissen des Operations Research auf Realprobleme besitzt dagegen der Lösungsaufwand, mit dem durchschnittlich für die Bewältigung von Problemausprägungen gerechnet werden muß, größere Bedeutung[76]. Die Untersuchung dieses Aspekts erfordert jedoch eine empirische Ergänzung zu den komplexitätstheoretischen Ansätzen.

Die average case-Analyse bildet den Zweig der Komplexitätstheorie, der sich mit einer solchen empirisch fundierten Betrachtung des durchschnittlichen Lösungsaufwands von Problemen befaßt und in jüngerer Zeit zunehmende Beachtung erfährt[77].

Bei der analytischen - oder (explizit) probabilistischen - Variante[78] der average case-Analyse wird die explizite Angabe einer Wahrscheinlichkeitsverteilung vorausgesetzt[79], die über dem Spektrum möglicher Problemausprägungen definiert ist. Sie schreibt jeder Ausprägung ihre Eintrittswahrscheinlichkeit in den Problemkontexten, die jeweils als relevant erachtet wer-

---

74) Vgl. z.B. Karp (1975b), S. 64; Simon (1980), S. 4; Bachem (1980), S. 817.

75) Vgl. die kritischen Anmerkungen zur worst case-Analyse bei Schnorr (1974), S. 182; Karp (1975), S. 64f.; Bachem (1980), S. 817; Fisher (1982), S. 17; Coffman (1982), S. 320; Lenstra (1982), S. 205.

76) Vgl. Simon (1980), S. 4.

77) Vgl. als Überblick über diese komplexitätstheoretischen Studien die annotierte Bibliographie von Karp (1985), S. 52ff.; vgl. auch Garey (1979), S. 148ff.

78) Vgl. Simon (1976), S. 286ff. u. 298; Lenstra (1976), S. 3f.; Valiant (1978), S. 333f.; Garey (1979), S. 148ff.; Lenstra (1982), S. 205f. Vgl. hierzu auch die Ausführungen zur probabilistischen Untersuchung von approximativen und heuristischen Lösungsalgorithmen für NP-vollständige Probleme auf S. 45.

79) Vgl. zur Problematik dieser Voraussetzung die Anmerkungen auf S. 53 zur Validität komplexitätstheoretischer Erkenntnisse.

den, in realitätskonformer Weise zu. Aufgrund dieser Verteilung wird durch analytisch-statistische Methoden der Erwartungswert des Lösungsaufwands als durchschnittliche Problemkomplexität abgeleitet.

Die simulative Variante[80)] der average case-Analyse geht von einer endlichen Menge exemplarischer Problemausprägungen aus, bezüglich derer angenommen wird, daß sie eine repräsentative Stichprobe aus der Menge aller möglichen Problemausprägungen darstellt. Der idealtypisch erforderliche Nachweis, daß die Repräsentativitätsprämisse von dieser Stichprobe tatsächlich erfüllt wird, könnte dadurch erfolgen, daß die Stichprobe - in Anlehnung an die analytische Variante der average case-Analyse - auf der Grundlage einer realitätskonformen Wahrscheinlichkeitsverteilung über allen möglichen Problemausprägungen erzeugt wird. Bei der praktischen Anwendung der simulativen Analysevariante wird jedoch in der Regel auf diesen Validitätsnachweis verzichtet[81)].

Mittels Computersimulationen wird untersucht, welchen durchschnittlichen Lösungsaufwand die Bewältigung des betrachteten Problems bereitet. Als Maßstab dient hierfür die durchschnittliche Rechenzeit für die Problemausprägungen der Stichprobe.

Worst und average case-Analysen unterscheiden sich wesentlich durch ihre theoretische Grenzfallbetrachtung einerseits und ihre praktische - zumindest idealtypische - Anknüpfung an die Eintrittswahrscheinlichkeiten unterschiedlicher Problemausprägungen in realen Problemkontexten anderseits. Entsprechend werden ihre Ergebnisse bezüglich des Lösungsaufwands von Problemen unter die Begriffe der theoretischen bzw. praktischen Komplexität[82)] subsumiert.

---

80) Vgl. Brown (1981), S. 588ff.; Hall (1986), S. 274f.; vgl. auch Schittkowski (1980), S. 51ff. u. 64ff., und Rardin (1982), S. 8ff., die besondere Aufmerksamkeit der Generierung von Problemausprägungen widmen.

81) Vgl. Simon (1976), S. 298; Rardin (1982), S. 12.

82) Diese Begriffsdifferenzierung wird später auf die Unterscheidung zwischen theoretischer bzw. praktischer (In-)Effizienz übertragen; vgl. zu dieser Dichotomie z.B. Derigs (1986), S. 48.

Die Komplexität von Problemen wird seitens der Komplexitätstheorie auf die Effizienz von Algorithmen zurückgeführt. Die Effizizienz eines Algorithmus wird hierbei als Inverse seines Ausführungsaufwands verstanden. Als Maßstab der Komplexität eines Problems gilt die Effizienz derjenigen Algorithmen, die bezüglich der Menge aller Algorithmen, die das betrachtete Problem zu lösen vermögen, die höchste Effizienz aufweisen (bestmögliche Algorithmen). Sofern diese Algorithmen konkret bekannt sind, werden sie als bestbekannte Algorithmen bezeichnet. Des weiteren läßt sich zwischen einer Problemkomplexität im strengen und einer im schwachen Sinne unterscheiden.

Bei der Komplexität im strengen Sinne[83)] wird durch die Effizienz der bestbekannten Algorithmen nur eine obere Grenze der Problemkomplexität festgelegt. Da grundsätzlich unbekannte Algorithmen mit höherer Effizienz als die bestbekannten Algorithmen existieren können, besteht die Möglichkeit, zukünftig zu entdecken, daß die tatsächliche Problemkomplexität infolge solcher effizienteren Algorithmen niedriger ist als bisher vermutet. Daher stellen bestbekannte Algorithmen immer nur eine vorläufige Obergrenze der Problemkomplexität dar. Wesentlich schwieriger[84)] - und dementsprechend seltener - ist der Beweis zu führen, daß grundsätzlich kein Algorithmus existieren kann, der ein Problem effizienter löst, als durch eine bestimmte untere Komplexitäts-Schranke[85)] angegeben wird[86)].

---

83) Vgl. Bachem (1980), S. 833; Traub (1982), S. 60; Schrader (1983), S. 1.

84) Vgl. Schnorr (1974), S. 182; Savage (1976), S. 347; Traub (1982), S. 60; Cook (1983), S. 403f.

85) Der Beweis muß nicht konstruktiv sein, d.h. er braucht nicht die Spezifizierung eines prinzipiell bestmöglichen Algorithmus zu umfassen; vgl. Valiant (1978), S. 331; Bachem (1980), S. 815f. u. 829. Deshalb kann der Fall eintreten, daß zwar eine obere Schranke der Effizienz möglicher Algorithmen aufgezeigt wird, aber ein Algorithmus mit einer solchen bestmöglichen Effizienz - vorläufig - unbekannt bleibt. Daher wird hier nicht von einer unteren Grenze, sondern nur von einer unteren Schranke der Problemkomplexität gesprochen.

86) Vgl. Hopcroft (1974), S. 625; Traub (1982), S. 60; Cook (1983), S. 403ff. Vgl. hierzu auch die Ausführungen auf S. 45f.

Die theoretische Problemkomplexität, die durch die Effizienz der bestmöglichen Algorithmen definiert wird, bleibt unbekannt. Es läßt sich im strengen Sinne nur feststellen, daß sie nicht oberhalb der Obergrenze der bestbekannten Algorithmen und - gegebenenfalls - nicht unterhalb der unteren Schranke unmöglicher Algorithmen liegt. Diese Problemkomplexität im strengen Sinne stellt die zeitinvariante Komplexitätsversion dar.

Die Komplexität im schwachen Sinne[87)] wird mit der Effizienz der bestbekannten Algorithmen identifiziert. Diese Füllung des Komplexitätsbegriffs erweist sich zwar "griffiger" als die vorgenannte, erlaubt aber nur temporär zutreffende Aussagen, weil die Gültigkeit eines Komplexitätsurteils von der jeweils aktuellen Gesamtheit aller bekannten Lösungsalgorithmen abhängt. Die Entwicklung neuartiger Algorithmen kann die Revision solcher vorläufigen Komplexitätsurteile erzwingen. Der einfacheren Diktion halber wird fortan - wenn nicht ausdrücklich anders vermerkt - von der Problemkomplexität im schwachen Sinne ausgegangen[88)].

87) Vgl. Peterson (1981), S. 118; Schrader (1983), S. 3.

88) Auf den hieraus resultierenden Validitätsmangel komplexitätstheoretischer Aussagen wird auf S. 50 eingegangen.

2.4 Komplexitätsklassen zur Beurteilung des Lösungsaufwands von Problemen

2.4.1 Die Klassen P und NP

Die Klasse der entscheidbaren Probleme läßt sich zunächst grob in die zwei (Sub-)Klassen der polynomial[89) beschränkten und der exponentiell[90) beschränkten Probleme (Klasse "P" bzw. "NP") aufgliedern. Mit dieser Klassifizierung ist zugleich eine Beurteilung der Problemschwierigkeit und der Effizienz der jeweils betroffenen, bestbekannten Lösungsalgorithmen verbunden[91). Probleme aus der Klasse P gelten als einfach oder leicht zu handhaben[92); die zugehörigen bestbekannten polynomial beschränkten Algorithmen werden als effizient bezeichnet. Zuweilen wird zwischen worst und average case-Analysen bezüglich solcher Probleme ein implikativer Zusammenhang vermutet[93): Wenn für ein Problem unter den theoretischen Annahmen von worst

89) Ein Polynomial g-ten Grades ist die Erweiterung eines Polynoms g-ten Grades in der Weise, daß jeder Summand eines Polynoms, der nur höchstens eine potenzierte Variable (mit dem maximalen Exponenten g) besitzen darf, durch ein Produkt aus endlich vielen potenzierten Variablen (mit den maximalen Exponenten g) substituiert wird; vgl. Hack (1975), S. 101. Es wird in der Komplexitätstheorie in der Regel ein erweiterter Polynomialbegriff verwendet, der nicht nur die Potenzierung, sondern auch die Logarithmierung von Variablen zuläßt; vgl. z.B. Peterson (1981), S. 118. Abweichender Ansicht sind Jones (1977), S. 280; Paul (1978), S. 91f., und Parker (1982a), S. 4, die logarithmisch beschränkte Komplexitätsfunktionen gleichberechtigt neben polynomial beschränkte stellen.

90) Analog zur vorangehenden Fußnote werden unter den Begriff der exponentiellen Terme auch solche subsumiert, die im allgemeinen nicht als Werte von Exponentialfunktionen definiert sind, aber immerhin noch die Struktur eines aus Basis und Exponent zusammengesetzten Terms aufweisen; vgl. Garey (1979), S. 6; French (1982), S. 140.

91) Vgl. zu den nachfolgenden Komplexitätsurteilen Karp (1972), S. 87; Lawler (1976), S. 5f.; Müller-Merbach (1976), S. 71; Ullman (1976), S. 140; Ecker (1977), S. 55; Lenstra (1977), S. 344; Garey (1979), S. 7f.; Bachem (1980), S. 821; Lenstra (1982), S. 202; French (1982), S. 142; Parker (1982a), S. 7.

92) Schönlein (1981), S. 116, spricht von einem "vertretbarem" Aufwand, der nur von polynomial beschränkten Algorithmen garantiert werden könne.

93) Vgl. Bland (1981), S. 1042.

case-Analysen ein polynomial beschränkter Algorithmus existiert, dann kann dieser Algorithmus auch in der Praxis effizient zur Problemlösung eingesetzt werden, was sich wiederum in entsprechenden average case-Analysen nachweisen lassen müßte.

Probleme aus der Klasse NP werden - sofern sie nicht in der Klasse P ebenso enthalten sind[94)] - als schwierig[95)] beurteilt; die zugehörigen bestbekannten exponentiell beschränkten Lösungsalgorithmen gelten als ineffizient.

Die Plausibilität der Annahmen, polynomiale und exponentielle Beschränktheit als indikatoren für die Effizienz bzw. Ineffizienz von Lösungsalgorithmen zu betrachten, läßt sich durch ein Beispiel[96)] verdeutlichen: Bei einem Problemumfang von n=60 verursacht ein polynomial beschränkter Algorithmus mit $O(n^2)$ einen Lösungsaufwand von etwa 0,0036 sec, während ein exponentiell beschränkter Algorithmus mit $O(2^n)$ ca. 366 Jahre zur Problemlösung benötigt[97)].

Die Klasse P[98)] umfaßt alle Probleme (kurz: P-komplexe Probleme)[99)], die durch einen deterministischen Turing-Automaten[100)] in polynomial beschränkter Zeit

---

94) Es wird auf S. 34 ausgeführt, daß die Klasse P eine Teilklasse der Klasse NP ist, so daß Probleme aus der Klasse NP zugleich zur Klasse P gehören können. Daher stellen die Klassen P und NP keine (disjunkte) Zerlegung der (Super-)Klasse aller entscheidbaren Probleme im mathematisch strengen Sinne dar.

95) Hierfür wird oftmals die plastische Bezeichnung "nicht-handhabbarer" ("intractable") Probleme verwendet; vgl. Lenstra (1979), S. 122; Garey (1979), S. 8, 11f. u. 34. Ähnlich äußert sich Ecker (1977), S. 55.

96) Vgl. zu weiteren Beispielen Garey (1979), S. 7f.; Bachem (1980), S. 820f.; French (1982), S. 142; Parker (1982a), S. 4; Kolata (1984), S. 1379.

97) Vgl. Garey (1979), S. 7, Figure 1.2.

98) Das Acronym "P" bezieht sich auf die o.a. "p"olynomiale Beschränkung der Komplexitätsfunktion; vgl. Parker (1982a), S. 8.

99) Vgl. zur Klasse der P-komplexen Probleme Karp (1972), S. 86ff.; Schnorr (1974), S. 132; Karp (1975b), S. 52; Garey (1979), S. 27; Brucker (1981), S. 29ff. u. 65ff.; Parker (1982a), S. 8.

100) Deterministische Turing-Automaten wurden bereits auf S. 12ff. vorgestellt.

gelöst werden können[101]. Die Komplexitätsfunktion läßt sich als ein Polynomial (einschließlich logarithmischer Terme) beliebigen Grades darstellen. Da gemäß der o.a. Church-Post-Turing-These jeder Algorithmus als Turing-Automat ausgedrückt werden kann, läßt sich äquivalent konstatieren: Alle Probleme aus der Klasse P sind dadurch gekennzeichnet, daß sie sich durch Algorithmen mit polynomial beschränktem Lösungsaufwand bewältigen lassen. P-komplexe Probleme werden daher - vereinfachend - auch als polynomial beschränkte Probleme bezeichnet. Aus der Sicht des Operations Research ist eines der bedeutsamsten p-komplexen Probleme dasjenige, das erfordert zu entscheiden, ob ein linear-homogenes, diophantisches Gleichungssystem der Form $\underline{Ax} = \underline{0}$ (mindestens) eine ganzzahlige Lösung $\underline{x}$ besitzt[102].

Innerhalb der Klasse der polynomial beschränkten Probleme erfolgt im allgemeinen[103] keine weitergehende komplexitätstheoretische Differenzierung. Daher kann das Konzept der polynomialen Komplexitätsfunktionen als Basiskonzept der Komplexitätstheorie betrachtet werden. Probleme, die sich so weit ähneln, daß ihre Komplexitätsfunktionen lediglich durch Polynomiale unterschiedlichen Grades ausgedrückt werden, gelten als äquivalent. Diese Äquivalenzklassenbildung nach dem Kriterium der Irrelevanz polynomial beschränkter Komplexitätsdifferenzen liegt auch allen nachstehend erörterten Konzepten der Komplexitätstheorie zugrunde[104].

---

101) Vgl. Ullman (1976), S. 145; Lenstra (1979), S. 123; Brucker (1981), S. 146ff.

102) Der polynomial beschränkte Lösungsalgorithmus für dieses Problem ist zugleich konstruktiv, d.h. er gibt auch die ganzzahlige Lösung konkret an, falls diese existiert. Vgl. hierzu von zur Gathen (1976), S. 64f.

103) Eine Ausnahme stellt die Betrachtung der logarithmisch beschränkten Probleme dar, die eine Subklasse der polynomial beschränkten Probleme bilden; vgl. Garey (1979), S. 177ff.

104) Vgl. Parker (1982a), S. 4.

Zur Klasse NP[105] gehören alle Probleme (kurz: NP-komplexe Probleme)[106], die sich mit der Hilfe von nondeterministischen Turing-Automaten[107] in polynomial beschränkter Weise lösen lassen[108]. Ein nondeterministischer Turing-Automat kann als ein Ensemble aus deterministischen (Sub-)Turing-Automaten vorgestellt werden. Bei der Problemlösung wird - wie z.B. bei branch and bound-Algorithmen - ein Suchbaum konstruiert. In jedem Knoten dieses Baums, der kein Endknoten ("Blatt") ist, wird je ein Sub-Automat auf die Analyse desjenigen Teilproblems angesetzt, das in diesem Knoten zu lösen ist. Da für ein Problem die räumliche Ausdehnung des Suchbaums - hier verstanden als die Anzahl seiner (verzweigten) Knoten - mit dem Problemumfang anwächst und dieser Umfang bei komplexitätstheoretischen Analysen beliebig groß sein darf, muß das Ensemble der Sub-Automaten potentiell unbegrenzt sein und mit dem Problemumfang exponentiell beschränkt anwachsen.

Da reale Automaten nur aus endlich vielen Komponenten bestehen können, läßt sich das Konzept der nondeterministischen Turing-Automaten grundsätzlich nicht verwirklichen. Es stellt eine automaten- und komplexitätstheoretische Fiktion dar[109]. Es ist allerdings möglich aufzuzeigen, daß die Funktionsweise solcher

---

105) Das Acronym "NP" kann in der Weise interpretiert werden, daß die Zeitkomplexität von deterministischen Turing-Automaten, die auf Probleme aus dieser Klasse angewendet werden, "n"icht "p"olynomial beschränkt ist. Parker (1982a), S. 9, erklärt das Akronym durch den Sachverhalt, daß Probleme aus dieser Klasse "n"ondeterministisch "p"olynomial seien.

106) Vgl. zur Klasse der NP-komplexen Probleme Karp (1972), S. 91ff.; Lenstra (1977), S. 344ff.; Garey (1979), S. 29ff.; Bachem (1980), S. 822ff., insbesondere S. 827; Brucker (1981), S. 41ff., insbesondere S. 47ff. u. 52; Parker (1982a), S. 8f.

107) Vgl. Schnorr (1974), S. 183f.; Karp (1975b), S. 45, 47 u. 58 (nondeterministische Algorithmen); Ullman (1976), S. 144f.; Häussler (1976), S. 20f.; Savage (1976), S. 348f.; Zervos (1977), S. 297ff.; Garey (1979), S. 30ff. i.V.m. S. 28f.; Lenstra (1979), S. 123; Brucker (1981), S. 42 u. 153ff.

108) Vgl. Karp (1972), S. 91f.; Schnorr (1974), S. 183; Ullman (1976), S. 145; Garey (1979), S. 13, 17, 29 u. 31f.; Lenstra (1979), S. 123; Bachem (1980), S. 823; Brucker (1981), S. 155f.; Li (1984), S. 212 (mittelbar).

109) Vgl. Lenstra (1979), S. 123; Garey (1979), S. 29.

fiktiven Automaten immer durch reale Automaten - die sich stets auf deterministische Turing-Automaten abbilden lassen - mit exponentiell beschränkter Zeitkomplexität simuliert werden kann[110]. Daher ist es möglich, NP-komplexe Probleme im schlechtesten Fall durch exponentiell beschränkte und real implementierbare Algorithmen zu lösen[111]. Aus diesem Grund werden Probleme aus der Klasse NP oftmals auch als exponentiell[112] - oder "superpolynomial"[113] - beschränkte Probleme bezeichnet. Es bleibt offen, ob sich für diese NP-komplexen Probleme auch effizientere als exponentiell beschränkte Lösungsalgorithmen finden lassen[114].

Die baumartige Struktur der nondeterministischen Turing-Automaten kann auch in der Weise interpretiert werden, daß NP-komplexe Probleme leicht zu verifizieren, aber schwer zu lösen sind[115]. Die leichte Verifizierungsmöglichkeit bedeutet, daß eine exogen vorgeschlagene Problemlösung mit polynomial beschränktem Aufwand hinsichtlich ihrer Zulässigkeit überprüft werden kann. Dies entspricht dem Einsatz eines deterministischen Turing-Automaten in einem Knoten des o.a. Suchbaums. Da aber die Anzahl möglicher Problemlösungen in Abhängigkeit vom Problemumfang exponentiell beschränkt anwächst, müssen zum Auffinden der Problemlösung im ungünstigsten Fall, daß alle vorgegebenen und überprüften Lösungen - bis auf die letzte Lösung - unzulässig sind, exponentiell beschränkt viele Verifizierungsschritte ausgeführt werden. Entsprechend wird im o.a. Bild der gesamte Suchbaum möglicher Problemlösungen erzeugt.

