FernUniversität in Hagen

Veröffentlichungen des Lehrstuhls für Betriebswirtschaftslehre, insbesondere Investitionstheorie und Unternehmensbewertung

Herausgeber:

Professor Dr. habil. Thomas Hering

Nr. 15

Thomas Hering

Lineare Optimierung und Sensitivitätsanalyse mit LinSen

Hagen (Westf.)

2018

Lineare Optimierung und Sensitivitätsanalyse mit LinSen

Nachfolgend wird als Faksimile die vor über einem Vierteljahrhundert geschriebene Diplomarbeit des Verfassers erstmals veröffentlicht.

HERING, TH.: Möglichkeiten und Probleme der EDV-gestützten Sensitivitätsanalyse linearer Programme, Diplomarbeit Münster (Westf.) 1990.

Motivation dafür war es, erstens die Zitierfähigkeit des Inhalts herzustellen und zweitens den vollständigen Quelltext von LinSen zu dokumentieren. Mit diesem (nach einer Vorläuferversion von 1988 in BASIC) 1990 in der Sprache TURBO-Pascal selbstverfaßten und bis 1994 weiterentwickelten EDV-Programm habe ich fast zwanzig Jahre lang alle Zahlenbeispiele für meine Publikationen konstruiert, in denen Anwendungen der linearen Optimierung und Dualitätstheorie sowie gelegentlich auch der gemischt-ganzzahligen linearen Optimierung eine Rolle spielten. Auch meine Koautoren und andere befreundete Wissenschaftler (sowie etliche interessierte Studenten) nutzten lange Zeit das (nach damaligen Maßstäben) besonders benutzerfreundliche und auf die einfache Handhabung kleiner Beispiele für Forschungs- und Lehrzwecke ausgerichtete Programm. Eine dem Fortschritt der Betriebssysteme angepaßte Version mit modernerer Benutzeroberfläche erreichte bis 2001 nur noch das Versuchsstadium, da die dafür einsetzbaren geringen Ressourcen (eine studentische Hilfskraft mit Informatikkenntnissen) angesichts der schnellen technischen Entwicklung unzureichend bleiben mußten und die Zeit der „selbstgestrickten“ Ein-Mann-Programme (jedenfalls für mich) vorbei war.

Die hier abgedruckte Diplomarbeit enthält naturgemäß die Version LinSen 1.0 von 1990. Kurz vor meiner Promotion entstand schließlich die perfektionierte und sinnvoll erweiterte, seitdem fünfzehn Jahre lang zuverlässig im Einsatz befindliche und nicht mehr angerührte Endversion 1.5 von 1994. Sie ist dokumentiert in:

HERING, TH.: Quelltext von LinSen 1.5 (1994), Veröffentlichungen des Lehrstuhls für Betriebswirtschaftslehre, insbesondere Investitionstheorie und Unternehmensbewertung, Fern-Universität Hagen, Hrsg. Th. Hering, Nr. 16, Hagen (Westf.) 2018.

Prof. Dr. Thomas Hering, Fern-Universität Hagen, Universitätsstraße 11, 58084 Hagen (Westf.)
Hering@FernUni-Hagen.de, http://www.fernuni-hagen.de/hering

D i p l o m a r b e i t

im Fach Unternehmensforschung

Möglichkeiten und Probleme der EDV-gestützten Sensitivitätsanalyse linearer Programme

Themensteller: Prof. Dr. Wolfgang von Zwehl

Ausgabetermin: 28. 6. 1990
Abgabetermin: 20. 9. 1990

vorgelegt dem Prüfungsamt für wirtschaftswissenschaftliche Prüfungen der Westfälischen Wilhelms-Universität Münster

von: Thomas H e r i n g

I n h a l t s v e r z e i c h n i s

Abkürzungsverzeichnis

BV	Basisvariable
EDV	elektronische Datenverarbeitung
f.	folgende (Seite)
ff.	fortfolgende (Seiten)
i.a.	im allgemeinen
KB	Kilobyte
LinSen	Lineare Optimierung und Sensibilitätsanalyse
LO	lineare Optimierung
LO-Problem	lineares Optimierungsproblem
LP	lineare Programmierung
max.	maximiere
ME	Mengeneinheiten
min.	minimiere
n.	nein
NBV	Nichtbasisvariable
OR	Operations Research
per def.	per definitionem
RS	rechte Seite(n)
Var.	Variable
WiSt	Wirtschaftswissenschaftliches Studium
w.z.z.w.	was zu zeigen war
ZE	Zeiteinheiten
ZfB	Zeitschrift für Betriebswirtschaft

Symbolverzeichnis

$\mathbf{0}$	Nullvektor
$\mathbf{A}$	Koeffizientenmatrix
$\mathbf{A}'$	erweiterte Koeffizientenmatrix
$\mathbf{a}^i$	i-ter Zeilenvektor von $\mathbf{A}$
a_{ij}	Tableaukoeffizient
$\mathbf{a}_j$	j-ter Spaltenvektor von $\mathbf{A}$
$\mathbf{b}$	Vektor der rechten Seiten
$\mathbf{b}'$	Vektor der Basisvariablenwerte
b_0	Absolutglied der Zielfunktion
b_0'	der Basislösung zugeordneter Wert von x_0
$\mathbf{c}$	Vektor der Zielfunktionsbeiträge

c'	Vektor der Zielfunktionskoeffizienten
D	Basismatrix
d	Vektor der Zielfunktionsbeiträge der Basisvariablen
δ	Änderungsvektor
D^{-1}	Inverse der Basismatrix
E	Einheitsmatrix
m	Anzahl der Restriktionen
n	Anzahl der Strukturvariablen
$\mathbb{R}^m$	arithmetischer Vektorraum (der Dimension m) über dem Körper der reellen Zahlen
s	Vektor der Schlupfvariablen
T	transponiert
t	Änderungsparameter
t_j	Änderungsparameter
v	Vektor der Basisvariablen
x	Vektor der Strukturvariablen
x^*	(optimale) (primale) Lösung
x'	Vektor der Struktur- und Schlupfvariablen
x_0	Zielfunktionswert
x_k	künstliche Variable
y	Vektor der dualen Strukturvariablen
y^*	(optimale) duale Lösung
y_0	dualer Zielfunktionswert
z	Vektor der dualen Schlupfvariablen

Abbildungsverzeichnis

1. Gegenstand der Arbeit

Betriebswirtschaftliche Planungsdaten sind häufig mit Unsicherheit behaftet, denn jede Planung ist notwendigerweise zukunftsgerichtet. Viele Größen schwanken im Zeitablauf, unterliegen Wahrscheinlichkeitsverteilungen oder sind nicht bzw. nur ungenau quantifizierbar. Andere können beeinflußt werden und hängen von zukünftigen Entscheidungen der Unternehmung ab. Oftmals sind nur Prognosen oder Schätzwerte verfügbar.[1]
In einer solchen Situation ist die Frage nach der Empfindlichkeit der Planungsergebnisse im Hinblick auf Veränderungen der Ausgangsdaten von besonderer Bedeutung. Ihre Beantwortung ist Gegenstand der Sensibilitätsanalyse.[2] Im Prinzip kann jedes Entscheidungsmodell einer derartigen Analyse unterzogen werden.[3] Die vorliegende Arbeit befaßt sich jedoch ausschließlich mit der Sensibilitätsanalyse linearer Optimierungsprobleme und ihrer EDV-technischen Umsetzung. Ein lineares Optimierungsproblem (LO-Problem) fordert die Maximierung oder Minimierung einer linearen "Zielfunktion" mehrerer Variabler, welche durch ein System linearer Ungleichungen beschränkt wird.[4]

1 Vgl. Kern, Empfindlichkeit, S. 49, Ellinger, OR, S. 89, Hillier/Lieberman, OR, S. 146.
2 Vgl. Dinkelbach, Analysen, S. 25, Runzheimer, OR, S. 95 f., Dürr/Kleibohm, OR, S. 81. In der Literatur hat sich der Begriff "Sensitivitätsanalyse" durchgesetzt. Müller-Merbach kritisiert ihn treffend als "buchstabengetreue, aber nicht wortgetreue Übersetzung" von "Sensitivity Analysis". Müller-Merbach, OR, S. 150.
3 Vgl. Dinkelbach, Analysen, S. 5 ff., 26 ff.
4 Vgl. Dantzig, LP und Erweiterungen, S. VII. Oft wird ein LO-Problem als "lineares Programm" bezeichnet, was zu Verwechslungen mit dem Begriff "EDV-Programm" führen kann. Bezeichnungen wie "lineare Programmierung" sollten deshalb vermieden werden. Vgl. Müller-Merbach, OR, S. 90 f., v. Zwehl, LP, S. 360, Zimmermann, OR, S. 48.

Die *Sensibilitätsanalyse* umfaßt Berechnungen, die sich an die Lösung einer linearen Optimierungsaufgabe anschließen. Ihr Ziel ist die Klärung folgender Fragen:[5]

1. Innerhalb welcher Grenzen dürfen Koeffizienten des LO-Problems schwanken, ohne daß sich die *Struktur* der optimalen Lösung ändert ?[6]
2. Wie wirken sich Änderungen der Ausgangsdaten auf die Lösung des LO-Problems aus, d.h.:
   Wie lautet die neue optimale Lösung (falls sie existiert) ?

Beide Fragen sollen unter Ausnutzung der Kenntnis des bisherigen Endtableaus beantwortet werden, ohne die abgeänderte Aufgabe von Anfang an neu zu lösen.[7]

Die hier gegebene Definition der Sensibilitätsanalyse ist sehr umfassend. Viele Autoren ordnen ihr vornehmlich die erste, andere nur die zweite Fragestellung zu.[8] Letztere unterscheidet sich von der parametrischen Optimierung dadurch, daß das geänderte Problem, ausgehend vom bisherigen Endtableau, für gegebene numerische Werte der Koeffizienten gelöst wird. Man untersucht also nicht allgemein und systematisch, für welche Parameterwerte welche Basislösung optimal ist.[9]

5 Vgl. Kern, Empfindlichkeit, S. 60 ff.
6 Der Übergang zu einer anderen Lösungsstruktur (Menge der Basisvariablen) wird auch als qualitative Änderung der Lösung bezeichnet. Vgl. Dinkelbach, Analysen, S. 48, 71 ff.
7 Vgl. Churchman/Ackoff/Arnoff, OR, S. 291.
8 Vgl. z.B. einerseits Dinkelbach, Analysen, S. 71 ff., Gal, LO, S. 195 ff., andererseits Hillier/Lieberman, OR, S. 150, Garvin, Introduction, S. 49-61, Beale, Mathematical Programming, S. 102 ff.
9 Die Abgrenzung der Begriffe Sensibilitätsanalyse und parametrische Optimierung ist in der Literatur nicht einheitlich. Vgl. z.B. Runzheimer, OR, S. 96, Wöhe, Allgemeine BWL, S. 148, Ellinger, OR, S. 90.

Die Sensibilitätsanalyse ermöglicht Aussagen über die Stabilität der gefundenen Lösung. Insbesondere können diejenigen Koeffizienten identifiziert werden, welche nur geringfügig schwanken dürfen, wenn die bisherige Lösungsstruktur optimal bleiben soll. Es empfiehlt sich dann, diese Parameter genauer zu schätzen oder gezielt zu beeinflussen.[10] Weiterhin lassen sich Basislösungen finden, die für möglichst viele denkbare Datenkonstellationen gute Ergebnisse liefern.[11] Schließlich erweist sich die Sensibilitätsanalyse als nützliches Instrument, um tatsächlich aufgetretene Datenänderungen auf einfache Weise nachträglich zu berücksichtigen.

