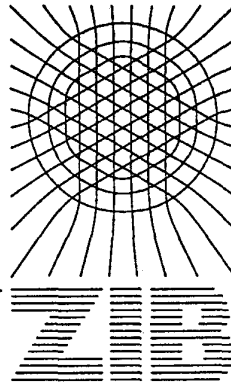

Konrad-Zuse-Zentrum für Informationstechnik Berlin
Heilbronner Straße 10, D-1000 Berlin 31



Robert E. Bixby*

Das Implementieren des Simplex-Verfahrens: Die Startbasis

* Rice University

Department of Mathematical Science
Houston, Texas

This paper was written while the author visited the Konrad-Zuse-Zentrum für Informationstechnik Berlin

Preprint SC 92-11 (April 1992)

Das Implementieren des Simplex-Verfahrens: Die Startbasis

Robert E. Bixby¹
Rice University

Kurzbeschreibung

Dieses Paper enthält die ersten beiden Teile einer geplanten Serie von Aufsätzen über die CPLEX²-Implementierung des Simplex-Verfahrens. Der erste Teil ist eine Einführung: er liefert eine kurze Beschreibung des Verfahrens für Probleme mit beschränkten Variablen, zusammen mit einer relativ ausführlichen Diskussion der numerischen Eigenschaften der Netlib-Probleme. Diese Probleme bilden auch das Fundament der rechnerischen Untersuchungen in den folgenden Teilen. Der zweite Teil enthält die Hauptergebnisse dieses Papers, eine Beschreibung der Methode, die von CPLEX verwendet wird, um eine Startbasis zu konstruieren.

Teil I: Einführung

1 Einführung

Vor zehn Jahren wurden die rechnerischen Aspekte der linearen Programmierung für ein fast vollständig entwickeltes Teilgebiet der numerischen Optimierung gehalten. Man glaubte, Dantzig's Simplex-Verfahren [3] sehr gut verstanden zu haben, und es gab davon mehrere leistungsfähige Implementierungen. Es ist wahrscheinlich so, daß die Mehrheit der Kenner dieses Gebiets keine weiteren grundlegenden Verbesserungen erwartete. Diese Erwartung ist nun widerlegt worden. Die erstaunlichen Weiterentwicklungen in der Computertechnik der letzten 10 Jahre haben es ermöglicht, sowohl etliche neue als auch einige alte Strategien neu zu implementieren und auch dazu geführt, daß wesentlich größere Probleme untersucht werden können. Diese Entwicklungen wurden auch noch durch Forderungen aus anderen Bereichen der Optimierung vorangetrieben, insbesondere der ganzzahligen Optimierung, wo die Fähigkeit, LP-Probleme schnell und zuverlässig lösen zu können, unentbehrlich ist. Darüberhinaus ist ausgehend von der Arbeit von Karmarkar [6] eine völlig neue Klasse von Methoden entwickelt worden, die sogenannten Innere-Punkte-Verfahren, die sich als Konkurrenz zum Simplex-Verfahren erwiesen hat. Obwohl Innere-Punkte-Verfahren in diesem

¹Die Vorbereitung dieses Aufsatzes wurde durch Grants der NSF (CCR-8815914) und der AFOSR (AFOSR-90-0273) unterstützt.

²CPLEX is a trademark of CPLEX Optimization, Inc.

Paper nicht behandelt werden, sind die neuesten Rechenergebnisse ebenso beeindruckend wie die des Simplex-Verfahrens ([1], [9]).

Dieses Paper ist das erste einer Serie von Aufsätzen, die verschiedene Aspekte der CPLEX-Implementierung des Simplex-Verfahrens beschreiben sollen, deren Hauptteile vom Author geschrieben wurden. Zwei weitere, neue und hervorragende Implementierungen des Simplex-Verfahrens sind die OSL Implementierung (Optimization System Library) von John Forrest [5] und MINOS 5.3 von Michael Saunders [11], die eine wesentliche Verbesserung von früheren MINOS-Versionen ist.

Der Hauptzweck dieses Papers und das Thema von Teil II ist die Beschreibung der Methode, die von CPLEX benutzt wird, um eine gute Startbasis zu konstruieren. Teil I enthält einige Grundlagen für diese Beschreibung: die Definition einer Basis, das Simplex-Verfahren für LP-Probleme mit beschränkten Variablen, und eine Untersuchung der numerischen Eigenschaften der Netlib-Probleme. Die Leser, die schon mit der Numerik der linearen Programmierung vertraut sind, werden es vielleicht vorziehen, sofort mit Teil II anzufangen.

In den folgenden Teilen dieser Serie werden andere Aspekte von CPLEX behandelt werden, unter anderem die Anwendung des Composite-Simplex-Verfahrens von Philip Wolfe in der Zulässigkeitsphase (Phase I), eine laufzeitparende Version des partiellen Pricings, ein einfaches Schranken-Perturbations-Verfahren zur Behandlung des Stallings, Vektorisierung und andere Themen.

2 Definition einer Basis

Ein Lineares Programm (LP) ist ein Optimierungsproblem der folgenden Form:

$$\begin{aligned} \min \quad & c^T x \\ & Ax = b \\ & l \leq x \leq u \end{aligned} \tag{1}$$

wobei c, A, b, l und u vorgegebene Matrizen der Dimensionen $n \times 1$, $m \times n$, $m \times 1$, $n \times 1$, und $n \times 1$ sind, und x ein $n \times 1$ Variablenvektor ist. Wir nehmen an, daß $\text{rang}(A) = m$ gilt. Falls nicht, ist dieser Mangel durch Hinzufügen von künstlichen Variablen leicht zu korrigieren.

Die lineare Funktion $c^T x$ heißt Zielfunktion, A Matrix der Nebenbedingungen oder Constraint-Matrix, b rechte Seite, l Vektor der unteren Schranken, und u Vektor der oberen Schranken. Wir setzen die obere (untere) Schranke einer Variable auf $+\infty$ ($-\infty$) falls diese Variable nach oben (unten) unbeschränkt ist. Es ist zu erwarten, daß $l_j = 0$ und $u_j = +\infty$ für die meisten Variablen gilt. Solche Variablen werden als nichtnegativ bezeichnet. Falls $l_j = -\infty$ und $u_j = +\infty$ gilt, wird x_j als frei bezeichnet, und falls $l_j = u_j$ gilt, wird x_j als fixiert bezeichnet.

Sei S eine Teilmenge von Spaltenindizes von A , so bezeichnet A_S "die" $m \times |S|$ Teilmatrix, die die entsprechenden Spalten von A enthält. Falls S eine geordnete Menge ist, gehen wir davon aus, daß die Spalten in der entsprechenden Reihenfolge

vorkommen. Sei d ein Vektor und S eine Teilmenge von Zeilenindizes, so bezeichnet d_S den entsprechenden Teilvektor von d . Falls S geordnet ist, gehen wir davon aus, daß die Einträge von d_S in der entsprechenden Reihenfolge vorkommen.

Definition. Eine *Basis* ist ein Tripel (B, N_l, N_u) mit den folgenden Eigenschaften:

(B1) $B = (B_1, \dots, B_m)$ ist eine geordnete Teilmenge der Spaltenindexmenge $\{1, \dots, n\}$ mit $\mathbf{B} = A_B$ nichtsingulär. B heißt *Basisindexmenge* und \mathbf{B} *Basismatrix*. Die Variablen $x_j (j \in B)$ heißen *Basisvariablen*. Die übrigen Variablen heißen *Nicht-Basisvariablen*. $N = \{j \in \{1, \dots, n\} : j \notin B\}$ bezeichnet die Menge der Indizes der Nicht-Basisvariablen.

(B2) Es gilt $N_l \cap N_u = \emptyset$, $N_l \cup N_u = \{j \notin B : x_j \text{ ist weder fixiert noch frei}\}$, $l_j > -\infty$ für alle $j \in N_l$, und $u_j < +\infty$ für alle $j \in N_u$.

Wir setzen $N_{fr} = \{j \in N : x_j \text{ ist frei}\}$ und $N_{fx} = \{j \in N : x_j \text{ ist fixiert}\}$. \square

Das Tripel (B, N_l, N_u) wird manchmal in abgekürzter Form als "die Basis B " bezeichnet. Die Bedingung, daß B geordnet ist, vereinfacht die Beschreibung des Simplex-Verfahrens. Es gibt keine entsprechende Veranlassung, N_l und N_u als geordnet zu betrachten.

Für jede Basis B definieren wir eine *Basislösung* X folgendermaßen:

$$\begin{aligned} X_{N_l} &= l_{N_l}, \\ X_{N_u} &= u_{N_u}, \\ X_{N_{fx}} &= l_{N_{fx}} = u_{N_{fx}}, \\ X_{N_{fr}} &= 0, \text{ und} \\ X_B &= \mathbf{B}^{-1}(b - A_N X_N). \end{aligned} \tag{2}$$

Die Basis B heißt *zulässig*, falls $l_B \leq X_B \leq u_B$.

3 Das Simplex-Verfahren mit beschränkten Variablen

Algorithmus 3.1. Eine Iteration des Simplex-Verfahrens für (1)

Input: Eine zulässige Basis B und die entsprechenden Werte der Basisvariablen X_B .

Schritt S1: Löse $\pi^T \mathbf{B} = c_B^T$.

Schritt S2: (Pricing) Definiere $d_N = c_N - A_N^T \pi$. Falls $d_j \geq 0$ für alle $j \in N_l$, $d_j \leq 0$ für alle $j \in N_u$, und $d_j = 0$ für alle $j \in N_{fr}$ gilt, STOP- B ist optimal; andernfalls, wähle eine in die Basis *eintretende Variable* x_{j_e} , $j_e \in N$, so daß d_{j_e} eine dieser Bedingungen verletzt.

Schritt S3: Löse $\mathbf{B}y = A_{j_e}$.

Schritt S4: (Ratio-Test) Falls $d_{j_e} < 0$ gilt, setze

$$\Theta_i = \begin{cases} +\infty & \text{falls } y_i = 0, \\ (X_{B_i} - l_{B_i})/y_i & \text{falls } y_i > 0, \text{ und} \\ (X_{B_i} - u_{B_i})/y_i & \text{falls } y_i < 0, \end{cases}$$

und falls $d_{j_e} > 0$ gilt, setze

$$\Theta_i = \begin{cases} +\infty & \text{falls } y_i = 0, \\ (u_{B_i} - X_{B_i})/y_i & \text{falls } y_i > 0, \text{ und} \\ (l_{B_i} - X_{B_i})/y_i & \text{falls } y_i < 0 \end{cases}$$

für $i = 1, \dots, m$. Setze

$$\Theta = \min\{\min_i \Theta_i, u_{j_e} - l_{j_e}\}.$$

Falls $\Theta = +\infty$ gilt, STOP-Problem (1) ist unbeschränkt.

Schritt S5: (Update) Falls $d_{j_e} < 0$ gilt, setze $X_B \leftarrow X_B - \Theta y$; falls nicht, setze $X_B \leftarrow X_B + \Theta y$.

(S5.1) Für $\Theta = u_{j_e} - l_{j_e}$: Falls $j_e \in N_u$ gilt, setze $N_l \leftarrow N_l \cup \{j_e\}$ und $N_u \leftarrow N_u \setminus \{j_e\}$; falls nicht, setze $N_u \leftarrow N_u \cup \{j_e\}$ und $N_l \leftarrow N_l \setminus \{j_e\}$.

(S5.2) Für $\Theta < u_{j_e} - l_{j_e}$: Sei $i_i \in B$, so daß $\Theta_{i_i} = \Theta$, und definiere $j_i = B_{i_i}(x_{j_i}$ ist die aus der Basis austretende Variable). Setze $B_{i_i} \leftarrow j_e$, und setze

$$X_{B_{i_i}} \leftarrow \begin{cases} l_{j_e} + \Theta & \text{falls } j_e \in N_l, \\ u_{j_e} - \Theta & \text{falls } j_e \in N_u, \\ \Theta & \text{falls } j_e \in N_{fr} \text{ und } d_{j_e} < 0, \text{ und} \\ -\Theta & \text{falls } j_e \in N_{fr} \text{ und } d_{j_e} > 0. \end{cases}$$

Falls x_{j_i} fixiert ist, setze $N_{fx} \leftarrow N_{fx} \cup \{j_i\}$; andernfalls, setze $N_l \leftarrow N_l \cup \{j_i\}$, wenn $d_{j_e} y_{i_i} < 0$ gilt, und setze $N_u \leftarrow N_u \cup \{j_i\}$, wenn $d_{j_e} y_{i_i} > 0$ gilt. Entferne j_e aus $N_{fr} \cup N_l \cup N_u$. \square

Problem (1) heißt *unbeschränkt*, wenn zu jeder reellen Zahl M eine zulässige Lösung x existiert mit $c^T x < M$. Eine Basis B heißt *optimal*, wenn $c^T X \leq c^T x$ gilt für alle zulässige Lösungen x , wobei X die zugehörige Basislösung ist.