Da P-komplexe Probleme durch einen deterministischen Turing-Automaten mit polynomial beschränktem Aufwand gelöst - nicht nur hinsichtlich einer vorgeschlagenen

---

110) Vgl. Cook (1971), S. 157; Ullman (1976), S. 145; Bachem (1980), S. 823 i.V.m. S. 828f.
111) Vgl. Garey (1979), S. 32f. (mit Beweis); Lenstra (1979), S. 122.
112) Vgl. Garey (1979), S. 6.
113) Vgl. Lenstra (1976), S. 1; Lenstra (1979), S. 122.
114) Näheres hierzu auf S. 38ff. zur Frage: P=NP?; vgl. auch die Anmerkungen auf S. 45.
115) Vgl. zum folgenden Parker (1982a), S. 8f.; Karp (1986), S. 104; weniger deutlich auch Garey (1979), S. 28 u. 30ff.

Lösung verifiziert - werden können, sind sie sowohl leicht zu verifizieren als auch leicht zu lösen. Erst recht lassen sie sich durch nondeterministische Turing-Automaten in polynomial beschränkter Weise lösen, weil deterministische Turing-Automaten (degenerierte) Spezialfälle ihrer nondeterministischen Pendants darstellen[116]. Daher ist die Klasse P notwendig eine - unter Umständen[117] unechte - Teilklasse der Klasse NP[118].

2.4.2 Die Klasse NP-vollständiger Probleme

Zur weiterreichenden Differenzierung der NP-komplexen Probleme werden die Klassen der NP-harten und der NP-vollständigen Probleme eingeführt.

Ein Problem ist genau dann NP-hart[119], wenn alle NP-komplexen Probleme auf dieses Problem reduziert werden können. Ein Problem P_i läßt sich auf ein Problem P_j reduzieren[120], wenn für jede Ausprägung von Problem P_i mit der Hilfe eines polynomial beschränkten Algorithmus eine Ausprägung von Problem P_j in der Weise konstruiert werden kann, daß sich jede Lösung der Ausprägung von Problem P_j in die Lösung einer Ausprägung von Problem P_i zurücktransformieren läßt. Die Reduzierbarkeit von Problem P_i auf Problem P_j bedeutet, daß Problem P_i als

116) Vgl. Häussler (1976), S. 21; Mehlhorn (1977), S. 193; Garey (1979), S. 32.
117) Näheres hierzu auf S. 38ff. zur Frage: P=NP?
118) Vgl. Lenstra (1977), S. 344; Garey (1979), S. 32; Bachem (1980); S. 823; Parker (1982a), S. 9; Lenstra (1982), S. 202; Karp (1986), S. 104.
119) Vgl. Hartmanis (1976a), S. 32; Mehlhorn (1977), S. 193; Garey (1979), S. 109ff.; Lenstra (1979), S. 123; Parker (1982b), S. 83; Clausen (1986), S. 32.
120) Vgl. Karp (1972), S. 86 u. 88; Karp (1975b), S. 54; Garey (1975), S. 399; Ullman (1976), S. 142; Brucker (1976b), S. 669; Paul (1978), S. 182; Valiant (1978), S. 330; Lenstra (1979), S. 123; Garey (1979), S. 34ff. u. 111; Brucker (1981), S. 54f.; Peterson (1981), S. 117 u. 119; Parker (1982a), S. 6f.; Goldberg (1984), S. 45.
Es wird hier nicht auf die - in den vorgenannten Quellen enthaltene - Differenzierung zwischen den beiden Konzepten der polynomial beschränkten Reduzierbarkeit und Transformierbarkeit eingegangen, weil diese Konzepte zu äquivalenten Ergebnissen führen; vgl. Garey (1979), S. 118.

ein Spezialfall von Problem P_j betrachtet werden kann und daß Problem P_j mindestens so schwierig zu lösen - mindestens so komplex - wie Problem P_i ist[121]. Wenn ein Lösungsalgorithmus für Problem P_j existiert, so ist ist es möglich, hieraus einen Lösungsalgorithmus für Problem P_i derart abzuleiten, daß die Ressourcenanforderungen beider Algorithmen nicht mehr als polynomial beschränkt voneinander abweichen. Infolge dieses Reduzierungszusammenhangs ist jedes NP-harte Problem mindestens so komplex wie jedes Problem aus der Klasse NP. Ein NP-hartes Problem braucht aber nicht selbst in der Klasse NP enthalten zu sein[122].

Ein Problem heißt NP-vollständig, wenn es NP-hart ist und darüber hinaus auch zur Klasse NP gehört[123]. Da einerseits NP-harte Probleme nicht notwendig zur Klasse NP zählen müssen, diese Zugehörigkeit aber anderseits für NP-vollständige Probleme vorausgesetzt wird, ist die Klasse der NP-vollständigen Probleme eine - echte oder unechte - Teilklasse der Klasse der NP-harten Probleme[124].

Cook bewies die zentrale komplexitätstheoretische Erkenntnis, daß alle NP-komplexen Probleme auf das Erfüllbarkeits-Problem der Aussagenlogik[125] reduziert

---

121) Vgl. Lenstra (1979), S. 123.

122) Vgl. Garey (1979), S. 109. Vgl. hierzu auch die Ausführungen in Fußnote 18) zu Optimierungsproblemen, deren zugrundeliegenden Entscheidungsprobleme zwar als NP-vollständig nachgewiesen sind, deren eigene NP-Vollständigkeit aber nicht gesichert ist. Diese Optimierungsprobleme sind mindestens so komplex wie ihre NP-vollständigen Entscheidungsprobleme, also NP-hart.

123) Vgl. Hartmanis (1976a), S. 32; Lenstra (1979), S. 123; Parker (1982b), S. 84; Clausen (1986), S. 32.

124) Vgl. Lenstra (1982), S. 201.

125) Das Erfüllbarkeits-Problem ist das Problem zu entscheiden, ob eine aussagenlogische Formel in konjunktiver Normalform den Wahrheitswert "wahr" für mindestens eine Belegung ihrer atomaren Aussagen mit den Wahrheitswerten "wahr" oder "falsch" annimmt. Eine solche Formel ist eine komplexe Aussage, die nur aus atomaren Aussagen, deren Negaten und dem logischen "und" zusammengesetzt ist. Vgl. hierzu Lenstra (1979), S. 123; Garey (1979), S. 38f.; Parker (1982b), S. 84.

werden können[126]. Dieses Erfüllbarkeits-Problem stellt somit den "Prototyp"[127] der NP-vollständigen Probleme dar.

Das Interesse der Komplexitätstheorie konzentriert sich auf die NP-vollständigen Probleme[128]. Sie stellen strenggenommen nur die schwierigsten Problem aus der Klasse NP dar[129], werden aber bei oberflächlicher Betrachtung als Inbegriff der schwierigsten Probleme überhaupt angesehen[130]. Auch aus dem Blickwinkel des Operations Research erweist sich die Mehrzahl "interessanter" Probleme als NP-vollständig[131]. Hierauf wird später ausführlicher eingegangen[132].

Wenn ein Problem als NP-vollständig nachgewiesen wird, gilt dies als Rechtfertigung, zu seiner Lösung nicht nach effizienten, d.h. polynomial beschränkten Algorithmen zu suchen, sondern sofort auf enumerative

---

126) Vgl. Cook (1971), S. 155f.; Karp (1972), S. 90 u. 92f.; Ullman (1976), S. 147ff.; Häussler (1976), S. 22ff., insbesondere S. 25; Savage (1976), S. 352ff.; Schuster (1976), S. 36ff.; Garey (1979), S. 39ff.; Brucker (1981), S. 160ff.; Parker (1982b), S. 83f.; Karp (1986), S. 104f.

127) Karp (1986), S. 104, spricht von einem "archetypischen" ("archetypal") Problem.

128) Vgl. die in Fußnote 106) angegebenen Quellen zu Problemen aus der Klasse NP, die als thematischen Schwerpunkt zumeist die NP-Vollständikeit von Problemen erörtern. Vgl. als spezielle Abhandlungen über NP-vollständige Probleme auch Garey (1979), S. 13f., 37ff., 118ff. (als Skizze der historischen Entwicklung der Analyse NP-vollständiger Probleme) u. 121ff. Diese Quelle gilt als Standardwerk der Theorie NP-vollständiger Problem. Vgl. des weiteren Karp (1972), S. 93ff.; Ullman (1973), S. 96ff.; Sahni (1974), S. 28ff.; Hartmanis (1976a), S. 30ff.; Ullman (1976), S. 139ff.; Savage (1976), S. 347ff., insbesondere S. 356ff.; Schuster (1976), S. 36ff.; Garey (1978), S. 499ff.; Brucker (1981), S. 145ff.
Abweichend stellen Karp (1976), S. 2ff., und Parker (1982b), S. 83ff., die - allerdings eng verwandten - NP-harten Probleme in den Vordergrund.

129) Vgl. Mehlhorn (1977), S. 201; Lenstra (1978), S. 23; Garey (1979), S. 37; Bachem (1980), S. 827; Lenstra (1982), S. 202.

130) Vgl. z.B. Valiant (1978), S. 330.

131) Die gleiche Ansicht vertritt Hartmanis (1976a), S. 30, ohne Bezug zum Operations Research.

132) Vgl. Abschnitt 3.

Algorithmen zurückzugreifen[133], die infolge ihrer strukturbedingten exponentiellen Beschränktheit als ineffizient eingestuft werden. Diese Rechtfertigung wird aus der Kongruenz zwischen der baumartigen Struktur von nondeterministischen Turing-Automaten, welche die NP-komplexen und somit auch die NP-vollständigen Probleme zu lösen vermögen, und der ebenfalls baumartigen Struktur von enumerativen Algorithmen gefolgert[134]. Die bekanntesten Vertreter solcher Algorithmen stammen aus den - nicht disjunkten - Klassen der branch and bound- und der backtracking-Algorithmen.

Aufgrund der NP-Vollständigkeit eines Problems kann aber nicht nur gerechtfertigt werden, ineffiziente Algorithmen zu seiner exakten Lösung heranzuziehen. Ebenso plausibel ist es, infolge der hohen Problemkomplexität auf eine exakte Lösung zu verzichten und stattdessen approximative[135] oder heuristische[136] Algorithmen anzuwenden[137].

Die beiden vorgenannten Argumentationen zeigen auf, wie aus klassifikatorischen Erkenntnissen der Komplexitätstheorie konkrete Handlungsempfehlungen für die Algorithmenauswahl im Bereich des Operations Research abgeleitet werden können. Allerdings sind diese Empfehlungen nur bedingt gerechtfertigt, weil sie auf der Voraussetzung beruhen, daß für NP-vollständige Probleme keine effizienten (exakten) Lösungsalgorithmen existieren. Tatsächlich konnte die Korrektheit dieser Annahme bisher jedoch weder bewiesen noch widerlegt werden.

---

133) Vgl. Brucker (1975), S. 5 u. 26; Ullman (1976), S. 139; Brucker (1976a), S. 366f.; Lenstra (1977), S. 345f. u. 360; Lenstra (1979), S. 122; Wehr (1980), S. 71; French (1982), S. 176; Lenstra (1982), S. 202.

134) Dies geht mittelbar aus Karp (1975b), S. 45, 47 u. 50, hervor.

135) Diese Algorithmen garantieren, eine Problem"lösung" zu finden, deren Abweichung von der exakten Lösung ein maximal vorgegebenes Maß nicht überschreitet.

136) Diese Algorithmen führen allenfalls zufällig zu den exakten Problemlösungen, erzeugen im Regelfall jedoch nur Näherungslösungen, deren Abstand von der exakten Lösung nicht nach oben beschränkt werden kann.

137) Vgl. Lenstra (1979), S. 122.

Die Beantwortung der Frage, ob für mindestens ein NP-vollständiges Problem ein polynomial beschränkter, d.h. effizienter Lösungsalgorithmus existiert, stellt ein fundamentales, noch ungelöstes (Meta-)Problem der Komplexitätstheorie dar[138]. Einerseits gehört jedes Problem, für das ein polynomial beschränkter Lösungsalgorithmus existiert, zur Klasse P. Anderseits sind infolge des Reduzierungszusammenhangs die Lösungsalgorithmen aller Probleme aus der Klasse NP höchstens so komplex wie der bestbekannte Lösungsalgorithmus für ein NP-vollständiges Problem. Daher würde die Existenz eines polynomial beschränkten Algorithmus für ein NP-vollständiges Problem zugleich alle NP-komplexen Probleme mit polynomial beschränktem Aufwand lösen können[139] und die Identität der Komplexitätsklassen P und NP bedeuten[140]. Da bisher eine Vielzahl von Problemen

---

138) Vgl. Brucker (1976b), S. 669; Hartmanis (1976a), S. 30; Lipton (1978), S. 193; Garey (1979), S. 32; Bachem (1980), S. 828; Korte (1985), S. 16 u. 25.

139) Vgl. Brucker (1975), S. 4; Karp (1975b), S. 50; Garey (1976a), S. 118; Ullman (1976), S. 142 u. 156f.; Mehlhorn (1977), S. 201; Ecker (1977), S. 243; Lenstra (1977), S. 360; Garey (1978), S. 499; Lenstra (1978), S. 23.
Strenggenommen kann nicht aus jedem Existenzbeweis eines polynomial beschränkten Lösungsalgorithmus für ein NP-vollständiges Problem die Lösung aller NP-vollständigen Probleme mit polynomial beschränktem Aufwand abgeleitet werden. Denn wenn dieser Existenzbeweis nicht-konstruktiver Art ist, liegen kein Informationen darüber vor, wie ein einzige dieser Lösungen ermittelt werden könnte. Einige komplexitätstheoretische Arbeiten weisen in die Richtung, daß ein Existenzbeweis - sofern er überhaupt geführt werden kann - nicht-konstruktiv sein muß; vgl. Hajek (1979), S. 227ff.; Bachem (1980), S. 829.

140) Vgl. Mehlhorn (1977), S. 200f.; Lenstra (1978), S. 23; Lenstra (1979), S. 122f.

als NP-vollständig nachgewiesen werden konnte[141)], ohne daß für ein einziges von ihnen ein polynomial beschränkter Algorithmus bekannt ist[142)], wird derzeit allgemein die Vermutung geteilt, daß die Beziehung P=NP nicht erfüllt ist[143)].

Da die Klasse P notwendig eine - echte oder unechte - Teilklasse der Klasse NP ist, würde die Gültigkeit der Vermutung P≠NP implizieren, daß die Klasse P eine echte Teilklasse der Klasse NP darstellt. Weil die NP-vollständigen Probleme per definitionem die komplexesten Probleme aus der Klasse NP sind, folgte aus der

141) Vgl. hierzu die überaus umfangreiche (95 Seiten umfassende!) Auflistung NP-vollständiger Probleme bei Garey (1979), S. 190ff., die neuerdings periodisch aktualisiert wird; vgl. Johnson (1981), S. 393ff. Nach Paul (1978), S. 195, und Bachem (1980), S. 827, sind etwa 2.000 NP-vollständige Probleme bekannt. Vgl. zu weiteren Zusammenstellungen von Problemen aus dieser Klasse Karp (1975a), S. 20ff.; Karp (1975b), S. 60ff.; Brucker (1975b), S. 670f.; Manders (1976), S. 23ff.; Hartmanis (1976a), S. 34ff.; Brucker (1976a), S. 364ff.; Schuster (1976), S. 36ff.; Brucker (1976c), S. 136ff.; Mehlhorn (1977), S. 202ff.; Lenstra (1977), S. 345; Lenstra (1979), S. 125ff.; Brucker (1981), S. 166ff., insbesondere S. 194ff. Vgl. auch die exemplarischen Nennungen in Abschnitt 3.1.

142) Vgl. Schnorr (1974), S. 184; Ullman (1975), S. 385; Garey (1976a), S. 118; Mehlhorn (1977), S. 194; Valiant (1978), S. 331; Garey (1978), S. 499; Garey (1979), S. u. 181; Lenstra (1979), S. 122; Bachem (1980), S. 828; Brucker (1981), S. 53.

143) Vgl. Schnorr (1974), S. 184; Brucker (1975), S. 4; Hartmanis (1976a), S. 30; Brucker (1976a), S. 357 u. 365; Lenstra (1977), S. 344 u. 360; Mehlhorn (1977), S. 193 u. 201; Lenstra (1978), S. 23; Lenstra (1979), S. 123; Garey (1979), S. 33; Bachem (1980), S. 828; Brucker (1981), S. 25 u. 53; Lenstra (1982), S. 202 u. 207; French (1982), S. 146 u. 148.
Vgl. zu den Schwierigkeiten, den Beweis für die Vermutung P≠NP zu leisten, Garey (1979), S. 181ff.; Lenstra (1982), S. 207f.; Parker (1982b), S. 85. Hieraus folgen Zweifel, diesen Beweis überhaupt erbringen zu können. Es drängt sich der Verdacht auf, es könne ein unentscheidbares Problem vorliegen; vgl. Hartmanis (1976b), S. 13ff.; Lipton (1978), S. 193; Bachem (1980), S. 828f.; Hopcroft (1984), S. 49; Korte (1985), S. 16; Karp (1986), S. 104.

Gültigkeit von $P \neq NP$ ebenso[144]), daß für kein NP-vollständiges Problem - und erst recht für kein NP-hartes Problem - ein polynomial beschränkter Lösungsalgorithmus existieren kann. Vielmehr wären alle Lösungsalgorithmen für NP-vollständige Probleme tatsächlich exponentiell beschränkt, da diese Probleme zu den im schlechtesten Falle exponentiell beschränkten NP-komplexen Problemen[145]) gehören. Für die Lösungsalgorithmen von NP-harten Problemen ließe sich sogar nur folgern, daß ihr Ausführungsaufwand mindestens exponentiell beschränkt ist, weil diese Probleme nicht zu den NP-komplexen Problemen zählen müssen. Infolge der allgemeinen Akzeptanz der Gültigkeit der o.a. Vermutung gelten diese Aussagen über den Lösungsaufwand von NP-vollständigen und NP-harten Problemen als nahezu gesicherte komplexitätstheoretische Erkenntnisse.

Unter der Voraussetzung der Gültigkeit von $P \neq NP$ (identisch mit $P \subset NP$) läßt sich die Klasse der NP-komplexen Probleme noch weiter differenzieren[146]). Sie umfaßt dann nicht nur alle P-komplexen Probleme, sondern darüber hinaus eine nicht-leere Klasse NPI von intermediären Problemen, die weder polynomial beschränkt (P-komplex) noch NP-vollständig sind. Offene Probleme, für die sich bisher polynomiale Beschränktheit oder NP-Vollständigkeit nicht nachweisen ließen[147]), werden als Probleme mit einer solchen intermediären Komplexität vermutet.

Lange Zeit galt das Problem der linearen Optimierung bei worst case-Analyse als ein aussichtsreicher Kandidat für solche intermediären Probleme aus der Klasse NP[148]), weil weder seine polynomiale noch seine exponentielle Beschränktheit nachgewiesen werden konnte.

---

144) Vgl. Garey (1979), S. 34, 37, 106, 109 u. 113. Die Vermutung, die Lösungsalgorithmen für NP-vollständige Probleme seien exponentiell beschränkt, teilen auch Ullman (1976), S. 144, und Sethi (1977), S. 322;

145) Vgl. S. 33.

146) Vgl. zum folgenden Garey (1979), S. 37, 77f., 81, insbesondere aber S. 154f.

147) Vgl. zu solchen offenen Problemen Brucker (1975), S. 26f.; Garey (1979), S. 285ff.; Lenstra (1979), S. 136; Lenstra (1982), S. 202ff.

148) Vgl. Garey (1979), S. 155ff.

Erst vor kurzem glückte der Beweis, daß es sich hierbei um ein P-komplexes Problem handelt[149]. Hierauf wird später ausführlich eingegangen[150].

2.4.3 Weiterführende Komplexitätsklassen

Die Vielzahl von NP-vollständigen Problemen hat das Bedürfnis geweckt, diese Problemklasse weiter zu zerlegen[151]. Die vorherrschende Klassifizierung von Problemen entweder als P-komplexe oder als NP-vollständige[152] Probleme[153] wird noch als zu grob, noch als kaum zufriedenstellend empfunden[154]. Da sich die Forschung der Komplexitätstheorie in dieser Richtung im Fluß befindet, werden nachfolgend nur einige Ansätze zur weiterführenden Differenzierung der Problemkomplexität kurz skizziert. Hinsichtlich anderer Konzepte, wie z.B. denen der Problem-Parametrisierung[155], der

---

149) Vgl. Johnson (1981), S. 399f.; Lenstra (1982), S. 204.

150) Vgl Abschnitt 3.2.2.

151) Vgl. Lenstra (1977), S. 360; Lenstra (1978), S. 24; Hansen (1979), S. 177. Dagegen war Hartmanis (1976a), S. 30ff., noch daran interessiert, die Gleichheit aller NP-vollständigen Probleme bis auf polynomial beschränkte Isomorphismen nachzuweisen.

152) Dies gilt nur in bezug auf Entscheidungsprobleme. Im Hinblick auf Optimierungsprobleme sind NP-äquivalente Probleme, auf die in Fußnote 19) eingegangen wurde, anstelle von NP-vollständigen Problemen anzusetzen; vgl. Parker (1982b), S. 85.