Die vorliegende Arbeit beschreibt grundlegende Verfahren der Sensibilitätsanalyse von LO-Problemen. Anhand des vom Verfasser erstellten EDV-Programms *LinSen*[12] soll demonstriert werden, wie sich die Sensibilitätsanalyse in ein Programmsystem zur linearen Optimierung integrieren läßt und wie bequem sich ihre praktische Anwendung gestaltet.
Nach der Formulierung von Effizienzkriterien der EDV-Umsetzung werden im folgenden die Grundzüge der linearen Optimierung dargestellt. Die darauf aufbauenden Abschnitte behandeln die Theorie der Sensibilitätsanalyse sowie zum Abschluß einige Erweiterungen. Der Anhang enthält neben der Programmdokumentation von LinSen ein Zahlenbeispiel zur Veranschaulichung der Ergebnisse.

10 Vgl. Dantzig, LP und Erweiterungen, S. 306 f., Dinkelbach, Sensitivitätsanalysen, S. 246.
11 Vgl. Hillier/Lieberman, OR, S. 82 f.
12 Der Name bedeutet "*Lin*eare Optimierung und *Sen*sibilitätsanalyse".

2. Elemente einer EDV-gestützten Sensibilitätsanalyse linearer Optimierungsprobleme

2.1 Effizienzkriterien eines EDV-Programms zur Sensibilitätsanalyse von LO-Problemen

Eine EDV-gestützte Konzeption zur Sensibilitätsanalyse muß gewissen Anforderungen genügen, die an jede Programmentwicklung gestellt werden. Folgende allgemeine Effizienzkriterien sind zu beachten:[13]
Zuverlässigkeit: Das Programm muß in allen denkbaren Situationen richtige Lösungen liefern. Daher wird im folgenden großer Wert auf vollständige und exakte Fallunterscheidungen gelegt.
Selbstdokumentation: Das Programm LinSen ist nach dem Prinzip der "strukturierten Programmierung" in der Sprache TURBO-Pascal verfaßt. Strukturierte Programmiersprachen erlauben es, ein komplexes Problem in viele kleine, überschaubare Module zu zerlegen, die unmittelbar in schematische Ablaufpläne (Struktogramme) überführbar sind.[14] Bei Verwendung aussagefähiger Variablen- und Prozedurnamen ergibt sich auf diese Weise ein übersichtlicher und leicht zu bearbeitender Quelltext.
Benutzerfreundlichkeit: Das Programm muß einfach zu bedienen sein. Eingabefehler des Benutzers sind umgehend abzufangen.[15] Dateneingabe und Ergebnisausgabe sollen in übersichtlicher Form erfolgen.[16]
Eine spezielle Anforderung ergibt sich aus dem Wesen der Sensibilitätsanalyse: Der aktuelle Stand der Ausgangsdaten muß jederzeit abrufbar sein, damit nicht nach mehreren Änderungen der Überblick verlorengeht.[17]

13 Vgl. Grob/Reepmeyer, EDV, S. 71 ff.
14 Vgl. ebenda, S. 92 ff., Biethahn, EDV, S. 151 ff., 156 ff., Steiner/Czerwinski, Lexikon, S. 439.
15 Vgl. Stahlknecht/Ohmann, LP auf dem PC, S. 1 f.
16 Vgl. Gal, LO, S. 250.
17 Vgl. in LinSen die Option "Aufgabe ausgeben".

2.2 Theorie und Rechentechnik der linearen Optimierung

2.2.1 Mathematische Begriffe und Grundlagen

Zur Lösung von LO-Problemen sind verschiedene Verfahren vorgeschlagen worden.[18] Im folgenden wird die auf Dantzig zurückgehende Simplexmethode zugrunde gelegt. Zunächst sind einige wesentliche Begriffe zu definieren.
Ein *LO-Problem* sei gegeben durch:[19]

$$\begin{aligned} \max.\ x_0;\ x_0 &:= \mathbf{c}^T\mathbf{x} + b_0 \\ \mathbf{Ax} &\leq \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned} \tag{1}$$

mit $\mathbf{x}:=(x_1,...,x_n)^T$ als Vektor der *Strukturvariablen*, $\mathbf{b}:=(b_1,...,b_m)^T$ als Vektor der *rechten Seiten* (RS), $\mathbf{c}:=(c_1,...,c_n)^T$ als Vektor der *Zielfunktionsbeiträge* und $\mathbf{A}:=(a_{ij})$, $1 \leq i \leq m$, $1 \leq j \leq n$ als *Koeffizientenmatrix* mit m Zeilenvektoren $\mathbf{a}^i := (a_{i1},...,a_{in})$ und n Spaltenvektoren $\mathbf{a}_j := (a_{1j},...,a_{mj})^T$. Das Problem (1) besteht darin, einen Vektor $\mathbf{x}$ zu finden, der den *Zielfunktionswert* x_0 unter Einhaltung der m *Restriktionen* $\mathbf{a}^i\mathbf{x} \leq b_i$ sowie der n *Nichtnegativitätsbedingungen* $x_j \geq 0$ maximiert. Definiert man einen Vektor $\mathbf{s}$ der *Schlupfvariablen* durch $\mathbf{s}:=(s_1,...,s_m)^T := \mathbf{b}-\mathbf{Ax} \geq \mathbf{0}$, so läßt sich (1) in Gleichungsform schreiben und lautet mit $\mathbf{E}$ als m-reihiger Einheitsmatrix und $\mathbf{0}$ als Nullvektor der jeweiligen Dimension wie folgt:

$$\begin{aligned} \max.\ x_0;\ x_0 &:= \mathbf{c}^T\mathbf{x} + b_0 \\ \mathbf{Ax} + \mathbf{Es} &= \mathbf{b} \\ \mathbf{x} \geq \mathbf{0},\ \mathbf{s} &\geq \mathbf{0} \end{aligned} \tag{2}$$

18 Vgl. Müller-Merbach, OR, S. 89, Krekó, Lehrbuch, S. 12 f., 302 ff., Joksch, LP, S. 143 ff. Zu neueren Entwicklungen vgl. Gal, LO, S. 251 f.

19 Vgl. Collatz/Wetterling, Optimierungsaufgaben, S. 4 f.

Jedes LO-Problem läßt sich in die Form (1) bzw. (2) überführen:[20] Minimierungsaufgaben werden durch Multiplikation der Zielfunktion mit -1 zu Maximierungsaufgaben. Liegen Bedingungen der Form $a^i\mathbf{x} \geq b_i$ vor, so multipliziert man sie ebenfalls mit -1. Gleichungsrestriktionen $a^i\mathbf{x} = b_i$ ersetze man durch $a^i\mathbf{x} \leq b_i$ und $a^i\mathbf{x} \geq b_i$, Variable $x_j \leq 0$ durch $x_j' := -x_j \geq 0$, nicht vorzeichenbeschränkte x_j durch $x_j' - x_j''$ mit $x_j', x_j'' \geq 0$.
Schreibt man (2) in Tabellenform, so ergibt sich das *Ausgangstableau* der Simplexmethode:[21]

BV	$\mathbf{x}^T$	$\mathbf{s}^T$	RS
x_0	$-\mathbf{c}^T$	$\mathbf{0}^T$	b_0
$\mathbf{s}$	$\mathbf{A}$	$\mathbf{E}$	$\mathbf{b}$

(3)

Es liegt in *kanonischer Form* vor. Diese ist dadurch gekennzeichnet, daß der Teil der Matrix, welcher in Tableau (3) durch **A** und **E** besetzt ist, alle m Einheitsvektoren des R^m als Spaltenvektoren enthält und die Koeffizienten der x_0-Zeile in den betreffenden Spalten null sind.[22] Ordnet man die Einheitsvektoren so um, daß sie die Matrix **E** bilden, so kann man die zu ihren Spalten gehörenden Variablen in dieser Reihenfolge zu einem Vektor **v** zusammenfassen, dem Vektor der *Basisvariablen* (BV). Die den Basisvariablen (Variablen von **v**) zugeordneten Spalten der Matrix $(\mathbf{A}|\mathbf{E})$ sind linear unabhängig und bilden eine *Basis* des R^m.[23] Sie werden in der durch **v** gegebenen Reihenfolge zur *Basismatrix* **D** zusammengefügt. Analog bildet man aus den Zielfunktionsbeiträgen der Basisvariablen einen Vektor **d**.[24] Alle

20 Vgl. Burkard, Methoden, S. 15 f.
21 Vgl. Krekó, Lehrbuch, S. 224.
22 Vgl. Dinkelbach, Analysen, S. 49 f.
23 Vgl. Dantzig, LP und Erweiterungen, S. 95.
24 Der Zielfunktionsbeitrag einer Schlupfvariablen ist null.

nicht in v enthaltenen Variablen heißen *Nichtbasisvariable* (NBV) des betrachteten Tableaus.[25]
Greift man aus (A|E) m linear unabhängige Spalten als Matrix D heraus, so lautet das dieser Basis zugeordnete Tableau in kanonischer Form wie folgt:[26]

BV	x^T	s^T	RS
x_0	$d^TD^{-1}A-c^T$	d^TD^{-1}	$d^TD^{-1}b+b_0$
v	$D^{-1}A$	D^{-1}	$D^{-1}b$

(4)

Tableau (4) ergibt sich aus (3) durch Äquivalenzumformungen, die sich als elementare Zeilentransformationen deuten lassen.[27]
Setzt man in (4) alle Nichtbasisvariablen gleich null, läßt sich dank der kanonischen Form eine spezielle Lösung des Systems der Restriktionsgleichungen ($Ax + s = b$) sofort ablesen: $v = D^{-1}b$ mit $x_0 = d^TD^{-1}b+b_0$. Sie heißt *Basislösung* zur Basis D.[28] Eine äquivalente Umformung von (4), die unter Beibehaltung der kanonischen Form einen Vektor der Matrix D durch einen anderen Vektor aus (A|E) ersetzt, wird als *Basistausch* oder *Basiswechsel* bezeichnet. Ein Basistausch verändert die Menge der Basisvariablen (Vektor v) und damit die Struktur der Basislösung.[29]

Das Tableau (4) heißt *zulässig*, wenn $D^{-1}b \geq 0$ gilt. Es heißt *optimal*, wenn für den Vektor der *Zielfunktionskoeffizienten* $(d^TD^{-1}A-c^T \mid d^TD^{-1})^T \geq 0$ erfüllt ist.

25 x_0 kann in jedem Tableau als Basisvariable angesehen werden.
26 Zur Herleitung vgl. Krekó, Lehrbuch, S. 225.
27 Vgl. Fischer, Lineare Algebra, S. 94 f.
28 Vgl. Dantzig, LP und Erweiterungen, S. 95.
29 Vgl. S. 2, Fußnote 6 dieser Arbeit.

Ein Vektor **x** ist eine *zulässige Lösung* des LO-Problems (1), falls er den Bedingungen $\mathbf{Ax} \leq \mathbf{b}$ und $\mathbf{x} \geq 0$ genügt. Als *optimale Lösung* bezeichnet man jede zulässige Lösung, die x_0 maximiert.[30]

Eine optimale Lösung des Problems (1) ist gefunden, wenn ein zulässiges und optimales Tableau (4) vorliegt. Die zugehörige Basislösung maximiert dann x_0.[31] Mit der Simplexmethode erhält man nach einer endlichen Anzahl von Rechenschritten ein solches *Optimaltableau*, oder man stellt fest, daß die Aufgabe nicht lösbar ist.[32] Es gibt also stets eine *optimale Basislösung*, wenn überhaupt eine optimale Lösung von (1) existiert.
Das Lösungsverfahren kann sich deshalb auf die Betrachtung von Basislösungen beschränken.[33]

2.2.2 Primaler Simplexalgorithmus

Der primale Simplexalgorithmus geht von einem zulässigen Tableau (3) bzw. (4) aus.[34] Seine Grundidee besteht darin, x_0 unter Beibehaltung der Zulässigkeit schrittweise durch Basiswechsel zu erhöhen, bis man ein optimales Tableau erreicht oder feststellt, daß die Zielfunktion nach oben nicht beschränkt ist.

