

Probabilistische Analyse von Optimierungs-Algorithmen

Karl Heinz Borgwardt

Universität Augsburg

1. Der Zweck

Optimierungsalgorithmen werden dazu konzipiert, um bestimmte Optimierungsprobleme (oder Klassen von solchen Problemen) effizient und zuverlässig zu lösen. In den meisten Fällen stehen zu diesem Zweck mehrere Algorithmen zur Auswahl. Wir wollen hier diskutieren, welche Gesichtspunkte für diese Auswahl maßgebend sein könnten. Insbesondere wollen wir auf solche Kriterien eingehen, die mit Hilfe von probabilistischer Analyse von Algorithmen gewonnen werden können.

Wir wollen hier nicht auf Komplexitätstheoretische Grundlagen eingehen und nehmen zur Vereinfachung an, daß unser Optimierungsproblem sich als Frage nach einem bestimmten Optimalwert oder Optimalvektor formulieren läßt.

Sei also A ein Algorithmus, mit dem sich die Probleme P einer bestimmten Problemklasse \mathcal{P} behandeln lassen. Dann stehen folgende Kriterien für die Güte von A im Vordergrund.

- Produziert der Algorithmus bei Eingabe von $P \in \mathcal{P}$ die gewünschte theoretische Lösung bzw. wie groß sind die Abweichungen?
- Wie viel Rechenzeit wird zur Produktion des Outputs benötigt?
- Wie viel Speicherplatz wird beansprucht?

Wir wollen uns auf diese Fragen konzentrieren, obgleich es viele weitere interessante und wertvolle Kriterien gibt.

Natürlich wird A nicht alle \mathcal{P} -Probleme gleich effizient behandeln können. So erwarten wir, daß "größere" Probleme schwieriger zu lösen sind als "kleine". Als Maß für die "Größe" eines Problems betrachten wir die Länge des Inputs, also die Anzahl der Bits, die benötigt werden, um dem Computer das spezielle Problem P zu beschreiben.

$L(P)$ sei diese Inputlänge und \mathcal{P}_n sei die Menge $\{P \in \mathcal{P} \mid L(P) = n\}$ für $n \in \mathbb{N}$.

Auf diese Weise werden die Probleme nach ihrer Inputlänge klassifiziert. (Oft verwendet man auch statt der Inputlänge problemtypische Dimensionsparameter, wie die Knotenzahl in einem Graphen.)

Eine sehr vorsichtige Vorgehensweise, die Güte von A zu messen, beruht auf der Analyse des schlechtestmöglichen Falls.

Wir bezeichnen mit $d_A(P)$ die Abweichung des Outputs von der gewünschten Lösung, mit $t_A(P)$ die benötigte Zeit, mit $s_A(P)$ den benötigten Speicherplatz bei Anwendung von A auf P .

Durch Supremumsbildung über die Teilklassen \mathcal{P}_n erhalten wir dann Funktionen von \mathbb{N} nach \mathbb{R} .

$$\bar{d}_A(n) = \text{Sup}\{d_A(P) \mid P \in \mathcal{P}_n\}$$

$$\bar{t}_A(n) = \text{Sup}\{t_A(P) \mid P \in \mathcal{P}_n\}$$

$$\bar{s}_A(n) = \text{Sup}\{s_A(P) \mid P \in \mathcal{P}_n\}.$$

Eine sehr interessante Frage betrifft die Geschwindigkeit, mit der diese Funktionen mit n wachsen. Insbesondere interessiert, ob es polynomiale Funktionen g gibt, mit denen obige Funktionen \bar{t} , \bar{s} nach oben abgeschätzt werden können. In diesem Fall spricht man von "polynomialen" Algorithmen. Dies ist ein Kriterium, das vor allem das Verhalten bei großen n beschreibt.

Die Worst-Case-Analyse zeichnet ein sehr pessimistisches Bild eines Algorithmus. Immerhin kann es sein, daß "schlechte" Elemente einer Problemklasse extreme Ausnahmefälle sind. In diesem Falle würde der Algorithmus zu schlecht dargestellt.

Die Praxis verlangt andere Kriterien. Man ist vorwiegend erfolgsorientiert, und strebt nicht nach größtmöglicher Allgemeinheit. Damit ist gemeint, daß es meist um die spezielle Lösung von P geht. Ob ein anderes Element $P' \in \mathcal{P}$ ebenfalls effizient gelöst werden könnte, ist meist uninteressant.