153) Vgl. Lenstra (1977), S. 346; Valiant (1978), S. 330; Parker (1982b), S. 85.

154) Vgl. Brucker (1975), S. 27; Rinnooy Kan (1976), S. 134 u. 140; Lenstra (1977), S. 346; Lenstra (1978), S. 23f.; Rardin (1982), S. 9; Clausen (1986), S. 35.

155) Vgl. Garey (1975), S. 410f.; Ecker (1977), S. 57; Garey (1979), S. 106f.; Fisher (1982), S. 29. Der Problemumfang n wird durch einen Vektor $\underline{n}'$ von charakteristischen Parametern, wie z.B. den Auftrags- und die Maschinenanzahlen bei Problemen der Maschinenbelegungsplanung, ersetzt. Für die modifizierte Komplexitätsfunktion $O(\underline{n}')$ resultiert u. U. eine polynomiale Beschränkung.

probabilistischen Algorithmen[156] und der Stopp-Algorithmen[157], wird auf die einschlägige Literatur verwiesen.

Ein Problem wird als schwach-NP-vollständig oder pseudopolynomial beschränkt bezeichnet, wenn es bei der üblichen binären Problemcodierung NP-vollständig ist, zugleich aber ein polynomial beschränkter Algorithmus existiert, der dieses Problem bei unärer Codierung zu lösen vermag[158]. Eine Problemcodierung heißt unär, wenn die Länge L(z) einer zu codierenden (ganzen) Zahl z aus der Problembeschreibung eine lineare Funktion

---

156) Vgl. Karp (1975a), S. 24ff.; Rabin (1976), S. 21ff.; Lenstra (1976), S. 1ff.; Solovay (1977), S. 84f.; Valiant (1978), S. 332f.; Bachem (1980), S. 838f.; Lenstra (1982), S. 205f.; Cook (1983), S. 405; Karp (1986), S. 108f.
Die probabilistischen Lösungsalgorithmen dürfen nicht mit probabilistischen average case-Analysen verwechselt werden. Während letzte Wahrscheinlichkeitsverteilungen über den Spektren möglicher Problemausprägungen voraussetzen, beruhen erste auf probabilistischen Komponenten (Zufallsgeneratoren) innerhalb der Lösungsalgorithmen. Solche Algorithmen können nicht mehr durch (non-)deterministische Turing-Automaten der o.a. Art realisiert werden, sondern erfordern den Übergang zum Typ der Orakel-Turing-Automaten. Bei diesen Automaten wird gelegentlich ein "Orakel" befragt, von dessen - automatenexterner - Beantwortung die Fortsetzung der Automatenoperationen abhängt. Im Falle der probabilistischen Algorithmen wird die Orakelantwort durch einen Zufallsgenerator erzeugt. Vgl. zu Orakel-Turing-Automaten (Orakel-Algorithmen) Garey (1979), S. 111ff.; Bachem (1980), 829ff. NP-vollständige Probleme, die durch probabilistische Algorithmen mit polynomial beschränktem Aufwand "gelöst" werden können, konstituieren die Klasse R.

157) Vgl. Simon (1976), S. 287f.
Es werden Stoppregeln hinsichtlich eines maximalen Lösungsaufwands eingeführt, bei dessen Erreichen - bis dahin erfolglose - Versuche der Problemlösung abgebrochen werden. Auf dieser Grundlage lassen sich die Wahrscheinlichkeit, daß eine Problemausprägung innerhalb dieser Aufwandsbegrenzung gelöst werden kann, und die Komplexität der Stopp-Algorithmen ermitteln.

158) Vgl. Rinnooy Kan (1976), S. 135 i.V.m. S. 70f.; Garey (1976a), S. 119; Lenstra (1977), S. 346; Gonzalez (1978), S. 37; Garey (1978), S. 500ff.; Lenstra (1978), S. 24; Garey (1979), S. 91ff.; Lenstra (1979), S. 133; Parker (1982a), S. 6; Parker (1982b), S. 86; Clausen (1986), S. 35. Ein sehr ähnliches Konzept "starker Polynomialität", das auch an der unären Problemcodierung anknüpft, wird von Tardos (1986), S. 250 u. 255f., vertreten.

darstellt[159]. Da sich pseudopolynomial beschränkte Probleme bei unärer Codierung wie P-komplexe Probleme verhalten, werden sie als weniger komplex betrachtet als die komplementären stark-NP-vollständigen Probleme, die trotz unärer Codierung NP-komplex bleiben. Es konnte nachgewiesen werden, daß in der Klasse der NP-vollständigen Probleme in diesem Sinne sowohl schwache (pseudopolynomiale)[160] als auch starke[161] Komplexitätsvarianten existieren. Das Konzept der schwach-NP-vollständigen Probleme erweist sich aus der Sicht des Operations Research besonders interessant, weil NP-vollständige Probleme nur dann pseudopolynomial sein können, wenn die Problemspezifizierung die Codierung von Zahlen umfaßt (numerische Probleme) und falls die o.a. Vermutung P≠NP korrekt ist[162].

Die Komplexität von Problemen, die sich bei der - bisher implizit unterstellten - Suche nach exakten Lösungen als NP-vollständig erweisen, läßt sich auch mit

159) Dies wird in der Regel dadurch erreicht, daß die Zahl z durch ein Wort aus z oder z+1 (Berücksichtigung der Null) identischen Symbolen codiert wird; vgl. Savage (1976), S. 182; Garey (1978), S. 506; Bachem (1980), S. 817; Parker (1982a), S. 6; Hopcroft (1984), S. 35.

160) Als erstes schwach-NP-vollständiges Problem wurde das knapsack-Problem nachgewiesen; vgl. Lenstra (1979), S. 133; Garey (1979), S. 135ff.; Parker (1982b), S. 86.

161) Das 3-Partitions-Problem war das erste NP-vollständige Problem, das trotz unärer Codierung NP-komplex blieb; vgl. Garey (1978), S. 504; Lenstra (1979), S. 133. Vgl. zu stark-NP-vollständigen Problemen Garey (1979), S. 95ff.

162) Vgl. Garey (1979), S. 95. Der Bezug auf numerische Probleme ergibt sich aus der Besonderheit, daß erst die Möglichkeit der unären Zahlencodierung die Existenz polynomial beschränkter Lösungsalgorithmen zuläßt.

der Hilfe von Approximations-Algorithmen[163] abstufen. Hierbei wird durch worst case-Analysen untersucht, welche Annäherungen an die exakten Lösungen garantiert werden können, wenn nur noch polynomial beschränkte Lösungsalgorithmen zugelassen werden. Beispielsweise gibt es unter diesen Voraussetzungen einige wenige einfach-NP-vollständige Probleme, für welche die maximale relative Abweichung von der exakten Lösung beliebig klein gewählt werden kann und der Lösungsaufwand bei Verminderung dieser Abweichung nur polynomial beschränkt anwächst[164]. Dagegen existiert auch eine große Anzahl von moderat-NP-vollständigen Problemen, bei denen sich nur eine konstante, von Null verschiedene Lösungsabweichung garantieren läßt[165]. Ferner gibt es schwer-NP-vollständige Probleme, für welche die Einhaltung keiner - sei sie auch noch so großen - maximalen relativen Abweichung von der exakten Lösung gewährleistet werden kann, sofern die Vermutung P≠NP richtig ist[166].

---

163) Vgl. Sahni (1974), S. 30ff.; Brucker (1976b), S. 670ff., insbesondere S. 672; Savage (1976), S. 364ff.; Sahni (1976), S. 116f. u. 123ff.; die annotierte Bibliographie von Garey (1976b), S. 43ff.; Ecker (1977), S. 56f. u. 61ff., 241 u. 250ff.; Lenstra (1977), S. 360; Valiant (1978), S. 334f.; Garey (1979), S. 123ff.; Lenstra (1979), S. 137; Graham (1979), S. 291ff. (passim); Fisher (1980), S. 1ff.; Fisher (1982), S. 16ff.; French (1982), S. 1177ff., insbesondere S. 183ff.; Parker (1982b), S. 86f.; Karp (1986), S. 106; Hall (1986), S. 275ff., dessen Analyse allerdings nicht die strengen Anforderungen der worst case-Analysen voll erfüllt, sondern nur eine näherungsweise, "heuristische" worst case-Analyse darstellt.

164) Das knapsack-Problem gehört zu dieser einfachen Variante der NP-vollständigen Probleme; vgl. zu solchen Problemen Brucker (1976b), S. 672ff.; Garey (1979), S. 134f.; Lenstra (1979), S. 137; Fisher (1980), S. 4ff.; Fisher (1982), S. 21ff.; French (1982), S. 1182ff. u. 188f.; Parker (1982b), S. 87.

165) Vgl. Karp (1976), S. 2f.; Lenstra (1979), S. 137; Parker (1982b), S. 87.

166) Hierzu zählt das (allgemeine) traveling salesman-Problem; vgl. Garey (1979), S. 147; Lenstra (1979), S. 137; Parker (1982b), S. 87. Vgl. auch Sahni (1974), S. 31f., der weitere dieser schwer-NP-vollständigen Probleme analysiert, und French (1982), S. 188.

Alternative probabilistische Ansätze[167] erstrecken sich auf average case-Analysen vom analytischen Typ, indem sie Wahrscheinlichkeitsverteilungen über den Spektren möglicher Problemausprägungen unterstellen. Hieraus läßt sich z.B.[168] ableiten, mit welcher Wahrscheinlichkeit von den approximativen oder heuristischen Algorithmen eine maximale relative Abweichung von den exakten Lösungen nicht überschritten wird.

Es wurde bereits oben darauf hingewiesen, daß die NP-vollständigen Probleme nur die schwierigsten Probleme in der Klasse NP darstellen, nicht aber notwendig die komplexesten Probleme schlechthin. Es sind seit längerem NP-harte Probleme bekannt, für die alle Versuche nachzuweisen, daß es sich um NP-komplexe Probleme handelt, fehlschlugen[169]. Bezüglich dieser Teilklasse von NP-harten Problemen, die per definitionem mindestens so komplex wie alle Probleme aus der Klasse NP - und somit auch wie alle NP-vollständigen Probleme - sind, wird vermutet, daß ihre Mitglieder tatsächlich schwieriger als alle NP-vollständigen Probleme ausfallen.

Für alle Probleme aus NP - und somit auch für die NP-vollständigen Probleme - stellt die exponentiell beschränkte Zeitkomplexität nur eine obere Komplexitätsschranke dar, weil unter der - obgleich höchst unwahrscheinlichen - Annahme der Identität der Klassen P und NP für diese Probleme auch polynomial beschränkte Lösungsalgorithmen existieren können. Im Hinblick auf einige Probleme, die außerhalb der Klasse PSPACE liegen, konnte dagegen streng bewiesen werden, daß ihre Zeitkomplexität (mindestens[170]) exponentiell be-

---

167) Vgl. Karp (1976), S. 3ff.; Karp (1977), S. 209ff.; Fisher (1980), S. 1f.; Fisher (1982), S. 16f.; Coffman (1982), S. 319ff.; Parker (1982b), S. 87f.; Karp (1986), S. 106ff.; Clausen (1986), S. 34f. Vgl. hierzu auch die Anmerkungen zu analytischen average case-Analysen des Simplex-Algorithmus auf S. 64.

168) Vgl. zu einer alternativen Auswertungsmöglichkeit Karp (1976), S. 3.

169) Vgl. Garey (1979), S. 161ff.

170) Es handelt sich hier also um eine untere Komplexitätsschranke.

schränkt ist[171]. Solche "inhärent exponentiell"[172] beschränkten Probleme können grundsätzlich niemals - auch nicht im unwahrscheinlichen Falle P=NP - durch polynomial beschränkte Algorithmen gelöst werden. Auch bezüglich dieser Probleme wird vermutet, daß sie tatsächlich schwieriger als NP-vollständige Probleme sind. Ferner existieren superexponentiell beschränkte Probleme[173], bezüglich derer bekannt ist, daß sie bei worst case-Analyse notwendig komplexer sind als ein bestimmter exponentieller Term[174].

Darüber hinaus sind Entscheidungsprobleme bekannt, die mit Sicherheit komplexer als die NP-vollständigen Probleme sind. Jedes NP-vollständige Problem kann - wie oben dargelegt - durch einen exponentiell beschränkten Algorithmus eines deterministischen Turing-Automaten gelöst werden. Exponentielle Komplexitätsfunktionen gehören zur Klasse der primitiv-rekursiven Funktionen[175]. Ex existieren aber Entscheidungsprobleme, deren Komplexitätsfunktionen außerhalb dieser Klasse primitiv-rekursiver Funktionen liegen und die infolgedessen noch nicht einmal exponentiell beschränkt sein können. Daher müssen diese Probleme komplexer als NP-vollständige Probleme sein. Zu diesen sehr schwierigen Problemen mit nicht-primitiv-rekursiv beschränkter Komple-

171) Vgl. Meyer (1972), S. 125ff.; Stockmeyer (1979), S. 89f.; Bachem (1980), S. 821f.; Lenstra (1982), S. 207; Cook (1983), S. 403f.
172) Stockmeyer (1979), S. 89.
173) Vgl. Fischer (1974), S. 28f. u. 38ff.; Lipton (1978), S. 194f., im Hinblick auf Entscheidungsprobleme aus dem Bereich der Presburger-Arithmetik (Addition von natürlichen Zahlen).
174) Obwohl dieser Term bereits recht groß gewählt ist, kann nicht ausgeschlossen werden, daß sich als untere Schranke des Lösungsaufwands für solche Entscheidungsprobleme ein größerer exponentieller Term bestimmen läßt. Folglich spricht Fischer auch - zweideutig - von "super"exponentiell beschränkter Komplexität.
175) Vgl. Peter (1957), S. 40f.
Vgl. zum Konzept der primitiv-rekursiven Funktionen, welche die einfachste Variante der rekursiven Funktionen darstellen, Kleene (1952), S. 219ff.; Peter (1957), S. 32ff.; Rogers (1967), S. 6ff.; Stegmüller (1973), S. 30ff.; Paul (1978), S. 13 u. 32ff.; Hermes (1978), S. 61ff.

xität gehört z.B.[176)]das Problem des "fleißigen Bibers"[177)].

176) Ein weiteres Exemplar wird später im Abschnitt 3.2.1 als Erreichbarkeitsproblem der Petrinetz-Theorie vorgestellt.

177) Ein "fleißiger Biber" ist ein Turing-Automat mit n Zuständen, der auf ein Band angesetzt wird, das zunächst nur Nullen enthält, nach endlicher Zeit anhält und in diesem Endzustand mindestens so viele Einsen auf das Band geschrieben hat wie jeder andere Turing-Automat mit derselben Zustandsanzahl. Die Rado-Funktion R(n) gibt die Anzahl von Einsen an, die von einem solchen Automaten mit n Zuständen maximal erzeugt werden können. Vgl. zu Turing-Automaten vom Typ des "fleißigen Bibers" Rado (1962), S. 878ff.; Boolos (1974), S. 27 u. 34ff.; Ludewig (1983), S. 4ff.; Dewdney (1984), S. 8ff.; Hopcroft (1984), S. 45ff.
Das Problem des "fleißigen Bibers" besteht darin, den Wert der Rado-Funktion R(n) für gegebene Zustandsanzahl zu bestimmen. Die Rado-Funktion ist nicht-rekusiv und somit erst recht nicht primitiv-rekusiv; vgl. Rado (1962), S. 881f.; Ludewig (1983), S. 8. Daher ist das Berechnungsproblem für diese Funktion weitaus komplexer als das Entscheidungsproblem für NP-vollständige Probleme. Für beliebig große n gilt dieses Problem als unentscheidbar; vgl. Boolos (1974), S. 35 u. 38f. Im Hinblick auf ein fest vorgegebenes (kleines) n kann aber zumindest das Ersatzproblem entschieden werden, ob der Wert der Rado-Funktion eine - ebenfalls fest vorgegebene - untere Schranke übersteigt. Die Entscheidung dieses Ersatzproblems erfordert aber die Berechnung unterer Schranken für die Rado-Funktion. Da diese Funktion nicht-rekursiv ist, kann die Berechnung unterer Schranken ihrer Funktionswerte nicht rekursiv - und somit auch nicht exponentiell - beschränkt sein.
Dieses (Ersatz-)Poblem wurde 1983 in Dortmund anläßlich der 6. Konferenz der Gesellschaft für Informatik über Theoretische Computerwissenschaft im Rahmen eines Wettbewerbs aufgegriffen, in dem es galt, eine höchste untere Schranke für den Wert der Rado-Funktion des "fleißigen Biber"-Automaten mit 5 Zuständen zu bestimmen; vgl. hierzu die o.a. Quellen.

Durch Rekurs auf die Raumkomplexität, die bisher aus den rein zeitkomplexen Betrachtungen ausgeklammert blieb, läßt sich ein analoges Resultät erzielen. Es ist möglich, alle oben angesprochenen Probleme, die aus den Klassen P oder NP stammen, mit deterministischen[178)] Turing-Automaten zu lösen, deren Speicherplatzbedarf nur in polynomial beschränkter Weise mit dem Problemumfang zunimmt[179)]. Solche Probleme werden in der (Super-)Klasse PSPACE zusammengefaßt[180)]. Analog zu den NP-vollständigen Problemen können innerhalb dieser Klasse PSPACE über den o.a. Reduzierungszusammenhang die PSPACE-vollständigen Probleme ausgezeichnet werden[181)]. Auch hier besteht ein "Prototyp"-Problem[182)], auf das sich alle anderen PSPACE-komplexen Probleme reduzieren lassen. Es handelt sich um das prädikatenlogische Analogon des Erfüllbarkeits-Problems des Aussagenlogik, das quantifizierte Erfüllbarkeits-Problem[183)].

178) Der übergang zu nondeterministischen Turing-Automaten mit polynomial beschränkter Raumkomplexität führt zu keinen wesentlich anderen Ergebnissen, da sich jedes Problem, das von einem nondeterministischen Turing-Automaten mit polynomial beschränktem (und mindestens logarithmisch mit dem Problemumfang anwachsenden) Speicherplatz gelöst werden kann, durch einen deterministischen Turing-Automaten mit dem quadratischen Speicherplatz, also ebenfalls mit polynomial beschränktem Speicherplatz lösen läßt; vgl. Garey (1979), S. 176.

179) Vgl. Garey (1979), S. 170; Lenstra (1979), S. 124; Goldberg (1984), S. 46.

180) Vgl. Garey (1979), S. 170ff.; Lenstra (1979), S. 124; Lenstra (1982), S. 206f.; Goldberg (1984), S. 46.

181) Vgl. Lenstra (1979), S. 124; Garey (1979), S. 171.

182) Weitere Beispiele hierfür finden sich bei Garey (1979), S. 172ff., insbesondere auch das verallgemeinerte HEX-Spiel, das größere Bekanntheit erlangt hat.

183) Vgl. Garey (1979), S. 171; Lenstra (1979), S. 124; Goldberg (1984), S. 46.
Es handelt sich um das Problem zu entscheiden, ob eine prädikatenlogische Formel, deren Variablen vollständig durch Quantoren gebunden sind, wahr ist. Eine solche Formel unterscheidet sich von einer aussagenlogischen Formel im wesentlichen dadurch, daß die atomaren Formelkomponenten keine strukturlosen atomaren Aussagen darstellen, sondern atomare Prädikate. Diese atomaren Prädikate besitzen eine innere Struktur, da sie über Variablen definiert sind. Hinzu kommen Existenz- und Allquantoren, welche die zulässigen Belegungen der Variablen mit Werten aus ihren Definitionsbereichen "quantifizieren".

Analog[184] zur unbeantworteten Frage, ob P≠NP gilt, erweist sich das (Meta-)Problem ungelöst, ob die Klassen P und NP mit der Klasse PSPACE identifiziert werden dürfen. Es wird jedoch allgemein vermutet, daß die Beziehungen P=PSPACE und NP=PSPACE nicht gelten. Daraus folgt die Vermutung, daß ein PSPACE-vollständiges Problem existiert, das weder P- noch NP-komplex ist[185]. Falls diese Annahme zutrifft, sind PSPACE-vollständige Probleme notwendig komplexer als NP-vollständige Probleme[186]. Auf jeden Fall sind sie aber mindestens so schwierig wie alle NP-vollständigen Probleme.

Darüber hinaus lassen sich sogar Probleme aufzeigen - wie z.B. das das Erreichbarkeitsproblem aus der Petrinetz-Theorie, auf das später näher eingegangen wird[187],- die noch nicht einmal zur Klasse PSPACE gehören[188], weil der Speicherplatz für ihre Lösung nicht mehr polynomial beschränkt werden kann. Ihr Lösungsaufwand übersteigt infolgedessen sogar den, der von PSPACE-vollständigen Problemen erfordert wird. Da die PSPACE-vollständigen Probleme mindestens so komplex sind wie die NP-vollständigen Probleme, müssen die o.a. Probleme, die nicht mehr zur Klasse PSPACE gehören, in der Tat schwieriger als die NP-vollständigen Probleme sein. Hier befindet sich zur Zeit die Erkenntnisgrenze der Komplexitätstheorie bezüglich der "schwierigsten" noch entscheidbaren Probleme.