30 Man beachte den Unterschied: Ein Tableau kann zugleich optimal und nicht zulässig sein, aber eine optimale Lösung ist per def. zulässig. Vgl. auch Dinkelbach, Analysen, S. 48.
31 Beweis: Nach Voraussetzung sind die Variablenwerte der Basislösung nichtnegativ und die Restriktionen und Nichtnegativitätsbedingungen von (2) also erfüllt. Aus der Nichtnegativität der Zielfunktionskoeffizienten folgt sodann $x_0 = (c^T - d^T D^{-1} A)x - d^T D^{-1} s + d^T D^{-1} b + b_0 \leq d^T D^{-1} b + b_0$, q.e.d.
Vgl. Dantzig, LP und Erweiterungen, S. 111 f.
32 Vgl. ebenda, S. 117, 139 ff.
33 Vgl. Dinkelbach, Analysen, S. 49.
34 Das Ausgangstableau (3) entspricht einem Tableau (4) mit $D=D^{-1}=E$, $d=0$, $v=s$.

Gegeben sei ein Tableau (4) in folgender Form:

BV	x'^T	RS
x_0	c'^T	b_0'
v	A'	b'

(5)

mit $b' \geq 0$. A', b', c', x' und b_0' sind zusammenfassende Schreibweisen für die einzelnen Elemente der Matrix (4).[35]

Der Algorithmus vollzieht sich in drei Schritten:[36]

Schritt 1: Eine Spalte mit dem absolut größten negativen c_j' wird zur *Pivotspalte* bestimmt. Sie habe den Index s. Gibt es kein negatives c_j', ist das Optimaltableau gefunden.

Schritt 2: Wegen $c_s'<0$ wächst x_0, wenn die zur Spalte s gehörende Nichtbasisvariable x_s' einen positiven Wert annimmt. Da das Tableau zulässig bleiben muß, sind dem Wachstum von x_s' i.a. Grenzen gesetzt. Gibt es in der Spalte s kein $a_{is}'>0$, so kann x_s' beliebig große Werte annehmen, und x_0 ist nicht nach oben beschränkt. Andernfalls bestimme man einen Zeilenindex r, für den gilt:

$b_r'/a_{rs}' := \min \{ b_i'/a_{is}' \mid a_{is}'>0, 1 \leq i \leq m \}$.

Nimmt x_s' den Wert b_r'/a_{rs}' an, so sinkt die der Zeile r zugeordnete Basisvariable (r-te Variable in v) von b_r' auf 0 und "verläßt die Basis"; die anderen Basisvariablen bleiben nichtnegativ. Zeile r heißt *Pivotzeile*, a_{rs}' ist *Pivotelement*.

Schritt 3: Basistausch. x_s' "tritt in die Basis ein" und wird an die r-te Stelle von v gesetzt. Man dividiert die Pivotzeile (einschließlich RS) durch das Pivotelement und subtrahiert von allen anderen Zeilen des Tableaus (einschließlich der x_0-Zeile) Vielfache der Pivotzeile, so daß in der Pivotspalte

35 Z.B. ist A' identisch mit $(D^{-1}A \mid D^{-1})$.
36 Vgl. Dantzig, LP und Erweiterungen, S. 110-117, Witte/Deppe/Born, LP, S. 77, Quelltext der Prozedur "Phase II" im Anhang D.

ein Einheitsvektor entsteht. Das neue Tableau besitzt wieder kanonische Form.
Die drei Schritte bilden einen *primalen Simplexschritt (Iteration)*. Sie werden bis zur Ergebnisfindung (optimale Lösung oder $x_0 \rightarrow \infty$) wiederholt.

Läßt man die Voraussetzung eines zulässigen Ausgangstableaus fallen, muß dem bis jetzt geschilderten Verfahren (*Phase II*) eine *Phase I* vorgeschaltet werden. Kommen Restriktionen vom Typ $a^i x \geq b_i$ oder $a^i x = b_i$ mit $b_i > 0$ vor, so sind einige rechte Seiten in (3) negativ (vgl. 2.2.1). Dies vermeidet man durch die Einführung *künstlicher Variabler* $x_k \geq 0$.[37] Bei Verwendung der Gleichungen $a^i x - s_i + x_k = b_i$ bzw. $a^i x + x_k = b_i$ liegt wieder ein zulässiges Ausgangstableau in kanonischer Form vor.[38] Die $\geq$-Restriktionen lassen sich vorab so umformen, daß für sie nur eine künstliche Variable benötigt wird.[39]
Eine zulässige Lösung der ursprünglichen Aufgabe ist erst dann gegeben, wenn alle künstlichen Variablen die Basis verlassen haben. Um dies sicher zu erreichen, verwendet man in der Phase I eine *Hilfszielfunktion*, welche die Summe der künstlichen Variablen minimiert. Die x_0-Zeile wird wie gewohnt mitgeführt, hat aber keinen Einfluß auf die Wahl der Pivotspalte. Da die Hilfszielfunktion durch die Zahl 0 nach unten beschränkt ist, liefert der primale Simplexalgorithmus eine optimale Lösung. Ist in dieser wenigstens eine künstliche Variable positiv, so besitzt das Ausgangsproblem keine zulässige Lösung. Befindet sich eine künstliche Variable mit dem Wert 0 in der Basis, läßt sie sich entweder ge-

37 Vgl. Dantzig, LP und Erweiterungen, S. 118 ff., Witte/Deppe/Born, LP, S. 81 ff.
38 Eine Gleichungsrestriktion braucht also nicht unbedingt in zwei Ungleichungen zerlegt zu werden.
39 Vgl. Hu, Ganzzahlige Programmierung, S. 60 sowie die LinSen-Prozedur "EINGABE".

gen eine nicht-künstliche Variable tauschen, oder die betreffende Zeile kann ersatzlos gestrichen werden, weil sie in den Spalten der nicht-künstlichen Variablen nur Nullen enthält.[40] Sind alle künstlichen Variablen aus der Basis entfernt, werden ihre Spalten sowie die Hilfszielfunktionszeile gestrichen. Phase II startet dann mit einer Basislösung des Ausgangsproblems, zu der ein zulässiges Tableau gehört.[41]
Neben der beschriebenen *Zweiphasenmethode* ist noch die *M-Methode* zu erwähnen. Sie hat ebenfalls die Ermittlung eines ersten zulässigen Tableaus zum Ziel. Die künstlichen Variablen werden mit einem hinreichend hohen "Kostensatz" M in der Zielfunktion bewertet, so daß sie aus der Basis austreten, falls dies überhaupt möglich ist. Die M-Methode ist für den Einsatz auf EDV-Rechnern ungeeignet, weil dann ein numerischer Wert für M vorzugeben wäre. Die "richtige" Wahl von M hängt aber immer von der Koeffizientenstruktur der zu lösenden Aufgabe ab: Ist M zu klein, erhält man keine zulässige Ausgangslösung; ein zu großes M führt zu erheblichen Rundungsfehlern und falschen Ergebnissen.[42]

Abschließend sei noch auf das Problem der Ausartung eingegangen. Wenn in einem primalen Simplexschritt die Pivotzeile nicht eindeutig bestimmt werden kann, so hat nach Durchführung des Basistauschs wenigstens eine Basisvariable den Wert 0. Eine solche

40 Letzterer Fall tritt auf, wenn es linear abhängige Restriktionen vom Typ $a^i x=b_i$ gibt.
41 Zur Phase I vgl. Hu, Ganzzahlige Programmierung, S. 59, Dantzig, LP und Erweiterungen, S. 120 ff. sowie die LinSen-Prozedur "Phase I".
42 Vgl. Witte/Deppe/Born, LP, S. 91. Die Koeffizienten von M entsprechen nach jeder Iteration genau den Koeffizienten der Hilfszielfunktion in Phase I. Zweiphasen- und M-Methode sind inhaltlich identisch.

Basislösung heißt *primal ausgeartet*.[43] Gilt aber für eine Pivotzeile $b_r'=0$, dann wird x_0 durch den folgenden Basistausch nicht verändert. Es ist möglich, daß sich anschließend ein Zyklus von derartigen Basiswechseln unendlich oft wiederholt.[44] Das Phänomen des "Kreiselns" tritt äußerst selten auf und ist eher von theoretischem Interesse. Festzuhalten bleibt, daß auch für primal ausgeartete Probleme immer eine endliche Folge von Simplexschritten existiert, die zur Entscheidung (optimale Lösung oder $x_0 \rightarrow \infty$) führt.[45]
Unproblematisch ist die *duale Ausartung*. Sie liegt vor, wenn der Zielfunktionskoeffizient einer Nichtbasisvariablen 0 beträgt, und bereitet dem primalen Simplexalgorithmus keinerlei Schwierigkeiten.[46]

2.2.3 Dualer Simplexalgorithmus

Der duale Simplexalgorithmus kehrt die primale Vorgehensweise zur Ermittlung eines zulässigen und optimalen Tableaus um.[47] Er ist anwendbar, wenn das Ausgangstableau bereits optimal ist, also in (3) $-c \geq 0$ gilt. Nun wird x_0 unter Beibehaltung der Optimalität schrittweise durch Basiswechsel gesenkt, bis man ein zulässiges Tableau erreicht oder fest-

43 Man spricht auch von Entartung oder Degeneration. Vgl. Gal, Sensitivitätsanalyse, S. 45.
44 Vgl. Dantzig, LP und Erweiterungen, S. 117. Das wohl bekannteste Beispiel stammt von Beale; vgl. Beale, Cycling, S. 269-275, Vajda, Mathematical Programming, S. 84.
45 Vgl. Dantzig, LP und Erweiterungen, S. 139 ff., 262 ff., 265. Dantzig zeigte weiterhin, daß ein Zyklus mit einer gegen 1 konvergierenden Wahrscheinlichkeit durchbrochen wird, wenn man die Pivotzeile unter den in Frage kommenden Zeilen zufällig auswählt. Vgl. ebenda, S. 143.
Die Zufallsauswahl ist im LinSen-Hauptmenü aktivierbar ("Parameter").
46 Vgl. Gal, Sensitivitätsanalyse, S. 46.
47 Vgl. Krekó, Lehrbuch, S. 220 ff., Witte/Deppe/Born, LP, S. 139-145.

stellt, daß das LO-Problem keine zulässige Lösung besitzt.
Dieses Vorgehen wird unmittelbar einsichtig, wenn man die Dualitätstheorie heranzieht. Zu jedem LO-Problem (1) gehört ein *duales Problem* (6):[48]

$$\begin{aligned} \text{min. } y_0;\ y_0 &:= \mathbf{b}^T\mathbf{y} + b_0 \\ \mathbf{A}^T\mathbf{y} &\geq \mathbf{c} \\ \mathbf{y} &\geq \mathbf{0} \end{aligned} \qquad (6)$$

Die duale Simplexmethode wählt Pivotzeile und Pivotspalte so aus, daß - gleichsam "im Hintergrund" - ein primaler Simplexschritt am zugehörigen dualen Problem ausgeführt wird.[49] Da mit dem dualen zugleich das *primale Problem* (1) gelöst wird,[50] ist das Verfahren nach endlich vielen Iterationen beendet.
Ausgehend vom Tableau (5) mit $\mathbf{c}' \geq \mathbf{0}$ sind folgende Schritte durchzuführen und bis zur Entscheidungsfindung (optimale oder keine zulässige Lösung) zu wiederholen:[51]
Schritt 1: Eine Zeile mit dem absolut größten negativen b_i' wird zur Pivotzeile bestimmt. Ihr Index sei r. Gibt es kein negatives b_i', ist das Optimaltableau gefunden.
Schritt 2: Wenn es in der Pivotzeile kein negatives a_{rj}' gibt, steht das Gleichungssystem im Widerspruch zu den Nichtnegativitätsbedingungen, und das LO-Problem besitzt keine zulässige Lösung.
Andernfalls bestimme man einen Spaltenindex s mit:
$c_s'/a_{rs}' := \max \{ c_j'/a_{rj}' \mid a_{rj}'<0,\ 1 \leq j \leq n+m \}$.
Spalte s wird zur Pivotspalte, a_{rs}' zum Pivotelement.