An verschiedenen Beispielen (etwa Lineare Optimierung) hat sich eine Kluft zwischen Beurteilung nach dem empirischen Verhalten eines Algorithmus durch die Praktiker (sehr schnell) und Beurteilung nach dem Worst-Case-Verhalten durch die Mathematiker (sehr langsam) entwickelt.

Um diese Kluft zu schließen, untersucht man das durchschnittliche Verhalten. Dazu ist es erforderlich, plausible bzw. realistische Annahmen über die Verteilung der auftretenden Probleme zu treffen. Basierend auf diesen Verteilungen können dann Erwartungswerte, Wahrscheinlichkeiten für bestimmte Laufzeiten oder Abweichungen ermittelt werden.

Weder die Einführung realistischer Modelle noch die erforderliche Auswertung sind leicht.

Im folgenden Abschnitt soll an einigen Beispielen aufgezeigt werden, wie Algorithmen probabilistisch analysiert, welche Ergebnisse dabei gewonnen und welche stochastischen Annahmen dazu verwendet wurden.

2. Beispiele und Ergebnisse

2.1 Lineare Optimierung

Lineare Optimierungsprobleme

$$\begin{array}{ll} \text{Maximiere} & v^T x \\ \text{unter} & a_1^T x \leq b^1, \dots, a_m^T x \leq b^m \\ \text{wobei} & x, v, a_1, \dots, a_m \in \mathbb{R}^n; b^1, \dots, b^m \in \mathbb{R}, m \geq n \end{array}$$

werden zumeist mit der Simplexmethode von G. B. Dantzig gelöst. Dieses Verfahren läuft in zwei analogen Phasen ab, wobei in jeder Phase ausgehend von einer Ecke des Zulässigkeitsbereichs eine Eckenfolge x_0, \dots, x_s konstruiert wird, so daß x_i und x_{i+1} jeweils benachbart sind und $v^T x_{i+1} > v^T x_i$ gilt (für $i = 0, \dots, s-1$).

Entscheidend für den Rechenaufwand ist nun die Anzahl der Eckenaustausch-Schritte s . Man klassifiziert am besten nach den Dimensionsparametern m und n und unterstellt eine Verteilung über die (m, n) -Probleme.

Borgwardt [3,4] untersucht eine parametrische Variante des Simplexalgorithmus, den sogenannten Schatteneckenalgorithmus. Bei diesem Algorithmus läßt sich die Frage nach dem Durchschnittswert von s auf ein Problem der stochastischen Geometrie zurückführen.

Sein stochastisches Modell (RSM) fordert, daß a_1, \dots, a_m, v identisch, unabhängig und rotationssymmetrisch verteilte Zufallsvektoren auf $\mathbb{R}^n \setminus \{0\}$ sind.

Er entwickelt u. a. folgende Resultate über die durchschnittliche Schrittzahl $E_{m,n}(s)$:

Satz. Bei $m \rightarrow \infty$ und festem n wächst $E_{m,n}(s)$ bei Unterstellung von zulässigen Verteilungen gemäß RSM

- | | |
|--|--------------------------------|
| 1) Gaußverteilung | $\sqrt{\ln m} n^{\frac{3}{2}}$ |
| 2) Gleichverteilung auf der Einheitskugel | $m^{\frac{1}{n+1}} n^2$ |
| 3) Gleichverteilung auf der Einheitssphäre | $m^{\frac{1}{n-1}} n^2$ |
| 4) Falls die Verteilung beschränkten Träger hat, gilt asymptotisch | |

$$E_{m,n}(s) \leq m^{1/n-1} n^2 C \text{ und } E_{m,n}(s) \rightarrow \infty.$$

- 5) Es gibt zulässige Verteilungen, so daß $E_{m,n}(s) \leq C(n)$ (bleibt beschränkt).

Die Unterschiede im Verhalten abhängig von der Verteilung beruhen auf unterschiedlichen Redundanzraten (ein gewisser Anteil der Restriktionen in der Aufgabenstellung ist für die Gestaltung des Zulässigkeitsbereichs überflüssig). In diesem Rotationssymmetrie-Modell sind Redundanzraten von 0 bis fast zu 1 realisierbar.