2.5 Validitätsprobleme der Komplexitätstheorie

Die Komplexitätstheorie erhebt den Anspruch, durch ihren aufwandsbezogenen Komplexitätsbegriff die Schwierigkeit von Problemen zu messen. Die Validität (Gültigkeit) dieses Meßkonzepts, d.h. die Übereinstimmung der zu messenden Problemschwierigkeit mit den tatsächlich

184) Vgl. zum folgenden Garey (1979), S. 171.
185) Vgl. Garey (1979), S. 171f.
186) Vgl. Garey (1979), S. 170 u. 172; Lenstra (1979), S. 125.
187) Vgl. Abschnitt 3.2.1.
188) Vgl. Meyer (1972), S. 125ff.; Lenstra (1979), S. 125; Lenstra (1982), S. 207.

gemessenen Sachverhalten[189]), muß jedoch in mehrfacher Weise eingeschränkt werden.

Erstens betrachtet die Komplexitätstheorie bei ihren Urteilen über die Schwierigkeiten von Problemen nicht diese Probleme selbst, sondern die Effizienz der jeweils bestbekannten Lösungsalgorithmen. Es wurde bereits herausgestellt, daß diese Effizienzurteile nicht allein von der Schwierigkeit der untersuchten Probleme determiniert weren, sondern auch vom jeweils aktuellen algorithmischen Kenntnisstand abhängen. Die aufwandsbezogene Komplexität im schwachen Sinne ist in dem Ausmaß invalide, in dem die Effizienz der bestbekannten Algorithmen hinter der Effizienz der noch unbekannten bestmöglichen Lösungsalgorithmen zurückbleibt. Dieser potentielle Validitätsmangel läßt sich grundsätzlich nicht (ex ante) feststellen, weil dies die - logisch widersprüchliche - Kenntnis noch unbekannter Algorithmen voraussetzte. Erst wenn neue, leistungsfähigere Algorithmen bekannt werden, kann ex post die Ungültigkeit früherer Komplexitätsurteile erkannt werden. Diese Problematik wird allerdings von der zeitinvarianten Komplexität im strengen Sinne vermieden.

Zweitens wird durch Komplexitätsfunktionen nur der asymptotische Lösungsaufwand für Probleme, deren Umfänge gegen unendlich konvergieren, betrachtet. Reale Problemausprägungen, die immer einen endlichen Umfang aufweisen, werden hierdurch grundsätzlich nicht exakt überdeckt[190]). Die Realitätsverzerrung fällt um so größer aus, je stärker der Lösungsaufwand für reale Problemumfänge vom asymptotischen Aufwand abweicht.

Drittens ist die Zuordnung der Qualitätsurteile "effizient" und "ineffizient" zu Problemen mit (bestbekannten) polynomial bzw. exponentiell beschränkten Algorithmen nur gültig, wenn der Problemumfang n eine "hinreichende" Größe erreicht[191]). Denn ein Polynomial

189) Vgl. zu diesem Validitätsverständnis Gass (1983), S. 605, 607 u. 609ff., insbesondere S. 609.
190) Vgl. Karp (1975b), S. 64.
191) Vgl. zu den Ausführungen dieses Abschnitts Lawler (1976), S. 5; Ullman (1976), S. 139; Bland (1981), S. 1041.

nimmt erst ab einer - vom Einzelfall abhängigen[192] - Untergrenze des Problemumfangs stets kleinere Werte an als ein exponentieller Term[193]. Daher trifft es nicht zu, polynomiale Beschränktheit eines Algorithmus bedeute generell, d.h. hier invariant bezüglich des Problemumfangs, eine höhere Effizienz als exponentielle Beschränktheit. Effizienzurteile von average case-Analysen[194] können ungültig sein, wenn die Umfänge der praktisch zu bewältigenden Probleme unterhalb der fallspezifischen Untergrenze bleiben und polynomial mit exponentiell beschränkten Algorithmen verglichen werden[195].

Auch das absolute Effizienzurteil, ein polynomial beschränkter Algorithmus gestatte eine effiziente Problemlösung, ist nicht immer valide. Denn Polynomiale mit großen Exponenten oder - sofern explizit angeführt - mit großen multiplikativen Konstanten können schon für einen kleinen Problemumfang zu so hohem Lösungsaufwand führen, daß der Algorithmus von seinen Anwendern nicht mehr im intuitiven Begriffsverständnis als effizient betrachtet wird. Für die praktische Effizienz -

---

192) Diese Untergrenze kann noch nicht einmal exakt angegeben werden, da die Komplexitätsfunktionen von additiven und multiplikativen Konstanten abstrahieren, welche die Lage der tatsächlichen Effizienzgrenze zwischen einem polynomial einem exponentiell beschränkten Algorithmen determinieren.

193) Aus diesem Sachverhalt kann zugunsten der Verwendung von asymptotischen Komplexitätsfunktionen gefolgert werden, daß nur durch die Betrachtung von Problemen, deren Umfänge gegen unendlich konvergieren, die o.a. Validitätsproblematik vermieden wird. Denn solche Probleme liegen per constructionem immer über der Untergrenze "hinreichenden" Problemumfangs. Allerdings bedeutet diese Argumentation zugleich das Entstehen des andersartigen Validitätsmangels, daß solche unendlich großen Probleme artifizielle Produkte von Grenzfallbetrachtungen, aber keine real existenten Probleme darstellen.

194) Für worst case-Analysen tritt diese Validitätsproblematik im allgemeinen nicht ein, weil hier nicht auf Problemumfänge rekurriert wird, die für die Praxis repräsentativ sein sollen.

195) Analog läßt sich aufzeigen, daß Aussagen über die relative Vorteilhaftigkeit zweier polynomial beschränkter Probleme nicht gültig sein müssen, wenn die betrachteten Komplexitätsfunktionen von - u.U. sehr großen - multiplikativen (oder additiven) Konstanten abstrahieren; vgl. Lawler (1976), S. 6.

im intuitiven Sinne - ist nicht die polynomiale Beschränktheit an sich ausschlaggebend[196], sondern die Tatsache, daß ein polynomial beschränkter Lösungsalgorithmus existiert, dessen Lösungsaufwand für real häufig auftretende Problemumfänge durch ein Polynomial mit kleinen Exponenten (und ggf. kleinen multiplikativen Konstanten) begrenzt wird[197].

Viertens stellt sich ein Validitätsdefizit ein, wenn Ergebnisse von worst case-Analysen als Urteile über die praktische Problemschwierigkeit ausgegeben werden. Denn worst case-Analysen erlauben per constructionem nur rein theoretische Komplexitätsurteile über den maximalen Lösungsaufwand für den schlechtest möglichen Fall. Invalide Komplexitätsfeststellungen dieser Art können zwar nicht der Komplexitätstheorie angelastet werden, weil sie aus der fehlerhaften Anwendung dieser Theorie resultieren. Doch verführt die vornehmliche Beschäftigung komplexitätstheoretischer Arbeiten - insbesondere auch ihrer populärwissenschaftlichen Rezeption - mit worst case-Analysen zu dem naheliegenden Fehlschluß, die derart gewonnenen (bedingten[198]) Komplexitätsurteile besäßen unbedingte Gültigkeit. Der Gefahr eines solchen Fehlschlusses ist auch die o.a.[199] Vermutung hinsichtlich eines implikativen Zusammenhangs zwischen worst und average case-Analysen ausgesetzt.

---

196) Vgl. Garey (1979), S. 9 u. 135; Dantzig (1979), S. 1 ("... polynomially bounded time says little.") u. 3; Karp, zitiert in Frenkel (1986), S. 110.

197) Vgl. Paul (1978), S. 181; Garey (1979), S. 9, der nur noch Polynomiale mit Exponenten, die den Wert 3 nicht überschreiten, als Effizienz-Indikatoren anerkennen möchte; Dantzig (1979), S. 2f.; Frenkel (1986), S. 110 (indirekt).

198) Die Bedingung der Urteilsgültigkeit liegt in der Voraussetzung der worst case-Betrachtungsweise.

199) Vgl. S. 29f.

Fünftens leiden average case-Analysen oftmals unter der Problematik[200], daß die empirische Gültigkeit der zugrundegelegten Wahrscheinlichkeitsverteilungen oder die Repräsentativität der Mengen ausgewählter Problemausprägungen unbekannt ist oder kommentarlos vorausgesetzt wird. Die Validität der analytisch bzw. simulativ gewonnenen Komplexitätsresultate ist dann aber nicht gewährleistet.

200) Vgl. Karp (1976), S. 3; Simon (1976), S. 298; Rabin (1976), S. 21; Valiant (1978), S. 333; Bachem (1980), S. 817; Brown (1981), S. 590f.; Rardin (1982), S. 12; Goldberg (1984), S. 53; vgl. auch Garey (1979), S. 150, bezüglich des Problems, eine Wahrscheinlichkeitsfunktion auszuwählen.

3 Anwendung der Komplexitätstheorie auf Probleme des Operations Research

3.1 Überblick

Die typischen Probleme des Operations Research sind über die Problemklassen der Komplexitätstheorie nicht annähernd gleichmäßig verteilt. Vielmehr konzentrieren sich die meisten Probleme auf die Klasse NP[201], und zwar in der Überzahl auf die Subklasse der NP-vollständigen Probleme[202]. Daher weisen die Standard-Konzepte der Komplexitätstheorie nur eine geringe Trennschärfe hinsichtlich der Beurteilung der Schwierigkeit von OR-Problemen auf. Die Ansätze der Komplexitätstheorie, jenseits der P-komplexen Probleme weitere Komplexitätsstufen zu differenzieren, haben noch zu keinen herausragenden Erkenntnisfortschritten im Hinblick auf Fragestellungen des Operations Research geführt[203]. Nur einige wenige NP-vollständige Probleme konnten als einfach im Sinne der Approximations-Algorithmen nachgewiesen werden[204]. Die worst case-Analyse von heuristischen Algorithmen bietet zur Zeit noch ein schwer zu überschauendes Bild, da die Ergebnisse stark mit der jeweils zugrundegelegten Problemspezifizierung variieren[205].

---

201) Vgl. zur Relevanz der NP-komplexen Probleme für das Operations Research Benito (1978), S. 81ff.; vgl. auch Karp (1975b), S. 57.
Bachem (1980), S. 824, vertritt die Ansicht, daß "... sich fast alle praxisrelevanten Operations Research Probleme in der Klasse NP befinden".

202) Bachem (1980), S. 827, führt an, derzeit seien 2.000 NP-vollständige OR-Probleme bekannt. Vgl. auch Brucker (1976c), S. 135ff.

203) Vgl. z.B. Fisher (1982), S. 23, mit dem Hinweis, daß für zahlreiche OR-Probleme grundsätzlich kein Algorithmus mit den wünschenswerten Approximations-Eigenschaften existieren kann, die den einfach-NP-vollständigen Problemen zukommen.

204) Vgl. Fisher (1982), S. 21ff. (in bezug auf knapsack-Probleme).

205) Vgl. Fisher (1982), S. 29. Vgl. auch die Ausführungen von Hall (1986), S. 275ff., in denen zwar eine worst case-Analyse für Heuristiken, welche der Reihenfolgeplanung von Aufträgen dienen, versucht wird. Doch werden derart restriktive Prämissen gesetzt, daß die "schwierigen" Planungsfälle a priori ausgeschlossen werden (S. 275), so daß dem inhaltlichen Anspruch einer worst case-Analyse implizit widersprochen wird.

Zur Klasse der NP-vollständigen Probleme aus dem Bereich des Operations Research zählen z.B. die folgenden - realproblem- oder modellbezogen abgegrenzten - Gebiete der[206]:

- Reihenfolge- und Terminplanung von Aufträgen, wie z.B. der Maschinenbelegungsplanung;
- Tourenplanung vom traveling salesman-Typ;
- knapsack-Modelle (z.B. für Verpackungsprobleme)
- quadratischen Zuordnungsmodelle (z.B. für die innerbetriebliche Standortplanung);
- Netzplan-Modelle (vor allem für die Projektplanung);
- Entscheidungsbaum-Modelle (z.B. für die flexible Investitionsplanung);
- rekursiven Funktionsgleichungs-Modelle der dynamischen Optimierungsrechnung (z.B. für mehrperiodige Lagerhaltungsprobleme).

Besonders intensiv wurden Probleme der Reihenfolgeplanung (scheduling) untersucht. Ein Grund für dieses große Interesse kann in dem Sachverhalt liegen, daß sich hinsichtlich dieser Ordinierungsprobleme Fragestellungen des betriebswirtschaftlich ausgerichteten Operations Research und der informatik-bezogenen Gestaltung von Computer-Betriebssystemen (Abarbeiten von Jobs) überschneiden[207]. Auf die vielfältige Literatur zu

---

206) Vgl. Karp (1972), S. 94f.; Brucker (1976b), S. 670f.; Müller-Merbach (1976), S. 71f.; Benito (1978), S. 84, 100ff. u. 109ff.; Garey (1979), S. 206ff. u. 236ff.; Bachem (1980), S. 824ff., insbesondere S. 827f.; Brucker (1981), S. 60ff. u. 194ff.

207) Vgl. Graham (1979), S. 320.

dieser Thematik kann hier nur verwiesen werden[208]. Am Rande sei ein Computer-Programm erwähnt, das die Komplexität von Problemen der Reihenfolgeplanung automatisch zu klassifizieren vermag[209].

3.2 Beispiele

3.2.1 Das Erreichbarkeitsproblem der Petrinetz-Theorie

Petrinetze sind gerichtete bipartite markierte Graphen. In der Basisvariante der Stelle/Transition-Netze[210] ist die Menge der Netzknoten in die disjunkten Teilmengen der S-Knoten ("Stellen") und der T-Knoten ("Transitionen") zerlegt. Die Stellen können von Objekten ("Marken") belegt werden. Eine Verteilung dieser Objekte über die Stellen eines Petrinetzes wird als Markierung bezeichnet. Die Transitionen lassen sich nach netzspezifischen Regeln "schalten", wobei sie - in Richtung der Kanten, die Stellen und Transitionen miteinander verbinden - von den unmittelbar vorgelagerten Stellen Marken abziehen und auf den unmittelbar nachgelagerten Stellen Marken ablegen. Die Menge aller Markierungen, die durch solches Schalten von einer gegebe-

---

208) Vgl. Ullman (1973), S. 96ff.; Ullman (1975), S.384ff.; Brucker (1975), S. 6ff., insbesondere Table I u. II auf S. 10f.; Garey (1975), S. 397ff.; Rinnooy Kan (1976), S. 5ff. u. 56ff., insbesondere S. 106ff. u. Table 7.1-7.3 auf S. 132f.; Brucker (1976a), S. 357ff., insbesondere S. 363ff.; Brucker (1976c), S. 140ff.; Garey (1976a), S. 117ff.; Sethi (1977), S. 320ff., insbesondere S. 322ff.; Lenstra (1977), S. 346ff., insbesondere S. 348ff.; Ecker (1977), S. 10, 58ff. u. 244ff., insbesondere S. 253ff.; Gonzalez (1978), S. 36ff.; Lenstra (1978), S. 25ff.; Graham (1979), S. 288ff.; Garey (1979), S. 236ff.; Blazewicz (1980), S. 1ff.; Brucker (1981), S. 25ff., insbesondere Tabellen 1 bis 5 auf S. 60ff., S. 65ff. u. 194ff.; Coffman (1982), S. 319ff.; French (1982), S. 137ff., insbesondere S. 150ff.; Pinedo (1982), S. 355ff.; Fisher (1982), S. 25ff.; Hall (1986), S. 272ff.

209) Vgl. Graham (1979), S. 319.

210) Vgl. zur präzisen Definition der hier nur skizzierten Stelle/Transition-Netze Mayr (1977), S. 7ff.; Jantzen (1980), S. 167ff.; Starke (1980), S. 50ff.; Reisig (1985), S. 62ff.

nen Ausgangsmarkierung erreicht werden können, wird als Erreichbarkeitsmenge bezeichnet.

Das Erreichbarkeitsproblem der Petrinetz-Theorie besteht in dem Problem zu entscheiden, ob von einer gegebenen (Ausgangs-)Markierung durch beliebig, aber endlich häufiges Schalten der Transitionen eine ebenfalls gegebene (End-)Markierung erreicht werden kann[211]. Dieses Erreichbarkeitsproblem kann in äquivalenter Weise im Kontext von Vektor-Additions-Systemen[212] formuliert werden, weil die Kalküle von Stelle/Transition-Netzen und Vektor-Additions-Systemen isomorph sind[213].

Die Bedeutung des Erreichbarkeitsproblems erstreckt sich auf zwei Aspekte. Erstens gehört es zu den wenigen Problemen, die einerseits derart komplex sind, daß sie lange Zeit ungelöst blieben, und die anderseits den Erkenntnisfortschritt der Komplexitätstheorie verdeutlichen.

Zweitens konnte aufgezeigt werden, daß sich eine Vielzahl von - ebenfalls ungelösten - Problemen, die vornehmlich aus den Bereichen der formalen Sprachen und mathematischen Logik (Metamathematik) stammen, auf das Erreichbarkeitsproblem für Petrinetze reduzieren läßt[214]. Daher würde die Lösung dieses Erreichbarkeitsproblems den Schlüssel zu zahlreichen anderen Pro-

---

211) Vgl. zu ausführlichen Darstellungen des Erreichbarkeitsproblems Mayr (1977), S. 10ff.; Peterson (1977), S. 240ff.; Starke (1980), S. 150ff.; Grabowski (1980), S. 209 u. 212ff.; Mayr (1980), S. 1f. u. 52ff.; Jantzen (1980), S. 174f. u. 183ff.; vgl. auch Niehuis (1986), S. 27.

212) Vgl. zur Darstellung solcher Vektor-Additions-Systeme Grabowski (1980), S. 208f.

213) Aus diesem Grund werden nachfolgend alle Ergebnisse bezüglich des Erreichbarkeitsproblems auf Petrinetze (Stelle/Transition-Netze) bezogen, auch wenn sie in den angeführten Quellen - exemplarisch sei auf Sacerdote (1977), S. 61ff., verwiesen - explizit im Hinblick auf Vektor-Additions-Systeme bezogen wurden.

214) Vgl. Crespi-Reghizzi (1977), S. 186ff. u. 190f.; Börger (1980), S. 123f. u. 129ff.; Mayr (1980), S. 52f.; Starke (1980), S. 150; Mayr (1984), S. 441.

blemen liefern[215]. Zumindest wäre durch sie die Existenz von Lösungen für die anderen Probleme bewiesen, auch wenn hiermit die tatsächliche Gestalt jener Lösungen - d.h. ihre Unentscheidbarkeit oder die Komplexität der Lösungsalgorithmen im Falle ihrer Entscheidbarkeit - noch nicht konstruktiv gegeben sein müßte.

Das Erreichbarkeitsproblem[216] blieb für lange Zeit ungelöst. Weder seine Entscheidbarkeit noch seine Unentscheidbarkeit waren bekannt. Zwar konnte die Entscheidbarkeit des Erreichbarkeitsproblems für die Spezialfälle von Petrinetzen mit höchstens 3 Stellen (van Leeuwen[217]) und später mit maximal 5 Stellen (Hopcroft und Pansiot[218]) nachgewiesen werden. Doch blieben diese Ansätze für das allgemeine Erreichbarkeitsproblem unfruchtbar, weil sie die Semilinearität der untersuchten Erreichbarkeitsmengen voraussetzten, während die Erreichbarkeitsmengen für Petrinetze mit mindestens 6 Stellen auch nicht-semilinear sein können[219].

Das eng verwandte (Inklusions-)Problem, ob die Erreichbarkeitsmenge eines Petrinetzes in derjenigen eines anderen Petrinetzes mit derselben Stellen-Anzahl (aber nicht notwendig mit derselben Ausgangsmarkierung) enthalten ist, wurde dagegen schon in der zweiten Hälfte der sechziger Jahre[220] von Rabin als unentscheidbar

---

215) Hack (1975), S. 72, bezeichnet das Erreichbarkeitsproblem als das mutmaßlich wichtigste ungelöste Problem im Kontext der Petrinetz-Theorie und verwandter mathematischer Konzepte. Ähnlich äußern sich Peterson (1981), S. 87f., und Mayr (1984), S. 441.

216) Vgl. zu dieser - lange Zeit den aktuellen Erkenntnisstand der Komplexitätstheorie richtig wiedergebenden - Ansicht: van Leeuwen (1974), S. 303; Cardoza (1976), S. 50; Crespi-Reghizzi (1976), S. 130 u. 137; Hack (1976), S. 77; Jones (1977), S. 297; Grabowski (1979), S. 341.

217) Vgl. van Leeuwen (1974), S. 307f.; vgl. auch Keramidis (1979), S. 134ff.

218) Vgl. Hopcroft (1979), S. 158 i.V.m. S. 140ff.

219) Vgl. Baker (1973), S. 6 u. 14; Hack (1975), S. 94; Hopcroft (1979), S. 135f. u. 145ff.; Mayr (1980), S. 2; Müller (1980b), S. 426; Grabowski (1980), S. 4; Mayr (1981), S. 238; Mayr (1984), S. 442; Müller (1985a), S. 379f.