Der folgende Satz zeigt die notwendigen Eigenschaften auf, so daß Algorithmus 3.1 iterativ anwendbar ist (d.h. die Input-Bedingungen erhalten bleiben) und die darin behaupteten Schlußfolgerungen stimmen. Ferner wird gezeigt, daß sich die Zielfunktionswerte monoton entwickeln. Allerdings wird nicht bewiesen, daß das gesamte Verfahren endlich ist, was-nebenbei erwähnt-für Version nicht stimmt [2].

Satz. Falls Algorithmus 3.1 im Schritt S2 aufhört, ist B eine optimale Basis. Falls er im Schritt S4 aufhört, ist Problem (1) unbeschränkt. Andernfalls genügen die im Schritt S5 produzierte Basis B und der Vektor X_B den Input-Bedingungen. Ferner ist der neue Zielfunktionswert nicht größer als der alte Wert.

Beweis. (Optimalität) Per Annahme ist B zulässig. Nehmen wir an, daß der Algorithmus im Schritt S2 aufhört. Sei $d = c - A^T \pi$, wobei π aus dem Schritt S1 stammt. Nun gilt für jede zulässige Lösung x :

$$d^T x = c^T x - \pi^T A x = c^T x - \pi^T b.$$

Daraus folgt, daß das Problem, $d^T x$ zu minimieren, äquivalent ist zu dem Problem, $c^T x$ zu minimieren. Aber für jede zulässige Lösung x gilt

$$\begin{aligned} d^T x &= d_B^T x_B + d_{N_l}^T x_{N_l} + d_{N_u}^T x_{N_u} + d_{N_{fr}}^T x_{N_{fr}} + d_{N_{fx}}^T x_{N_{fx}} \\ &= d_B^T X_B + d_{N_l}^T x_{N_l} + d_{N_u}^T x_{N_u} + d_{N_{fr}}^T X_{N_{fr}} + d_{N_{fx}}^T X_{N_{fx}} \\ &\geq d_B^T X_B + d_{N_l}^T l_{N_l} + d_{N_u}^T u_{N_u} + d_{N_{fr}}^T X_{N_{fr}} + d_{N_{fx}}^T X_{N_{fx}} \\ &= d^T X. \end{aligned}$$

Und zwar gilt die zweite Gleichung, weil $d_B = 0$, $d_{N_{fr}} = 0$, und $x_{N_{fx}} = X_{N_{fx}}$ ist. Die Ungleichung gilt, weil $x_{N_l} \geq l_{N_l}$, $d_{N_l} \geq 0$, $x_{N_u} \leq u_{N_u}$, und $d_{N_u} \leq 0$ ist. Also, ist B optimal.

(Unbeschränktheit) Nehmen wir an, daß $d_{j_e} < 0$. Für $\alpha \geq 0$

$$\begin{aligned} X_{j_e}^\alpha &= X_{j_e} + \alpha, \\ X_B^\alpha &= X_B - \alpha y, \text{ und} \\ X_j^\alpha &= X_j \text{ für } j \notin B \cup \{j_e\}, \end{aligned}$$

wobei $By = A_{j_e}$ gilt, und X durch (2) gegeben ist. Offensichtlich gilt $AX^\alpha = b$. Wenn $\alpha \leq \Theta$, gilt $l_{j_e} \leq X_{j_e}^\alpha \leq u_{j_e}$. Das impliziert $l_N \leq X_N \leq u_N$. Wenn zusätzlich $0 \leq \alpha \leq \min_i \Theta_i$ erfüllt ist, gilt $l_B \leq X_B^\alpha \leq u_B$, woraus folgt, daß X^α zulässig ist. Darüberhinaus gilt

$$\begin{aligned} c^T X^\alpha &= c^T X + \alpha c_{j_e} - \alpha c_B^T y \\ &= c^T X + \alpha c_{j_e} - \alpha \pi^T A_{j_e} \\ &= c^T X + \alpha d_{j_e}. \end{aligned}$$

Falls $\Theta = +\infty$, können wir folgern, daß (1) unbeschränkt ist. Analog, falls $d_{j_e} > 0$ (statt $d_{j_e} < 0$) gilt, definieren wir $X_{j_e}^\alpha = X_{j_e} - \alpha$ und $X_B^\alpha = X_B + \alpha y$. Der Beweis läuft dann genau so wie im ersten Fall.

(Update) Nehmen wir an, daß Schritt S5 durchgeführt wird: Das bedeutet also, daß der Algorithmus weder im Schritt S2 noch im Schritt S4 aufhört. Dann müssen wir zeigen, daß die im Schritt S5 erzeugte geordnete Menge B eine Basis ist und daß X_B die entsprechende Basislösung darstellt. Zunächst stellen wir fest, daß X_B gleich X_B^α ist mit $\alpha = \Theta$, wobei X_B^α wie in dem Beweis der Unbeschränktheit definiert ist. Es folgt, daß, falls B eine Basis ist, X_B die zugehörige Basislösung ist. Es ist dann auch leicht zu zeigen angesichts der gewählten d_{j_e} und Θ Werte, daß $c^T X^\Theta \leq c^T X$ gilt.

Falls B unverändert bleibt, gibt es nichts mehr zu beweisen. Andernfalls, müssen wir noch zeigen, daß B nicht singulär ist. Sei \bar{B} die neue Matrix B , es gilt

$$\bar{B} = BE,$$

wobei

$$E = I + (y - e_{i_i})e_{i_i}^T.$$

I stellt dabei eine $m \times m$ Einheitsmatrix, und e_{i_i} einen m -dimensionalen Einheitsvektor (mit i_i -ter Komponente gleich Eins) dar.

Da B per Annahme nicht singularär ist, impliziert $BEv = 0$, daß

$$v + v_{i_i}(y - e_{i_i}) = Ev = 0$$

ist. Daraus können wir $v = 0$ folgern, weil, gemäß Definition von Θ , $y_{i_i} \neq 0$ ist. B ist also nicht singularär. \square

4 Die Netlib-LP-Probleme

Für unsere rechnerischen Untersuchungen, verwenden wir die sogenannten Netlib-Probleme³. So weit wir wissen, enthält diese Liste keine zufällig erzeugten Probleme. Alle darin aufgenommenen Probleme sind durch echte Modellierungsversuche entstanden.

Ein paar allgemeine Bemerkungen zu der Liste erscheinen hier angebracht. Erstens sind einige Probleme nur deshalb in die Liste aufgenommen worden, weil sie zu irgendwelchen Schwierigkeiten für das Simplex-Verfahren geführt haben. Solche Probleme sind sehr nützlich, wenn es um die Entwicklung eines robusten Codes geht. Sie sind aber nicht besonders gut geeignet zur Bestimmung von Parameterwerten und dergleichen.

Ein zweiter Punkt ist, daß die Netlib-Liste mehrere Klassen von Problemen enthält. Als Beispiele, sind the Pilot-Probleme (pilot 4, perold, pilotwe, pilotnov, pilotja, pilots and pilot87)) und die Fit-Probleme (fit1p, fit1d, fit2p und fit2d) zu erwähnen. Das Lösen einer solchen Klasse zeigt zwar gut auf, wie das Simplex-Verfahren sich verhält, wenn die Größe des Problems wächst. Es hat aber wenig Sinn, eine ganze Klasse zu lösen, ohne einige kleinere Exemplare zuerst näher zu untersuchen. Die 'd' und 'p' Versionen der Fit-Probleme stellen z.B. äquivalente, duale Formulierungen desselben Modells dar. Nachdem es sich gezeigt hat, daß fit1d viel leichter mit dem Simplex-Verfahren zu behandeln ist als fit1p, gibt es wenig praktischen Grund, fit2p, dem größeren Bruder von fit1p, weitere Aufmerksamkeit zu schenken. Ebenfalls ist es in den Fachkreisen sehr wohl bekannt, daß die Pilot-Probleme schlecht skaliert sind. Diese Klasse läßt sich also viel besser behandeln, wenn man ein "stärkeres" Skalierungsverfahren anwendet, als das, was von CPLEX per Default benutzt wird⁴.

³Die Netlib-LP-Liste wurde von David Gay zusammengestellt. Sie ist zur Zeit durch "anonymous ftp" unter der Adresse 'research.att.com' erhältlich (userid: anonymous, password: <blank>). Die Daten sind im Unterverzeichnis 'dist/lpdata' zu finden.

⁴Das von CPLEX per Default verwendete Skalierungsverfahren ist sehr einfach. Nur die Nebenbedingungsmatrix A wird skaliert. Zunächst wird jede nichtleere Zeile durch den, in der Zeile enthaltenen, größten absoluten Betrag dividiert. Anschließend wird jede Spalte in gleicher Weise behandelt.

Am Ende dieses Abschnitts, sind drei Tabellen aufgeführt (I, II, III). Tabelle I enthält allgemeine Daten über die Netlib-Probleme. Tabellen II und III enthalten numerische Information über die von CPLEX gelieferten optimalen Lösungen. Ein Leerzeichen entspricht jeweils einer Null. Die Probleme sind nach der Anzahl der in der Nebenbedingungsmatrix enthaltenen Nicht-Null-Einträge sortiert. Bilder der einzelnen Matrizen sind in [7] und [8] vorzufinden (der Tech-Report [8] enthält viel mehr Daten als die veröffentlichte Version).

In Tabelle I ist die Anzahl der verschiedenen Zeilentypen (Constrainttypen) und Spaltentypen (Variablentypen) angegeben. Eingeschachtelte Variablen sind Variablen mit endlichen oberen und unteren Schranken. Fixierte Variablen sind Variablen, die $l_j = u_j$ erfüllen. Wie zu erwarten ist, sind die meisten Variablen nichtnegativ.

Zu den üblichen Zeilentypen sind zusätzlich die Anzahl der Range-Zeilen und freien Zeilen aufgelistet. Range-Zeilen sind Zeilen, die Bedingungen folgender Art erfüllen:

$$\underline{b}_i \leq a_i^T x \leq \bar{b}_i$$

Diese lassen sich durch Hinzufügen einer mit entsprechenden Schranken versehenen Variablen y modellieren:

$$\begin{aligned} a_i^T x - y &= 0 \\ \underline{b}_i &\leq y_i \leq \bar{b}_i \end{aligned}$$

Freie oder unbeschränkte Zeilen werden normalerweise von dem Modellhersteller entweder als eine Art Buchführungsmittel eingeführt (d.h., die veranlassen, daß das entsprechende innere Produkt $a_i^T x$ automatisch von dem LP-Code berechnet wird) oder als Mittel zum Speichern von alternativen Zielfunktionen. Solche Zeilen werden während der Optimierungsphase einfach ignoriert.

Eine der ersten Aufgaben, bei der Implementierung der meisten numerisch ausgerichteten Algorithmen, ist die Bestimmung verschiedener Toleranzen. Diese Toleranzen ersetzen häufig Nullen, die in der mathematischen Beschreibung vorkommen. (Eine sorgfältige Behandlung der Toleranzbestimmung für das Simplex-Verfahren würde den Rahmen dieses Papers sprengen, und wird hier also nicht unternommen.) Zwei wichtige Toleranzen stellen die Zulässigkeits- und Optimalitätstoleranzen dar. Die Zulässigkeits-toleranz bestimmt, inwieweit die Werte der Basisvariablen ihre Schranken verletzen dürfen, damit die Basis noch als zulässig zu bezeichnen ist. Die Optimalitätstoleranz bestimmt, inwieweit die Werte der reduzierten Kosten die Optimalitätsbedingung verletzen dürfen, damit eine Basis noch als optimal zu bezeichnen ist. Diese Toleranzen haben den Defaultwert 1.0E-6 (0.000001), können aber, unabhängig von einander, bis auf den Wert 1.0E-9 neu gesetzt werden.

Eine dritte wichtige Toleranz ist die Pivotschwelle, die benutzt wird, um die Zerlegung der Basismatrix zu berechnen. Das von CPLEX verwendete Zelegungsverfahren basiert auf die Markowitz-Methode [10]. Diese Methode wurde nach dem in [12] beschriebenen Muster implementiert. Darin wird eine Relaxierung von Partialpivoting benutzt, die Schwellenpivoting heißt [4]. Eine Pivotschwelle α wird also angegeben, wobei $0 < \alpha \leq 1$ gelten muß. Eine Pivot p wird nur dann akzeptiert, falls $|p| \geq \alpha q$ gilt, wobei q der größte absolute Betrag innerhalb der von p bestimmten Zeile ist. CPLEX

benutzt per Default den Wert $\alpha = 0.01$. Die Pivotschwelle α kann neu gesetzt werden innerhalb des Intervalles $[0.0001, 0.99999]$ ($\alpha = 1.0$ wird aus numerischen Gründen nicht erlaubt).