Der nichtasymptotische Fall konnte ebenfalls behandelt werden.

Satz. Für alle Paare (m, n) mit $m \geq n$ und für alle Verteilungen gemäß RSM gilt

- 1) $E_{m,n}(s) \leq O(m^{\frac{1}{n-1}} n^3)$
- 2) $E_{m,n}$ (Gesamtstrittzahl für Phase I + Phase II) = $O(m^{\frac{1}{n-1}} n^4)$.

Ein anderes stochastisches Modell, das Sign-Invariance-Modell, wird von verschiedenen Autoren [1,2,7,11] zugrundegelegt. Hierbei betrachtet man die m Restriktionen

$$a_1^T x \leq b^1, \dots, a_m^T x \leq b^m$$

und führt m unabhängige Bernoulliexperimente mit Ausgang 0 oder 1 und Wahrscheinlichkeiten $P(0) = \frac{1}{2} = P(1)$ aus. Ergibt sich im i -ten Experiment der Wert 1, dann klappt man die i -te Restriktion um, d. h. man ersetzt sie durch $a_i^T x \geq b^i$. Beim Ausgang 0 wird die Restriktion nicht verändert. Somit erhält man 2^m verschiedene Probleme, die alle gleiche Wahrscheinlichkeit besitzen.

Man untersucht eine parametrische Variante, die mit der Borgwardt'schen fast identisch ist. Mit einfachen kombinatorischen Methoden kommt man zu überraschend niedrigen Wachstumsraten in der Dimension. Dabei wird über die 2^m Probleme gemittelt.

Satz. Unter dem Sign-Invariance-Modell gilt

- a) $E_{m,n}(s) \leq n \frac{m-n+2}{m+1}$
- b) $E_{m,n}$ (Gesamtzahl für Phase I + Phase II) = $O(n^2)$.

Auf den ersten Blick überrascht die niedrige Ordnung. Aber man muß berücksichtigen, daß für $m \gg n$

- ein großer Teil der erzeugten Probleme einen leeren Zulässigkeitsbereich besitzt,
- die Redundanzrate sehr hoch ist,
- die mittlere Anzahl von Ecken pro Problem sehr niedrig ist.

All diese Einflüsse verringern die Zahl von Iterationsschritten, egal welcher Algorithmus angewendet wird. Somit beruhen die "guten" Ergebnisse auf der Auswahl des Modells und nicht auf der Güte des Simplexverfahrens. Dies belegt eine erstaunliche Sensitivität gegenüber der Wahl des stochastischen Modells.

2.2 Rucksackprobleme

Eines der bekanntesten Probleme der ganzzahligen Optimierung ist das Rucksackproblem.

Gegeben seien n Gegenstände $j = 1, \dots, n$ mit Gewichten $a_j \in \mathbb{R}$ und jeweiligem Wert (Nutzen) $c_j \in \mathbb{R}$ sowie eine Obergrenze $b \in \mathbb{R}$. Bestimme $x = (x_1, \dots, x_n)$ mit $x_j \in \{0, 1\}$ so, daß unter der Nebenbedingung $\sum_{j=1}^n a_j x_j \leq b$ der Ausdruck $\sum_{j=1}^n c_j x_j$ maximiert wird.

Lifschitz [9] entwickelt eine enumerative Lösungsmethode für dieses Problem. Er nutzt dabei aus, daß in einer Optimallösung niemals $x_i > x_j$ gelten kann wenn ($a_i > a_j$ und $c_i < c_j$) gilt. Somit scheiden viele der 2^n möglichen Kandidaten x aus.

Falls die a_j und die c_j identisch verteilte, nichtnegative unabhängige Zufallsvariablen mit Dichtefunktion sind, dann kann Lifschitz beweisen, daß im Durchschnitt nur $O(e^{2\sqrt{n}})$ Kandidaten untersucht werden müssen.

Tinhofer [10] gibt eine Analyse eines einfachen Algorithmus zur Lösung des verwandten **Subset-Sum-Problems** an.

Hier gilt jeweils $a_j = c_j$ und die a_j sind gleichverteilte, unabhängige Variablen über $\{1, 2, \dots, d(n)\}$, wobei $d(n) \in \mathbb{N}$. Die rechte Seite b sei unabhängig von den a_j und gleichverteilt über $\{1, \dots, nd(n)\}$.