220) Das Jahr 1967 wird von Hack (1973), S. 1, angeführt, während Sacerdote (1977), S. 61, das Jahr 1969 nennt.

nachgewiesen[221]. Der Beweis läßt sich auf die o.a., von Matijasevic aufgezeigte Unentscheidbarkeit des 10. Problems von Hilbert zurückführen[222]. Analog hierzu wurde vermutet, auch das Erreichbarkeitsproblem sei im allgemeinen Fall unentscheidbar[223].

Seit Mitte der siebziger Jahre kam dagegen vermehrt der Verdacht auf, das Erreichbarkeitsproblem könne doch entscheidbar sein[224]. Es wurden für den Fall, daß sich diese Vermutung als richtig herausstellen sollte, untere Schranken für die Komplexität der Algorithmen zur Lösung des Erreichbarkeitsproblems aufgestellt. Sie zeigten, daß ein solcher Lösungsalgorithmus sehr aufwendig ausfallen müßte, weil die unteren Schranken mit der Petrinetz-Größe exponentiell anwachsen[225].

Ein aufsehenerregendes Ereignis war im Jahr 1977 die Arbeit von Sacerdote und Tenney, in der - angeblich - der Beweis für die Entscheidbarkeit des Erreichbarkeitsproblems erbracht wurde[226]. Nach der anfangs breiten Anerkennung für die Leistung, dieses zentrale Problem der Graphen- und Komplexitätstheorie endlich

---

221) Rabin selbst publizierte seine Beweisführung nicht in schriftlicher Form; vgl. Hack (1973), S. 1; Sacerdote (1977), S. 61. Aber der Unentscheidbarkeits-Beweis kann entnommen werden: Baker (1973), S. 14ff.; Hack (1973), S. 1ff.; Hack (1975), S. 118ff. i.V.m. S. 96f. u. 114ff.; Hack (1976), S. 87ff. Vgl. auch Hack (1975), S. 121; Hack (1976), S. 77 u. 91; Araki (1976), S. 20; Mayr (1977), S. 5; Peterson (1977), S. 242; Peterson (1981), S. 133.

222) Vgl. Hack (1973), S. 1ff.; Baker (1973), S. 14ff.; Hack (1975), S. 118ff. i.V.m. S. 96f. u. 114ff.; Hack (1976), S. 87ff.; Peterson (1981), S. 133. Vgl. auch Araki (1976), S. 22ff., und Peterson (1981), S. 133ff., in bezug auf das strengere Problem der Gleichheit der betroffenen Erreichbarkeitsmengen.
Bereits Rabin griff im Jahr 1972 auf das Unentscheidbarkeits-Resultat von Matijasevic zurück, als er eine Vereinfachung seines Beweises der Unentscheidbarkeit des Inklusionsproblems mündlich vortrug; vgl. Hack (1973), S. 1.

223) Vgl. Byrn (1974), S. 2/61.

224) Vgl. Hack (1975), S. 172; Hopcroft (1979), S. 135; Peterson (1981), S. 146.

225) Vgl. Hack (1975), S. 170f.; Peterson (1977), S. 242f.; Jones (1977), S. 293; Peterson (1981), S. 146 u. 148.

226) Vgl. Sacerdote (1977), S. 62ff.

gelöst zu haben[227], äußerten sich zunehmend Stimmen, welche die Korrektheit der Beweisführung anzweifelten[228]. Die angekündigte Beweiskorrektur wurde niemals erbracht[229].

Im Jahr 1980 stellte Mayr einen neuen Beweis für die Entscheidbarkeit des Erreichbarkeitsproblems vor[230]. Nach den enttäuschenden Erfahrungen mit dem Beweisversuch von Sacerdote und Tenney blieb die Reaktion auf diesen neuen Beweisansatz zunächst zurückhaltend[231]. Die Ausführungen von Mayr wurden als derart kompliziert eingeschätzt, daß ihre Stringenz auch in der Folgezeit keine Anerkennung fand[232], obwohl weder Beweislücken noch -fehler aufgezeigt werden konnten[233].

Den entscheidenden Beitrag zur - allseits akzeptierten - Lösung des Erreichbarkeitsproblems lieferte Kosaraju im Jahr 1982. Sein Beweis der Entscheidbarkeit dieses Problems[234] wies zwar noch eine Argumentationslücke auf[235]. Diese konnte aber alsbald von Müller geschlossen werden[236], ohne daß hierdurch die grundle-

---

227) Vgl. Peterson (1977), S. 242 u. 247; Araki (1977b), S. 111; Murata (1977), S. 413; Hughes (1978), S. 201; Peterson (1978), S. 147f.

228) Vgl. Hopcroft (1979), S. 135 u. 138; Mayr (1980), S. 2f.; Müller (1980), S. 426; Heinemann (1980), S. 93; Peterson (1981), S. 146 u. 274; Mayr (1981), S. 238; Müller (1982), S. 4; Müller (1983a), S. 174; Müller (1983b), S. 4; Mayr (1984), S. 442; Müller (1985a), S. 377; Niehuis (1986), S. 27.

229) Vgl. Peterson (1981), S. 274; Niehuis (1986), S. 27.

230) Vgl. Mayr (1980), S. 51 i.V.m. S. 39ff.; Mayr (1981), S. 245 i.V.m. S. 243f.; Mayr (1984), S. 454 i.V.m. S. 452ff. u. 454ff.

231) Heinemann (1980), S. 93, z.B. äußert in bezug auf den Beweis von Mayr, es könne "... mit einer gewissen Berechtigung angenommen werden ..., daß er korrekt ist".

232) Vgl. Müller (1982), S. 4; Müller (1983a), S. 174; Müller (1983b), S. 4; Müller (1985a), S. 377.

233) Vgl. Müller (1983c).

234) Vgl. Kosaraju (1982), S. 281 i.V.m. S. 270ff.; Müller (1983a), S. 177ff.; Müller (1985a), S. 382ff. u. 388ff.

235) Vgl. Müller (1983a), S. 179ff.; Müller (1983b), S. 4; Müller (1985a), S. 378 u. 385f.

236) Vgl. zu den Verbesserungsvorschlägen, die zu einer vervollständigten, nunmehr korrekten Beweisversion führten, Müller (1982), S. 5ff.; Müller (1983a), S. 180ff.; Müller (1985a), S. 386f.

gende - und nun auch als transparent anerkannte[237] - Beweisstruktur von Kosaraju hätte modifiziert werden müssen[238].

Die o.a. Vermutung, daß Lösungsalgorithmen für das Erreichbarkeitsproblem äußerst komplex ausfallen müßten, falls sie überhaupt existieren, konnte durch den konstruktiven Beweis von Kosaraju verifiziert werden. Sein Entscheidungsalgorithmus besitzt eine Zeitkomplexität, die durch keine primitiv-rekursive Funktion beschränkt werden kann. Da die Klasse der primitiv-rekursiven Funktionen auch alle exponentiellen Funktionen umfaßt, ist der Algorithmus zur Lösung des Erreichbarkeitsproblems noch nicht einmal exponentiell beschränkt[239]. Dies ist insofern erstaunlich, als für alle bisher intensiv analysierten, NP-vollständigen Probleme des Operations Research exponentiell beschränkte Lösungsalgorithmen formuliert werden konnten.

---

237) Vgl. Müller (1982), S. 5.

238) Vgl. Müller (1983b), S. 4, Müller (1983c). Es mußte lediglich in einem Lemma zusätzlich der starke Netzzusammenhang gefordert werden, der aber für Petrinetze ohnehin immer erfüllt ist.

239) Die Komplexität dieses Algorithmus ist laut Müller (1983c) und Müller (1985a), S. 387, und Müller (1985b), S. 245, mindestens so groß wie die der nicht-primitiv-rekursiven Ackermann-Funktion für das Argument (2,n). Vgl. zur Ackermann-Funktion Kleene (1952), S. 271f.; Hermes (1961), S. 83ff.; Stegmüller (1973), S. 45; Cardoza (1976), S. 53. Dieses Komplexitäts-Resultat ändert sich nach Müller (1983c) auch dann nicht, wenn die von Müller (1983a), S. 182f., und Müller (1985a), S. 387, vorgeschlagene Vereinfachung des Lösungsalgorithmus von Kosaraju vorgenommen wird. (Durch diese Modifizierung läßt sich der Teil-Algorithmus zur Konstruktion eines Überdeckungsgraphen von nicht-primitiver-rekusiver auf exponentiell beschränkte Raumkomplexität vereinfachen.)
Jantzen (1983), S. 24f., befaßt sich mit einem Problem, das mit dem Erreichbarkeitsproblem zwar nicht identisch, aber dennoch eng verwandt ist. Es handelt sich um die Berechnung der Funktion "busy" (in Anlehnung an das Problem des "fleißigen Bibers"?; vgl. S. 46), welche die Anzahl der Marken angibt, die unter einer erreichbaren Markierung maximal alle Stellen eines Petrinetzes simultan zu belegen vermögen. Auch dieses Berechnungsproblem besitzt mindestens die Zeitkomplexität wie die Berechnung der o.a. Ackermann-Funktion und ist infolgedessen wie das Erreichbarkeitsproblem nicht primitiv-rekursiv beschränkt. Vgl. hierzu auch Niehuis (1986), S. 28.

Darüber hinaus besitzt das Erreichbarkeitsproblem mindestens[240] exponentiell beschränkte Raumkomplexität[241,242]. Damit ist es auf jeden Fall komplexer als alle NP-vollständigen Probleme, die höchstens polynomial beschränkte Raumkomplexität besitzen[243].

Somit erweist sich die Lösung des Erreichbarkeitsproblems - sowohl aus der Sicht seiner Zeit- als auch aus der Perspektive seiner Raumkomplexität - als ein äußerst schwieriges Unterfangen. Der Lösungsalgorithmus aus dem Beweis von Kosaraju (und Müller) stellt sich wesentlicher komplexer heraus als die Algorithmen zur Lösung NP-vollständiger Probleme, die bislang oftmals als die komplexesten Problemstellungen des Operations Research betrachtet wurden[244]. Der Nachweis der Entscheidbarkeit des Erreichbarkeitsproblems der Petrinetz-Theorie läßt - wie bereits oben konstatiert - das Bedürfnis deutlich hervortreten, die Komplexitätstheorie bei der Analyse "sehr schwieriger" Probleme über die Klasse der NP-vollständigen Probleme hinaus fortzuentwickeln.

---

240) Peterson (1981), S. 146 u. 148, stellt klar heraus, daß der tatsächlich von einem Turing-Automaten benötigte Speicherplatz (die Raumkomplexität) auch noch schneller als exponentiell beschränkt mit dem Problemumfang wachsen könnte. Jones (1977), S. 293, äußert die Vermutung, daß die Raumkomplexität nicht exponentiell beschränkt werden könne, falls sich das Erreichbarkeitsproblem als entscheidbar herausstelle. Der von Mayr vorgestellte Beweisansatz basiert auf einem Algorithmus, dessen Raumkomplexität nicht mehr durch eine primitiv-rekursive Funktion beschrieben werden kann; vgl. Mayr (1980), S. 53f. Folglich läßt sich der Speicherplatzbedarf dieses Algorithmus - obgleich er das Erreichbarkeitsproblem noch nicht endgültig zu lösen vermag - erst recht nicht exponentiell beschränken.

241) Vgl. zum Konzept der exponentiell beschränkten Raumkomplexität Cardoza (1976), S. 52; Garey (1979), S. 170ff.

242) Vgl. Hack (1975), S. 171; Lipton (1976); Cardoza (1976), S. 50; Lenstra (1979), S. 125; Jantzen (1980), S. 187; Mayr (1980), S. 53; Peterson (1981), S. 146ff.; Müller (1985a), S. 377.

243) Vgl. die Ausführungen auf S. 48f.

244) Vgl. hierzu die Ausführungen in Abschnitt 3.1.

3.2.2 Das Problem der linearen Optimierung

Am Beispiel der rationalzahligen linearen Optimierungsprobleme, die im Rahmen des Operations Research als lineare Programme modelliert werden, läßt sich die Tragweite der Unterscheidung zwischen worst und average case-Analyse verdeutlichen. Nachfolgend wird als Lösungsaufwand eines Algorithmus nur dessen Zeitkomplexität betrachtet.

Zur Lösung von linearen Programmen werden seit langem verschiedene Varianten des Simplex-Algorithmus eingesetzt. Obwohl "pathologische" Fälle bekannt sind, in denen er zu extrem hohem Lösungsaufwand führt[245], hat sich dieser Algorithmus dennoch praktisch durchgesetzt[246]. Sein durchschnittlicher Lösungsaufwand wird von den Algorithmusanwendern als akzeptabel empfunden. Der Simplex-Algorithmus gilt als "praktisch" effizient[247]. Komplexitätstheoretische average case-Analysen legen den Erwartungswert für den Lösungsaufwand dieses Algorithmus auf ein Polynomial niedriger Ordnung fest[248]. Die Durchschnitts-Komplexität von linearen Optimierungsproblemen mit n Variablen und m Restriktionen wird zumeist durch die Polynomiale[249] $O(m,n) = m(\log(2+n/m))$[250], $O(m) = am$ mit einer Konstanten a, deren Wert zwischen 2 und 3 liegt[251], oder $O(m,n) =$

---

245) Vgl. Smale (1983), S. 530f.; Fricker (1985), S. 32; Hooker (1986), S. 76. Vgl. hierzu den Verweis auf S. 64f. bezüglich der von Klee und Minty konstruierten unendlich großen Menge von Problemausprägungen, deren Lösungsaufwand mit dem Problem(ausprägungs)umfang exponentiell beschränkt wächst. Wenn einfache Varianten des Simplex-Algorithmus benutzt werden, kann es sogar zu einer Entartung kommen, bei welcher sich der Algorithmus in einer Endlosschleife verfängt; vgl. Dantzig (1966), S. 262ff.

246) Vgl. Karp (1975a), S. 24

247) Vgl. Karp (1975a), S. 24; Karp (1975b), S. 65; Johnson (1981), S. 400; Lenstra (1982), S. 204; Smale (1983), S. 530; Borgwardt (1985), S. 650.

248) Vgl. Johnson (1981), S. 400; Schrader (1982), S. 267; Lenstra (1982), S. 206; Parker (1982a), S. 7; Schrader (1983), S. 3.

249) Unter dem Symbol "log" wird hier der Logarithmus zur Basis 2 verstanden.

250) Vgl. Dantzig (1979), S. 1.

251) Vgl. Dantzig (1966), S. 185; Klee (1972), S. 174f.; Dantzig (1979), S. 1.

m(log n)[252] geschätzt. Borgwardt[253] hat in einem komplexen (probabilistisch-)analytischen Ansatz sogar streng nachgewiesen, daß die durchschnittliche Komplexität (einer Variante) des Simplex-Algorithmus durch das Polynomial $O(m,n) = mn^2(n+1)^2$ gegeben ist[254].

Im Gegensatz hierzu haben worst case-Analysen zu dem Ergebnis geführt, daß der Simplex-Algorithmus "theoretisch" ineffizient ist[255], weil sein Lösungsaufwand für den schlechtest möglichen Fall nicht polynomial, sondern nur exponentiell[256] beschränkt werden kann. Er beträgt bei n Variablen und m=2n Restriktionen des zu lösenden linearen Optimierungsproblems $O(n) = 2^{n-1}$, wie erstmals für die Variante der "approximate steepest ascent"-Pivotierungsregel[257] von Klee und Minty nachge-

---

252) Vgl. Schrader (1982), S. 267; Schrader (1983), 2; Pan (1986), S. 134.

253) Vgl. Borgwardt (1982), S. 176 i.V.m. S. 160 (Prämissen des probabilistischen Modells) u. 161ff. Bei Borgwardt (1985), S. 656ff., finden sich differenzierte Ergebnisse für unterschiedliche Prämissen hinsichtlich der Wahrscheinlichkeitsverteilungen.

254) Eine ähnliche analytische average case-Analyse stellt Smale (1983), S. 531ff., vor. Sein Komplexitätsresultat (S. 532) ist jedoch weniger anschaulich, weil es sich nicht auf die Parameter n und m als explizite unabhängige Variable beschränkt und die Restriktionen-Anzahl m konstant hält. Vgl. zu weiteren average case-Analysen die annotierte Bibliographie bei Karp (1985), S. 77ff.; vgl. auch Karp (1986), S. 108.

255) Vgl. Karp (1975a), S. 24

256) Vgl. Bachem (1980), S. 833; Schönlein (1981), S. 116f.; Johnson (1981), S. 400; Lenstra (1982), S. 204; Derigs (1986), S. 49.

257) Spätere Untersuchungen auf der Basis von alternativen Pivotierungsregeln führten zum gleichen Ergebnis einer nur exponentiell beschränkten Komplexität; vgl. Goldfarb (1979), S. 278 u. 284 i.V.m. S. 278ff., mit dem gleichen Ergebnis (s.o.) für die "steepest edge"-Pivotierungsregel, Bland (1981), S. 1081; Schrader (1982), S. 267; Schrader (1983), S. 2.

wiesen wurde[258]. Dieses exponentiell beschränkte Anwachsen des Lösungsaufwands gilt nicht nur für eine kleine, endliche Zahl von "pathologischen" Problemausprägungen, sondern konnte für eine unendlich große Menge ähnlicher Ausprägungen festgestellt werden.

Der praktisch akzeptable Ressourceneinsatz für den Simplex-Algorithmus trotz seiner theoretischen Ineffizienz hat lange Zeit Irritationen ausgelöst[259]. Es besteht anscheinend das Bedürfnis, Probleme, für die durchschnittlich effiziente Lösungsalgorithmen bekannt sind, auch in den schlechtest möglichen Fällen polynomial beschränkt bewältigen zu können. Tatsächlich gelang es seit Mitte der siebziger Jahre, mehrere Algorithmen für die Lösung von linearen Programmen zu konstruieren, die sich bei worst case-Analysen als effizient herausstellen. Zu den bekanntesten dieser Algorith-

---

258) Vgl. Klee (1972), S. 174 i.V.m. S. 161ff.; Bland (1981), S. 1081; Schrader (1982), S. 267; Weber (1982), S. B/229; Schrader (1983), S. 2; Borgwardt (1985), S. 650; Pan (1986), S. 134. Mitunter wird hier auch der - um den Subtrahenden 1 - reduzierte Term 2^n angeführt.
Borgwardt (1985), S. 650, führt zusätzlich eine präzisere Betimmung der Komplexität des Simplex-Algorithmus an, die aber auch polynomial beschränkt bleibt (für beliebige Parameter m und n; die Restriktion m=2n braucht nicht mehr erfüllt zu werden). Vgl. auch die dort angegebenen weiterführenden Quellen.

259) Vgl. etwa die Äußerung von Schönlein (1981), S. 117: "Für das Phänomen des nahezu polynomialen 'worst case behaviour' in der Praxis gibt es bis heute keine Erklärung." Derigs (1986), S. 48, spricht von dem "Widerspruch 'praktische Effizienz contra theoretische Ineffizienz'", der die wesentlichen Arbeiten zum Problem der linearen Programmierung in den vergangenen Jahre, beherrscht habe. Den gleichen Widerspruch führt auch Schönlein (1986), S. 344, an. Vgl. ebenso Parker (1982a), S. 7; Hooker (1986), S. 76f.

men[260] zählen die Lösungsmethoden von Khachiyan[261] und Karmarkar[262]. Ihre Komplexitätsfunktionen sind durch Polynomiale beschränkt[263].

Diese Polynome nehmen zwar infolge der binären Codierung der Koeffizienten eines linearen Programms eine etwas unübersichtliche Form an. Doch läßt sich ihre komplexitätstheoretisch wesentliche Aussage dadurch herausstellen, daß von der Koeffizientencodierung (und von technischen Ganzzahligkeitsaspekten) abstrahiert

---

260) Ein weiterer polynomial beschränkter Algorithmus, der sich von den nachfolgend angeführten deutlich unterscheidet, wird von Tardos (1986), S. 251ff., beschrieben.

261) Vgl. Khachiyan,L.G. (1979), S. 191ff.; Padberg (1980), S. 3ff.; Bachem (1980), S. 833ff.; Bland (1981), S. 1044ff.; Grötschel (1981), S. 172ff.; Gacs (1981), S. 62ff.; Schönlein (1981), S. 117ff.; Schrader (1982), S. 274ff.; Weber (1982), S. B/230ff.; Schrader (1983), S. 4ff.
Grötschel (1981), S. 170 u. 177ff., und Schrader (1983), S. 10 u. 12, heben die Bedeutung des Khachiyan-Algorithmus für die Komplexitätstheorie hervor, die über den Nachweis der polynomial beschränkten Komplexität des Problems der linearen Optimierung weit hinausreicht. Diese Bedeutung erstreckt sich vor allem auf die Funktion, den Weg für die Suche nach polynomial beschränkten Lösungsalgorithmen für kombinatorische (ganzzahlige) Probleme zu bereiten.

262) Vgl. Karmarkar (1984), S. 377ff.; Schönlein (1986), S. 344ff.; Derigs (1986), S. 51ff.; Hooker (1986), S. 77ff. Eine Fortentwicklung von Karmarkars Algorithmus, die auf Multiprozessor-Automaten parallel ausgeführt werden kann, stellen Pan und Reif in Pan (1986), S. 132ff., vor.