Die in Tabellen II und III angegebenen Zielfunktionswerte wurden auf folgende Weise berechnet. Zunächst wurden die Probleme per Default gelöst. Anschließend wurde α auf 0.9999 und die Zulässigkeits- und Optimalitätstoleranzen auf 1.0E-9 gesetzt. Daraufhin wurde nochmals optimiert. In der letzten Phase wurden die alten, per Default berechneten, optimalen Basen als Startbasen eingesetzt. Die Tests wurden auf einer SPARCstation 2 mit IEEE Gleitkomma-Arithmetik durchgeführt. Das liefert 15-17 Stellen Genauigkeit. Die Tabellen enthalten Daten für die Default-Basen und die mit höheren Genauigkeit berechneten Basen. Es werden auch Daten für die skalierten und nichtskalierten Versionen jedes Problems aufgelistet (obwohl zu bemerken ist, daß CPLEX das Lösen eines Problems in der nichtskalierten Form nicht zuläßt).

Die Konditionszahlen in Tabelle II wurden in der L_∞ -Norm berechnet [4]. Die relative Zielfunktionsabweichung ist durch $|(z_d - z_g)/z_g|$ definiert, wobei z_g den Default-Zielfunktionswert und z_d der genauere Zielfunktionswert darstellt.

Die aufgezeigten Abweichungen wurden in der Maxnorm berechnet: $\|AX - b\|_\infty$ und $\|c_B^T - \pi^T B\|_\infty$, wobei X durch Anwendung von (2) und π durch Lösen des Systems $\pi^T B = c_B^T$ bestimmt wurden.

Die Unzulässigkeiten für die Schranken und die reduzierten Kosten wurden wie folgt berechnet:

$$\begin{aligned} \text{Schrankenunzulässigkeit} &= \max\{\max\{X_j - u_j, l_j - X_j, 0\} : j \in B\}, \text{ und} \\ \text{Unzulässigkeit der reduzierten Kosten} &= \max\{\Delta_l, \Delta_u, \Delta_f\}, \end{aligned}$$

wobei

$$\begin{aligned} \Delta_l &= \max\{\max\{c_j - \pi^T A_j, 0\} : j \in N_l\}, \\ \Delta_u &= \max\{\max\{\pi^T A_j - c_j, 0\} : j \in N_u\}, \text{ und} \\ \Delta_f &= \max\{|c_j - \pi^T A_j| : j \in N_{fr}\}. \end{aligned}$$

Die oben angegebenen Formeln für Δ_l und Δ_u nutzen die Eigenschaft aus, daß alle Netlib-Probleme als Minimierungsprobleme formuliert sind. Wie bereits erwähnt wurde, sind die Defaultwerte der Zulässigkeits- und Optimalitätstoleranzen gleich 1.0E-6. Die Unzulässigkeiten in den Schranken und reduzierten Kosten müssen also kleiner gleich 1.0E-6 sein, damit die Lösung als optimal bezeichnet werden kann.

TABELLE I
Netlib-Problem-Daten

PROBLEM	Optimal Value	Cons. Vars.		Nonzeros			Constraint Types				Variable Types						
				Cona.	Obj.	RHS	Less	Greater	Equal	Range	Nonneg.	Box	Free	Fix	Other		
1 afiro	-4.6475314286E+02	27	32	83	5	7	19		8			32					
2 sc50b	-7.0000000000E+01	50	48	118	1	5	30		20			48					
3 sc50a	-6.4575077059E+01	50	48	130	1	10	30		20			48					
4 sc105	-5.2202061212E+01	105	103	280	1	20	60		45			103					
5 kb2	-1.7499001299E+03	43	41	286	5	0	12	15	16			32	9				
6 adlittle	2.2549496316E+05	56	97	383	82	37	40	1	15			97					
7 scagr7	-2.3313898243E+06	129	140	420	133	53	38	7	84			140					
8 stocfor1	-4.1131976219E+04	117	111	447	27	8	48	6	63			111					
9 blend	-3.0812149846E+01	74	83	491	30	8	31		43			83					
10 sc205	-5.2202061212E+01	205	203	551	1	38	114		91			203					
11 recipe	-2.6661600000E+02	91	180	663	89	0	6	18	67			85	69		26		
12 share2b	-4.1573224074E+02	96	79	694	36	24	83		13			79					
13 vtpbase	1.2983146246E+05	198	203	908	6	59	133	10	55			87	65	1	18		32
14 lotfi	-2.5264706062E+01	153	308	1078	8	49	42	16	95			308					
15 share1b	-7.6589318579E+04	117	225	1151	31	103	28		89			225					
16 boeing2	-3.1501872802E+02	166	143	1196	143	39	1	142	23	19		89	54				
17 scorpion	1.8781248227E+03	388	358	1426	282	76	48	60	280			358					
18 bore3d	1.3730803942E+03	233	315	1429	96	0	19		214			302	11		1		1
19 scagr25	-1.4753433061E+07	471	500	1554	475	179	146	25	300			500					
20 sctapl	1.4122500000E+03	300	480	1692	360	154		180	120			480					
21 capri	2.6900129138E+03	271	353	1767	19	130	75	54	142			192	131	14	16		
22 brandy	1.5185098965E+03	220	249	2148	2	54	54		166			249					
23 israel	-8.9664482186E+05	174	142	2269	89	171	174					142					
24 finnis	1.7279106560E+05	497	614	2310	404	116	302	148	47			492	36		45		41
25 gfrdpnc	6.9022359995E+06	616	1092	2377	1090	68	68		548			834	258				
26 scsd1	8.66666666743E+00	77	760	2388	760	1			77			760					
27 etamacro	-7.5571523337E+02	400	688	2409	80	24	48	80	272			426	135		82		45
28 agg	-3.5991767287E+07	488	163	2410	131	432	405	47	36			163					
29 bandm	-1.5862801845E+02	305	472	2494	165	118			305			472					
30 e226	-1.8751929066E+01	223	282	2578	189	99	185	5	33			282					
31 scfxm1	1.8416759028E+04	330	457	2589	23	116	143		187			457					
32 grow7	-4.7787811815E+07	140	301	2612	21	0			140			21	280				
33 standata	1.2576995000E+03	359	1183	3031	7	7	199		160			1063	104		16		
34 scrs8	9.0429695380E+02	490	1169	3182	847	77	59	47	384			1169					
35 beaconfd	3.3592485807E+04	173	262	3375	101	67	33		140			262					
36 boeing1	-3.3521356751E+02	351	384	3485	380	146	4	249	9	89		228	156				
37 shell	1.2088253460E+09	536	1775	3556	1344	2	2		534			1399	117		250		9
38 standmps	1.4060175000E+03	467	1075	3679	7	115	199		268			955	104		16		
39 stair	-2.5126695119E+02	356	467	3856	1	70	147		209			373	6	6	82		
40 degen2	-1.4351780000E+03	444	534	3978	471	243	223		221			534					
41 agg2	-2.0239252356E+07	516	302	4284	231	472	456		60			302					
42 agg3	1.0312115935E+07	516	302	4300	231	467	456		60			302					
43 scsd6	5.0500000077E+01	147	1350	4316	1350	9			147			1350					
44 ship04s	1.7987147004E+06	402	1458	4352	1458	263	40	8	354			1458					
45 seba	1.5711600000E+04	515	1028	4352	522	9		1	507	7		521	507				
46 tuff	2.9214776509E-01	333	587	4520	3	0	15	26	292			559	26	2			
47 forplan	-6.6421896127E+02	161	420	4563	353	15	50	20	90	1		396	21		3		
48 bnll	1.9776295615E+03	643	1175	5121	1008	276	205	206	232			1175					
49 pilot4	-2.5811392589E+03	410	1000	5141	4	165	26	97	287			635	247	88	30		
50 scfxm2	3.6660261565E+04	660	914	5183	46	237	286		374			914					

TABELLE I
Netlib-Problem-Daten

PROBLEM	Optimal Value	Cons. Vars.		Nonzeros			Constraint Types				Variable Types						
				Cons.	Obj.	RHS	Less	Greater	Equal	Range	Nonneg.	Box	Free	Fix	Other		
51 growl5	-1.0687094129E+08	300	645	5620	45	0			300			45	600				
52 perold	-9.3807552782E+03	625	1376	6018	8	215	40	90	495			951	266	88	64		7
53 mmt800	5.5567956482E+05	524	854	6227	8	204	93	81	350			854					
54 ship04l	1.7933245380E+06	402	2118	6332	2118	263	40	8	354			2118					
55 sctap2	1.7248071429E+03	1090	1880	6714	1410	521			620	470		1880					
56 ganges	-1.0958573613E+05	1309	1681	6912	109	491	25		1284			1277	397				7
57 ship08s	1.9200982105E+06	778	2387	7114	2387	413	72	8	698			2387					
58 sierra	1.5394362184E+07	1227	2036	7302	1950	781	633	66	528				2016			20	
59 scfxm3	5.4901254550E+04	990	1371	7777	69	358	429		561			1371					
60 ship12s	1.4892361344E+06	1151	2763	8178	2763	653	101	5	1045			2763					
61 grow22	-1.6083433648E+08	440	946	8252	66	0			440			66	880				
62 stocfor2	-3.9024408538E+04	2157	2031	8343	1149	8	888	126	1143			2031					
63 scsd8	9.0499999993E+03	397	2750	8584	2750	15			397			2750					
64 sctap3	1.4240000000E+03	1480	2480	8874	1860	682			860	620		2480					
65 pilotwe	-2.7201075328E+06	722	2789	9126	92	73	30	109	583			2335	294	80	78		2
66 maros	-5.8063743701E+04	846	1443	9614	392	42	399	124	323			1405			35		3
67 fitlp	9.1463780924E+03	627	1677	9868	1026	627			627			1278	399				
68 25fv47	5.5018458883E+03	821	1571	10400	727	287	305		516			1571					
69 czprob	2.1851966989E+06	929	3523	10669	3504	860	38	1	890			3294				229	
70 ship08l	1.9090552114E+06	778	4283	12802	4283	413	72	8	698			4283					
71 pilotnov	-4.4972761882E+03	975	2172	13057	72	332	151	123	701			1628	340		204		
72 nesm	1.4076036488E+07	662	2923	13288	700	542	94		480	88		1009	1508		175		231
73 fitld	-9.1463780924E+03	24	1026	13404	1026	0	12	11	1				1026				
74 bnl2	1.8112365404E+03	2324	3489	13999	2125	441	482	515	1327			3489					
75 pilotja	-6.1131364654E+03	940	1988	14698	8	309	151	128	661			1250	339	88	311		
76 ship12l	1.4701879193E+06	1151	5427	16170	5427	653	101	5	1045			5427					
77 cycle	-5.2263930249E+00	1903	2857	20720	602	0	146	368	1389			2773	77	7			
78 80bau3b	9.8722419241E+05	2262	9799	21002	8061	346	35		2227			6244	2986		498		71
79 degen3	-9.8729400000E+02	1503	1818	24646	1584	594	786		717			1818					
80 truss	4.5881584719E+05	1000	8806	27836					1000			8806					
81 greenbea	-7.2555248130E+07	2392	5405	30877	622	0	107	86	2199			4996	290		103		16
82 greenbeb	-4.3022602612E+06	2392	5405	30877	622	0	107	86	2199			4978	291	4	115		17
83 d2q06c	1.2278421081E+05	2171	5167	32417	3257	874	664		1507			5167					
84 woodw	1.3044763331E+00	1098	8405	37474	4	34	9	4	1085			8405					
85 pilots	-5.5748972928E+02	1441	3652	43167	53	282	1191	17	233			2320	1040		203		89
86 fit2p	6.8464293294E+04	3000	13525	50284	10500	1500			3000			6025	7500				
87 stocfor3	-3.9976783944E+04	16675	15695	64875	9129	8	6866	980	8829			15695					
88 wood1p	1.4429024116E+00	244	2594	70215	1	2			1	243		2594					
89 pilot87	3.0171034733E+02	2030	4883	73152	652	274	1708	89	233			2971	1578		220		114
90 fit2d	-6.8464293294E+04	25	10500	129018	9000	0	10	14	1				10500				