Sein Algorithmus durchläuft die a_j ($i = 1, \dots, n$) und addiert sie auf. Wenn die laufende Summe bei Hinzunahme von a_j größer als b würde, streicht man a_j und setzt $x_j = 0$. Andernfalls wird a_j hinzuaddiert und $x_j = 1$ gesetzt. Nach Durchlaufen von $j = 1, \dots, n$ hat man das Ergebnis.

Tinhofer deutet den Ablauf dieses Algorithmus als Markoff-Prozeß und untersucht die entstehende Lücke zwischen b und dem Summenwert. Damit gewinnt er auch eine Abschätzung für die Wahrscheinlichkeit, das Optimum zu erzielen.

Satz. Unter der Bedingung, daß $\sum_{j=1}^n a_j \geq b$, gilt

$$P(\text{Lücke wird } 0) = 1 - O\left(\frac{1}{n}\right).$$

2.3 Scheduling-Probleme

Für die Produktionsplanung bedeutsam sind Maschinenbelegungspläne (sog. Scheduling-Probleme).

n Arbeiten J_1, \dots, J_n müssen von einer Maschine hintereinander (ohne Unterbrechungen) bearbeitet werden. J_i trifft zum Zeitpunkt r_i ein und benötigt p_i Zeiteinheiten zur Bearbeitung.

Wir versuchen durch optimale Planung der Startzeiten für die Jobs die Summe der Ausführungszeiten (Zeitabstand zwischen Eintreffen und Vollendung des Jobs) zu minimieren.

Für dieses Problem führt Gazmuri [6] eine Analyse durch. Seine Annahmen sind:

- i) Die Zeiten p_i sind identisch verteilte, unabhängige, ganzzahlige Variablen, die durch M nach oben beschränkt sind.
- ii) Die Abstände zwischen den Ankunftszeiten $t_i = r_{i+1} - r_i$ sind identisch und unabhängig verteilte ganzzahlige Variable, deren Momentenerzeugende Funktion irgendwo endlich wird.
- iii) Die Variablen p_1, \dots, p_n und t_1, \dots, t_n sind kollektiv unabhängig.
- iv) $E(p) < E(t)$

Sein Algorithmus zerlegt die Menge der Jobs in kleine Teilmengen, die man enumerativ behandeln kann (z. B. $\lfloor \ln n \rfloor$ Jobs). Diese unabhängige Lösung wird bis zu einem vorgegebenen Punkt übernommen. Die restlichen Jobs werden nach der Regel "kürzester Job zuerst" bearbeitet und soweit erforderlich, nach hinten verschoben.

Satz. Der Algorithmus liefert mit Wahrscheinlichkeit $1 - O[(\ln n)^2 \frac{1}{n}]$ einen relativen Fehler von höchstens $O[(\ln \ln n)^\alpha]$ mit $\alpha \in (-1, 0)$.

2.4 Zuordnungsprobleme

Beim quadratischen Zuordnungsproblem sind zwei $n \times n$ -Matrizen (c_{ij}) und (d_{ij}) gegeben. Die Aufgabe besteht darin, unter allen Permutationen δ von $\{1, \dots, n\}$ diejenige herauszufinden, die

$$Q(\delta, n) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} d_{\delta(i)\delta(j)} \text{ minimal macht.}$$

Wir nennen die minimierende Permutation δ_* , die maximierende δ^* .

Burkard und Fincke haben folgendes Modell untersucht. Die Werte d_{ij} seien bestimmt durch den Abstand zwischen X_i und X_j , wobei die X_i unabhängig und gleichverteilte Punkte über dem Einheitsquadrat sind. Die c_{ij} seien identisch und unabhängig verteilt über $[0, 1]$ mit $E(c_{ij}) = c > 0$.

Der relative Abstand zwischen bestem und schlechtestem Wert kann folgendermaßen abgeschätzt werden

Satz.

$$P \left\{ \left(\frac{Q(\delta^*, n) - Q(\delta_*, n)}{Q(\delta_*, n)} \right) \leq \frac{3 + \alpha}{c} n^{-0.18} \right\} \rightarrow 1 \text{ bei } n \rightarrow \infty, \text{ für alle } \alpha > 0.$$

Zum Beweis wird das Quadrat unterteilt in $m(n)$ kongruente Teilquadrate. Grenzwertsätze der Wahrscheinlichkeitstheorie zeigen, daß mit hoher Wahrscheinlichkeit

in jedem dieser Teilquadrate ungefähr $\frac{n}{m(n)}$ Punkte zu liegen kommen. Die Wahrscheinlichkeiten für Abweichungen können abgeschätzt werden. Ansonsten werden die Punkte ersetzt durch die jeweiligen Mittelpunkte ihrer Teilquadrate. Nun wird für die vereinfachte Situation eine Optimallösung bestimmt und der dabei in Kauf genommene Fehler abgeschätzt.