48 Vgl. z.B. Krekó, Lehrbuch, S. 213.
Das duale Problem zu (6) ist (1).
49 Vgl. Hillier/Lieberman, OR, S. 263, 265, Dantzig, LP und Erweiterungen, S. 276-279, Witte/Deppe/Born, LP, S. 144.
50 Vgl. Abschnitt 2.3.1 dieser Arbeit.
51 Vgl. LinSen-Prozedur "DUAL SIMPLEX".

<u>Schritt 3:</u> Basistausch wie in Abschnitt 2.2.2.

Die drei Schritte bilden einen *dualen Simplexschritt* und sind in Anbetracht der Dualität nicht weiter erläuterungsbedürftig; sie stellen genau das "Spiegelbild" eines primalen Simplexschrittes dar.[52] Man sieht daher auch sofort ein, daß für den dualen Simplexalgorithmus die primale Ausartung unschädlich ist, während im Falle der dualen Ausartung Maßnahmen zur Verhinderung des Kreiselns erforderlich werden können.

Bei Durchführung der Sensibilitätsanalyse treten manchmal Tableaus auf, die weder zulässig noch optimal sind. Um die Einführung künstlicher Variabler zu vermeiden, kann man vorübergehend eine Hilfszielfunktion verwenden, die aus der x_0-Zeile z.B. in der folgenden Weise abgeleitet wird: Man vergrößert die Zielfunktionskoeffizienten aller Nichtbasisvariablen um denselben Wert, so daß sie sämtlich positiv werden. Die Zielfunktionskoeffizienten der Basisvariablen werden unverändert übernommen; sie betragen auch in der Hilfszeile 0. Die Voraussetzung des dualen Simplexverfahrens ist dann gegeben. Nach dessen Abschluß entfällt die Hilfszielfunktionszeile. Existiert ein zulässiges Tableau, so wird die Rechnung mit dem primalen Simplexalgorithmus fortgesetzt.[53]

52 <u>Hillier/Lieberman</u>, OR, S. 263.
53 Vgl. LinSen-Prozedur "DUAL HILF". Auf diese Weise ließe sich auch das allgemeine Problem (1) ohne Verwendung künstlicher Variabler lösen.

2.2.4 Eine auf die Erfordernisse der Sensibilitätsanalyse abgestimmte Berechnungsprozedur

Die Berechnungsprozedur ist der Kern eines Programms zur linearen Optimierung und muß für Zwecke der Sensibilitätsanalyse eine hohe Flexibilität aufweisen. Wie in Kapitel 2.3 gezeigt wird, können die Tableaueigenschaften Zulässigkeit und Optimalität in jeder möglichen Kombination auftreten. Die Algorithmen zur Behandlung aller vorkommenden Fälle wurden in den Abschnitten 2.2.2 und 2.2.3 bereitgestellt und sind Grundlage des LinSen-Unterprogramms "BERECHNUNG". Abb. 1 zeigt den Aufbau dieser Prozedur.

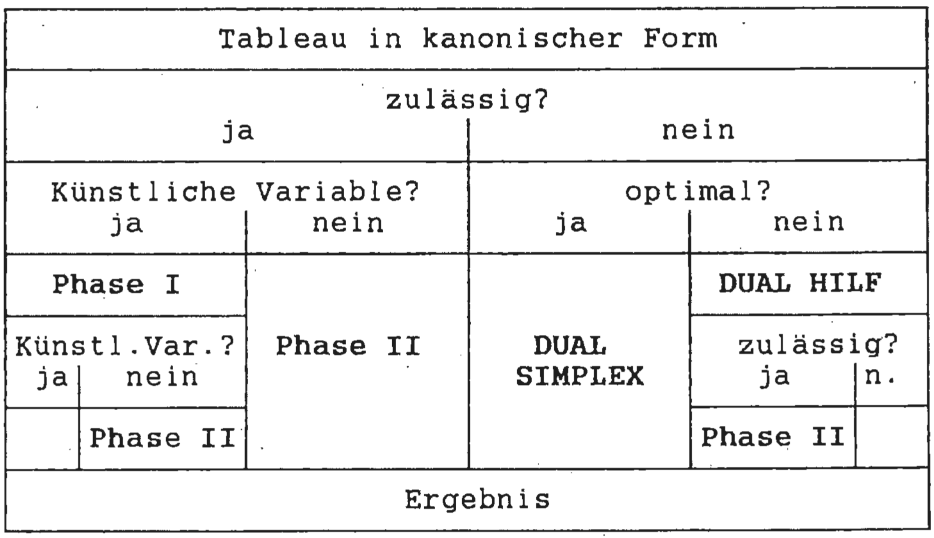

Abb. 1: Aufbau der Prozedur BERECHNUNG[54]

Künstliche Variable werden im Programm LinSen nur für die erstmalige Lösung einer Aufgabe verwendet. Es bleibt zu klären, welche Auswirkungen sie auf die Durchführung der Sensibilitätsanalyse haben.

54 Die Abfragen "zulässig?/optimal?" beziehen sich auf das vorliegende Tableau, das auch künstliche Variable enthalten kann.
Die Frage "Künstl(iche) Var(iable)?" bedeutet: Befinden sich künstliche Variable in der Basis ?

Das Auftreten von Restriktionen des Typs $a^i x=b_i$ hat zur Folge, daß nach dem Streichen der Spalten der künstlichen Variablen am Ende von Phase I die häufig benötigte Inverse der Basismatrix nicht mehr aus dem Endtableau abgelesen werden kann, weil einige der im Ausgangstableau zur Matrix E gehörenden Variablen nicht länger mitgeführt werden.[55] Die Inverse läßt sich in diesem Fall nur noch durch Matrizeninvertierung bestimmen. Als Alternative kann man natürlich auch die Spalten der künstlichen Variablen weiter mitführen und dafür sorgen, daß keine von ihnen Pivotspalte wird. Dann ist in jedem Tableau die vollständige Inverse der Basismatrix ablesbar. Das Programm LinSen umgeht die angesprochenen Probleme auf einfache Weise: Die Option "Sensibilitätsanalyse" steht nur zur Verfügung, wenn das LO-Problem keine Restriktionen des Typs $a^i x=b_i$ enthält. Da sich aber jede Gleichung äquivalent durch zwei Ungleichungen darstellen läßt (vgl. 2.2.1), bedeutet das Voraussetzen von Ungleichungsrestriktionen in (1) keinerlei Einschränkung der Allgemeinheit.

Besteht das Restriktionensystem nur aus ≤- und ≥-Bedingungen, so gibt es maximal eine künstliche Variable x_k (vgl. 2.2.2). Falls diese Variable in Phase I die Basis verlassen hat, liegt ein Tableau der Form (4) vor. Besitzt das LO-Problem dagegen keine zulässige Lösung, so hat x_k im Endtableau einen positiven Wert. Um zu einem Tableau (4) zu gelangen, muß die künstliche Variable vor Beginn der Sensibilitätsanalyse aus der Basis entfernt werden.[56] In der betreffenden Zeile muß sich hierfür ein Pivotelement finden: Läßt man die x_k-Spalte

55 Die Inverse der Basismatrix D ist in (4) deshalb ablesbar, weil *alle* Restriktionen Ungleichungen sind und daher die Matrix E in (3) nur aus Spalten nicht-künstlicher Variabler besteht.

56 Vgl. LinSen-Prozedur "SENSIBILITÄTSANALYSE".

außer Betracht, so sind die Zeilenvektoren des verbleibenden Ausgangstableaus offensichtlich linear unabhängig, denn jede Zeile enthält eine Schlupfvariable, die in allen anderen Zeilen nicht auftritt. Da der Rang einer Matrix durch elementare Zeilentransformationen nicht verändert wird,[57] kann eine Zeile im Endtableau als Koeffizienten von x und s nicht nur Nullen enthalten. Also läßt sich die Basisvariable x_k gegen eine nicht-künstliche Variable tauschen, w.z.z.w.
Als Ergebnis ist festzuhalten, daß die Sensibilitätsanalyse für *jedes* LO-Problem ein Ausgangstableau (3) und ein Endtableau (4) voraussetzen kann. Künstliche Variable brauchen in allen folgenden Überlegungen nicht mehr berücksichtigt zu werden.

2.2.5 Die revidierte Simplexmethode

Für die effiziente Lösung großer LO-Probleme auf EDV-Anlagen wurde die revidierte Simplexmethode entwickelt. Sie benötigt weniger Speicherplatz und Rechenzeit und wirkt der Kumulation von Rundungsfehlern entgegen.[58]
Der revidierten Simplexmethode liegt die Idee zugrunde, in (4) nicht mehr die ganze Matrix $D^{-1}A$, sondern nur noch die für den nächsten primalen Simplexschritt benötigte Pivotspalte zu berechnen.[59] Außerdem erfolgt nach einer gewissen Anzahl von Iterationen eine Neuberechnung des ganzen Tableaus (4) direkt aus den Ausgangsdaten von (1). Dies ist möglich, weil die jeweilige Basis **D** bekannt ist und damit auch D^{-1} durch Gauß-Jordan-Invertierung

57 Vgl. Fischer, Lineare Algebra, S. 52 f.
58 Vgl. Dürr/Kleibohm, OR, S. 69 f., Neumann, OR, S. 124.
59 Vgl. Krekó, Lehrbuch, S. 226 f., Gass, LP, S. 106-125.

leicht ermittelt werden kann. Das Zurückgreifen auf die ursprünglichen Daten verhindert eine weitere Fortpflanzung von Rundungsfehlern.[60] Rundungsfehler entstehen auf EDV-Anlagen vornehmlich durch zu geringe Rechengenauigkeit bei der Übertragung von Dezimalzahlen in das Binärsystem.[61] Sie können die Wirksamkeit des Simplexverfahrens beeinträchtigen.[62] Wichtig in bezug auf die Abbruchkriterien des Simplexalgorithmus (vgl. 2.2.2, 2.2.3) ist insbesondere die Entscheidung, wann eine Zahl als 0 angesehen wird. Im allgemeinen gibt man einen hinreichend kleinen Absolutwert als Rundungstoleranz vor. Betragsmäßig kleinere Zahlen werden dann wie Nullen behandelt.[63]
Das Programm LinSen verzichtet auf die Verwendung der revidierten Simplexmethode und bietet Einstellparameter für eine bedingte Rundung an, die der Größenordnung der Koeffizienten angepaßt werden können.[64] Obwohl im Zyklusbeispiel von Beale (vgl. 2.2.2) während der Rechnung periodische Dezimalbrüche auftreten, ist auch nach über 500000 Iterationen kein Ende des Kreiselns wegen kumulierter Rundungsfehler abzusehen. Dies mag als Hinweis auf hinreichende numerische Stabilität von LinSen gewertet werden.

60 Vgl. Gal, LO, S. 250, Künzi/Tzschach/Zehnder, Mathematische Optimierung, S. 84 f.
61 Vgl. Herschel, Pascal, S. 73 f.
62 Vgl. Müller-Merbach, Round-Off Errors, S. 1 ff.
63 Vgl. Stahlknecht/Ohmann, LP auf dem PC, S. 86 ff., Künzi/Tzschach/Zehnder, Mathematische Optimierung, S. 80.
64 Vgl. LinSen-Prozeduren "PARAMETER" und "GLATT".