TABELLE II
Stabilität von optimalen Basen

PROBLEM	Reoptimized Basis Condition Number		Default Settings Objective Value	Reoptimized Objective Value	Relative Objective Error
	Unscaled	Scaled			
1 afiro	3.21E+01	3.13E+01	-4.647531428571E+02	-4.647531428571E+02	
2 sc50b	1.23E+02	1.23E+02	-7.000000000000E+01	-7.000000000000E+01	
3 sc50a	1.28E+02	9.52E+01	-6.457507705856E+01	-6.457507705856E+01	
4 sc105	4.83E+02	3.28E+02	-5.220206121171E+01	-5.220206121171E+01	
5 kb2	1.56E+06	2.87E+03	-1.749900129906E+03	-1.749900129906E+03	
6 adlittle	5.47E+03	4.35E+02	2.254949631624E+05	2.254949631624E+05	
7 scagr7	1.04E+04	4.24E+03	-2.331389824331E+06	-2.331389824331E+06	
8 stocfor1	5.07E+05	1.93E+02	-4.113197621944E+04	-4.113197621944E+04	
9 blend	1.33E+04	4.07E+02	-3.081214984583E+01	-3.081214984583E+01	
10 sc205	2.30E+03	1.75E+03	-5.220206121171E+01	-5.220206121171E+01	
11 recipe	4.82E+05	1.04E+02	-2.666160000000E+02	-2.666160000000E+02	
12 share2b	1.29E+06	7.49E+03	-4.157322407414E+02	-4.157322407414E+02	
13 vtpbase	1.85E+09	9.23E+04	1.298314624614E+05	1.298314624614E+05	
14 lotfi	1.05E+07	3.76E+04	-2.526470606188E+01	-2.526470606188E+01	
15 share1b	2.39E+07	3.97E+03	-7.658931857919E+04	-7.658931857919E+04	
16 boeing2	4.89E+06	5.34E+04	-3.150187280152E+02	-3.150187280152E+02	
17 scorpion	3.24E+03	2.38E+03	1.878124822738E+03	1.878124822738E+03	
18 bore3d	1.54E+07	6.43E+04	1.373080394208E+03	1.373080394208E+03	
19 scagr25	1.38E+04	4.18E+03	-1.475343306077E+07	-1.475343306077E+07	
20 sctapl	6.82E+04	1.68E+02	1.412250000000E+03	1.412250000000E+03	
21 capri	1.13E+06	3.84E+04	2.690012913768E+03	2.690012913768E+03	
22 brandy	1.24E+05	1.32E+03	1.518509896488E+03	1.518509896488E+03	
23 israel	1.53E+07	2.66E+04	-8.966448218630E+05	-8.966448218630E+05	
24 finnis	9.91E+04	3.66E+03	1.727910655956E+05	1.727910655956E+05	
25 gfrdpnc	1.94E+05	1.38E+05	6.902235999549E+06	6.902235999549E+06	
26 scsd1	7.65E+01	7.65E+01	8.666666674333E+00	8.666666674333E+00	
27 etamacro	5.53E+06	5.66E+03	-7.557152298914E+02	-7.557152333749E+02	4.610E-09
28 agg	5.08E+07	2.04E+05	-3.599176728658E+07	-3.599176728658E+07	
29 bandm	2.39E+05	5.16E+03	-1.586280184501E+02	-1.586280184501E+02	
30 e226	1.57E+07	1.11E+04	-1.875192906637E+01	-1.875192906637E+01	
31 scfxm1	2.88E+06	3.93E+03	1.841675902835E+04	1.841675902835E+04	
32 grow7	4.16E+02	3.64E+02	-4.778781181471E+07	-4.778781181471E+07	
33 standata	3.23E+04	8.53E+02	1.257699500000E+03	1.257699500000E+03	
34 scrs8	5.65E+05	2.78E+03	9.042969538008E+02	9.042969538008E+02	
35 beaconfd	1.04E+04	4.02E+01	3.359248580720E+04	3.359248580720E+04	
36 boeing1	5.86E+07	2.29E+05	-3.352135675071E+02	-3.352135675071E+02	
37 shell	1.20E+02	1.20E+02	1.208825346000E+09	1.208825346000E+09	
38 standmps	3.28E+04	2.73E+03	1.406017500000E+03	1.406017500000E+03	
39 stair	8.94E+04	2.16E+04	-2.512669511930E+02	-2.512669511930E+02	
40 degen2	3.80E+03	3.80E+03	-1.435178000000E+03	-1.435178000000E+03	
41 agg2	1.58E+05	6.71E+03	-2.023925235598E+07	-2.023925235598E+07	
42 agg3	3.43E+05	1.40E+04	1.031211593509E+07	1.031211593509E+07	
43 scsd6	9.31E+02	9.31E+02	5.050000007826E+01	5.050000007714E+01	2.218E-11
44 ship04s	2.29E+03	2.65E+02	1.798714700445E+06	1.798714700445E+06	
45 seba	3.78E+06	1.67E+05	1.571160000000E+04	1.571160000000E+04	
46 tuff	1.00E+08	7.63E+03	2.921477650936E-01	2.921477650936E-01	
47 forplan	8.68E+07	7.82E+04	-6.642189612722E+02	-6.642189612722E+02	
48 bn1	2.12E+07	1.37E+06	1.977629561523E+03	1.977629561523E+03	
49 pilot4	5.48E+10	4.95E+06	-2.581139259543E+03	-2.581139258885E+03	2.549E-10
50 scfxm2	4.09E+06	3.95E+03	3.666026156500E+04	3.666026156500E+04	

TABELLE II
Stabilität von optimalen Basen

PROBLEM	Reoptimized Basis Condition Number		Default Settings Objective Value	Reoptimized Objective Value	Relative Objective Error
	Unscaled	Scaled			
51 grow15	2.76E+03	2.68E+03	-1.068709412936E+08	-1.068709412936E+08	3.198E-13
52 perold	1.37E+10	4.54E+06	-9.380755278232E+03	-9.380755278235E+03	
53 mmt800	4.72E+10	1.39E+07	5.556795648175E+05	5.556795648175E+05	
54 ship04l	8.73E+02	7.13E+01	1.793324537970E+06	1.793324537970E+06	
55 sctap2	5.20E+04	2.11E+02	1.724807142857E+03	1.724807142857E+03	
56 ganges	1.98E+05	1.32E+05	-1.095857361293E+05	-1.095857361293E+05	
57 ship08s	1.03E+04	1.48E+02	1.920098210535E+06	1.920098210535E+06	
58 sierra	1.00E+15	1.23E+02	1.539436218363E+07	1.539436218363E+07	
59 scfxm3	4.09E+06	3.95E+03	5.490125454975E+04	5.490125454975E+04	
60 ship12s	1.01E+04	1.44E+02	1.489236134406E+06	1.489236134406E+06	
61 grow22	2.32E+03	2.15E+03	-1.608343364826E+08	-1.608343364826E+08	1.397E-11
62 stocfor2	9.55E+06	7.16E+03	-3.902440853788E+04	-3.902440853788E+04	
63 scsd8	1.67E+03	1.67E+03	9.049999999255E+02	9.049999999255E+02	
64 sctap3	2.20E+04	1.70E+02	1.424000000000E+03	1.424000000000E+03	
65 pilotwe	1.39E+10	1.49E+08	-2.720107532765E+06	-2.720107532803E+06	
66 maros	6.93E+09	1.19E+05	-5.806374370113E+04	-5.806374370113E+04	
67 fit1p	1.22E+08	6.42E+05	9.146378092421E+03	9.146378092421E+03	
68 25fv47	1.94E+06	2.03E+04	5.501845888287E+03	5.501845888287E+03	
69 czprob	1.48E+04	5.82E+02	2.185196698857E+06	2.185196698857E+06	
70 ship08l	1.73E+04	3.21E+02	1.909055211389E+06	1.909055211389E+06	
71 pilotnov	4.31E+13	1.28E+08	-4.497276188219E+03	-4.497276188219E+03	1.718E-11
72 nesm	1.34E+06	2.51E+06	1.407603648756E+07	1.407603648756E+07	
73 fit1d	1.08E+06	1.10E+03	-9.146378092421E+03	-9.146378092421E+03	
74 bnl2	5.66E+07	3.02E+06	1.811236540359E+03	1.811236540359E+03	
75 pilotja	5.37E+12	2.28E+09	-6.113136465533E+03	-6.113136465428E+03	
76 ship12l	1.01E+04	1.44E+02	1.470187919329E+06	1.470187919329E+06	
77 cycle	1.25E+08	5.19E+05	-5.226393024894E+00	-5.226393024894E+00	
78 80bau3b	2.28E+04	6.01E+02	9.872241924091E+05	9.872241924091E+05	
79 degen3	5.35E+04	5.35E+04	-9.872940000000E+02	-9.872940000000E+02	
80 truss	8.57E+02	8.57E+02	4.588158471856E+05	4.588158471856E+05	
81 greenbea	4.53E+08	3.21E+07	-7.255524812985E+07	-7.255524812985E+07	4.149E-08
82 greenbeb	2.96E+06	1.58E+05	-4.302260261207E+06	-4.302260261207E+06	
83 d2q06c	7.38E+08	1.46E+06	1.227842108142E+05	1.227842108142E+05	
84 woodw	8.30E+04	1.55E+02	1.304476333084E+00	1.304476333084E+00	
85 pilots	3.30E+08	2.96E+07	-5.574897061534E+02	-5.574897292840E+02	
86 fit2p	1.24E+09	2.77E+07	6.846429329383E+04	6.846429329383E+04	
87 stocfor3	2.28E+08	2.49E+05	-3.997678394365E+04	-3.997678394365E+04	
88 wood1p	2.32E+04	4.65E+01	1.442902411573E+00	1.442902411573E+00	
89 pilot87	5.32E+08	1.43E+07	3.017103666613E+02	3.017103473331E+02	
90 fit2d	1.98E+04	2.77E+03	-6.846429329383E+04	-6.846429329383E+04	

TABELLE III

Abweichungen der optimalen Basen

PROBLEM	Default Settings Optimization								Reoptimized								
	Bound		Reduced-Cost		AX-b		c_B'-pi'B		Bound		Reduced-Cost		AX-b		c_B'-pi'B		
	Infeasibilities		Infeasibilities		Max. Residual		Max. Residual		Infeasibilities		Infeasibilities		Max. Residual		Max. Residual		
	Unsc.	Sc.	Unsc.	Sc.	Unsc.	Sc.	Unsc.	Sc.	Unsc.	Sc.	Unsc.	Sc.	Unsc.	Sc.	Unsc.	Sc.	
1	afiro					1E-14	1E-14	6E-17	6E-17					2E-14	2E-14	6E-17	6E-17
2	sc50b					1E-13		1E-16						1E-13		1E-16	
3	sc50a					2E-13	1E-13	1E-16	1E-16					3E-14	1E-14	1E-16	1E-16
4	sc105					9E-13	5E-13	1E-16	1E-16					3E-14	1E-14	1E-16	1E-16
5	kb2					8E-12	7E-14	1E-14	1E-14					2E-12	2E-14	2E-15	2E-15
6	adlittle			1E-13	1E-13	2E-13	1E-13	2E-13	2E-13		3E-13	3E-13		1E-13	1E-13	2E-13	2E-13
7	scagr7					1E-12	6E-13	5E-13	5E-13					1E-12	6E-13	5E-13	5E-13
8	stocfor1					5E-13	7E-15	9E-14	9E-14					5E-13	7E-15	1E-13	1E-13
9	blend					2E-13	4E-15	1E-15	1E-15					1E-14	4E-15	2E-15	2E-15
10	sc205					1E-12	6E-13	1E-16	1E-16					2E-13	1E-13	1E-16	1E-16
11	recipe																
12	share2b			4E-17	4E-15	4E-13	4E-15	7E-14	7E-14		9E-16	9E-16		4E-13	4E-15	6E-14	6E-14
13	vtpbase	1E-14	5E-17			3E-11	6E-14	4E-12	4E-12	1E-14	5E-17			3E-11	6E-14	7E-12	7E-12
14	lotfi	2E-13	4E-15			9E-10	9E-13	8E-17	8E-17	1E-12	1E-14			1E-12	1E-14	8E-17	8E-17
15	share1b					8E-10	9E-13	7E-15	7E-15					4E-10	2E-12	1E-14	1E-14
16	boeing2			9E-18	9E-16	7E-13	5E-13	1E-15	1E-15	9E-16	9E-16			6E-13	2E-13	2E-15	2E-15
17	scorpion	2E-16	2E-16			4E-16	4E-16	1E-13	1E-13	5E-17	8E-17			4E-16	4E-16	1E-13	1E-13
18	bore3d					6E-13	5E-13	2E-13	2E-13		6E-17	1E-16		6E-13	2E-13	9E-15	9E-15
19	scagr25					2E-12	1E-12	1E-12	1E-12					2E-12	9E-13	1E-12	1E-12
20	sctapl	2E-15	2E-16	7E-16	1E-14	9E-15	2E-16	7E-15	1E-14	2E-16	2E-16	6E-16	1E-14	2E-14	2E-16	7E-15	7E-15
21	capri					2E-12	2E-12	9E-14	9E-14	1E-14	1E-14			5E-13	5E-13	3E-14	4E-14
22	brandy					4E-13	1E-14	2E-14	2E-14					4E-13	1E-14	3E-15	3E-15
23	israel			2E-15	9E-15	1E-10	2E-12	9E-14	9E-14		4E-15	7E-15		1E-10	9E-13	3E-13	3E-13
24	finnis	7E-15	2E-15			5E-13	1E-13	3E-13	3E-13	7E-15	2E-15	3E-14	9E-14	4E-13	2E-13	3E-13	3E-13
25	gfrdpnc	2E-12	1E-15			5E-12	2E-12	7E-12	7E-12	1E-12	1E-15			5E-12	2E-12	7E-12	1E-11
26	scsd1					1E-16		1E-15		1E-17				1E-16		5E-15	
27	etamacro	3E-16	3E-16	7E-07	7E-07	1E-12	2E-13	3E-14	3E-14	3E-17	3E-17			1E-12	2E-13	3E-14	3E-14
28	agg					5E-10	9E-10	6E-14	6E-14					5E-10	9E-10	2E-13	2E-13
29	bandm					3E-12	3E-14	2E-13	2E-13					4E-13	6E-15	2E-15	2E-15
30	e226					1E-13	2E-15	1E-15	1E-15					7E-14	2E-15	9E-16	9E-16
31	scfxm1	2E-15	2E-15			2E-12	7E-13	9E-15	9E-15	2E-15	2E-15			3E-12	2E-13	3E-15	3E-15
32	grow7			2E-11	2E-11	2E-08	2E-08	3E-13	3E-13					2E-10	2E-10	2E-15	2E-15
33	standata					5E-13	2E-14	2E-15	2E-15					5E-13	2E-14	2E-15	2E-15
34	scrs8	2E-16	2E-16			3E-14	9E-16	9E-13	9E-13	2E-16	2E-16			6E-14	2E-15	4E-13	4E-13
35	beaconfd					3E-11	3E-13	2E-15	2E-15					3E-11	3E-13	2E-15	2E-15
36	boeing1			5E-18	8E-17	6E-12	3E-12	9E-15	9E-15		9E-18	2E-16		6E-12	9E-13	3E-15	3E-15
37	shell																
38	standmps					2E-13	5E-15	2E-15	2E-15					2E-13	5E-15	4E-15	4E-15
39	stair					2E-11	6E-12	6E-14	7E-14					2E-13	9E-14	2E-15	2E-15
40	degen2	2E-16				9E-16		7E-15		2E-16				9E-16		1E-14	
41	agg2					7E-11	3E-10	1E-14	1E-14					1E-10	5E-10	1E-14	1E-14
42	agg3					6E-11	5E-10	1E-14	1E-14					1E-10	5E-10	1E-14	1E-14
43	scsd6			2E-08		2E-16		4E-15						3E-16		2E-15	
44	ship04s	2E-14	2E-14	1E-13	1E-13	5E-14	5E-14			2E-14	2E-14			5E-14	5E-14		
45	seba					2E-12	6E-14							2E-12	6E-14		
46	tuff					2E-12	2E-14	9E-18	9E-18					5E-12	4E-14	2E-18	2E-18
47	forplan					2E-12	9E-13	1E-14	1E-14					2E-12	9E-13	1E-14	1E-14
48	bnll	3E-17	3E-17	4E-14	3E-13	4E-13	6E-14	2E-13	2E-13	1E-12	1E-12	1E-12	9E-12	3E-13	6E-14	1E-13	1E-13
49	pilot4	2E-07	5E-08	1E-13	1E-13	2E-07	2E-09	4E-13	4E-13	9E-13	1E-15			3E-07	8E-10	1E-13	1E-13
50	scfxm2	2E-15	2E-15	2E-15	4E-16	6E-12	1E-12	6E-14	6E-14	2E-15	2E-15			4E-12	7E-13	1E-14	1E-14