2.5 Travelling Salesman-Probleme

Hierbei geht es darum, in einem gewichteten Graphen einen Kreis durch alle Knoten mit minimalem Gewicht zu finden.

Im Euklidischen Fall denkt man sich die Knoten als Punkte in der Ebene und die Gewichte der Kanten als die jeweiligen Abstände.

Karp und Steele [8] geben als stochastisches Modell folgendes vor: v_1, \dots, v_n seien zufällige Punkte, die unabhängig und gleichverteilt sind über dem Einheitsquadrat.

Ihr Algorithmus beruht wiederum auf einer Zerlegung in Teilquadrate. Innerhalb der Teilquadrate werden optimale Touren bestimmt (mit einer exakten Methode). Betrachte die Optimaltour durch die $m(n)$ Mittelpunkte der Teilquadrate (einfach zu bestimmen). Verbinde die Teiltouren entsprechend eben erwähnter Optimaltour und streiche überflüssige Kanten.

Die Autoren kommen zu folgenden Ergebnissen

Satz. Falls $m(n)$ sich verhält wie $\frac{n}{\delta(n)}$ bei einer unbeschränkt wachsenden Funktion $\delta(n)$, dann gilt

$$\sum_{n=1}^{\infty} P(\text{relativer Fehler} \geq +\varepsilon) < \infty, \text{ für alle } \varepsilon > 0.$$

Also approximiert die Heuristik die Optimallösung fast überall. Der Zeitbedarf für diese Heuristik kann ebenfalls abgeschätzt werden.

Satz. Für die spezielle Wahl $\delta = \lfloor \log n \rfloor$ ist die erwartete Rechenzeit $O(n^2 \log n)$.

3. Offene Fragen und Schwierigkeiten

Es fällt auf, daß die meisten der genannten Ergebnisse asymptotische Aussagen sind. Dies hat unter anderem folgende Gründe:

Viele Heuristiken arbeiten auf der Basis von einfachen und wenigen Repräsentanten, für die das entsprechende Problem einfach zu lösen wäre. Bei großer Zahl n garantieren bestimmte Grenzwertsätze der Wahrscheinlichkeitstheorie ein "einigermaßen reguläres" Verhalten der Mehrzahl der Zufallsvariablen. In diesem Fall können die Zufallsvariablen mit ihren Repräsentanten identifiziert werden, ohne daß der Fehler zu groß wird.

Außerdem treten die gewünschten Erwartungswerte oder Wahrscheinlichkeiten zunächst nach der Beschreibung der Wirkungsweise des Algorithmus als komplizierte Reihen oder Integrale auf. Ihre Auswertung ist oft nur im asymptotischen Fall möglich.

Natürlich ist diese Situation nicht befriedigend. Denn meist liegen in der Praxis Probleme kleiner oder mittlerer Größenordnung vor. Für diese sind die asymptotischen Resultate kaum anwendbar. Verschlimmert wird die Situation noch, wenn zudem die entsprechende Aussage nur mit gegen 1 konvergierender Wahrscheinlichkeit nachweisbar ist. Wünschenswert wären Beschreibungen und Ergebnisse auch für den Fall moderater Dimensionen, wie sie bei Linearer Optimierung erzielt wurden.

Sorge muß außerdem die hohe Sensitivität gegenüber der Modellwahl bereiten. Niemand weiß, wie die Probleme in der Realität tatsächlich verteilt sind. Uneinigkeit in der Frage nach der Verteilung kann deshalb leicht zu völlig unterschiedlichen Meinungen über Algorithmen führen. Es liegt auf der Hand, daß Mathematiker einfache, elegante und leicht handhabbare Modelle bevorzugen, weil sich nur diese als auswertbar erweisen. Reale Strukturen (wie z. B. Steckernetze) zeigen aber oft sehr wenig Ähnlichkeit mit den idealistischen Modellen.