263) Vgl. - in bezug auf den Khachiyan-Algorithmus - Khachiyan (1979), S. 191 u. 194; Bland (1981), S. 1046; Schönlein (1981), S. 116 u. 120

wird, indem nur noch der Umfang L[264] des bereits codiert vorliegenden Problems betrachtet wird[265]. Für ein lineares Programm mit n Variablen und m Restriktionen ergibt sich dann als Komplexität des Khachiyan-Algorithmus das Polynomial $O(L,m,n) = L(m+n^2)n^3$ [266] oder - bei variierender Algorithmuskonstruktion - ein ähnliches, im wesentlichen von L linear und von n quadra-

264) Vgl. zur Codierung der Problembeschreibung und zum hieraus abgeleiteten Problemumfang L Khachiyan (1979), S. 191; Padberg (1980), S. 1; Bland (1981), S. 1047 u. 1079; Weber (1982), S. B/231; Karmarkar (1984), S. 377 u. 393; Derigs (1986), S. 50. Die dort ausgewiesenen Terme für den Problemumfang differieren leicht - je nach den zugrundegelegten Codierungsprämissen -, so daß die resultierenden Komplexitätsmaße für die Algorithmen entsprechend variieren. Von diesen codierungsbedingten, für die Einordnung in Komplexitätsklassen irrelevanten Nuancen wird fortan abgesehen. Eine Vorstellung von der Größenordnung des Terms L vermittelt Dantzig (1979), S. 2: Bei einem Problem mit n Variablen, m Restriktionen, einer Spannweite der Koeffizientenwerte von 100.000 und einer Genauigkeit der numerischen Darstellung von 5 signifikanten Dezimalstellen gilt L = 30mn. Abweichender Ansicht ist Karmarkar (1984), S. 393, der L in praktischen Anwendungsfällen für deutlich kleiner als n hält.

265) Diese Vorgehensweise auf der Grundlage des codierten Problemumfangs L kann in Zweifel gezogen werden. Vgl. hierzu die subtile, von L abhängige Unterscheidung zwischen reell- und ganzzahligen Berechnungskonzepten bei Traub (1982), S. 60ff. Dort wird aufgezeigt, daß der Problemumfang L nur für das ganzzahlige Standard-Berechnungskonzept der Komplexitätstheorie Relevanz besitzt. Der Aufwand für die Ausführung einer Berechnungsoperation ist hier proportional zum Umfang der Operanden. Für das reellzahlige Alternativkonzept, in dem der Ausführungsaufwand konstant, also gegenüber der Operandenlänge invariant ist, spielt dagegen der Problemumfang keine Rolle mehr. Diese Differenzierung führt zu der wesentlichen Konsequenz, daß der Khachiyan-Algorithmus im reellzahligen alternativen Berechnungskonzept nicht mehr polynomial beschränkt, darüber hinaus sogar unbeschränkt ist; vgl. Traub (1982), S. 61f.; Schrader (1983), S. 7. Johnson (1981), S. 399, hält die Komplexität des Problems der linearen Programmierung im reellzahligen Berechnungskonzept für noch unbekannt.
In dieser Arbeit wird aber stets vom komplexitätstheoretischen Standardansatz des ganzzahligen Berechnungskonzepts ausgegangen. Das Alternativkonzept wird hingegen z.B. von Smale (1983), S. 530, und Tardos (1986), S. 250, vorausgesetzt.

266) Vgl. Khachiyan (1979), S. 191.

tisch abhängendes Polynomial[267]. Das entsprechende Polynomial für den Karmarkar-Algorithmus lautet dagegen $O(L,n) = L^2(\log L)(\log(\log L))n^{3,5}$ [268] oder - bei leicht abweichender Komplexitätsermittlung[269] - $O(L,m,n) = L^2m^2n^{1,5}$ [270]. Somit ist der Karmarkar-Algorithmus - im Falle der worst case-Analyse und im Hinblick auf die Variablenanzahl n - effizienter als der Khachiyan-Algorithmus[271].

Diese Entwicklung neuer Lösungsalgorithmen für das Problem der linearen Optimierung läßt das o.a. Validitätsdefizit der Komplexitätstheorie für das Konzept der Problemkomplexität im schwachen Sinne deutlich hervortreten. Obwohl sich das zugrundeliegende Problem in keiner Weise änderte, verschob sich das Urteil über

---

267) Vgl. Dantzig (1979), S. 2; Padberg (1980), S. 1; Bland (1981), S. 1048; Schrader (1982), S. 277; Schrader (1983), S. 6. Diese Autoren berücksichtigen (zunächst) nur die unabhängigen Variablen n und L (nicht aber m), gelangen diesbezüglich jedoch zum qualitativ gleichwertigen Ergebnis einer polynomial beschränkten Komplexität. Dantzig gibt z.B. das Polynomial $O(L,n) = 4L(n+1)^2$, gerundet auf $4Ln^2$ an, während Padberg $O(L,n) = 16Ln^2$ nennt. Schrader (1982) führt das Polynomial $O(L,n) = L6n(n+1)$ an. Nach Schrader (1982), S. 288 u. 296, besitzen auch alle nachträglichen Modifizierungen des Khachiyan-Algorithmus (die auf S. 279ff. vorgestellt werden) die charakteristische, polynomial beschränkte Komplexität der Größenordnung $O(L,n) = Ln^2$.
Karmarkar (1984), S. 377, und Hooker (1986), S. 84, konstatieren dagegen auch eine quadratische Abhängigkeit vom Problemumfang, indem sie die Polynomiale $O(L,n) = L^2(\log L)(\log(\log L))n^6$ bzw. $O(L,n) = L^2n^6$ als Komplexität des Khachiyan-Algorithmus ausweisen.

268) Vgl. Karmarkar (1984), S. 376. Hooker (1986), S. 84 u. 87ff., beschränkt seine Angabe $O(L,n) = L^2n^{3,5}$ auf die nicht-logarithmischen Terme. Pan (1986), S. 133, gibt die abweichende, in der Größenordnung aber ähnliche Komplexität $O(L,m) = Lm^{3,5}$ an, die durch algorithmische Verfeinerungen noch auf das Polynomial $O(L,m) = Lm^{3,165}$ reduziert wird (S. 134).

269) Karmarkar und Derigs messen den codierten Problemumfang L in unterschiedlicher Weise; Karmarkar nimmt in seiner Komplexitätsargumentation auf die Restriktionen-Anzahl m keinen Bezug.

270) Vgl. Derigs (1986), S. 54. Ein abweichendes, nicht auf den codierten Problemumfang L, dafür aber auf die Lösungsgenauigkeit Bezug nehmendes Polynomial gibt Schönlein (1986), S. 352, an.

271) Vgl. Karmarkar (1984), S. 373 u. 377; Hooker (1986), S. 77; Tardos (1986), S. 250.

"seine" Komplexität durch neue algorithmische Konzepte. Mußte das lineare Optimierungsproblem ehemals - auf der Basis der damals bekannten Lösungsalgorithmen - noch zur Klasse NP gerechnet werden[272], so ist es heute als ein Mitglied der Klasse P erwiesen.

Um so überraschender ist es, daß sich diese theoretisch effizienten Algorithmen bei der Anwendung auf einzelne Ausprägungen linearer Optimierungsprobleme keineswegs praktisch effizienter als der Simplex-Algorithmus herausstellten. Mehrere empirische Analysen wurden ausgeführt, in denen der Lösungsaufwand dieser Algorithmen anhand von "repräsentativen" Testproblemen untersucht wurde. Zumeist erfolgte ein Vergleich mit dem Lösungsaufwand des Simplex-Algorithmus als Maßstab für die praktische Effizienz. Aus den Ergebnissen dieser Studien wird allgemein der Schluß gezogen, die beiden o.a. polynomial beschränkten Algorithmen seien dem Simplex-Algorithmus in der Praxis - zumindest derzeit -

272) Die Komplexität des linearen Programmierungsproblems konnte zwar früher als polynomial beschränkt vermutet werden. Entsprechend rechneten einige Autoren dieses Problem auch noch nicht der Klasse NP zu, sondern bezeichneten seine Zugehörigkeit zur Klasse P als offenes Problem; vgl. etwa Bachem (1980), S. 833; Schrader (1983), S. 1. Doch widersprechen sich die Offenheit bezüglich der Zugehörigkeit zur Klasse P und die Gewißheit der Mitgliedschaft zur Klasse NP nicht, weil P eine Teilklasse von NP bildet. Folgerichtig bezeichnete Garey (1979), S. 155, das Problem der linearen Optimierung noch als NP-komplex.

deutlich unterlegen[273]. Allerdings liegen im Hinblick auf den Karmarkar-Algorithmus erste Hinweise vor, daß sein durchschnittlicher Lösungsaufwand für reale Probleme tatsächlich geringer als der des Simplex-Algorithmus sein könnte[274].

---

273) Vgl. die Ergebnisse von empirischen Effizienzstudien bezüglich des Khachiyan-Algorithmus, die bei Schönlein (1981), S. 120f., wiedergegeben werden. Vgl. auch Bland (1981), S. 1077; Weber (1982), S. B/230; Lenstra (1982), S. 204; Schrader (1983), S. 12; Kolata (1984), S. 1379; Derigs (1986), S. 51; Hooker (1986), S. 77. Schönlein (1981) führt die Ansicht von Rosen und Frawley an, der Khachiyan-Algorithmus sei "- zumindest für kleine Probleme - nicht nur ineffizient, sondern auch numerisch instabil" (S. 120). Er ergänzt die Ansicht von Mc Call, dieser Algorithmus besäße "keinerlei praktische Bedeutung" (S. 121). Im Sinne der praktischen Bedeutungslosigkeit äußern sich auch Traub (1982), S. 59; Parker (1982a), S. 7; Schönlein (1986), S. 344; Derigs (1986), S. 51. Vgl. des weiteren Dantzig (1979), S. 2f., und das artifiziell anmutende, aber Schwächen des Khachiyan-Algorithmus herauskehrende Beispiel bei Bland (1981), S. 1085f.
Vgl. hinsichtlich der empirischen Analysen des Karmarkar-Algorithmus, die zumindest dessen angebliche Überlegenheit noch nicht zu bestätigen vermochten, die skeptischen Anmerkungen von Schönlein (1986), S. 344 u. 353; Derigs (1986), S. 55; Hooker (1986), S. 77 u. 85. Ungeachtet dieser empirischen Erfahrungen soll der Anspruch erhoben werden, der Karmarkar-Algorithmus sei dem Simplex-Algorithmus nicht nur theoretisch, sondern - bei großen Problemausprägungen - auch praktisch überlegen, d.h. der erste weise einen geringeren durchschnittlichen Lösungsaufwand als der zweite auf; vgl. Fricker (1985), S. 32; Hooker (1986), S. 75ff.; Derigs (1986), S. 51 u. 55; Schönlein (1986), S. 344 u. 353. Bei Karmarkar (1984) findet sich allerdings keine klare Aussage, welche den behaupteten Überlegenheitsanspruch decken würde. Dieser rührt vermutlich von der etwas euphorischen Darstellung in Kolata (1984), S. 1379f., her.

274) Vgl. die Ausführungen von Hooker (1986), S. 86, zu den Tests von Adler et al. Besonders interessant ist hierbei der Umstand, daß die relative Geschwindigkeit des Karmarkar-Algorithmus, die auf die (absolute) Lösungsgeschwindigkeit des Simplex-Algorithmus bezogen ist, mit zunehmendem Problemumfang wächst. Allerdings muß auch festgehalten werden, daß die Implementierung von Adler et al. als affine Skalierungs-Methode mit dem Karmarkar-Algorithmus - einer projektiven Skalierungs-Methode - nicht identisch, sondern mit diesem nur konzeptionell eng verwandt ist; vgl. zu diesem Unterschied Hooker (1986), S. 76 u. 86. Vgl. zu den stimulierenden Effizienztests des Karmarkar-Algorithmus auch Frenkel (1986), S. 111.

Während sich der Simplex-Algorithmus bei der average case-Analyse deutlich effizienter als unter den Annahmen der worst case-Analyse verhält[275], liegen die Relationen von durchschnittlicher und schlechtest möglicher Effizienz bei den polynomial beschränkten Algorithmen nicht derart eindeutig fest. Zumindest die Komplexität des Khachiyan-, u.U. auch die des Karmarkar-Algorithmus fällt für den average case nicht offensichtlich niedriger aus als für den worst case[276].

Hiermit wird die oben vorgetragene[277] komplexitätstheoretische Vermutung empirisch widerlegt, ein Lösungsalgorithmus sei (genau) dann praktisch effizient, wenn er sich polynomial beschränkt erweist. In dieser Hinsicht haben neuartige Algorithmen des Operations Research befruchtend - hier: korrigierend - auf die Komplexitätstheorie zurückgewirkt.

Ein derzeit noch nicht überzeugend eingelöstes Forschungsvorhaben der Komplexitätstheorie erstreckt sich daher auf die Suche nach Lösungsalgorithmen für lineare Optimierungsprobleme, die einerseits theoretisch effizient sind und sich anderseits praktisch mindestens genau so effizient verhalten wie der Simplex-Algorithmus. Solange diese Suche noch zu keinem allgemein anerkannten Erfolg geführt hat, kann die Vermutung, daß ein trade off zwischen durchschnittlicher und minimaler Effizienz der Lösung von Problemen bestehe, nicht widerlegt werden. Die o.a. neuesten Resultate bezüglich der Komplexität des Karmarkar-Algorithmus[278] bei average case-Analysen lassen es jedoch begründet erscheinen, diese Widerlegung in näherer Zukunft zu erwarten.

275) Vgl. Schrader (1982), S. 267; vgl. auch das plastische Beispiel bei Klee (1972), S. 175.

276) Vgl. Dantzig (1979), S. 3, in bezug auf den Khachiyan-Algorithmus.

277) Vgl. S. 29f.

278) Von der Differenzierung zwischen affiner und projektiver Skalierungsmethode, die in Fußnote 274) angesprochen wurde, wird hier abgesehen.

Literaturverzeichnis

Appel,K. u. W. Haken: The Solution of the Four-Color-Map Problem, in: Scientific Amaerican, Vol. 237 (1977), No. 4, S. 108-121.

Araki,T. u. T. Kasami: Some Undecidable Problems for Petri Nets, in: Systems, Computers, Controls, Vol. 7 (1976), No. 1, S. 20-28.

Araki,T. u. T. Kasami: Some Decision Problems Related to the Reachability Problem for Petri Nets, in: Theoretical Computer Science, Vol. 3 (1977), S. 85-104 (a).

Araki,T. u. T. Kasami: Decidable Problems on the Strong Connectivity of Petri Net Reachability Sets, in: Theoretical Computer Science, Vol. 4 (1977), S. 99-119 (b).

Asser,G.: Turing-Maschinen und Markowsche Algorithmen, in: Zeitschrift für mathematische Logik und Grundlagen der Mathematik, Bd. 5. (1959), S. 346-365.

Bachem,A.: Komplexitätstheorie im Operations Research, in: Zeitschrift für Betriebswirtschaft, 50. Jg. (1980), S. 812-844.

Baker,H.G.: Rabin's Proof of the Undecidability of the Reachability Set Inclusion Problem of Vector Addition Systems, Computation Structures Group Memo 79, Project MAC am Massachusetts Institute of Technology, o.O. (Cambridge/Massachusetts) 1973.

Baur,W.: Zeitlich beschränkte Turingmaschinen und polynomiale Reduktion, in: Specker,E. u. V. Strassen (Hrsg.): Komplexität von Entscheidungsproblemen, Lecture Notes in Computer Science 43, Berlin - Heidelberg - New York 1976, S. 11-19.

Beensen,R.: Komplexitätsbeherrschung in den Wirtschaftswissenschaften, Berlin 1971.

Benito,F. u. H. Gröflin: Optimierungsprobleme mit nicht polynomial begrenzten Algorithmen, in: Liebling,T.M. u. M. Rössler (Hrsg.): Kombinatorische Entscheidungsprobleme: Methoden und Anwendungen, Lecture Notes in Economics and Mathematical Systems 153, Berlin - Heidelberg - New York 1978, S. 81-121.

Bland,R.G., D. Goldfarb u. M.J. Todd: The Ellipsoid Method: A Survey, in: Operations Research, Vol. 29 (1981), S. 1039-1091.

Blazewicz,J., J.K. Lenstra u. A.H.G. Rinnooy Kan: Scheduling Subject to Resource Constraints: Classification and Complexity, Preprint BW 127/80, stichting mathematisch centrum, afdeling mathematische besliskunde, Amsterdam 1980.

Böhling,K.H. u. B. von Braunmühl: Komplexität bei Turingmaschinen, Mannheim - Wien - Zürich 1974.

Börger,E. u. H. Kleine Büning: The Reachability Problem for Petri Nets and Decision Problems for Skolem Arithmetic, in: Theoretical Computer Science, Vol. 11 (1980), S. 123-143.

Boolos,G. u. R. Jeffrey: Computability and Logic, Cambridge (Großbritannien) 1974.

Borgwardt,K.-H.: The Average Number of Pivot Steps Required by the Simplex-Method is Polynomial, in: Zeitschrift für Operations Research, Vol. 26 (1982), S. 157-177.

Borgwardt,K.H.: Der durchschnittliche Rechenaufwand beim Simplexverfahren, in: Ohse,D., A.C. Esprester, H.-U. Küpper, P. Stähly u. H. Steckhan (Hrsg.): Operations Research Proceedings 1984, DGOR - Vorträge der 13. Jahrestagung, 12.-14.09.1984 in Sankt Gallen, Berlin - Heidelberg - New York - Tokyo 1985, S. 647-660.

Brauer,W. u. K. Indermark: Algorithmen, rekursive Funktionen und formale Sprachen, Mannheim - Wien - Zürich 1968.

Brown,C.A. u. P.W. Purdom: How to Search Efficiently, in: Drinan,A. (Hrsg.): Proceedings of the Seventh International Joint Conference on Artificial Intelligence, IJCAI-81, 24.-28.08.1981 in Vancouver, Vol. 1, o.O. (Menlo Park) 1981, S. 588-594.

Brucker,P., J.K. Lenstra u. A.H.G. Rinnooy Kan: Complexity of Machine Scheduling Problems, Preprint BW 43/75, stichting mathematisch centrum, afdeling mathematische besliskunde, Amsterdam 1975.

Brucker,P.: Die Komplexität von Scheduling Problemen, in: Kohlas,J., O. Seifert, P. Stähly u. H.-J. Zimmermann (Hrsg.): Proceedings in Operations Research 5, Vorträge der Jahrestagung 1975 - DGOR, Würzburg - Wien 1976, S. 357-368 (a).

Brucker,P.: Anmerkungen zu heuristischen Verfahren, in: Dathe,H.N., P. Mertens, F.D. Peschanel, H. Späth u. H.-J. Zimmermann (Hrsg.): Proccedings in Operations Research 6, Vorträge der Jahrestagung 1976 - DGOR, Würzburg - Wien 1976, S. 668-676 (b).

Brucker,P.: NP-vollständige Scheduling-Probleme, in: Noltemeier,H. (Hrsg.): Graphen, Algorithmen, Datenstrukturen, Ergebnisse des Workshops WG 76, 2. Fachtagung über Graphentheoretische Konzepte der Informatik, 16.-18.06.1976 in Göttingen, München - Wien 1976, S. 135-148 (c).

Brucker,P.: Scheduling, Wiesbaden 1981.

Burks,A.W. (Hrsg. und vervollständigender Verf.): Theory of Self-Reproducing Automata - John von Neumann, Urbana - London 1966.

Byrn,W.H.: Sequential Processes, Deadlocks, and Semaphore Primitives, Dissertation am Department of Applied Mathematics, Harvard University, Cambridge (Massachusetts) 1974.

Cardoza,E., R. Lipton u. A.R. Meyer: Exponential Space Complete Problems for Petri Nets and Commutative Semigroups: Preliminary Report, in: Conference Record of The Eigth Annual ACM Symposium on Theory of Computing, 3.-5.05.1976 in Hershey, New York 1976, S. 50-54.

Church,A.: An Unsolvable Problem of Elementary Number Theory, in: American Journal of Mathematics, Vol. 58 (1936), S. 345-363 (a).

Church,A.: A Note on the Entscheidungsproblem, in: The Journal of Symbolic Logic, Vol. 1 (1936), S. 40-41 (b).

Church,A.: Correction to A Note on the Entscheidungsproblem, in: The Journal of Symbolic Logic, Vol. 1 (1936), S. 101-102 (c).

Clausen,J. u. J. Krarup: Combinatorial Optimization: Challenges and Trends, in: Streitferdt,L., H. Hauptmann, A.W. Marusev, D. Ohse u. U. Pape (Hrsg.): Operations Research Proceedings 1985, DGOR - Vorträge der 14. Jahrestagung, Berlin - Heidelberg - New York - Tokyo 1986, S. 24-46.

Coffman,E.G., G.N. Frederickson u. G.S. Lueker: Probabilistic Analysis of the LPT Processor Scheduling Heuristic, in: in: Dempster,M.A.H., J.K. Lenstra u. A.H.G. Rinnooy Kan (Hrsg.): Deterministic and Stochastic Scheduling, Proceedings of the NATO Advanced Study and Research Institute on Theoretical Approaches to Scheduling Problems, 6.-17.07.1981 in Durham, Dordrecht - Boston - London 1982, S. 319-331.