TABELLE III
Abweichungen der optimalen Basen

PROBLEM	Default Settings Optimization								Reoptimized								
	Bound		Reduced-Cost		AX-b		c_B-piB		Bound		Reduced-Cost		AX-b		c_B-piB		
	Infeasibilities		Infeasibilities		Max. Residual		Max. Residual		Infeasibilities		Infeasibilities		Max. Residual		Max. Residual		
	Unsl.	Sl.	Unsl.	Sl.	Unsl.	Sl.	Unsl.	Sl.	Unsl.	Sl.	Unsl.	Sl.	Unsl.	Sl.	Unsl.	Sl.	
51	grow15			7E-12	7E-12	3E-09	3E-09	6E-12	7E-12			2E-13	2E-13	2E-09	2E-09	2E-13	2E-13
52	perold	4E-12	9E-15	1E-13	1E-13	2E-06	2E-09	1E-11	1E-11	4E-12	9E-15	8E-14	1E-13	3E-09	3E-12	1E-13	1E-13
53	mmf800					1E-10	2E-11	3E-11	3E-11					2E-10	3E-11	3E-11	3E-11
54	ship041			4E-13	4E-13	2E-14	3E-14					4E-13	4E-13	2E-14	3E-14		
55	sctap2			5E-16	1E-14	9E-15	2E-16	7E-15	1E-14	6E-16	6E-16	4E-16	7E-15	9E-15	2E-16		
56	ganges					1E-11	1E-11	6E-14	6E-14			1E-13	1E-13	1E-11	1E-11	8E-14	8E-14
57	ship08s	5E-16	5E-16			2E-14	2E-14			5E-16	5E-16			2E-14	2E-14		
58	sierra	2E-13	2E-13	6E-07	6E-07	9E-13	9E-13	4E-12	4E-12	2E-13	2E-13	4E-12	4E-12	9E-13	9E-13	4E-12	4E-12
59	scfxm3	2E-15	2E-15	2E-15	7E-16	4E-12	7E-13	1E-13	1E-13	2E-15	2E-15	2E-15	7E-16	7E-12	7E-13	1E-14	1E-14
60	ship12s	2E-15	3E-14			3E-14	6E-14	9E-13	9E-13	2E-15	4E-14			3E-14	6E-14	9E-13	9E-13
61	grow22			5E-11	5E-11	3E-07	3E-07	8E-12	9E-12			7E-12	7E-12	2E-08	2E-08	5E-12	7E-12
62	stocfor2	7E-16	7E-16			5E-12	2E-14	6E-14	6E-14	7E-16	7E-16			2E-12	1E-14	9E-14	9E-14
63	scsd8	4E-16				4E-15		1E-14		2E-16				2E-15		2E-14	
64	sctap3			4E-16	7E-15	9E-15	2E-16	2E-14	2E-14			4E-16	7E-15	9E-15	2E-16	1E-14	1E-14
65	pilotwe	7E-12	2E-15			4E-07	5E-09	6E-10	6E-10	7E-12	2E-15			4E-05	8E-09	1E-09	1E-09
66	maros	2E-10	1E-14	5E-13	5E-13	5E-10	5E-12	3E-12	3E-12	8E-13	1E-13			9E-10	9E-13	2E-14	3E-14
67	fit1p					5E-14	2E-16	2E-12	2E-12					5E-14	2E-16	2E-12	2E-12
68	25fv47					1E-11	1E-11	2E-13	2E-13					2E-12	3E-13	2E-14	2E-14
69	czprob					6E-13	6E-13	3E-14	3E-14					2E-12	2E-12	3E-14	3E-14
70	ship081	4E-15	4E-15			4E-14	4E-14	9E-13	9E-13	4E-15	4E-15			2E-14	2E-14	9E-13	9E-13
71	pilotnov	1E-16	1E-16			6E-09	1E-10	3E-16	3E-16	2E-12	2E-15			5E-07	4E-09	1E-16	1E-16
72	nesm	1E-08	1E-08	2E-08	2E-07	1E-11	7E-13	2E-12	2E-12	9E-13	8E-14	7E-11	7E-11	3E-12	9E-13	8E-13	8E-13
73	fit1d					7E-13	9E-16	7E-15	7E-15					7E-13	9E-16	7E-15	7E-15
74	bnl2	4E-14	4E-14	7E-07	7E-07	2E-13	8E-14	1E-14	1E-14	2E-14	2E-14	1E-15	4E-15	2E-13	1E-13	1E-14	1E-14
75	pilotja	7E-12	3E-16	3E-15	3E-15	7E-05	3E-09	2E-12	2E-12	7E-12	3E-16	2E-13	2E-13	2E-04	6E-09	2E-11	2E-11
76	ship121	3E-15	3E-15			5E-14	6E-14	5E-13	5E-13	3E-15	3E-15			5E-14	6E-14		
77	cycle					3E-12	3E-14	3E-17	3E-17	3E-13	7E-15	4E-19	4E-19	2E-12	6E-14	3E-17	3E-17
78	80bau3b	2E-13	1E-13			2E-12	2E-12	3E-14	3E-14	2E-13	1E-13			2E-12	2E-12	1E-14	1E-14
79	degen3	4E-15				1E-14		4E-15		2E-15		4E-16		7E-15		4E-15	
80	truss	4E-15				2E-13		2E-12		4E-15				2E-13		2E-12	
81	greenbea	2E-13	2E-13	1E-12	1E-12	1E-08	1E-08	2E-11	2E-11	2E-13	2E-13	1E-12	1E-12	1E-08	1E-08	3E-11	3E-11
82	greenbeb	1E-13	1E-13	7E-17	1E-16	7E-12	7E-12	4E-12	4E-12	1E-13	1E-13	3E-13	3E-13	3E-12	3E-12	4E-12	4E-12
83	d2q06c			2E-15	2E-15	8E-10	7E-12	2E-13	2E-13			6E-16	2E-15	5E-10	9E-13	2E-14	2E-14
84	woodw					1E-13	1E-16	3E-16	3E-16					1E-13	1E-16	3E-16	3E-16
85	pilots	3E-08	2E-08	1E-08	5E-08	6E-10	8E-11	2E-12	2E-12	1E-13	2E-14	2E-16	2E-16	7E-10	7E-11	9E-14	9E-14
86	fit2p	1E-14	6E-16			7E-14	4E-15	3E-11	3E-11	1E-14	6E-16			7E-14	4E-15	3E-11	3E-11
87	stocfor3	1E-15	1E-15			2E-11	7E-14	4E-14	4E-14	3E-15	3E-15			9E-12	3E-14	1E-13	1E-13
88	wood1p	4E-14	4E-17			1E-13	3E-16	4E-15	4E-15	2E-14	2E-17			9E-14	7E-16	3E-15	3E-15
89	pilot87	2E-12	1E-13	2E-07	4E-07	6E-08	6E-09	2E-12	2E-12	2E-12	1E-13			1E-09	1E-10	9E-15	1E-14
90	fit2d					1E-13	4E-16	1E-14	1E-14					1E-13	4E-16	1E-14	1E-14

Teil II: Die Startbasis

5 Konstruieren der Startbasis

Das Simplex-Verfahren erfordert als Input eine zulässige Basis. Falls keine vorhanden ist, ist es üblich, ein entsprechendes Zusatzproblem zu konstruieren. Dieses Zusatzproblem sollte die Eigenschaft haben, daß es leicht ist eine zulässige Basis zu definieren. Diese zulässige Basis wird zunächst als Startbasis verwendet, um eine optimale Basis für das Zusatzproblem zu berechnen. Dieser Schritt heißt Phase I, und das Zusatzproblem heißt Phase I Problem. Anschliessend wird die optimale Phase I-Basis als Startbasis eingesetzt, um das ursprüngliche Problem zu lösen.

Wir werden in diesem Abschnitt vier verschiedene Startbasen für Phase I beschreiben. Erstens wird die sogenannte künstliche Basis definiert. Diese Basis ist praktisch die Startbasis, die in den meisten Lehrbüchern erklärt wird. Als nächstes wird die zulässige Schlupfbasis und anschließend die Schlupfbasis definiert. Diese zwei Basen entsprechen etwa einer Zwischenstation auf dem Weg zur Entwicklung der CPLEX-Basis. Zuletzt wird die CPLEX-Basis eingeführt.

Sei (B, N_l, N_u) eine unzulässige Basis. Für $j = 1, \dots, n$ definieren wir

$$p_j(x) = \begin{cases} x - u_j & \text{falls } x > u_j, \\ 0 & \text{falls } l_j \leq x \leq u_j, \text{ und} \\ l_j - x & \text{falls } l_j > x. \end{cases} \quad (3)$$

Das entsprechende Phase I Problem ist stückweise linear:

$$\min \sum_{j=1}^n p_j(x_j) \\ Ax = b \quad (4)$$

B ist für (4) offensichtlich zulässig, und es ist einfach zu sehen, daß (1) dann und nur dann eine zulässige Lösung hat, wenn (4) eine Lösung mit Zielfunktionswert 0.0 hat. Es ist zu bemerken, daß die CPLEX-Basis anfangs zur Verletzung der Schranken führen kann. Aus diesem Grund ist das dadurch entstandene Problem in der Form (4) zu betrachten. Andererseits führen künstliche Basen und zulässige Schlupfbasen zu Phase I Problemen mit echt linearen Zielfunktionen.

Künstliche Basis

Die einfachste Startbasis ist die künstliche Basis, die wir nunmehr definieren wollen. Sei

$$\begin{aligned} N &= \{1, \dots, n\}, \\ N_l &= \{j \in N \setminus (N_{fr} \cup N_{fx}) : |l_j| \leq |u_j|\}, \\ N_u &= \{j \in N \setminus (N_{fr} \cup N_{fx}) : |l_j| > |u_j|\}, \text{ und} \\ \bar{b} &= b - A_N X_N, \end{aligned} \quad (5)$$

und betrachte das Problem

$$\begin{aligned} \min \quad & e^T z \\ & Ax + Dz = b \\ & l \leq x \leq u \\ & z \geq 0, \end{aligned} \tag{6}$$

wobei $z^T = (x_{n+1}, \dots, x_{n+m})$, $e^T = (1, \dots, 1)$, und $D = \text{diag}(\delta_1, \dots, \delta_m)$ mit

$$\delta_i = \begin{cases} +1 & \text{falls } \bar{b}_i \geq 0, \text{ und} \\ -1 & \text{andernfalls.} \end{cases}$$

Die Variablen z_i heißen künstliche Variablen. Problem (1) ist offensichtlich zulässig dann und nur dann, wenn Problem (6) den optimalen Wert 0.0 besitzt.