2.3 Komponenten der Sensibilitätsanalyse

2.3.1 Die Bedeutung der Dualitätstheorie im Rahmen der Sensibilitätsanalyse

Die Dualitätstheorie beschreibt die Beziehungen zwischen den LO-Problemen (1) und (6). Mit ihrer Hilfe läßt sich in vielen Fällen sehr einfach erkennen, ob nach einer Änderung der Ausgangsdaten die bisherige Basislösung optimal bleibt.[65] Für Zwecke der Sensibilitätsanalyse sind die beiden folgenden Sätze von besonderer Bedeutung:

> Satz 1 (Optimalitätssatz):
>
> $\mathbf{x}^*$ und $\mathbf{y}^*$ seien zulässige Lösungen von (1) bzw. (6). Gilt $\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*$, so sind $\mathbf{x}^*$ und $\mathbf{y}^*$ optimale Lösungen des primalen bzw. des dualen Problems.

> Satz 2 (Dualitätstheorem):
>
> Wenn eines der Probleme (1) und (6) eine optimale Lösung besitzt, dann auch das andere Problem: Beide haben den gleichen optimalen Zielfunktionswert. Aus dem Optimaltableau eines Problems lassen sich die optimalen Basislösungen beider Probleme ablesen. Insbesondere gilt:
> Die Zielfunktionskoeffizienten der primalen Schlupfvariablen (Strukturvariablen) entsprechen den Werten der dualen Strukturvariablen (Schlupfvariablen).

Beweise dieser Sätze liefern u.a. Krekó und Dantzig.[66]

Änderungen der Ausgangsdaten betreffen sowohl das primale als auch das duale Problem. Da $\mathbf{x}^*$ und $\mathbf{y}^*$ dem Optimaltableau entnommen werden können und $\mathbf{c}^T\mathbf{x}^*=\mathbf{b}^T\mathbf{y}^*$ gilt (Satz 2), läßt sich häufig Satz 1

65 Vgl. Hillier/Lieberman, OR, S. 143 ff.
66 Vgl. Krekó, Lehrbuch, S. 213-215, Dantzig, LP und Erweiterungen, S. 166 f., Müller-Merbach, OR, S. 133 ff., Collatz/Wetterling, Optimierungsaufgaben, S. 58 ff.

heranziehen, um die Optimalität der bisherigen Lösung für das abgeänderte Problem zu überprüfen. Der Dualitätszusammenhang kann beim Hinzufügen und Streichen von Variablen und Restriktionen sowie bei Änderungen der Koeffizientenspalte von Nichtbasisvariablen vorteilhaft ausgenutzt werden.

2.3.2 Sensibilitätsanalyse der Zielfunktionsbeiträge

2.3.2.1 Isolierte Schwankungsbreiten der Zielfunktionsbeiträge

Gegeben sei ein zulässiges und optimales Tableau (5). Wie sich aus (4) ergibt, kann eine Änderung der Zielfunktionsbeiträge nur die Optimalität, nicht aber die Zulässigkeit des Tableaus beeinflussen: $\mathbf{D}^{-1}\mathbf{b}$ bleibt unberührt. Zuerst soll die Frage beantwortet werden, in welchen Grenzen jeweils ein einzelner Zielfunktionsbeitrag c_j schwanken darf, ohne daß die Optimalitätseigenschaft des bisherigen Endtableaus verlorengeht und ein Basiswechsel erforderlich wird.[67]

Im Problem (1) werde c_s in c_s+t mit $t \neq 0$ abgeändert, d.h. die x_s-Spalte in (3) enthält nunmehr den Zielfunktionskoeffizienten $-c_s-t$. Da die x_0-Zeile nie Pivotzeile ist, wird sie nur additiv verändert; im Endtableau (5) steht daher an derselben Stelle $c_s'-t$. Ist die Variable x_s in (5) Nichtbasisvariable, so bleibt das Tableau optimal, wenn ihr Zielfunktionskoeffizient nichtnegativ ist, also $t \leq c_s'$ gilt. Eine untere Begrenzung für t gibt es nicht. Die gesuchte isolierte Schwankungsbreite des Zielfunktionsbeitrages von x_s beträgt mithin $]-\infty;c_s+c_s']$. Ist dagegen x_s Basisvariable in der

67 Vgl. Dinkelbach, Analysen, S. 71-74.
Bei Minimierungsproblemen gibt LinSen die Intervalle für die ursprünglichen, noch nicht mit -1 multiplizierten Zielfunktionsbeiträge aus.

Zeile r, so gilt $c_s'=0$, und der Koeffizient $c_s'-t = -t \neq 0$ zerstört die kanonische Form. Sie wird wiederhergestellt, indem man die Zeile r t-mal zur x_0-Zeile addiert.[68] Die neuen Zielfunktionskoeffizienten lauten $c_j'+t\cdot a_{rj}'$ ($j\neq s$; für $j=s$ ergibt sich, wie beabsichtigt: $-t+t\cdot 1=0$) und müssen nichtnegativ sein, wenn das Tableau optimal bleiben soll. Hieraus folgt $t \geq -c_j'/a_{rj}'$ für $a_{rj}'>0$ bzw. $t \leq -c_j'/a_{rj}'$ für $a_{rj}'<0$. Die Untergrenze für t beträgt also $t_u := \max \{-c_j'/a_{rj}' \mid a_{rj}'>0,\ 1\leq j\leq n+m,\ j\neq s\}$ bzw. $-\infty$, wenn keines der a_{rj}' ($j\neq s$) positiv ist. Analog ergibt sich die Obergrenze für t zu $t_o := \min \{-c_j'/a_{rj}' \mid a_{rj}'<0,\ 1\leq j\leq n+m,\ j\neq s\}$ bzw. $+\infty$, wenn es kein negatives a_{rj}' gibt.[69] Die gesuchte isolierte Schwankungsbreite beträgt somit $[c_s+t_u; cs+t_o]$.[70] Je kleiner das Intervall ist, desto "sensibler" reagiert die optimale Lösung auf Änderungen des betrachteten Zielfunktionsbeitrages.

Im Falle der Ausartung des Optimaltableaus treten Interpretationsschwierigkeiten auf, die im folgenden behandelt werden sollen.
Die duale Ausartung verursacht kaum Probleme. Falls sich ein positives Pivotelement findet, gibt es mehr als eine optimale Basislösung, denn eine Nichtbasisvariable mit dem Zielfunktionskoeffizienten 0 kann in die Basis eintreten, ohne den Wert von x_0 zu verändern. Konvex-Kombinationen dieser Basislösungen sind ebenfalls wieder optimale Lösungen.[71] Da die Sensibilitätsanalyse die Stabilität einer gegebenen Lösungsstruktur untersucht, ist es möglich, sie für jedes der verschiedenen Opti-

68 Vgl. Dürr/Kleibohm, OR, S. 84.
69 Vgl. Gal, LO, S. 200 ff.
70 Das Intervall heißt auch "Empfindlichkeitsbereich" oder "kritischer Bereich". Gal/Gehring, Planungstechniken, S. 90.
71 Vgl. Gal, Sensitivitätsanalyse, S. 46 f., Krekó, Lehrbuch, S. 54 ff.

maltableaus gesondert durchzuführen. Diese Vorgehensweise ist sinnvoll, weil zu unterschiedlichen Tableaus i.a. auch unterschiedliche Lösungen **x** gehören.[72] Es kann von Interesse sein, diese Lösungen hinsichtlich ihrer Stabilität zu vergleichen. Die Ergebnisse weisen eine erhöhte Empfindlichkeit aus: Bestimmte Zielfunktionsbeiträge besitzen in einer oder in beiden Richtungen keinerlei Variationsspielraum, weil oftmals c_s' oder t_u oder t_o den Wert 0 haben.[73] Konvex-Kombinationen bleiben optimal, solange die sie erzeugenden Basislösungen optimal sind.

Problematischer gestaltet sich der Fall der primalen Ausartung. Findet sich ein negatives Pivotelement, so kann eine Basisvariable mit dem Wert 0 die Basis verlassen, ohne daß sich Zielfunktionswert und optimale Lösung ändern.[74] Zur selben optimalen Lösung **x** gehört also mehr als eine optimale Basis. Da die voneinander verschiedenen Optimaltableaus materiell dieselben Handlungsmöglichkeiten (Vektor **x**) beschreiben, ist im Gegensatz zum Fall der dualen Ausartung ihre getrennte Untersuchung nicht mehr sinnvoll. Vielmehr liegt es nahe, für jeden Zielfunktionsbeitrag c_j die Schwankungsintervalle bezüglich aller alternativen Optimaltableaus zu ermitteln und die *Vereinigung* dieser Intervalle als "richtige" isolierte Schwankungsbreite von c_j zu definieren.[75] Damit wird dem Umstand Rechnung getragen, daß sich die optimale Lösung **x** auf ver-

72 Liegt gleichzeitig primale Ausartung vor und ändert sich die optimale Lösung **x** durch den Basistausch nicht, ist anders zu verfahren.

73 Vgl. Gal, LO, S. 204.

74 Vgl. Gal, Sensitivitätsanalyse, S. 45 f.

75 Vgl. Gal, Shadow Prices, S. 66 f. Falls die primale Ausartung durch eine "schwach redundante" Restriktion verursacht wird, kann man nach Weglassen dieser Restriktion wie im nicht ausgearteten Fall vorgehen und kommt zu denselben Ergebnissen. Vgl. ebenda, S. 67 f.

schiedene Art und Weise als Basislösung darstellen läßt und man an der Stabilität dieser Lösung, nicht aber der formalen Basisstrukturen interessiert ist. Es ist für die Interpretation der Optimallösung gleichgültig, ob die Nullvariablen als Basis- oder als Nichtbasisvariable auftreten.[76] Solange irgendeines der verschiedenen Tableaus optimal bleibt, ist die alte Lösung **x** optimal.
Die korrekte Durchführung der Sensibilitätsanalyse unter Ausartung setzt, wie gezeigt, die Kenntnis der verschiedenen Optimaltableaus voraus. Das Programm LinSen prüft aus diesem Grunde eine optimale Basislösung stets auf primale und duale Ausartung und ermöglicht die Berechnung alternativer Optimaltableaus, für die dann ebenfalls Schwankungsintervalle ermittelt werden können (vgl. unten, 2.4.1.).