Cohors-Fresenborg,E.: Mathematik mit Kalkülen und Maschinen, Braunschweig 1977.

Cook,S.A.: The Complexity of Theorem-Proving Procedures, in: o.V.: Conference Record of the 3rd Annual ACM Symposium on Theory of Computing, 1971 in Urbana, o.O. (New York) 1971, S. 151-158.

Cook,S.A.: An Observation on Time-Storage Trade Off, in: o.V.: Conference Record of the 5th Annual ACM Symposium on Theory of Computing, New York 1973, S. 29-33.

Cook,S.A.: An Overview of Computational Complexity, in: Communications of the ACM, Vol. 26 (1983), S. 401-408.

Crespi-Reghizzi,S. u. D. Mandrioli: Some Algebraic Properties of Petri Nets, in: Alta Frequenza, Vol. 45 (1976), N. 2, S. 130-137.

Crespi-Reghizzi,S. u. D. Mandrioli: Petri Nets and Szilard Languages, in: Information and Control, Vol. 33 (1977), S. 177-192.

Dantzig,G.B.: Lineare Programmierung und Erweiterungen, Berlin - Heidelberg - New York 1966.

Dantzig,G.B.: Comments on Khachiyan's Algorithm for Linear Programming, Technical Report SOL 79-22 am Systems Development Laboratory, Stanford University, Stanford 1979.

Davis,M.: Computability and Unsolvability, New York - Toronto - London 1958.

Davis,M.: Hilbert's Tenth Problem is Unsolvable, in: The American Mathematical Monthly, Vol. 80 (1973), S. 233-269 (a).

Davis,M.: Hilbert's 10th Problem, in: Scientific American, Vol. 229 (1973), No. 5, S. 84-91 (b).

Derigs,U.: Neuere Ansätze in der Linearen Programmierung - Motivation, Konzepte und Verfahren, in: Streitferdt,L., H. Hauptmann, A.W. Marusev, D. Ohse u. U. Pape (Hrsg.): Operations Research Proceedings 1985, DGOR - Vorträge der 14. Jahrestagung, Berlin - Heidelberg - New York - Tokyo 1985, S. 47-58.

Dewdney,A.K.: Computer-Kurzweil: Eine Computerfamilie für den fleißigen Biber ..., in: Spektrum der Wissenschaft, o.Jg. (1984), Heft 11, S. 8-16.

Ecker,K.: Organisation von parallelen Prozessen - Theorie deterministischer Schedules, Mannheim - Wien - Zürich 1977.

Ernst,D., K. Garbrecht, E. Golling, F. Gudden, E. Hofmeister, H. Kiemle, H. Morgenbrod, P. Müller-Stoy, H. Stegmeier u. W. Urbach: Chancen mit Chips - Zwischenbilanz einer Basistechnologie, Berlin - München 1984.

Fischer,M.J. u. M.O. Rabin: Super-Exponential Complexity Of Presburger Arithmetic, in: SIAM-AMS Proceedings, Vol. 7 (1974), S. 27-41.

Fischer,P.C.: On Formalisms for Turing Machines, in: Journal of the Association for Computing Machinery (ACM), Vol. 12 (1965), S. 570-580.

Fisher,M.L.: Worst-Case Analysis of Heuristic Algorithms, in: Management Science, Vol. 26 (1980), S. 1-17.

Fisher,M.L.: Worst-Case Analysis of Heuristic Algorithms for Scheduling and Picking, in: Dempster,M.A.H., J.K. Lenstra u. A.H.G. Rinnooy Kan (Hrsg.): Deterministic and Stochastic Scheduling, Proceedings of the NATO Advanced Study and Research Institute on Theoretical Approaches to Scheduling Problems, 6.-17.07.1981 in Durham, Dordrecht - Boston - London 1982, S. 15-34.

French,S.: Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop, New York - Chichester - Brisbane - Toronto 1982.

Frenkel,K.A.: Piecing Together Complexity, in: Communications of the ACM, Vol. 29 (1986), S. 110-111.

Frickler,F.: Komplexe Optimierungsaufgaben schneller lösen, in: Frankfurter Allgemeine Zeitung, Ausgabe vom 27.03.1985, S. 32.

Gacs,P. u. L. Lovasz: Khachiyan's Algorithm for Linear Programming, in: Mathematical Programming Studies, Vol. 14 (1981), S. 61-68.

Garey,M.R. u. D.S. Johnson: Complexity Results for Multiprocessor Scheduling under Resource Constraints, in: SIAM Journal on Computing, Vol. 4 (1975), S. 397-411.

Garey,M.R., D.S. Johnson u. R. Sethi: The Complexity of Flowshop and Jobshop Scheduling, in: Mathematics of Operations Research, Vol. 1 (1976), S. 117-129 (a).

Garey,M.R. u. D.S. Johnson: Approximation Algorithms for Combinatorial Problems: An Annotated Bibliography, in: Traub,J.F. (Hrsg.): Algorithms and Complexity - New Directions and Recent Results, New York - San Francisco - London 1976, S. 41-49 (b).

Garey,M.R. u. D.S. Johnson: "Strong" NP-Completeness Results: Motivation, Examples, and Implications, in: Journal of the Association for Computing Machinery (ACM), Vol. 25 (1978), S. 499-508.

Garey,M.R. u. D.S. Johnson: Computers and Intractability - A Guide to the Theory of NP-Completeness, San Francisco 1979.

Gass,S.I.: Decision-Aiding Models: Validation, Assessment, and Related Issues for Policy Analysis, in: Operations Research, Vol. 31 (1983), S. 603-631.

Gödel,K.: Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I, in: Monatshefte für Mathematik und Physik, 38. Bd. (1931), S. 173-198.

Goldberg,A. u. I. Pohl: Is Complexity Theory of Use to AI?, in: Elithorn,A. u. R. Banerji (Hrsg.): Artificial and Human Intelligence, Edited Review Papers presented at the International NATO Symposium on Artificial and Human Intelligence, im Oktober 1981 in Lyon, Amsterdam - New York - Oxford 1984, S. 43-55.

Goldfarb,D. u. W.Y. Sit: Worst Case Behavior of the Steepest Edge Simplex Method, in: Discrete Applied Mathematics, Vol. 1 (1979), S. 277-285.

Gonzalez,T. u. S. Sahni: Flowshop and Jobshop Schedules: Complexity and Approximation, in: Operations Research, Vol. 26 (1978), S. 36-52.

Grabowski,J.: On Hack's Conjecture Concerning Reachability in Petri Nets, in: Elektronische Informationsverarbeitung und Kybernetik, Vol. 15 (1979), S. 339-354.

Grabowski,J.: Linear Methods in the Theory of Vector Addition Systems I, in: Elektronische Informationsverarbeitung und Kybernetik, Vol. 16 (1980), S. 207-236.

Graham,R.L., E.L. Lawler, J.K. Lenstra u. A.H.G. Rinnooy Kan: Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey, in: Hammer, P.L., E.L. Johnson u. B.H. Korte (Hrsg.): Discrete Optimization II, Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications, im August 1977 in Banff und Vancouver, zugleich: Annals of Discrete Mathematics, Vol. 5 (1979), Amsterdam - New York - Oxford 1979, S. 287-326.

Grötschel,M., L. Lovasz u. A. Schrijver: The Ellipsoid Method and its Consequences in Combinatorial Optimization, in: Combinatorica, Vol. 1 (1981), S. 169-197.

Hack,M.(H.T.): A Petri Net Version of Rabin's Undecidability Proof for Vector Addition Systems, Computer Structures Group Memo 94, Project MAC am Massachusetts Institute of Technology, Cambridge (Massachusetts) 1973.

Hack,M.H.T.: Decidability Questions for Petri Nets, Dissertation am Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge (Massachusetts) 1975.

Hack,M.(H.T.): The Equality Problem for Vector Addition Systems is Undecidable, in: Theoretical Computer Science, Vol. 2 (1976), S. 77-95.

Hajek,P.: Arithmetical Hierarchy and Complexity of Computation, in: Theoretical Computer Science, Vol. 8 (1979), S. 227-237.

Hall,N.G. u. W.T. Rhee: Average and worst-case analysis of heuristics for the maximum tardiness problem, in: European Journal of Operational Research, Vol. 26 (1986), S. 272-277.

Hansen,P. u. B. Simeone: Report of the Session on Structural Aspects of Discrete Problems, in: Hammer, P.L., E.L. Johnson u. B.H. Korte (Hrsg.): Discrete Optimization I, Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications, im August 1977 in Banff und Vancouver, zugleich: Annals of Discrete Mathematics, Vol. 4 (1979), Amsterdam - New York - Oxford 1979, S. 177-181.

Hartmanis,J. u. R.E. Stearns: On the Computational Complexity of Algorithms, in: Transactions of the American Mathematical Society, Vol. 117 (1965), S. 285-306.

Hartmanis,J. u. L. Berman: On Isomorphisms and Density of NP and Other Complete Sets, in: o.V.: Conference Record of The Eigth Annual ACM Symposium on Theory of Computing, 3.-5.05.1976 in Hershey, New York 1976, S. 30-40 (a).

Hartmanis,J. u. J.E. Hopcroft: Independence results in computer science, in: SIGACT Newsletter, No. 8 (1976), S. 13-24 (b).

Häussler,A.: Polynomial beschränkte nichtdeterministische Turingmaschinen und die Vollständigkeit des aussagenlogischen Erfüllungsproblems, in: Specker,E. u. V. Strassen (Hrsg.): Komplexität von Entscheidungsproblemen, Lecture Notes in Computer Science 43, Berlin - Heidelberg - New York 1976, S. 20-35.

Heinemann,B.: Teilklassen der selbst-modifizierenden Netze, Bericht Nr. 69 am Fachbereich Informatik der Universität Hamburg, Hamburg 1980.

Hermes,H.: Definite Begriffe und berechenbare Zahlen, in: Semesterberichte zur Pflege des Zusammenhangs von Universität und Schule aus den mathematischen Seminaren, 10. Bd. (1937), S. 110-123.

Hermes,H.: Maschinen zur Entscheidung von mathematischen Problemen, in: Mathematisch=physikalische Semesterberichte, Bd. II (1952), S. 179-189.

Hermes,H.: Die Universalität programmgesteuerter Rechenmaschinen, in: Mathematisch=physikalische Semesterberichte, Bd. IV (1954), S. 42-53.

Hermes,H.: Aufzählbarkeit, Entscheidbarkeit, Berechenbarkeit - Einführung in die Theorie der rekursiven Funktionen, 3. Aufl., Berlin - Göttingen - Heidelberg 1978.

Herschel,R.: Einführung in die Theorie der Automaten, Sprachen und Algorithmen, München - Wien 1974.

Hilbert,D.: Mathematische Probleme. Vortrag, gehalten auf dem internationalen Mathematiker-Kongreß zu Paris 1900., in: Nachrichten von der Königl(ichen) Gesellschaft der Wissenschaften zu Göttingen. Mathematisch-physikalische Klasse aus dem Jahre 1900, Göttingen 1900, S. 253-297.

Hooker,J.N.: Karmarkar's Linear Programming Algorithm, in: Interfaces, Vol. 16 (1986), No. 4, S. 75-90.

Hopcroft,J.E.: Complexity of Computer Computations, in: Rosenfeld,J.L. (Hrsg.): Information Processing 74, Proceedings of the IFIP Congress 1974, 5.-10.08.1974 in Stockholm, Amsterdam - London - New York 1974, S. 620-626.

Hopcroft,J.(E.) u. J.-J. Pansiot: On the Reachability Problem for 5-Dimensional Vector Addition Systems, in: Theoretical Computer Science, Vol. 8 (1979), S. 135-159.

Hopcroft,J.E.: Turingmaschinen, in: Spektrum der Wissenschaft, o.Jg. (1984), Heft 7, S. 34-49.

Hughes,C.E.: The Equivalence of Vector Addition Systems to a Subclass of Post Canonical Systems, in: Information Processing Letters, Vol. 7 (1978), S. 201-204.

Jantzen,M. u. R. Valk: Formal Properties of Place/Transition Nets, in: Brauer,W. (Hrsg.): Net Theory and Applications, Proceedings of the Advanced Course on General Net Theory of Processes and Systems, 8.-19.10.1979 in Hamburg, Lecture Notes in Computer Science 84, Berlin - Heidelberg - New York 1980, S. 165-212.

Jantzen,M.: The Large Markings Problem, in: Special Interest Group "Petri Nets and Related System Models" der Gesellschaft für Informatik, Newsletter 14 (1983), S. 24-25.

Jeroslow,R.G.: There Cannot be any Algorithm for Integer Programming with Quadratic Constraints, in: Operations Research, Vol. 21 (1973), S. 221-224.

Johnson,D.S.: The NP-Completeness Column: An Ongoing Guide, in: Journal of Algorithms, Vol. 2 (1981), S. 393-405.

Jones,N.D., L.H. Landweber u. Y.E. Lien: Complexity of Some Problems in Petri Nets, in: Theoretical Computer Science, Vol. 4 (1977), S. 277-299.

Kaphengst,H.: Eine abstrakte programmgesteuerte Rechenmaschine, in: Zeitschrift für mathematische Logik und Grundlagen der Mathematik, Bd. 4 (1958), S. 366-379.

Kalmar,L.: Über ein Problem, betreffend die Definition des Begriffes der allgemein-rekursiven Funktion, in: Zeitschrift für Mathematische Logik und Grundlagen der Mathematik, Bd. 1 (1955), S. 93-96.

Karmarkar,N.: A New Polynomial-Time Algorithm for Linear Programming, in: Combinatorica, Vol. 4 (1984), S. 373-395.

Karp,R.M.: Reducibility Among Combinatorial Problems, in: Miller,R.E. u. J.W. Thatcher (Hrsg.): Complexity of Computer Computations, Proceedings of a Symposium on the Complexity of Computer Computations, 20.-22.03.1972 in New York, New York - London 1972, S. 85-103.

Karp,R.M.: The Fast Approximate Solution of Hard Combinatorial Problems, in: o.V.: Proceedings of the 6th South-Eastern Conference on Combinatorics, Graph Theory, and Computing, Winnipeg 1975, S. 15-31 (a).

Karp,R.M.: On the Computational Complexity of Combinatorial Problems, in: Networks, Vol. 5 (1975), S. 45-68 (b).

Karp,R.M.: The Probabilistic Analysis of Some Combinatorial Search Algorithms, in: Traub,J.F. (Hrsg.): Algorithms and Complexity - New Directions and Recent Results, New York - San Francisco - London 1976, S. 1-19.

Karp,R.M.: Probabilistic Analysis of Partitioning Algorithms for the Traveling-Salesman Problem in the Plane, in: Mathematics of Operations Research, Vol. 2 (1977), S. 209-224.

Karp,R.M., J.K. Lenstra, C.J.H. McDiarmid u. A.H.G. Rinnooy Kan: Probabilistic Analysis, in: O'hEigeartaigh,M., J.K. Lenstra u. A.H.G. Rinnooy Kan (Hrsg.): Combinatorial Optimization - Annotated Bibliographies, Chichester - New York - Brisbane - Toronto - Singapore 1985, S. 52-88.

Karp,R.M.: Combinatorics, Complexity, and Randomness, in: Communications of the ACM, Vol. 29 (1986), S. 98-109.

Kawamura,K. u. D.W. Malone: Probing Complexity in Social Systems Through Interpretive Structural Modeling, in: Finsterbusch,K. u. C.P. Wolf (Hrsg.): Methodology of Social Impact Assessment, Stroudsburg 1977, S. 347-354.

Keramidis,S. u. W. Grote: Beiträge zur Lösung des Verklemmungsproblems in prioritätsfreien Betriebsmittelmaschinen und Petri-Netzen, Arbeitsberichte des Instituts für Mathematische Maschinen und Datenverarbeitung (Informatik) an der Universität Erlangen-Nürnberg, Bd. 12, Nr. 9, Erlangen 1979.

Khachiyan,L.G.: A Polynomial Algorithm in Linear Programming, in: Soviet Mathematics Doklady, Vol. 20 (1979), No. 1, S. 191-194. (Anmk.: Die Schreibweise "Khachiyan" lehnt sich an die überwiegend übliche Transkription des kyrillischen Originals an. In der o.a. Quelle wird die seltenere Transkription "Hacijan" verwendet.)

Kindervater,G.A.P. u. J.K. Lenstra: Parallel Algorithms, in: O'hEigeartaigh,M., J.K. Lenstra u. A.H.G. Rinnooy Kan (Hrsg.): Combinatorial Optimization - Annotated Bibliographies, Chichester - New York - Brisbane - Toronto - Singapore 1985, S. 106-128.

Kirsch,W.: Die Handhabung von Entscheidungsproblemen, München 1978.

Klee,V. u. G.J. Minty: How Good Is the Simplex Algorithm?, in: Shisha,O. (Hrsg.): Symposium on Inequalities III, New York 1972, S. 159-175.

Kleene,S.C.: λ-Definability and Recursiveness, in: Duke Mathematical Journal, Vol. 2 (1936), S. 340-353.

Kleene,S.C.: Introduction to Metamathematics, Amsterdam - Groningen 1952.

Kolata,G.: A Fast Way to Solve Hard Problems, in: Science, Vol. 225 (1984), S. 1379-1380.

Korte,B.: Was ist kombinatorische Optimierung?, Report No. 85370-OR am Institut für Ökonometrie und Operations Research, Abteilung Operations Research an der Universität Bonn, Bonn 1985.

Kosaraju,R.: Decidability of Reachability in Vector Addition Systems - Perliminary Version, in: o.V.: Conference Record of the 14th Annual ACM Symposium on Theory of Computing, o.O. (New York) 1982, S. 267-281.

Lawler,E.L.: Combinatorial Optimization: Networks and Matroids, New York - Chicago - ... - London - Sydney 1976.

Lenstra,J.K. u. A.H.G. Rinnooy Kan: A Note on the Expected Performance of Branch-and-Bound Algorithms, Prepublication BW 63/76, stichting mathematisch centrum, afdeling mathematische besliskunde, Amsterdam 1976.

Lenstra,J.K., A.H.G. Rinnooy Kan u. P. Brucker: Complexity of Machine Scheduling Problems, in: Hammer,P.L., E.L. Johnson, B.H. Korte u. G.L. Nemhauser (Hrsg.): Annals of Discrete Mathematics, Vol. 1 (1977), Studies in Integer Programming, Amsterdam - New York - Oxford 1977, S. 343-362.

Lenstra,J.K. u. A.H.G. Rinnooy Kan: Complexity of Scheduling under Precedence Constraints, in: Operations Research, Vol. 26 (1978), S. 22-35.

Lenstra,J.K. u. A.H.G. Rinnooy Kan: Computational Complexity of Discrete Optimization Problems, in: Hammer, P.L., E.L. Johnson u. B.H. Korte (Hrsg.): Discrete Optimization I, Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications, im August 1977 in Banff und Vancouver, zugleich: Annals of Discrete Mathematics, Vol. 4 (1979), Amsterdam - New York - Oxford 1979, S. 121-140.

Lenstra,J.K., A.H.G. Rinnooy Kan u. P. van Emde Boas: An appraisal of computational complexity for operations researchers, in: European Journal of Operational Research, Vol. 11 (1982), S. 201-210.

Li,G. u. B.W. Wak: How to Cope With Anomalies in Parallel Approximate Branch-And-Bound Algorithms, in: o.V.: Proceedings of the Fourth Annual National Conference on Artificial Intelligence (AAAI-84), 1984 in Austin, Los Altos 1984, S. 212-215.

Lipton,R.L.: The Reachability Problem Requires Exponential Space, Report No. 62 am Department of Computer Science, Yale University, New Haven 1975.

Lipton,R.L.: Model Theoretic Aspects of Computational Complexity, in: o.V.: Proceedings of 19th Annual Symposium on Foundations of Computer Science, New York 1978, S. 193-200.

Ludewig,J., U. Schult u. F. Wankmüller: Chasing the Busy-Beaver - Notes and Observations on a Competition to Find the 5-State Busy Beaver, Bericht Nr. 159 an der Abteilung Informatik II der Universität Dortmund, Dortmund 1983.

Lucas,J.R.: Minds, Machines, and Gödel, in: Philosophy, Vol. 36 (1961), S. 112-127.

Luhmann,N.: Komplexität, in: Grochla,E. (Hrsg.): Handwörterbuch der Organisation, 2. Aufl., Stuttgart 1980, Sp. 1064-1070.

Manders,K. u. L. Adleman: NP-Complete Decision Problems for Quadratic Polynomials, in: o.V.: Conference Record of The Eigth Annual ACM Symposium on Theory of Computing, 3.-5.05.1976 in Hershey, New York 1976, S. 23-29.

Matijasevic,J.V.: Enumerable Sets Are Diophantine, in: Soviet Mathematics Doklady, Vol. 11 (1970), S. 354-358.

Mayr,E.W.: The Complexity of the Finite Containment Problem for Petri Nets, Technical Report TR-181 am Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge (Massachusetts) 1977. (Anmk.: auch veröffentlicht in: Journal of the Association for Computing Machinery (ACM), Vol. 28 (1981), S. 561-576.)