Auf eine weitere Gefahr im Zusammenhang mit der Modellwahl sei ebenfalls hingewiesen. Man muß sich darüber klar sein, was man eigentlich will. Will man, weil man ein Modell für angemessen hält, einen Algorithmus unter diesem Modell analysieren und seine Effizienz feststellen? Oder will man einen Eindruck, den man empirisch von einem Algorithmus hat, in der Analyse bestätigt haben? Dies würde unter Umständen implizieren, daß man Modelle nach der Analyse so lange verwirft, bis die gewünschte Effizienz sichtbar wird.

Ein weiterer Nachteil liegt in der Kompliziertheit der Analyse. Während worst-case Abschätzungen noch relativ einfach durch Verkopplung ungünstiger Fälle zu gewinnen sind, ist es hier nötig, den Verlauf des Algorithmus sehr einfach zu charakterisieren. Dies bedeutet, daß meist nur sehr einfach strukturierte Algorithmen stochastisch analysierbar sind. Die wirklich effizienten und hochgradig ausgetüftelten Algorithmen entziehen sich der Analyse (noch). Dies ist auch der Fall bei linearer Optimierung, wo nur parametrische Varianten untersucht werden konnten. Es gibt aber weit bessere Varianten.

In jedem Falle darf man die Erwartungen an die probabilistische Analyse nicht zu hoch ansetzen. Sie sollte, ja sie muß sogar (siehe Abschnitt 1) in die Überlegungen eingeschlossen werden, aber sie kann (wenigstens heute noch) nicht als das entscheidende Gütekriterium dienen. Vielleicht liegt der Nutzen von probabilistischer Analyse von Algorithmen gar nicht einmal so sehr in den genauen Größenordnungen, sondern vielmehr im Erkennen eines "normalen Verhaltens" dieses Algorithmus, was zum Verständnis wesentlich beitragen kann.

4. Literatur

- [1] I. Adler, R. M. Karp, R. Shamir [1983]: "A simplex variant solving an $m \times d$ linear program in $O(\min(m^2, d^2))$ expected number of pivot steps", Computer Science Division, University of California, Berkeley 1983.
- [2] I. Adler, N. Megiddo [1983]: "A simplex algorithm whose average number of steps is bounded between two quadratic functions of the smaller dimension", Department of

Industrial Engineering and Operations Research, University of California, Berkeley 1983.

- [3] K. H. Borgwardt [1982 a]: “Some distribution-independent results about the asymptotic order of the average number of pivot steps of the simplex method”, *Mathematics of Operations Research* 7 (1982), 441–462.
- [4] K. H. Borgwardt [1982 b]: “The average number of pivot steps required by the simplex method is polynomial”, *Zeitschrift für Operations Research* 26 (1982), 157–177.
- [5] R. E. Burkard, U. Fincke [1983]: “Probabilistic asymptotic properties of quadratic assignment problems”, *Zeitschrift für Operations Research* 27 (1983), 73–81.
- [6] P. G. Gazmuri [1985]: “Probabilistic analysis of a machine scheduling problem”, *Mathematics of Operations Research*, 10 (1985), 328–339.
- [7] Haimovich, M. [1983]: “The simplex algorithm is very good! — On the expected number of pivot steps and related properties of random linear programs”, 415 Uris Hall, Columbia University, New York, April 1983.
- [8] R. M. Karp, J. M. Steele [1985]: “Probabilistic analysis of heuristics”, in: E. L. Lawler, J. K. Lenstra, A. H. G. Rinnoy Kan, D. B. Shmoys (eds.), *The Traveling Salesman Problem*, Wiley, Chichester (1985), Ch. 6..
- [9] V. Lifschitz [1980]: “The efficiency of an algorithm of integer programming: A probabilistic analysis”, *Proceedings of the American Mathematical Society*, Volume 79, Number 1 (1980) 72–76.
- [10] G. Tinhofer [1984]: “Probabilistische Ansätze in der diskreten Optimierung”, Bericht Nr. 230 der Mathematisch-Statistischen Sektion in der Forschungsgesellschaft Joanneum (1984).
- [11] M. J. Todd [1983]: “Polynomial expected behaviour of a pivoting algorithm for linear complementarity and linear programming problems”, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York 1983.