2.3.2.2 Simultane Änderung mehrerer Zielfunktionsbeiträge

Die in 2.3.2.1 hergeleiteten isolierten Schwankungsbreiten sind i.a. nur von begrenztem praktischen Wert, weil sie voraussetzen, daß jeweils alle anderen Zielfunktionsbeiträge genau bekannt und konstant sind. Diese Annahme dürfte häufig unrealistisch sein; man denke z.B. an folgenden Fall: Wenn in einem Produktionsplanungsproblem ein Rohstoffpreis falsch geschätzt wurde, verändert eine nachträgliche Korrektur meist die Zielfunktionsbeiträge mehrerer Variabler, so daß die "ceteris paribus"-Bedingung der isolierten Intervalle nicht erfüllt ist.[77] Können darüber hinaus die c_j unabhängig voneinander schwanken, ist eine simultane, mehrparametrische Betrachtung unvermeidlich.
Die Vorgehensweise ist dieselbe wie in 2.3.2.1, wobei jedoch nunmehr gleichzeitig alle c_j durch c_j+t_j ersetzt werden. Für jede Variable, die im Optimal-

76 Vgl. <u>Gal</u>, Sensitivitätsanalyse, S. 284.
77 Vgl. <u>Sandor</u>, Ranging, S. 29 f.

tableau Basisvariable ist, wird die beschriebene Wiederherstellung der kanonischen Form vorgenommen. Die neuen Zielfunktionskoeffizienten enthalten dann Terme, welche linear von den n Parametern t_j abhängen. Die vorliegende Basislösung bleibt optimal, wenn alle Zielfunktionskoeffizienten nichtnegativ sind, wodurch ein lineares Ungleichungssystem gegeben ist. Dessen Lösungsmenge läßt sich für n=3 nur noch schlecht und für n>3 überhaupt nicht mehr graphisch darstellen. Für die praktische Anwendung ist durch die Formulierung solcher "kritischen Gebiete" wenig gewonnen, da sich im allgemeinen Fall die Optimalität der Basislösung nur durch Einsetzen konkreter Parameterwerte in sämtliche Ungleichungen überprüfen läßt.[78]

Für n=2 ist das "Optimalitätsgebiet" eine Fläche in der (t_1, t_2)-Ebene. Sie kann wegen ihrer linearen Begrenzung unmittelbar aus zwei isolierten Schwankungsintervallen konstruiert werden: Man berechnet z.B. die isolierte Schwankungsbreite von t_2 für zwei verschiedene Werte von t_1, trägt beide Intervalle als Strecken in ein Koordinatensystem ein und verbindet die Randpunkte durch Geraden; vgl. Abb. 2 auf der folgenden Seite.

Die "Unhandlichkeit" simultaner Schwankungsbereiche hat zur Entwicklung neuer Konzepte der Sensibilitätsanalyse geführt. Als Beispiel sei der "Toleranzansatz" genannt.[79] Sein Ziel ist es, den maximalen Prozentsatz zu ermitteln, um den die betrachteten Koeffizienten simultan und unabhängig voneinander von ihren ursprünglichen Werten abwei-

78 Vgl. Dinkelbach, Analysen, S. 137,
Wendell, Sensitivity Analysis, S. 568 f.
LinSen verzichtet auf die Angabe solcher Ungleichungen, zumal sich diese auf einfachste Weise aus dem Optimaltableau ergeben.

79 Vgl. Wendell, Sensitivity Analysis, S. 564 ff., 574 ff.

chen dürfen, wenn die bisherige Lösungsstruktur optimal bleiben soll.

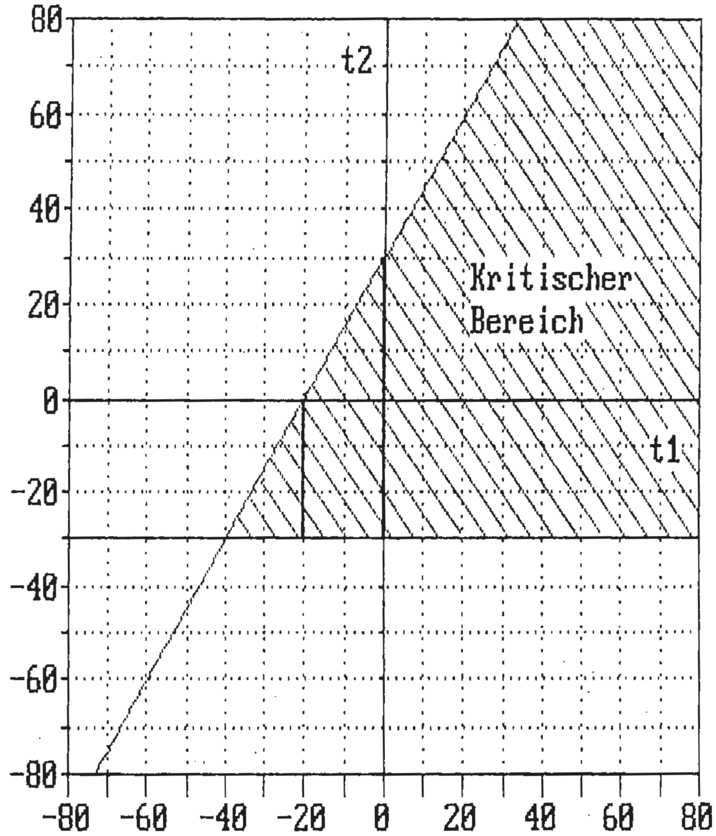

Abb. 2: Konstruktion des kritischen Bereichs

Es verbleibt noch die zweite Frage der eingangs definierten Sensibilitätsanalyse zu beantworten: Wie lautet das neue Optimaltableau, wenn Änderungen im Vektor c zu berücksichtigen sind ? Im allgemeinen wird die neue optimale Lösung wesentlich näher am bisherigen Optimum als am Koordinatenursprung der **x**-Hyperebene liegen. Es ist daher vorteilhaft, vom Endtableau (4) der bisherigen Aufgabe auszugehen und nicht das abgeänderte Problem von Anfang an neu zu lösen.[80]

80 Vgl. Hillier/Lieberman, OR, S. 80, Krekó, Lehrbuch, S. 233, Shetty, Analyses, S. 103, Kern, Empfindlichkeit, S. 67.

Im Prinzip hat man nur die x_0-Zeile in (4) mit den geänderten Werten von **c** und **d** neu zu berechnen, um das zur alten optimalen Basis gehörende Tableau der neuen Aufgabe zu erhalten. Noch einfacher ist aber folgende Vorgehensweise:[81] Die alte x_0-Zeile in (4) wird gestrichen und durch die neue Zielfunktionszeile in der Form des Ausgangstableaus (3) überschrieben. Dies führt zur Zerstörung der kanonischen Form, wenn danach die Zielfunktionskoeffizienten der Basisvariablen nicht mehr alle gleich 0 sind. Da das einer Basis zugeordnete Tableau (4) eindeutig ist,[82] genügt es, die kanonische Form wiederherzustellen. Durch Addition geeigneter Vielfacher der Tableauzeilen zur Zielfunktionszeile (vgl. oben und 2.3.2.1) erhält man das gesuchte Tableau. Ist es optimal, so maximiert die bisherige Basislösung auch die neue Zielfunktion. Andernfalls wird der primale Simplexalgorithmus auf das Tableau angewendet, und man erhält nach einigen Iterationen das Ergebnis der abgeänderten Aufgabe.

2.3.3 Sensibilitätsanalyse der rechten Seiten

2.3.3.1 Isolierte Schwankungsbreiten der rechten Seiten

Ausgangspunkt sei ein zulässiges und optimales Tableau (5). Aus (4) ergibt sich, daß eine Änderung der rechten Seiten **b** nur Auswirkungen auf die Zulässigkeit, nicht aber auf die Optimalität des Tableaus haben kann: Die Zielfunktionskoeffizienten hängen nicht von **b** ab.[83] Zunächst ist zu untersuchen, in welchen Grenzen jeweils eine einzelne rechte Seite b_i variieren darf, ohne die Zulässig-

81 Vgl. LinSen-Prozedur "NEUBERECHNUNG ZLFKT".
82 Vgl. Dantzig, LP und Erweiterungen, S. 94.
83 Vgl. Dinkelbach, Analysen, S. 74-76, Dürr/Kleibohm, OR, S. 86-88.

keit des bisherigen Optimaltableaus zu zerstören und einen Basistausch notwendig zu machen.
Sei δ ein Nullvektor des R^m, zu dessen r-ter Koordinate ein Parameter $t \neq 0$ addiert wird, d.h. $\delta := (0...t...0)^T$. Ersetzt man in (1) einen Koeffizienten b_r durch b_r+t, so lauten die rechten Seiten im Ausgangstableau (3) $b+\delta$ und im Endtableau (4) $D^{-1}(b+\delta) = b'+D^{-1}\delta$. Das Produkt $D^{-1}\delta$ ist der mit t multiplizierte r-te Spaltenvektor von D^{-1}. Die Werte der Basisvariablen bleiben nichtnegativ und damit zulässig, wenn $b'+D^{-1}\delta \geq 0$ gilt, bzw. koordinatenweise: $b_i'+a_{in+r}'t \geq 0$ für alle i.[84] Es folgt $t \geq -b_i'/a_{in+r}'$ für $a_{in+r}'>0$ bzw. $t \leq -b_i'/a_{in+r}'$ für $a_{in+r}'<0$. Die Untergrenze für t ist folglich $t_u := \max \{-b_i'/a_{in+r}' \mid a_{in+r}'>0, 1 \leq i \leq m\}$ bzw. $-\infty$, wenn keines der a_{in+r}' positiv ist. Die Obergrenze lautet $t_o := \min \{-b_i'/a_{in+r}' \mid a_{in+r}'<0, 1 \leq i \leq m\}$ bzw. $+\infty$, wenn es keine negativen a_{in+r}' gibt.[85] Die gesuchte isolierte Schwankungsbreite ergibt sich zu $[b_r+t_u; b_r+t_o]$.[86] Sie läßt sich besonders einfach ermitteln, wenn die zur Restriktion r gehörende Schlupfvariable im Optimaltableau Basisvariable ist. Dann ist der r-te Spaltenvektor von D^{-1} ein Einheitsvektor, und alle a_{in+r}' sind null, mit Ausnahme eines Koeffizienten $a_{ln+r}'=1$. Die Untergrenze ergibt sich sofort zu $t_u := -b_l'$; eine Obergrenze existiert nicht.[87]

84 A' und b' wurden in (5) definiert.
85 Vgl. Gal, LO, S. 196 ff.
86 Die von LinSen berechneten Intervalle für die rechten Seiten von $\geq$-Restriktionen beziehen sich auf die ursprünglichen, noch nicht mit -1 multiplizierten b_i.
87 Dies leuchtet unmittelbar ein: Die Schlupfvariable gibt z.B. einen Kapazitätsüberschuß an. b_r könnte also maximal um die überschüssige Kapazität verringert sowie beliebig vergrößert werden, ohne daß irgendeine Basisvariable negativ würde. Vgl. Dürr/Kleibohm, OR, S.87, Dinkelbach, Analysen, S.75.

Es sei noch auf eine interessante zusätzliche Anwendungsmöglichkeit der isolierten Schwankungsbreiten der rechten Seiten hingewiesen: Mit ihrer Hilfe lassen sich "redundanzverdächtige" Restriktionen identifizieren. Näheres hierzu findet man im Aufsatz von Gal und Zimmermann.[88]

Welche Besonderheiten ergeben sich, wenn das Optimaltableau ausgeartet ist ?
Die duale Ausartung ist unproblematisch. Jede einzelne der verschiedenen Optimallösungen kann einer eigenständigen Sensibilitätsanalyse unterzogen werden (vgl. 2.3.2.1).
Bei primaler Ausartung ist der optimalen Lösung **x** i.a. mehr als ein Optimaltableau zugeordnet, so daß die zu diesen Tableaus gehörenden isolierten Schwankungsintervalle zusammengefaßt werden müssen.[89] Bewegt sich der Wert der betrachteten rechten Seite innerhalb der Vereinigung dieser Intervalle, so bleibt immer wenigstens eines der alternativen Endtableaus zulässig und damit optimal.[90]
Die verschiedenen optimalen Lösungsstrukturen reagieren wegen der primalen Ausartung besonders empfindlich auf Änderungen der rechten Seiten:[91] Häufig ist $t_o=0$ und/oder $t_u=0$. In solchen Fällen verliert die übliche Deutung der Zielfunktionskoeffizienten als "Schattenpreise" ihre Gültigkeit, weil einige rechte Seiten nicht den geringsten Variationsspielraum nach unten bzw. nach oben besitzen.[92]
Es ist dann notwendig, i.a. voneinander abweichende "links- und rechtsseitige Schattenpreise" zu bestimmen, indem man zum jeweiligen "Nachbartableau"

88 Vgl. Zimmermann/Gal, Redundanz, S. 232 ff.
89 Die Tableaus sind alle gleichwertig; man kann nicht eines den anderen vorziehen. Vgl. 2.3.2.1 und Gal, Shadow Prices, S. 66.
90 Vgl. ebenda, S. 67.
91 Vgl. Gal, Sensitivitätsanalyse, S. 284 f.
92 Vgl. Stahlknecht/Ohmann, LP auf dem PC, S. 85 f.