Mayr,E.W.: Ein Algorithmus für das allgemeine Erreichbarkeitsproblem bei Petrinetzen und damit zusammenhängende Probleme, Dissertation and der Technischen Universität München, München 1980.

Mayr,E.W.: An Algorithm for the General Petri Net Reachability Problem, in: o.V.: Conference Record of the 13th Annual ACM Symposium on Theory of Computing, 1981 in Milwaukee, o.O. (New York) 1981, S. 238-246.

Mayr,E.W.: An Algorithm for the General Petri Net Reachability Problem, in: SIAM Journal on Computing, Vol. 13 (1984), S. 441-460.

Mehlhorn,K.: Effiziente Algorithmen, Stuttgart 1977.

Meyer,A.R. u. L.J. Stockmeyer: The Equivalence Problem for Regular Expressions with Squaring Requires Exponential Space, in: o.V.: Proceedings of the 13th Annual IEEE Symposium on Switching and Automata Theory, o.O. 1972, S. 125-129.

Müller,H.: Decidability of Reachability in Persistent Vector Replacement Systems, in Dembinski,P. (Hrsg.): Mathematical Foundations of Computer Science 1980, Proceedings of the 9th Symposium, 1.-5.09.1980 in Rydzyna, Lecture Notes in Computer Science 88, Berlin - Heidelberg - New York 1980, S. 426-438.

Müller,H.: Filling a gap in Kosaraju's proof for the decidability of the reachability problem in VAS, in: Special Interest Group "Petri Nets and Related System Models" der Gesellschaft für Informatik, Newsletter 12 (1982), S. 4-10.

Müller,H.: On Kosaraju's Proof of the Reachability Problem for Vector Addition Systems, in: Priese,L. (Hrsg.): Report on the 1st GTI-Workshop, 10-16.10.1982 in Paderborn, Bericht Nr. 13 der Reihe Theoretische Informatik an der Universität Paderborn, Paderborn 1983, S. 174-183 (a).

Müller,H.: The Reachability Problem for VAS, in: o.V.: Papers presented at the 4th European Workshop on Applications and Theory of Petri Nets, 26.-29.09.1983 in Toulouse, o.O. 1983, S. 4-5 (b).

Müller,H.: ergänzender Vortrag, gehalten am 27.09.1983 anläßlich des 4th European Workshop on Applications and Theory of Petri Nets, 26.-29.09.1983 in Toulouse (ohne Skript) (c).

Müller,H.: The Reachability Problem for VAS, in: Rozenberg,G. (Hrsg.): Advances in Petri Nets 1984, Lecture Notes in Computer Science 188, Berlin - Heidelberg - New York - Tokyo 1985, S. 376-391 (a).

Müller,H.: Weak Petri Net Computers for Ackermann Functions, in: Elektronische Informationsverarbeitung und Kybernetik, Vol. 21 (1985), S. 236-246 (b).

Müller-Merbach,H.: Morphologie heuristischer Verfahren, in: Zeitschrift für Operations Research, Bd. 20 (1976), S. 69-87.

Murata,T.: State Equation, Controllability, and Maximal Matchings of Petri Nets, in: IEEE Transactions on Automatic Control, Vol. AC-22 (1977), S. 412-416.

Myhill,J.: Some Philosophical Implications of Mathematical Logic, in: The Review of Metaphysics, Vol. 6 (1952), S. 165-198.

Nagel,E. u. J.R. Newman: Der Gödelsche Beweis, Wien - München 1964.

Niehuis,S. u. F. Victor: Modellierung von Pr/T-Netzen in Prolog, Arbeitspapiere der GMD (Gesellschaft für Mathematik und Datenverarbeitung mbH/Bonn), Nr. 231, Sankt Augustin 1986.

O'hEigeartaigh,M., J.K. Lenstra u. A.H.G. Rinnooy Kan (Hrsg.): Combinatorial Optimization - Annotated Bibliographies, Chichester - New York - Brisbane - Toronto - Singapore 1985.

Ottmann,T.: Einfache universelle mehrdimensionale Turingmaschinen, Habilitationsschrift an der Universität Karlsruhe, Karlsruhe 1975.

Padberg,M.W. u. M.R. Rao: The Russian Method for Linear Inequalities and Linear Optimization, Report an der New York University Graduate School of Business Administration, New York 1980.

Pan,V. u. J. Reif: Efficient Parallel Linear Programming, in: Operations Research Letters, Vol. 5 (1986), No. 3, S. 127-135.

Papadimitriou,C.H.: Computational Complexity, in: O'hEigeartaigh,M., J.K. Lenstra u. A.H.G. Rinnooy Kan (Hrsg.): Combinatorial Optimization - Annotated Bibliographies, Chichester - New York - Brisbane - Toronto - Singapore 1985, S. 39-51.

Parker,R.G. u. R.L. Rardin: An Overview of Complexity Theory in Discrete Optimizations: Part I. Concepts, in: AIIE Transactions, Vol. 14 (1982), S. 3-10 (a).

Parker,R.G. u. R.L. Rardin: An Overview of Complexity Theory in Discrete Optimization: Part II. Results and Implications, in: AIIE Transactions, Vol. 14 (1982), S. 83-89 (b).

Paul,W.J.: Komplexitätstheorie, Stuttgart 1978.

Peter,R.: Rekursive Funktionen, 2. Aufl., Berlin 1957.

Peterson,J.L.: Petri Nets, in: Computing Surveys, Vol. 9 (1977), S. 223-252.

Peterson,J.L.: An Introduction to Petri Nets, in: Tranter,W.H. (Hrsg.): Proceedings of the National Electronics Conference, Vol. 32, 16.-18.10.1978 in Chicago, Oak Brook 1978, S. 144-148.

Peterson,J.L.: Petri Net Theory and the Modeling of Systems, Englewood Cliffs 1981.

Pfohl,H.-C.: Problemorientierte Entscheidungsfindung in Organisationen, Berlin - New York 1977.

Pinedo,M.: On the Computational Complexity of Stochastic Scheduling Problems, in: Dempster,M.A.H., J.K. Lenstra u. A.H.G. Rinnooy Kan (Hrsg.): Deterministic and Stochastic Scheduling, Proceedings of the NATO Advanced Study and Research Institute on Theoretical Approaches to Scheduling Problems, 6.-17.07.1981 in Durham, Dordrecht - Boston - London 1982, S. 355-365.

Post,E.L.: Finite Combinatory Processes - Formulation I, in: The Journal of Symbolic Logic, Vol. 1 (1936), S. 103-105.

Post,E.L.: Recursively Enumerable Sets of Positive Integers and Their Decision Problems, in: Bulletin of the American Mathematical Society, Vol. 50 (1944), S. 284-316.

Priese,L.: Towards a Precise Characterization of the Complexity of Universal and Nonuniversal Turing Machines, in: SIAM Journal on Computing, Vol. 8 (1979), S. 508-523.

Putnam,H.: Minds and Machines, in: Hook,S. (Hrsg.): Dimensions of Mind, 3. Aufl, New York - London 1966, S. 138-164.

Putnam,H.: Recursive Functions and Hierarchies, in: Papers in the Foundation of Mathematics, published as a supplement to the American Mathematical Monthly, Vol. 80 (1973), S. 68-86.

Rabin,M.O.: Probabilistic Algorithm, in: Traub,J.F. (Hrsg.): Algorithms and Complexity - New Directions and Recent Results, New York - San Francisco - London 1976, S. 21-39.

Rado,T.: On Non-Computable Functions, in: The Bell System Technical Journal, Vol. 41 (1962), S. 877-884.

Rardin,R.L. u. B.W. Lin: Test Problems for Computational Experiments -- Issues and Techniques, in: Mulvey, J.M. (Hrsg.): Evaluating Mathematical Programming Techniques, Proceedings of a Conference, Held at the National Bureau of Standards, 5.-6.01.1981 in Boulder, Lecture Notes in Economics and Mathematical Systems 199, Berlin - Heidelberg - New York 1982, S. 8-15.

Reisig,W.: Petri Nets - An Introduction, EATCS Monographs on Theoretical Computer Science Vol. 4, Berlin - Heidelberg - New York - Tokyo 1985.

Rinnooy Kan,A.H.G.: Machine Scheduling Problems - Classification, complexity and computations, Dissertation an der Universität Amsterdam 1976, Leiden 1976.

Rogers,H.: Theory of Recursive Functions and Effective Computability, New York - Saint Louis - ... - London - Sydney 1967.

Sacerdote,G.S. u. R.L. Tenney: The Decidability of the Reachability Problem for Vector Addition Systems, in: o.V.: Conference Record of the 9th Annual ACM Symposium on Theory of Computing, 2.-4.05.1977 in Boulder (Colorado), New York 1977, S. 61-76.

Sahni,S. u. T. Gonzalez: P-Complete Problems and Approximate Solutions, in: o.V.: Proceedings of the 15th IEEE Annual Symposium on Switching and Automata Theory, New York 1974., S. 28-32.

Sahni,S.: Algorithms for Scheduling Independent Tasks, in: Journal of the Association for Computing Machinery (ACM), Vol. 23 (1976), S. 116-127.

Savage,J.E.: The Complexity of Computing, New York - London - Sydney - Toronto 1976.

Schittkowski,K.: Nonlinear Programming Codes, Lecture Notes in Economics and Mathematical Systems 183, Berlin - Heidelberg - New York 1980.

Schnorr,C.P.: Rekursive Funktionen und ihre Komplexität, Stuttgart 1974.

Schnorr,C.P.: The Network Complexity and the Turing Machine Complexity of Finite Functions, in: Acta Informatica, Vol. 7 (1976), S. 95-107.

Schönlein,A.: Der Algorithmus von Khachian, in: Angewandte Informatik, 23. Jg. (1981), S. 115-121.

Schönlein,A.: Der Algorithmus von Karmarkar - Idee, Realisation, Beispiel und numerische Erfahrungen, in: Angewandte Informatik, 28. Jg. (1986), S. 344-353.

Schrader,R.: Ellipsoid Methods, in: Korte,B. (Hrsg.): Modern Applied Mathematics, Optimization and Operations Research, Amsterdam - New York - Oxford 1982, S. 265-311.

Schrader,R.: The Ellipsoid Method and Its Implications, in: OR Spektrum, Bd. 5 (1983), S. 1-13.

Schuster,P.: Probleme, die zum Erfüllungsproblem der Aussagenlogik polynomial äquivalent sind, in: Specker, E. u. V. Strassen (Hrsg.): Komplexität von Entscheidungsproblemen, Lecture Notes in Computer Science 43, Berlin - Heidelberg - New York 1976, S. 36-48.

Selman,A.L. (Hrsg.): Structure in Complexity Theory, Proceedings of the Conference of Structure in Complexity Theory, 2.-5.06.1986 in Berkeley, Berlin - Heidelberg - New York - Tokyo 1986.

Sethi,R.: On the Complexity of Mean Flow Time Scheduling, in: Mathematics of Operations Research, Vol. 2 (1977), S. 320-330.

Shepherdson,J.C. u. H.E. Sturgis: Computability of Recursive Functions, in: Journal of the Association for Computing Machinery (ACM), Vol. 10 (1963), S. 217-255.

Simon,H.A.: The Architecture of Complexity, in: Proceedings of the American Philosophical Society, Vol. 106 (1962), S. 467-482.

Simon,H.A. u. J.B. Kadane: Problems of Computational Complexity in Artificial Intelligence, in: Traub,J.F. (Hrsg.): Algorithms and Complexity - New Directions and Recent Results, New York - San Francisco - London 1976, S. 281-299.

Simon,H.: Grenzen der Rationalität in Entscheidungsprozessen, in: Journal für Betriebswirtschaftslehre, 30. Jg. (1980), S. 2-17.

Smale,S.: The Problem of the Average Speed of the Simplex Method, in: Bachem,A., M. Grötschel u. B. Korte (Hrsg.): Mathematical Programming - The State of the Art - Bonn 1982, Berlin - Heidelberg - New York - Tokyo 1983, S. 530-539.

Solovay,R. u. V. Strassen: A Fast Monte-Carlo Test for Primality, in: SIAM Journal on Computing, Vol. 6 (1977), S. 84-85.

Starke,P.H.: Petri-Netze - Grundlagen, Anwendungen, Theorie, Berlin (Ost) 1980.

Stegmüller,W.: Unvollständigkeit und Unentscheidbarkeit - Die metamathematischen Resultate von Gödel, Church, Kleene, Rosser und ihre erkenntnistheoretische Bedeutung, 3. Aufl., Wien - New York 1973.

Stockmeyer,L.J. u. A.K. Chandra: Probleme mit nicht auffindbaren Lösungen, in: Spektrum der Wissenschaft, o.Jg. (1979), Heft 7, S. 86-90.

Szyperski,N. u. M. Eul-Bischoff: Interpretative Strukturmodellierung (ISM), Braunschweig - Wiesbaden 1983.

Tardos,E.: A Strongly Polynomial Algorithm to Solve Combinatorial Linear Programs, in: Operations Research, Vol. 34 (1986), S. 250-256.

Traub,J.F. u. H. Wozniakowski: Complexity of Linear Programming, in: Operations Research Letters, Vol. 1 (1982), S. 59-62.

Turing,A.M.: On Computable Numbers, With an Application to the Entscheidungsproblem, in: o.V.: Proceedings of the London Mathematical Society, Second Series, Vol. 42, London 1937, S. 230-265 (a).

Turing,A.M.: On Computable Numbers, With an Application to the Entscheidungsproblem. A Correction, in: o.V.: Proceedings of the London Mathematical Society, Second Series, Vol. 43, London 1937, S. 544-546 (b).

Turing,A.M.: Computability and λ-Definability, in: The Journal of Symbolic Logic, Vol. 2 (1937), S. 153-163 (c).

Ullman,J.D.: Polynomial Complete Scheduling Problems, in: o.V.: Proceedings of the 4th Symposium on Operating System Principles, Yorktown Heights 1973, S. 96-101.

Ullman,J.D.: NP-Complete Scheduling Problems, in: Journal of Computer and System Sciences, Vol. 10 (1975), S. 384-393.

Ullman,J.D.: Complexity of Sequencing Problems, in: Coffman,E.G. (Hrsg.): Computer and Job-Shop Scheduling Theory, New York - London - Sydney - Toronto 1976, S. 139-164.

Ulrich,H.: Die Unternehmung als produktives soziales System, 2. Aufl., Bern - Stuttgart 1970.

Valiant,L.G.: The Complexity of Combinatorial Computations: An Introduction, in: Giloi,W.K. (Hrsg.): GI - 8. Jahrestagung, Berlin - Heidelberg - New York 1978, S. 326-337.

van Leeuwen,J.: A Partial Solution to the Reachability Problem For Vector-Addition Systems, in: Conference Record: Papers presented on the 6th Annual ACM Symposium on Theory of Computing, 1974 in Seattle, New York 1974, S. 303-309.

von zur Gathen,J. u. M. Sieveking: Weitere zum Erfüllungsproblem polynomial äquivalente kombinatorische Aufgaben, in: Specker,E. u. V. Strassen (Hrsg.): Komplexität von Entscheidungsproblemen, Lecture Notes in Computer Science 43, Berlin - Heidelberg - New York 1976, S. 49-71.

Wang,H.: A Variant to Turing's Theory of Computing Machines, in: Journal of the Association for Computing Machinery (ACM), Vol. 4 (1957), S. 63-92.

Weber,H.H.: Khachiyan's Algorithmus, in: Zeitschrift für Operations Research, Bd. 26 (1982), S. B/229-B/240.

Wehr,G.: Das M,N-Job-Shop-Scheduling Problem: Eine Branch- and Bound-Methode mit verbesserten unteren Schranken, neuen Verzweigungsstrategien und einem effektiven Einsatz heuristischer Lösungsmethoden, Dissertation an der Technischen Universität München 1980, München 1980.

Weizenbaum,J.: Die Macht der Computer und die Ohnmacht der Vernunft, 3. Aufl., Frankfurt 1982.

Yamada,H.: Real-Time Computation and Recursive Functions Not Real-Time Computable, in: IRE Transactions on Electronic Computers, Vol. EC-11 (1962), S. 753-760.

Zelewski,S.: Das Leistungspotential der Künstlichen Intelligenz - eine informationstechnisch-betriebswirtschaftliche Analyse, Dissertation am Seminar für Allgemeine Betriebswirtschaftslehre und Fertigungswirtschaft der Universität Köln 1985, Witterschlick/Bonn 1986.

Zervos,C.R. u. K.B. Irani: Colored Petri Nets: Their Properties and Applications, Dissertation und Technical Report No. RADC-TR-77-246 am Department of Electrical Engineering, University of Michigan, Ann Arbor 1977.

Verzeichnis der Arbeitsberichte des Seminars für Allgemeine Betriebswirtschaftslehre, Industriebetriebslehre und Produktionswirtschaft der Universität zu Köln (bis Sommer 1986: Seminar für Allgemeine Betriebswirtschaftslehre und Fertigungswirtschaft).

Nr. 1: ZELEWSKI, STEPHAN: Entscheidungsmodelle zur Verschrottung von Fertigungshilfsmitteln, Köln 1984.

Nr. 2: KERN, WERNER; ZELEWSKI, STEPHAN: Ein Zuordnungsmodell für Meßgeräte in Energie-Informations-Systemen, Köln 1985.

Nr. 3: KERN, WERNER; PETERS, ULRICH: Energiebewirtschaftung in industriellen Betrieben - Bericht über eine Befragung, Köln 1985.

Nr. 4: BOOS, JOCHEN: Lokalisierung von Meßstellen für ein Informations-System zur Energiebewirtschaftung in industriellen Betrieben - Entwicklung eines OR-Modells mit einem Lösungsvorschlag -, Köln 1986.

Nr. 5: ZELEWSKI, STEPHAN: Ansätze der Künstlichen Intelligenz-Forschung zur Unterstützung der Netzplantechnik, Köln 1986.

Nr. 6: ZELEWSKI, STEPHAN: Schnittstellen bei betrieblichen Informationssystemen - eine Darstellung aus systemtheoretischer und betriebswirtschaftlicher Sicht -, Köln 1986.

Nr. 7: ZELEWSKI, STEPHAN: Konzepte für Frühwarnsysteme und Möglichkeiten zu ihrer Fortentwicklung durch Beiträge der Künstlichen Intelligenz, Köln 1986.

Nr. 8: ZELEWSKI, STEPHAN: Das Konzept der unscharfen Mengen unter besonderer Berücksichtigung ihrer linguistischen Interpretation - eine Lösung für unscharfe Probleme? -, Köln 1986.

Nr. 9: ZELEWSKI, STEPHAN: Der tau-Wert: Aspekte eines neueren spieltheoretischen Ansatzes zur fairen Preisbildung aus kostenrechnerischer Perspektive, Köln 1986.

Nr. 10: ZELEWSKI, STEPHAN: Competitive Bidding aus der Sicht des Ausschreibers - ein spieltheoretischer Ansatz -, Köln 1986.

Nr. 11: ZELEWSKI, STEPHAN: Netztheoretische Ansätze zur Konstruktion und Auswertung von logisch fundierten Problembeschreibungen, Köln 1986.

Nr. 12: ZELEWSKI, STEPHAN: Netztheoretische Fundierung von parallelen Algorithmen für die Lösung linear-ganzzahliger OR-Modelle, Köln 1986.

Nr. 13: ZELEWSKI, STEPHAN: Intelligente Informationsbanksysteme - benutzerfreundliche Instrumente für die Informationsvermittlung? -, Köln 1986.

Nr. 14: ZELEWSKI, STEPHAN: Komplexitätstheorie - ihr Beitrag zur Klassifizierung und Beurteilung von Problemen des Operations Research -, Köln 1986.

Nr. 15: ZELEWSKI, STEPHAN: Der Informationsbroker, Köln 1986.

Nr. 16: ZELEWSKI, STEPHAN: Soziale Verantwortbarkeit von Technologien, Köln 1986.

Nr. 17: ZELEWSKI, STEPHAN: Expertensysteme - Übersicht über Konzeptionen und betriebswirtschaftliche Anwendungsmöglichkeiten -, Köln 1986.

Nr. 18: ZELEWSKI, STEPHAN: Das Leistungspotential der Künstlichen Intelligenz für Industrieanwendungen - Ein Überblick -, Köln 1987.

Nr. 19: ZELEWSKI, STEPHAN: Expertensysteme im "Büro der Zukunft" - Ein Überblick über Anwendungsperspektiven und Bewertungsaspekte -, Köln 1987.

Nr. 20: KUMMER, SEBASTIAN: Computerunterstützung schöpferischer Forschungs- und Entwicklungsaktivitäten, Köln 1987.

Nr. 21: ZELEWSKI, STEPHAN: Betriebswirtschaftliche Aspekte des industriellen Einsatzes von Expertensystemen - Anwendungsmöglichkeiten und Bewertung -, Köln 1988.

Nr. 22: ZELEWSKI, STEPHAN: Expertensysteme für Prozeßplanung und -steuerung in der Fabrik der Zukunft - Ein Überblick über Konzepte und erste Prototypen -, Köln 1988.

Nr. 23: ZELEWSKI, STEPHAN: Expertensysteme zur Sicherung der Betriebsbereitschaft in der Fabrik der Zukunft, Köln 1988.