Die künstliche Basis für (6) ist (B, N_l, N_u) , wobei $B = (n+1, \dots, n+m)$ ist, und N_l und N_u durch (5) gegeben sind.

Zulässige Schlupfbasis

Die meisten LPs weisen in ihrer ursprünglichen Form einige Ungleichungen auf:

$$\begin{aligned} \min \quad & c^T x \\ & A_1 x \leq b_1 \\ & A_2 x \geq b_2 \\ & A_3 x = b_3 \\ & l \leq x \leq u \end{aligned} \tag{7}$$

Nehmen wir an, daß b_i ein m_i -Vektor ist ($i = 1, 2, 3$), so daß $m = m_1 + m_2 + m_3$. Problem (1) läßt sich durch Hinzufügen von Schlupfvariablen $s_1^T = (x_{n+1}, \dots, x_{n+m_1})$ und $s_2^T = (x_{n+m_1+1}, \dots, x_{n+m_1+m_2})$ in die Form (1) transformieren

$$\begin{aligned} \min \quad & c^T x \\ & A_1 x + s_1 = b_1 \\ & A_2 x - s_2 = b_2 \\ & A_3 x = b_3 \\ & l \leq x \leq u \\ & s_1 \geq 0, s_2 \geq 0 \end{aligned}$$

Um eine zulässige Schlupfbasis zu bestimmen, konstruieren wir in einem ersten Schritt eine künstliche Schlupfbasis für das obige Problem. Es ist zu bemerken, daß alle Schlupfvariablen zunächst Elemente der Menge N_l sind. Anschließend führen wir die folgenden Veränderungen aus: Für alle $i \in \{1, \dots, m_1\}$ mit $\delta_i = 1$ und für alle $i \in \{m_1 + 1, \dots, m_1 + m_2\}$ mit $\delta_i = -1$ ersetzen wir in B die künstliche Variable $x_{n+m_1+m_2+i}$ durch die Schlupfvariable x_{m+i} . D.h. alle vorhandenen nichtnegativen Schlupfvariablen werden in die Basis aufgenommen.

Schlupfbasis

In diesem Fall benutzen wir alle Schlupfvariablen, unabhängig von ihren Werten. Wir betrachten also das Problem

$$\begin{aligned} \min \quad & \sum_{j=1}^{n+m} p_j(x_j) \\ & A_1x + s_1 = b_1 \\ & A_2x - s_2 = b_2 \\ & A_3x + z = b_3 \end{aligned}$$

Die Funktionen p_j sind durch (3) definiert, wobei die Schranken $s_1 \geq 0$, $s_2 \geq 0$ und $z = 0$ zu verwenden sind. Die entsprechende Schlupfbasis ist dann (B, N_l, N_u) , wobei $B = (n+1, \dots, m)$ ist, und N_l und N_u durch (5) gegeben sind.

CPLEX-Basis

Das folgende Verfahren ist etwas komplizierter als die bis jetzt vorgeführten. Es ist aber ganz leicht zu implementieren. Wir wollen eine Basis mit den folgenden Eigenschaften konstruieren: Sie soll dünn besetzt und numerisch stabil sein. Zusätzlich soll sie möglichst viel Bewegungsfreiheit aufweisen, d.h. die Variablenschranken sollen möglichst groß sein. Ferner soll die Basis möglichst wenige künstliche Variablen enthalten. Die Hauptschritte des Verfahrens sehen so aus: Zunächst wird eine Reihenfolge der Variablen bestimmt. Anschließend wird diese Reihenfolge verwendet, um die Basis zu konstruieren. Wie aus den Rechenergebnissen ersichtlich werden wird, kann diese Basis zu einer erheblichen Reduzierung der Anzahl der Iterationen führen, besonders bei einfacheren Problemen. Für "schwierigere" Probleme übt das Verfahren weniger Einfluß aus.

Es ist vielleicht ganz natürlich, genau das Gegenteil zu erwarten, daß bei schwierigeren Problemen eine gute Startbasis wirksamer sein sollte. Diese Überlegung gab in der Tat den Anlaß, sogenannte Crash-Prozeduren zu entwickeln. Solche Prozeduren können für LP Probleme mit bestimmten Strukturen besonders wirksam sein. Als typisches Beispiel erwähnen wir Modelle mit durch den Zeitablauf bestimmten Mehrfach-Perioden-Strukturen. Auf der anderen Seite ist, nach Ansicht des Autors, die CPLEX-Basis nicht als Crash-Prozedur zu betrachten, sondern einfach als eine Default-Startbasis. Im allgemeinen kann das Verwenden einer Crash-Prozedur für ein Problem unbekannter Herkunft katastrophale Folgen haben. Mit der CPLEX-Basis suchen wir also nicht unbedingt nach einer guten Basis, sondern vielmehr wollen wir eine schlechte vermeiden. Insbesondere wollen wir künstliche Variablen vermeiden, die durch ihre Schranken sehr eingengt sind, und sowieso mit wenigen Ausnahmen aus der Basis entfernt werden müssen.

Nehmen wir an, daß das ursprüngliche Problem in der Form (7) gegeben ist und in die Form (1) durch Hinzufügen von Schlupfvariablen transformiert wurde. Die oben erwähnte Reihenfolge der Variablen wird wie folgt bestimmt. Definieren wir Mengen C_1, C_2, C_3, C_4 , wobei die Elemente aus C_i gegenüber den Elementen aus

C_{i+1} ($i = 1, 2, 3$) als bevorzugt zu betrachten sind:

$$\begin{aligned} C_1 &= \{n+1, \dots, n+m_1+m_2\}, \\ C_2 &= \{j : x_j \text{ frei}\}, \\ C_3 &= \{j \leq n : \text{genau eine von } l_j, u_j \text{ ist endlich}\}, \text{ und} \\ C_4 &= \{j : -\infty < l_j, u_j < +\infty\}. \end{aligned}$$

Die Vereinigung dieser Mengen enthalten offensichtlich alle Variablenindizes. Es ist auch zu bemerken, daß C_1 genau die Indexmenge der Schlupfvariablen ist. Diese Variablen werden gegenüber den freien Variablen aufgrund ihrer Sparse-Struktur und wegen der guten numerischen Eigenschaften der Einheitsvektoren bevorzugt. Für $j \in \{1, \dots, n+m_1+m_2\}$ definieren wir Strafwerte \bar{q}_j wie folgt:

$$\bar{q}_j = \begin{cases} 0 & \text{falls } j \in C_2, \\ l_j & \text{falls } j \in C_3 \text{ und } u_j = +\infty, \\ -u_j & \text{falls } j \in C_3 \text{ und } l_j = -\infty, \text{ und} \\ l_j - u_j & \text{falls } j \in C_4. \end{cases}$$

Sei $\gamma = \max\{|c_j| : 1 \leq j \leq n\}$ und definiere

$$c_{max} = \begin{cases} 1000\gamma & \text{falls } \gamma \neq 0 \\ 1 & \text{andernfalls} \end{cases}$$

Ferner definiere für $j \in \{1, \dots, n\}$

$$q_j = \bar{q}_j + c_j/c_{max}.$$

Die Indizes aus den einzelnen Mengen C_1, C_2 und C_3 werden nun nach den Werten q_j in ansteigender Reihenfolge sortiert. Anschließend werden die dadurch erzeugten Listen in der Reihenfolge C_1, C_2, C_3, C_4 aneinandergesetzt. Sei (j_1, \dots, j_n) die dadurch entstandene Gesamtliste. Zusammenhängend läßt sich sagen, daß diese Liste zunächst bezüglich der Werte der Schranken und in einem zweiten Kriterium bezüglich der Zielfunktionswerte sortiert wurde. Wir sind nun in der Lage, die CPLEX-Basis zu konstruieren. Nachdem diese Konstruktion durchgeführt worden ist, werden die Indizes aus $N \setminus (B \cup N_{fr} \cup N_{fx})$ den Mengen N_l und N_u entsprechend zugeordnet. Dazu verwenden wir (5). Wie schon erwähnt, wird die Nebenbedingungsmatrix von CPLEX skaliert. Eine wichtige Eigenschaft dieser Skalierung ist, daß der größte absolute Betrag innerhalb jeder Nicht-Null-Zeile und jeder Nicht-Null-Spalte 1 ist.

Schritt 1: Für $i = 1, \dots, m_1 + m_2$ setze

$$\begin{aligned} I_i &\leftarrow 1, \text{ und} \\ r_i &\leftarrow 1. \end{aligned}$$

Setze $B \leftarrow \{n+1, \dots, n+m_1+m_2\}$. Für $i = m_1 + m_2 + 1, \dots, m$ setze $I_i \leftarrow 0$ und $r_i \leftarrow 0$. Für $i = 1, \dots, m$ setze $v_i \leftarrow +\infty$.

Schritt 2: Sei $C = (j_i, \dots, j_n)$. Für $k = 1, \dots, n$ führe die Schritte (a) und (b) aus:

(a) Sei $\alpha = \max\{|A_{lj_k}| : r_l = 0\}$. Falls $\alpha \geq 0.99$, wähle l' , so daß $\alpha = |A_{l'j_k}|$ und $r_{l'} = 0$ ist. Setze

$$\begin{aligned} B &\leftarrow B \cup \{j_k\}, \\ I_{l'} &\leftarrow 1, \\ v_{l'} &\leftarrow \alpha, \text{ und} \\ r_l &\leftarrow r_l + 1 \text{ für alle } l \text{ mit } |A_{lj_k}| \neq 0. \end{aligned}$$

Fahre mit dem nächsten k -Wert fort.

(b) Falls es ein l gibt, so daß $|A_{lj_k}| > 0.01v_l$ gilt, fahre mit dem nächsten k -Wert fort. Falls nicht, sei $\alpha = \max\{|A_{lj_k}| : I_l = 0\}$. Falls $\alpha = 0$ gilt, fahre mit dem nächsten k -Wert fort. Falls nicht, wähle l' , so daß $I_{l'} = 0$ und $\alpha = |A_{l'j_k}|$ ist. Setze

$$\begin{aligned} B &\leftarrow B \cup \{j_k\}, \\ I_{l'} &\leftarrow 1, \\ v_{l'} &\leftarrow \alpha, \text{ und} \\ r_l &\leftarrow r_l + 1 \text{ für alle } l \text{ mit } |A_{lj_k}| \neq 0. \end{aligned}$$

Schritt 3: Für jedes $i = m_1 + m_2 + 1, \dots, m$ setze $B \leftarrow B \cup \{n + m_1 + m_2 + i\}$ falls $I_i = 0$ ist. D.h., für jede noch nicht "abgedeckte" Zeile fügen wir die entsprechende künstliche Variable der Basis hinzu. \square

Die Werte v aus der obigen Konstruktion könnte man als Pseudo-Pivot-Werte bezeichnen. Durch die Bedingung $|A_{lj_i}| \leq 0.01v_l$ aus Schritt 2(b) versuchen wir, eine annähernd untere Dreiecksmatrix zu erzeugen. Beispielsweise ist leicht einzusehen, daß, bis auf Permutation der Zeilen und Spalten, B genau eine untere Dreiecksmatrix darstellt, wenn 0.01 durch 0 ersetzt wird. In ähnlicher Weise ist es leicht zu beweisen, daß alle Werte v gleich 1 bleiben, wenn v für alle Schlupfvariablen den Anfangswert 1 annimmt und in Schritt 2(a) 0.99 durch 1 ersetzt wird.

6 Rechenergebnisse

Alle Tests wurden auf eine 40-Mhz-SPARCstation 2 ausgeführt. CPLEX ist ausschließlich in C geschrieben und wurde durch Verwendung des "bundled" C -Compilers mit der Angabe '-O4 -cg89 -libmil -dalign' übersetzt.

Die CPLEX-Prozedur zum Konstruieren einer Startbasis läßt sich in folgender Weise zusammenfassen. Zunächst werden alle Schlupfvariablen in die Basis aufgenommen. Dann werden die übrigen Variablen bezüglich der Schrankenbreiten sortiert. Anschließend wird die dadurch erzeugte Reihenfolge zusammen mit einem heuristischen Zerlegungsverfahren verwendet (siehe Schritt 2 am Ende des letzten Abschnitts), um Pivotelemente zu finden, die alle Gleichungen abdecken.

Die Schlupfvariablen werden zunächst aufgrund numerischer Eigenschaften der Einheitsvektoren in die Basis aufgenommen. Zum anderen erwarten wir, daß ein wesentlicher Prozentsatz der Schlupfvariablen noch in der optimalen Basis vorhanden sein wird. Aus den Resultaten, die in Tabelle IV enthalten sind, ist festzustellen, daß durchschnittlich 52.9% der Schlupfvariablen der optimalen Basis angehören. In nur sechs der insgesamt 90 Fälle war der Prozentsatz weniger als 10%.

Das heuristische Zerlegungsverfahren war sehr erfolgreich im Vermeiden künstlicher Variablen. Im Durchschnitt wurden nur 13.1% der Gleichungen durch solche Variablen repräsentiert. In den optimalen Basen blieben fast keine künstlichen Variablen. Eine Ausnahme bildet 'cycle', wobei 130 künstliche Variablen, vermutlich aufgrund redundanter Gleichungen, noch in der optimalen Basis vorhanden waren.