übergeht. Ein Tableau liefert den "Verkaufs-", das andere den "Ankaufs-Schattenpreis" eines Begrenzungsfaktors.[93]
Die Sensibilitätsanalyse der rechten Seiten wirft bei primaler Ausartung noch ein weiteres Problem auf: Es kann Tableaus geben, die formal nicht optimal sind (wenigstens ein negativer Zielfunktionskoeffizient), aber dennoch bereits die optimale Lösung liefern ! Durch einen dualen Simplexschritt lassen sie sich bei unveränderten Werten der Basisvariablen in optimale Tableaus überführen. Eine offene Frage ist, ob auch diese formal nicht optimalen Tableaus der optimalen Lösung zugerechnet werden sollen. Als Folge müßten auch für sie Schwankungsintervalle ermittelt und mit den übrigen Intervallen vereinigt werden.[94]

2.3.3.2 Simultane Änderung mehrerer rechter Seiten

Die in 2.3.3.1 angewendete Vorgehensweise läßt sich sofort auf den mehrparametrischen Fall übertragen. Können die rechten Seiten gleichzeitig und unabhängig voneinander schwanken, lautet der Änderungsvektor δ nunmehr $\delta := (t_1, t_2, ..., t_m)^T$. Für die Werte der Basisvariablen im Endtableau (4) ist $D^{-1}(b+\delta) = b'+D^{-1}\delta \geq 0$ zu fordern, wodurch ein lineares Ungleichungssystem von m Variablen festgelegt ist. Die im Zusammenhang mit derartigen Ungleichungssystemen auftretenden Schwierigkeiten sind völlig analog zu den in Abschnitt 2.3.2.2 geschilderten und sollen hier nicht ein zweites Mal dargestellt werden.[95]
Wird ein geänderter Vektor b vorgegeben, erhebt sich wieder die Frage, wie man ohne großen zusätzlichen Rechenaufwand zur neuen optimalen Lösung ge-

93 Siehe hierzu Gal, Shadow Prices, S. 62, 70.
94 Vgl. ebenda, S. 62, 67, 70.
95 Vgl. Wendell, Sensitivity Analysis, S. 568 f. Zum Toleranzansatz vgl. ebenda, S. 572 ff.

langt.[96] Tableau (4) zeigt, daß lediglich der Zielfunktionswert ($d^TD^{-1}b+b_0$) sowie die Werte der Basisvariablen ($D^{-1}b$) neu zu berechnen sind. D^{-1}, d und b_0 haben sich nicht geändert und sind dem alten Endtableau bzw. dem Ausgangsproblem (1) direkt zu entnehmen. Ist das auf diese Weise gewonnene Tableau nach wie vor zulässig, bleibt die bisherige Lösungsstruktur auch für die abgeänderte Aufgabe optimal. Andernfalls dient die vorliegende Basis als Ausgangspunkt für den dualen Simplexalgorithmus, der in wenigen Iterationen zum neuen Endtableau führt.[97]

2.3.4 Sensibilitätsanalyse der Tableaukoeffizienten

2.3.4.1 Probleme bei der Berechnung isolierter Schwankungsbreiten der Tableaukoeffizienten

Gegeben sei wieder ein Optimaltableau (5) bzw. (4). Ändert man ein Element der Koeffizientenmatrix **A**, das zu einer Nichtbasisvariablen x_s von (4) gehört, bleiben **b**, c, **d** sowie die Basis **D** und ihre Inverse D^{-1} unberührt.[98] Tableau (4) ist zu entnehmen, daß dann die bisherige Lösung weiterhin zulässig ist und lediglich geprüft werden muß, ob der Zielfunktionskoeffizient von x_s nichtnegativ bleibt: Die Änderung betrifft nur die x_s-Spalte. Man ersetzt in (1) a_{rs} durch $a_{rs}+t$, addiert also zum Spaltenvektor a_s den aus 2.3.3.1 bekannten Vektor δ. Der zu prüfende Zielfunktionskoeffizient in (4) lautet dann $d^TD^{-1}(a_s+\delta)-c_s = c_s'+d^TD^{-1}\delta$.[99] Weil $D^{-1}\delta$ der mit t multiplizierte r-te Spaltenvektor von D^{-1} ist, entspricht $d^TD^{-1}\delta$ folglich dem mit t multiplizierten Zielfunktionskoeffizienten c_{n+r}' der Schlupfvaria-

96 Vgl. Hillier/Lieberman, OR, S. 151 f., Krekó, Lehrbuch, S. 233.
97 Vgl. LinSen-Prozedur "RECHTE SEITEN".
98 Vgl. Dinkelbach, Analysen, S. 76-78.
99 Per def. gilt $c_s' := d^TD^{-1}a_s-c_s$, vgl. (4), (5).

blen s_r in (5).[100] Mit den Bezeichnungen von (5) ergibt sich also der neue Zielfunktionskoeffizient von x_s zu $c_s'+c_{n+r}'t$. Nach Voraussetzung ist das bisherige Tableau optimal; es gilt also $c_s'\geq 0$, $c_{n+r}'\geq 0$. Es bleibt optimal, wenn $c_s'+c_{n+r}'t \geq 0$ ist. Die Untergrenze für t lautet mithin $t_u:= -c_s'/c_{n+r}'$, sofern $c_{n+r}'>0$ gilt. Ist $c_{n+r}'=0$, gibt es keine Untergrenze: $t_u:= -\infty$. Eine Obergrenze für t existiert nicht. Ergebnis: Gehört a_{rs} zu einer Nichtbasisvariablen des Optimaltableaus, heißt das isolierte Schwankungsintervall: $[a_{rs}+t_u;+\infty[$.
Wenn die Änderung einen Koeffizienten von **A** betrifft, der zu einer Basisvariablen in (5) gehört, bedeutet dies eine Revision der Basismatrix und ihrer Inversen. Aus (4) ergibt sich, daß sowohl die Optimalität als auch die Zulässigkeit des Tableaus u.U. nicht mehr gegeben sind. Erschwerend kommt hinzu, daß durch die Koeffizientenänderung die Spalten von **D** linear abhängig werden können.[101] Wenn aber **D** singulär ist, existiert $\mathbf{D}^{-1}$ nicht mehr. Unter der einschränkenden Voraussetzung, daß letzterer Fall nicht eintritt, **D** also invertierbar bleibt, lassen sich isolierte Schwankungsbreiten herleiten. Hierzu sei auf Dinkelbach verwiesen.[102]
Die ohnehin begrenzte Aussagefähigkeit isolierter Schwankungsintervalle (vgl. 2.3.2.2) tritt im Falle der Tableaukoeffizienten in verschärfter Form auf: Häufig repräsentieren die a_{ij} Produktionskoeffizienten. Ist die Zielfunktion gewinn- oder kostenorientiert, so beeinflußt eine Änderung des Faktorverbrauchs pro Einheit i.a. gleichzeitig den zugehörigen Zielfunktionsbeitrag c_j; eine isolierte Betrachtung der a_{ij} ist dann nicht sinnvoll.[103]
Schließlich lassen sich die gewonnenen Intervalle

100 Vgl. (4), (5). s_r heißt in (5) per def. x_{n+r}'.
101 Vgl. <u>Hadley</u>, LP, S. 423.
102 Vgl. <u>Dinkelbach</u>, Analysen, S. 78-82.
103 Vgl. <u>Stepan</u>/<u>Fischer</u>, Optimierung, S. 99.

auch nicht zur Konstruktion simultaner Schwankungsbereiche verwenden. Dies liegt daran, daß die Zielfunktionskoeffizienten und Basisvariablenwerte nicht mehr linear von den Änderungsparametern abhängen, wie es noch in 2.3.2.2 und 2.3.3.2 der Fall war (vgl. 2.3.4.2).
Wegen der einschränkenden Voraussetzungen und eng begrenzten Anwendungsmöglichkeiten wird im Programm LinSen auf die Berechnung isolierter Schwankungsintervalle der a_{ij} verzichtet.

2.3.4.2 Simultane Änderung der Koeffizienten einer Spalte

Die im Abschnitt 2.3.4.1 geschilderten Schwierigkeiten verstärken sich, wenn in der Koeffizientenmatrix mehrere a_{ij} unabhängig voneinander variieren.[104] Ein Parameter enthaltender Term kann zum Pivotelement werden, wodurch im Simplextableau an jeder Stelle Quotienten auftreten können, deren Nenner von Parametern abhängen. Insbesondere werden die Vektoren **c'** und **b'** in (5) zu nichtlinearen Funktionen mehrerer Parameter. Gal kommt zu dem Schluß: "Bis jetzt gibt es in der Literatur praktisch keinen vernünftigen Algorithmus zur Lösung dieser Probleme".[105] Als Ausweg bietet sich neben der Untersuchung von Spezialfällen der Übergang zum Toleranzansatz der Sensibilitätsanalyse an.[106]

Im folgenden soll wieder die zweite Frage der Sensibilitätsanalyse in den Vordergrund treten. Wie erhält man, ausgehend vom alten Optimaltableau (4), die Lösung der abgeänderten Aufgabe, wenn für eine ganze Spalte von (1) neue Werte vorgegeben werden ? Es erweist sich für Anwendungen als zweckmäßig,

104 Vgl. Dinkelbach, Analysen, S. 134 f., 137.
105 Gal, Sensitivitätsanalyse, S. 241.
106 Vgl. z.B. Barnett, Stability, S. 226 ff.; zum Toleranzansatz vgl. 2.3.2.2 und Ravi/Wendell, Tolerance, S. 1106 ff.

zugleich eine Änderung des Zielfunktionsbeitrages der betreffenden Spalte zuzulassen.[107]
Die gestellte Frage läßt sich auch dann beantworten, wenn die modifizierte Spalte zu einer Basisvariablen des Optimaltableaus gehört. In keinem Fall ist es erforderlich, das Problem von Beginn an neu zu lösen.[108]
Zuerst sei angenommen, daß die zur Spalte s von (1) gehörende Variable x_s im Optimaltableau (4) Nichtbasisvariable ist und daher in der optimalen Basislösung den Wert 0 hat. Dann kann eine Änderung des Spaltenvektors a_s sowie des Zielfunktionsbeitrages c_s nur die Optimalität, nicht aber die Zulässigkeit des bisherigen Tableaus beeinflussen (vgl. 2.3.4.1). Mit Hilfe der Dualitätssätze läßt sich leicht herausfinden, ob die alte Basislösung optimal bleibt: Letzteres ist genau dann der Fall, wenn die gemäß Satz 2 ablesbare Lösung $y^*=(d^TD^{-1})^T$ des zugehörigen Dualproblems nach wie vor zulässig ist.[109] Der Abänderung einer Spalte im Primalproblem (1) entspricht die Änderung einer Zeile im Dualproblem (6). Erfüllt die bisherige duale Lösung y^* die modifizierte duale Restriktion (Zeile) $a_s^T y \geq c_s$, so sind die alten Lösungen x^* und y^* des Primal- bzw. des Dualproblems für die abgeänderten Probleme nach wie vor zulässig und führen zum glei-

107 Auf diese Weise kann in Ansätzen zur Produktionsplanung ein gesamtes Produktionsverfahren einschließlich der zugehörigen Deckungsspanne geändert werden. Vgl. 2.3.4.1 und Anhang A. Die simultane Änderung einer *Zeile* wird hier nicht behandelt; vgl. dazu Shetty, Analyses, S. 93 f. Mit LinSen läßt sich das Problem am schnellsten lösen, indem man die Restriktion streicht und mit den geänderten Koeffizienten wieder anfügt (vgl. 2.3.6).