Jedes der 90 Netlib-Probleme wurde mit allen vier Startbasen gelöst: künstliche Basis (K), zulässige Schlupfbasis (ZS), Schlupfbasis (S) und CPLEX-Basis (C). Alle Tests wurden in 'batch mode' auf einem alleinstehenden System durchgeführt. Keine andere Jobs liefen gleichzeitig mit den Testläufen.

Die angegebenen Laufzeiten sind 'user times', die mit der 'times' Library-Funktion gemessen wurden. Die gesamten Phase I Iterationen und Laufzeiten sind in der letzten Zeile der Tabelle zu finden. Diese Zahlen wurden durch eine kleinere Zahl schwierigerer Probleme sehr stark beeinflußt und sind nicht so aufschlußreich wie vielleicht zu hoffen wäre. Dennoch zeigen sie ganz deutlich, daß die CPLEX-Basis die künstliche Basis dominiert. Wie vielleicht zu erwarten war, kommt der größte Beitrag zur Dominanz aus der Phase I. Zusätzlich ist zu bemerken, daß der Unterschied wesentlich größer ist, als der Unterschied in den Laufzeiten. Die CPLEX-Basis ist sehr dünn besetzt (*Sparse*), aber nicht so einfach zu behandeln wie eine Basis, die aus lauter Einheitsvektoren besteht.

Um den Einfluß der schwierigeren Probleme auszugleichen, wurden der Tabelle V zwei Brüche hinzugefügt: Für jede der drei Basen (K), (ZS) und (S) wurde der Quotient aus Anzahl der Iterationen des entsprechenden Verfahrens durch die Anzahl der Iterationen für die CPLEX-Basis berechnet. Analog wurden für jedes der 90 Probleme auch die entsprechenden Laufzeitverhältnisse berechnet. Hätten wir die Quotienten auch bezüglich der Phase I Iterationen gebildet, so wäre das Ergebnis von geringer Bedeutung, da sich künstliche Variablen überwiegend in der Phase I auswirken. Durchschnittsergebnisse erscheinen in der untersten Zeile der Tabelle. Diese Zahlen könnten vielleicht irreführend sein, wenn zu viele davon wesentlich kleiner als 1 wären. Das ist aber nicht der Fall. Die CPLEX-Basis führte zu größeren Laufzeiten in 20 der 90 Fällen. Darunter sind 12 Fälle in denen die Laufzeiten um weniger als 10% größer sind als die entsprechenden Schlupfbasis-Laufzeiten. In nur einem Beispiel ist die Laufzeit um mehr als 30% höher, als dies unter Verwendung von Schlupfvariablen der Fall ist. Die durchschnittliche Verbesserung der CPLEX-Basis gegenüber der Schlupfbasis beträgt etwa 35%.

Wie in Abschnitt 1.5 schon diskutiert wurde, liefert die CPLEX-Basis in der Regel keinen grossen Gewinn für Probleme, die schwierig sind. Wenn wir die Schranken außer Acht lassen, ist die Dimension der Basis gleich der Anzahl der Zeilen. Somit

ist es als gut zu bewerten, wenn die Anzahl Iterationen ungefähr der Anzahl Zeilen entspricht, vorausgesetzt Spalten aus der optimalen Basis können nicht im voraus erraten werden, dann brauchen wir mindestens so viel Iterationen, wie die Anzahl der Zeilen, um die richtigen Variablen in die Basis aufzunehmen.

Unter den größeren Problemen ist Stocfor3 im obigen Sinne einfach. Obwohl die einzelnen Iterationen viel Zeit beanspruchen, ist die gesamte Anzahl der Iterationen kleiner als die Anzahl der Zeilen. Für dieses Problem bedeutet die CPLEX-Basis eine wesentliche Verbesserung. Die Probleme 'greenbea', 'greenbeb', 'pilots', 'fit2p' und 'pilot87' sind dagegen nicht einfach. Die CPLEX-Basis hat dementsprechend wenig Einfluß. Das Problem 'fit2d' liefert zu diesem Thema kein gutes Beispiel, weil es eine verhältnismäßig grosse Zahl an Nicht-Basisvariablen besitzt, die von einer Schranke zur anderen versetzt werden müssen. Das Verfahren zur Bestimmung der Werte der Nicht-Basisvariablen ist für alle vier Startbasen gleich.

Anerkennung

Ich möchte Robert Weismantel danken, daß er mir bei der Übersetzung dieses Aufsatzes aus dem Englischen behilflich war.

TABELLE IV

Künstliche und Schlupf-Variablen in den optimalen Basen

PROBLEM	Constraints		Initial Basis	Optimal Basis	
	Eq.	Ineq.	Artifs	Slacks	Artifs
1 afiro	8	19		7	
2 sc50b	20	30	1	2	
3 sc50a	20	30	1	4	
4 sc105	45	60	1	10	
5 kb2	16	27	3	16	
6 adlittle	15	41	4	10	
7 scagr7	84	45	7	32	
8 stocfor1	63	54	15	38	
9 blend	43	31	3	15	
10 sc205	91	114	1	9	
11 recipe*	63	24	15	22	7
12 share2b	13	83	2	43	
13 vtpbase	55	143	28	102	
14 lotfi	95	58	6	45	
15 share1b	89	28	27	23	
16 boeing2	23	143		87	
17 scorpion	280	108	73	16	
18 bore3d	214	19	36	13	6
19 scagr25	300	171	29	120	
20 sctapl	120	180		99	
21 capri	142	129	31	44	
22 brandy*	139	54	52	31	3
23 israel		174		105	
24 finnis	47	450		178	
25 gfrdpnc	548	68	113	17	1
26 scsd1	77		18		
27 etamacro	272	128	49	26	2
28 agg	36	452	6	408	
29 bandm	305		54		
30 e226	33	190	2	78	
31 scfxm1	187	143	39	75	
32 grow7	140		39		
33 standata	160	199	2	177	
34 scrs8	384	106	22	23	
35 beaconfd	140	33	1	33	
36 boeing1	9	253		129	
37 shell	534	2	99	2	1
38 standmps	268	199	22	176	
39 stair	209	147	16	6	
40 degen2	221	223	25	101	2
41 agg2	60	456	3	387	
42 agg3	60	456	7	386	
43 scsd6	147		38		
44 ship04s*	312	48	46	29	
45 seba	507	1	1	1	
46 tuff*	261	41	50	37	
47 forplan	90	70	11	47	
48 bnl1*	231	411	31	149	
49 pilot4	287	123	24	22	
50 scfxm2	374	286	73	148	

TABELLE IV

Künstliche und Schlupf-Variablen in den optimalen Basen

PROBLEM	Constraints		Initial Basis	Optimal Basis	
	Eq.	Ineq.	Artifs	Slacks	Artifs
51 grow15	300		81		
52 perold	495	130	82	25	
53 fmm800*	350	174	17	99	2
54 ship041*	312	48	46	32	
55 sctap2	470	620		379	
56 ganges	1284	25	254	24	
57 ship08s*	632	80	74	47	
58 sierra*	523	699	123	634	10
59 scfxm3	561	429	109	218	
60 ship12s*	936	106	133	50	
61 grow22	440		130		
62 stocfor2	1143	1014	147	644	
63 scsd8	397		72		
64 sctap3	620	860		516	
65 pilotwe	583	139	48	24	
66 maros*	322	522	20	252	
67 fit1p	627				
68 25fv47*	516	305	117	172	1
69 czprob	890	39	10	22	
70 ship081*	632	80	74	53	1
71 pilotnov*	677	274	182	163	9
72 nesm	480	94	8	58	
73 fit1d	1	23		12	
74 bnl2	1327	997	212	543	
75 pilotja*	645	279	154	149	
76 ship121*	936	106	121	46	
77 cycle*	1376	514	279	416	130
78 80bau3b		2262		133	
79 degen3	717	786	33	321	2
80 truss	1000		148		
81 greenbea*	2196	193	206	129	
82 greenbeb*	2196	193	202	122	3
83 d2q06c*	1507	664	153	382	2
84 woodw*	1085	13	2	9	1
85 pilots	233	1208	11	100	
86 fit2p	3000				
87 stocfor3	8829	7846		4532	
88 wood1p*	243	1	132		1
89 pilot87	233	1797	17	150	
90 fit2d	1	24		5	
Avg. % artifs/eq. rows in initial basis			13.1%		
Avg. % slacks in optimal basis				52.9%	
*Empty rows removed					

TABELLE V

Vergleich der Anfangsbasen:

Artificial(A), Feasible Slack (FS), Slack (S), CPLEX (C)
(Künstlich(A), Zulässige Schlupf (FS), Schlupf (S))

PROBLEM	Phase I Iterations				Total Iterations				Run Times (seconds)				Total Iterations Ratios			Run Time Ratios		
	A	FS	S	C	A	FS	S	C	A	FS	S	C	A/C	FS/C	S/C	A/C	FS/C	S/C
1 afiro	30	7	7	1	31	18	18	10	0.0	0.0	0.0	0.0	3.10	1.80	1.80	1.64	1.14	1.14
2 sc50b	52	0	0	0	70	48	48	32	0.1	0.1	0.1	0.1	2.19	1.50	1.50	1.33	1.18	1.18
3 sc50a	43	0	0	0	68	44	44	28	0.1	0.1	0.1	0.0	2.43	1.57	1.57	1.49	1.22	1.22
4 scl05	102	0	0	0	145	91	91	56	0.3	0.2	0.2	0.2	2.59	1.63	1.63	1.46	1.06	1.06
5 kb2	0	0	0	0	81	59	59	33	0.1	0.1	0.1	0.1	2.45	1.79	1.79	1.94	1.48	1.48
6 adlittle	78	29	29	16	116	82	82	94	0.2	0.1	0.1	0.1	1.23	0.87	0.87	1.19	0.99	0.99
7 scagr7	146	117	117	51	179	141	141	87	0.3	0.3	0.3	0.2	2.06	1.62	1.62	1.80	1.70	1.70
8 stocfor1	163	76	76	15	194	93	93	31	0.4	0.2	0.2	0.1	6.26	3.00	3.00	5.12	2.12	2.12
9 blend	85	0	0	0	100	98	98	58	0.2	0.2	0.2	0.2	1.72	1.69	1.69	0.92	1.03	1.03
10 sc205	190	0	0	0	320	190	190	129	0.9	0.6	0.6	0.7	2.48	1.47	1.47	1.43	0.96	0.96
11 recipe	68	51	51	12	94	73	73	41	0.1	0.1	0.1	0.1	2.29	1.78	1.78	1.90	1.45	1.45
12 share2b	136	69	69	76	181	107	107	111	0.4	0.2	0.2	0.2	1.63	0.96	0.96	1.46	0.92	0.92
13 vtpbase	248	100	129	87	274	130	151	114	0.7	0.3	0.4	0.3	2.40	1.14	1.32	2.15	0.88	1.12
14 lotfi	183	105	105	68	284	230	230	199	0.6	0.5	0.5	0.6	1.43	1.16	1.16	1.07	0.75	0.75
15 share1b	168	162	162	85	233	240	240	175	0.7	0.7	0.7	0.6	1.33	1.37	1.37	1.11	1.13	1.13
16 boeing2	176	122	97	108	242	188	147	178	0.6	0.5	0.4	0.5	1.36	1.06	0.83	1.22	0.96	0.70
17 scorpion	365	317	306	131	399	382	378	168	2.0	1.7	1.5	0.8	2.38	2.27	2.25	2.45	2.08	1.85
18 bore3d	213	198	198	91	235	221	221	113	0.5	0.5	0.5	0.4	2.08	1.96	1.96	1.26	1.23	1.23
19 scagr25	523	421	418	233	751	565	541	400	5.3	3.7	3.5	3.3	1.88	1.41	1.35	1.61	1.13	1.06
20 sctapl	362	199	180	121	534	350	299	207	2.1	1.4	1.1	0.9	2.58	1.69	1.44	2.39	1.55	1.22
21 capri	461	321	321	344	537	414	417	438	2.3	1.4	1.6	2.2	1.23	0.95	0.95	1.08	0.64	0.73
22 brandy	244	208	208	81	348	326	326	174	1.7	1.5	1.5	1.1	2.00	1.87	1.87	1.51	1.38	1.38
23 israel	298	8	7	7	464	172	204	204	2.4	0.9	1.0	1.0	2.27	0.84	1.00	2.32	0.89	0.99
24 finnis	670	246	255	229	966	461	489	517	5.7	2.4	2.5	3.0	1.87	0.89	0.95	1.93	0.79	0.85
25 gfrdpnc	910	846	846	293	1156	1103	1103	525	6.3	6.2	6.2	3.1	2.20	2.10	2.10	2.00	1.97	1.97
26 scsd1	99	99	99	38	181	181	181	161	0.5	0.5	0.5	0.5	1.12	1.12	1.12	0.90	0.90	0.90
27 etamacro	614	511	489	423	933	764	722	718	4.7	3.6	3.4	3.7	1.30	1.06	1.01	1.28	0.99	0.93
28 agg	585	89	77	54	640	136	116	99	3.9	0.7	0.6	0.6	6.46	1.37	1.17	6.76	1.24	1.10
29 bandm	426	426	426	135	594	594	594	283	4.2	4.2	4.2	1.9	2.10	2.10	2.10	2.18	2.18	2.18
30 e226	290	91	97	34	449	383	360	361	1.9	2.1	1.8	1.9	1.24	1.06	1.00	1.03	1.11	0.97
31 scfxm1	416	258	258	174	520	415	415	324	2.2	1.7	1.7	1.5	1.60	1.28	1.28	1.44	1.13	1.13
32 grow7	0	0	0	0	309	309	309	275	2.2	2.2	2.2	2.5	1.12	1.12	1.12	0.90	0.90	0.90
33 standata	440	179	179	49	522	223	223	153	1.8	0.8	0.8	0.7	3.41	1.46	1.46	2.54	1.11	1.11
34 scrs8	857	879	860	339	1117	1148	1168	561	7.3	7.3	7.7	4.6	1.99	2.05	2.08	1.60	1.60	1.69
35 beaconfd	161	128	128	0	189	163	163	31	0.4	0.4	0.4	0.1	6.10	5.26	5.26	3.58	3.25	3.25
36 boeing1	572	334	341	226	997	710	673	563	4.5	3.4	3.1	3.2	1.77	1.26	1.20	1.38	1.06	0.95
37 shell	619	615	615	260	774	793	793	494	3.8	3.8	3.8	3.0	1.57	1.61	1.61	1.26	1.29	1.29
38 standmps	585	316	313	209	698	455	443	317	2.7	1.8	1.8	1.6	2.20	1.44	1.40	1.70	1.10	1.10
39 stair	492	386	377	227	643	505	568	417	9.2	6.1	7.3	6.6	1.54	1.21	1.36	1.39	0.92	1.11
40 degen2	674	673	659	547	1125	1107	1041	895	11.9	12.7	11.4	9.9	1.26	1.24	1.16	1.19	1.28	1.34
41 agg2	642	72	72	34	769	215	215	125	4.6	1.0	1.0	0.7	6.15	1.72	1.72	6.72	1.53	1.53
42 agg3	614	71	71	36	732	241	241	134	4.7	1.3	1.3	0.7	5.46	1.80	1.80	6.65	1.77	1.77
43 scsd6	221	221	221	114	589	589	589	427	1.7	1.7	1.7	1.4	1.38	1.38	1.38	1.23	1.23	1.23
44 ship04s	413	348	348	182	537	484	484	276	2.0	1.7	1.7	1.2	1.95	1.75	1.75	1.68	1.45	1.45
45 seba	659	659	659	138	853	853	853	234	5.0	5.0	5.0	1.4	3.65	3.65	3.65	3.56	3.56	3.56
46 tuff	377	384	384	366	474	428	428	427	1.9	2.0	2.0	2.4	1.11	1.00	1.00	0.83	0.83	0.83
47 forplan	326	327	327	166	465	472	472	336	1.5	1.6	1.6	1.3	1.38	1.40	1.40	1.14	1.21	1.21
48 bn1l	2241	2147	2147	2401	2511	2384	2384	2642	33.0	30.4	30.4	36.6	0.95	0.90	0.90	0.90	0.83	0.83
49 pilot4	1063	851	729	322	1744	1549	1334	1015	26.5	22.4	18.8	14.6	1.72	1.53	1.31	1.82	1.54	1.29
50 scfxm2	858	602	602	400	1116	884	884	669	9.3	6.3	6.3	5.3	1.67	1.32	1.32	1.74	1.17	1.17

TABELLE V

Vergleich der Anfangsbasen:

Artificial(A), Feasible Slack (FS), Slack (S), CPLEX (C)
 (Künstlich(A), Zulässige Schlupf (FS), Schlupf (S))

PROBLEM	Phase I Iterations				Total Iterations				Run Times (seconds)				Total Iterations Ratios			Run Time Ratios			
	A	FS	S	C	A	FS	S	C	A	FS	S	C	A/C	FS/C	S/C	A/C	FS/C	S/C	
51 grow15	0	0	0	0	849	849	849	621	11.8	11.8	11.8	11.3	1.37	1.37	1.37	1.04	1.04	1.04	
52 perold	3614	3298	3428	1370	4629	4416	4679	2457	100.1	94.3	102.9	52.3	1.88	1.80	1.90	1.91	1.80	1.97	
53 mm800	1106	1058	917	541	1290	1220	1064	683	7.4	7.1	5.9	4.5	1.89	1.79	1.56	1.63	1.58	1.30	
54 ship041	383	360	360	223	565	525	525	353	2.2	2.1	2.1	1.6	1.60	1.49	1.49	1.32	1.30	1.30	
55 sctap2	1356	679	611	252	2328	1655	1505	548	29.0	20.1	18.2	4.7	4.25	3.02	2.75	6.24	4.31	3.92	
56 ganges	1591	1579	1579	358	1714	1727	1727	605	21.1	21.7	21.7	9.6	2.83	2.85	2.85	2.21	2.27	2.27	
57 ship08a	772	697	697	202	1073	955	955	429	7.5	6.7	6.7	3.6	2.50	2.23	2.23	2.07	1.85	1.85	
58 sierra	2175	577	577	206	3068	841	841	455	36.3	7.2	7.2	4.4	6.74	1.85	1.85	8.20	1.62	1.62	
59 scfxm3	1240	906	906	596	1681	1280	1280	1050	18.8	12.3	12.3	11.6	1.60	1.22	1.22	1.62	1.06	1.06	
60 ship12s	1108	1102	1102	316	1346	1287	1287	528	12.6	11.5	11.5	6.3	2.55	2.44	2.44	1.99	1.82	1.82	
61 grow22	0	0	0	0	1224	1224	1224	892	23.5	23.5	23.5	21.8	1.37	1.37	1.37	1.08	1.08	1.08	
62 stocfor2	2657	1503	1503	411	3508	2182	2182	1061	78.6	46.0	46.0	26.5	3.31	2.06	2.06	2.97	1.74	1.74	
63 scsd8	1039	1039	1039	468	2023	2023	2023	1320	12.0	12.0	12.0	8.2	1.53	1.53	1.53	1.47	1.47	1.47	
64 sctap3	1703	951	869	337	2896	2273	2221	786	46.5	40.0	39.1	9.2	3.68	2.89	2.83	5.04	4.33	4.24	
65 pilotwe	1618	1626	1626	675	3533	3737	3737	2652	75.5	83.4	83.4	76.2	1.33	1.41	1.41	0.99	1.09	1.09	
66 maros	1215	742	742	760	1912	1367	1367	1476	22.3	15.7	15.7	18.8	1.30	0.93	0.93	1.19	0.84	0.84	
67 fit1p	837	837	837	450	1249	1249	1249	809	16.8	16.8	16.8	11.2	1.54	1.54	1.54	1.49	1.49	1.49	
68 25fv47	1760	1414	1377	979	3734	3438	3439	2679	94.1	76.3	77.6	65.1	1.39	1.28	1.28	1.44	1.17	1.19	
69 czprob	1421	1245	1245	547	2321	2083	2083	1178	17.8	16.6	16.6	10.8	1.97	1.77	1.77	1.65	1.53	1.53	
70 ship081	771	732	732	326	1387	1283	1283	804	10.3	9.6	9.6	6.6	1.73	1.60	1.60	1.56	1.46	1.46	
71 pilotnov	2864	2701	2614	2262	3161	3228	3237	2767	69.5	74.8	75.8	71.2	1.14	1.17	1.17	0.98	1.05	1.06	
72 nesm	1777	1951	1951	961	5029	5307	5307	4094	36.6	38.2	38.2	31.0	1.23	1.30	1.30	1.18	1.23	1.23	
73 fit1d	0	0	0	0	1161	989	989	947	4.7	4.0	4.0	3.9	1.23	1.04	1.04	1.21	1.05	1.05	
74 bnl2	4640	4051	4051	2858	6819	5564	5564	4422	250.5	186.0	186.0	148.2	1.54	1.26	1.26	1.69	1.25	1.25	
75 pilotja	4583	3259	3543	2011	7065	5459	6004	4365	207.1	157.7	182.0	139.0	1.62	1.25	1.38	1.49	1.13	1.31	
76 ship12l	1220	1147	1147	579	1648	1502	1502	969	18.3	15.8	15.8	12.5	1.70	1.55	1.55	1.46	1.26	1.26	
77 cycle	0	0	0	0	2416	2034	2034	1207	38.6	32.2	32.2	27.1	2.00	1.69	1.69	1.42	1.19	1.19	
78 80bau3b	4056	2195	2227	2227	11107	10173	10635	10635	173.8	163.0	166.2	166.2	1.04	0.96	1.00	1.05	0.98	1.00	
79 degen3	3188	2964	3107	2873	5471	4860	5043	4633	254.7	202.0	204.5	201.4	1.18	1.05	1.09	1.26	1.00	1.02	
80 truss	3482	3482	3482	2205	11511	11511	11511	10622	206.8	206.8	206.8	192.4	1.08	1.08	1.08	1.07	1.07	1.07	
81 greenbea	3688	3473	3473	2966	7582	8115	8115	7973	282.8	320.3	320.3	341.5	0.95	1.02	1.02	0.83	0.94	0.94	
82 greenbeb	3695	3661	3661	3171	6717	7037	7037	6121	250.5	268.5	268.5	241.8	1.10	1.15	1.15	1.04	1.11	1.11	
83 d2q06c	4778	3770	2944	1427	10922	10258	9832	9649	684.4	668.6	655.8	684.6	1.13	1.06	1.02	1.00	0.98	0.96	
84 woodw	1099	1026	1113	369	2791	2160	2440	1701	38.1	28.6	31.3	25.5	1.64	1.27	1.43	1.50	1.12	1.23	
85 pilots	5495	4552	4530	4034	8599	8039	7698	7402	822.7	833.1	757.8	768.3	1.16	1.09	1.04	1.07	1.08	0.99	
86 fit2p	5271	5271	5271	3076	12585	12585	12585	10030	796.0	796.0	796.0	690.5	1.25	1.25	1.25	1.15	1.15	1.15	
87 stocfor3	21140	12220	12220	5015	28698	18443	18443	10813	4288.5	2762.6	2762.6	1817.5	2.65	1.71	1.71	2.36	1.52	1.52	
88 wood1p	211	211	211	222	459	459	459	749	8.7	8.7	8.7	14.0	0.61	0.61	0.61	0.62	0.62	0.62	
89 pilot87	5213	4211	4633	4053	10835	10296	9995	9696	2179.0	2176.9	2105.1	2087.2	1.12	1.06	1.03	1.04	1.04	1.01	
90 fit2d	2	0	0	0	12593	12041	12041	11726	131.1	125.0	125.1	120.0	1.07	1.03	1.03	1.09	1.04	1.04	
TOTALS	122926	94758	94384	58249	226982	197485	197434	157966	11572	9752	9627	8276	2.09	1.55	1.54	1.90	1.36	1.35	
AVERAGES																			

Literaturverzeichnis

- [1] Adler, I., N. Karmarkar, M. G. C. Resende, und G. Veiga (1989), An implementation of Karmarkar's algorithm for linear programming, *Mathematical Programming* 44 No. 3, 297–336.
- [2] Chvátal, Vašek (1983), *Linear Programming*, W. H. Freeman, New York.
- [3] Dantzig, G. B. (1951), Maximization of a linear function of variables subject to linear inequalities, in : *Activity Analysis of Production and Allocation*, R.C. Koopmans (ed.), Wiley, New York, 1951, 339–347.
- [4] Golub, G. H. und C. F. Van Loan (1983), *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Maryland.
- [5] Optimization Subroutine Library: Guide and Reference (1990), IBM Corporation, April 1990.
- [6] Karmarkar, N. (1984), A new polynomial time algorithm for linear programming, *Combinatorica* 4, 375–395.
- [7] Lustig, I. (1987), An analysis of an available set of linear programming test problems, Technical Report SOL 87-11, Systems Optimization Lab, Stanford University, Stanford, California.
- [8] Lustig, I. (1987), An analysis of an available set of linear programming test problems, *Comput. Opns Res.* 16 173–184.
- [9] Lustig, I. J., R. E. Marsten, und D. F. Shanno (1990), On implementing Mehrotra's predictor-corrector interior-point method for linear programming, Technical Report SOR 90-03, Department of Civil Engineering and Operations Research, Princeton University, Princeton, NJ.
- [10] Markowitz, H. M. (1957), The elimination form of the inverse and its application to linear programming, *Management Science* 3 255–269.
- [11] Murtagh, Bruce A. und Michael Saunders (1987), Minos 5.1 User's Guide, Technical Report SOL 83-20R, Systems Optimization Lab, Stanford University, Stanford, California.
- [12] Suhl, Uwe H. und Leena M. Suhl (1990), Computing sparse LU factorizations for large-scale linear programming bases, *ORSA Journal on Computing* 2 325–335.