108 Vgl. im Gegensatz hierzu Hadley, LP, S. 423, Churchman/Ackoff/Arnoff, OR, S. 294.

109 Vgl. 2.3.1 und Hillier/Lieberman, OR, S. 143 f., 152 f.

chen Zielfunktionswert.[110] Dann sind sie nach Satz 1 optimal. Ist dagegen die duale Schlupfvariable und damit (Satz 2) der Zielfunktionskoeffizient der primalen Strukturvariablen x_s negativ (d.h. gilt $a_s^T y^* - c_s < 0$), so muß wenigstens ein primaler Simplexschritt durchgeführt werden, um die Optimalität des Tableaus wiederherzustellen. Hierzu ist vorab die geänderte Spalte $D^{-1}a_s$ in (4) zu berechnen. Der neue, negative Zielfunktionskoeffizient von x_s ist aus der Dualitätsüberlegung bereits bekannt und läßt sich durch Ausrechnen von $d^T D^{-1} a_s - c_s$ bestätigen.[111]

Wenn x_s im Optimaltableau Basisvariable ist, wird ganz analog zunächst die zugehörige Spalte in (4) neu berechnet, und zwar unter Verwendung der bisherigen Werte von D^{-1} und d sowie der abgeänderten Werte von a_s und c_s. Diese Operation zerstört die kanonische Form des Simplextableaus, denn i.a. ist $D^{-1}a_s$ kein Einheitsvektor mehr und $d^T D^{-1} a_s - c_s$ ist ungleich null. Man hat nun durch elementare Zeilentransformationen den ursprünglich in der x_s-Spalte von (4) stehenden Einheitsvektor zu rekonstruieren. Die x_s-Spalte ist dabei Pivotspalte, und die ursprünglich den Koeffizienten 1 des Einheitsvektors enthaltende Zeile wird zur Pivotzeile. Mit den Rechenregeln des Basistauschs (vgl. Schritt 3 in 2.2.2) stellt man die kanonische Form wieder her. Auf das nach Abschluß der Umformung vorliegende Tableau, das u.U. weder zulässig noch optimal ist,[112] wird die in Abschnitt 2.2.4 vorgestellte Berechnungsprozedur angewendet. Nach einigen Iterationen erhält man die Lösung des abgeänderten Problems.

110 Die c_j der Basisvariablen sowie die b_i wurden nicht geändert, und x^* ist wegen $x_s^* = 0$ weiterhin zulässige Lösung des Primalproblems.

111 Wegen $y^* = (d^T D^{-1})^T$ gilt $a_s^T y^* - c_s = d^T D^{-1} a_s - c_s$. D^{-1} und d sind unverändert, da x_s nicht in der Basis ist.

112 Vgl. Hillier/Lieberman, OR, S. 149 f., 153 ff.

Die Wiederherstellung des Einheitsvektors ist nicht möglich, falls das zur Umformung erforderliche Pivotelement null ist. Dies kommt vor, wenn die Spalten der Matrix D durch die Änderung von a_s linear abhängig geworden sind und somit keine Basis mehr bilden.[113] Der fehlende Einheitsvektor muß dann in einer anderen Tableauspalte rekonstruiert werden. Diese Rechenoperation ist gleichbedeutend mit dem Basisaustritt der Variablen x_s.
Abb. 3 faßt den gesamten Algorithmus zusammen.[114]

<table>
<tr><td colspan="3">Neuberechnung der x_s-Spalte</td></tr>
<tr><td colspan="3">Ist x_s Basisvariable?</td></tr>
<tr><td colspan="2">ja</td><td>nein</td></tr>
<tr><td colspan="2">Ist das Pivotelement 0?</td><td rowspan="3"></td></tr>
<tr><td>ja</td><td>nein</td></tr>
<tr><td>x_s aus Basis entfernen: Basistausch</td><td>kanonische Form wiederherstellen</td></tr>
<tr><td colspan="3">BERECHNUNG (Abb. 1)</td></tr>
</table>

Abb. 3: Änderung einer Koeffizientenspalte

Es bleibt noch zu zeigen, daß der Basisaustritt von x_s möglich ist. Ein Basistausch kann durchgeführt werden, wenn sich ein von 0 verschiedener Koeffizient in der Pivotzeile findet.[115] Diese Voraussetzung ist immer gegeben: Läßt man die x_s-Spalte außer Betracht, sind die Zeilenvektoren des Ausgangstableaus linear unabhängig (Einheitsmatrix in (3) !) und können somit nicht durch Simplexschritte in ein Vektorsystem überführt werden, das den

113 Vgl. 2.3.4.1 und Hillier/Lieberman, OR, S. 154, Fußnote 5.
114 Vgl. LinSen-Prozedur "TABLEAUKOEFF" und die Zahlenbeispiele im Anhang A.
115 Vgl. LinSen-Prozedur "AUSTRITT STRUKTURVAR".

Nullvektor enthält.[116] Es muß also auch außerhalb der x_s-Spalte in jeder Zeile wenigstens ein Nicht-Nullelement geben, w.z.z.w.
Der erzwungene Basistausch kann Zulässigkeit und Optimalität des Tableaus beeinflussen.

2.3.5 Änderung der Anzahl der Strukturvariablen

2.3.5.1 Hinzufügen einer Strukturvariablen

In das LO-Problem (1) werde nachträglich eine neue, bislang nicht berücksichtigte Strukturvariable x_{n+1} mit einem Zielfunktionsbeitrag von c_{n+1} und einer Koeffizientenspalte a_{n+1} eingeführt. Wie wirkt sich diese Änderung der Ausgangsdaten auf die optimale Lösung aus ?[117]
Mit $x_{n+1}:=0$ bleibt die bisherige Lösung zulässig; es stellt sich also ausschließlich die Frage nach ihrer Optimalität für das erweiterte Problem. Der Einführung einer neuen primalen Strukturvariablen entspricht das Hinzutreten einer weiteren Restriktion im Dualproblem (6): $a_{n+1}^T y \geq c_{n+1}$. Wird diese Restriktion von der bisherigen dualen Optimallösung y^* erfüllt, so sind die Voraussetzungen von Satz 1 gegeben, und die um $x_{n+1}=0$ erweiterte bisherige Lösung des Problems (1) bleibt optimal.[118] Es läßt sich leicht untersuchen, in welchen Grenzen die Koeffizienten der neuen Variablen schwanken dürfen, ohne daß die duale Restriktion verletzt wird.[119]
Wenn die neue duale Nebenbedingung nicht erfüllt

116 Vgl. die sehr ähnliche Überlegung in 2.2.4.
117 Vgl. Dinkelbach, Analysen, S. 83 f., Bloech, Änderung, S. 193, Gal, LO, S. 182 f.
118 Vgl. Hillier/Lieberman, OR, S. 144 f., 153: Die Überlegung ist dieselbe wie in 2.3.4.2; sie entspricht dem Vorgehen bei Änderung von Koeffizienten einer NBV. Der einzige Unterschied liegt darin, daß hier die zu ändernde Spalte nur Nullen enthält, weil x_{n+1} im alten Problem gar nicht vorkommt.
119 Vgl. Hillier/Lieberman, OR, S.145.

ist, so hat die zu ihr gehörende duale Schlupfvariable einen negativen Wert. Nach Satz 2 ist dieser Wert zugleich der Zielfunktionskoeffizient von x_{n+1} im bisherigen, um eine Spalte erweiterten Optimaltableau.[120] Der primale Simplexalgorithmus sorgt dann dafür, daß x_{n+1} in die Basis eintritt.
Durch die Einführung der neuen Variablen wird das alte Optimaltableau (4) wie folgt modifiziert: Für x_{n+1} ist eine Spalte einzufügen, deren Koeffizienten sich zu $D^{-1}a_{n+1}$ ergeben. D^{-1} (und d) können unverändert übernommen werden, da x_{n+1} nicht zur Basis gehört. Der Zielfunktionskoeffizient von x_{n+1} ist aus der Dualitäts-Vorüberlegung bekannt, läßt sich aber ebensogut direkt mit Hilfe von (4) berechnen und lautet $d^T D^{-1} a_{n+1} - c_{n+1}$.
Auf das erhaltene Tableau wird sodann der primale Simplexalgorithmus angewendet.[121]

2.3.5.2 Streichen einer Strukturvariablen

Als Umkehrung zu Abschnitt 2.3.5.1 stellt sich die Frage, welchen Einfluß der Wegfall einer Strukturvariablen x_s auf die optimale Lösung des LO-Problems hat.[122]
Ist x_s im bisherigen Optimaltableau Nichtbasisvariable, kann die zugehörige Spalte ersatzlos gestrichen werden.[123] Das verbleibende Tableau ist weiterhin zulässig, optimal und in kanonischer Form. Befindet sich die Variable x_s dagegen in der optimalen Basis, muß zunächst ihr Austritt erzwungen werden. Am Ende von Abschnitt 2.3.4.2 wurde gezeigt, daß immer ein geeignetes Pivotelement existiert. Nach Durchführung des Basistauschs kann die

120 Er läßt sich als mit -1 multiplizierte *wertmäßige Deckungsspanne* deuten, wenn x_{n+1} ein neues Produkt repräsentiert.
121 Vgl. 2.3.4.2 und LinSen-Prozedur "STRUKTURVAR".
122 Vgl. LinSen-Prozedur "STRUKTURVAR".
123 Im Dualproblem entfällt eine Restriktion. Aus Satz 1 folgt die Optimalität der alten Basis.

x_s-Spalte gestrichen werden. Das Resttableau ist in kanonischer Form, aber u.U. nicht zulässig und/oder nicht optimal. Durch Anwendung der Berechnungsprozedur (vgl. 2.2.4, Abb. 1) wird das reduzierte Problem gelöst.
Abschließend sei erwähnt, daß eine Strukturvariable auch einfach gestrichen werden kann, indem man sämtliche zu ihrer Spalte gehörenden Koeffizienten in (1) gleich null setzt. Insofern liegt ein Spezialfall des Abschnitts 2.3.4.2 vor.[124]

2.3.6 Änderung der Anzahl der Restriktionen

2.3.6.1 Hinzufügen einer Restriktion

Dem Problem (1) werde nachträglich eine Restriktion der Form $a^{m+1}x \leq b_{m+1}$ hinzugefügt. Eine Einschränkung des Raums der zulässigen Lösungen kann niemals zu einer Verbesserung des Zielfunktionswertes führen, sondern allenfalls die bisherige optimale Lösung x^* unzulässig machen.[125]
Erfüllt x^* die neue Restriktion, dann bleibt die bisherige Basislösung für das erweiterte Problem zulässig und damit optimal. Dies ergibt sich wieder unmittelbar aus Satz 1: Durch die Einführung einer zusätzlichen primalen Restriktion tritt im Dualproblem (6) eine neue Variable y_{m+1} hinzu. Man ergänzt die alte duale Optimallösung y^* um $y_{m+1}{}^*=0$ und erkennt, daß dann x^* und y^* die Voraussetzungen des Optimalitätssatzes erfüllen.
Um die neue Restriktion im Optimaltableau (4) zu berücksichtigen, kann man die um eine Zeile und

124 Die Spalte enthält dann in jedem Tableau den Nullvektor und kann durch Simplexschritte nicht mehr verändert werden. Die Mitführung einer überflüssigen Spalte kostet aber Rechenzeit und Speicherplatz, weshalb eine gesonderte Behandlung in LinSen gerechtfertigt ist.

125 Vgl. Dürr/Kleibohm, OR, S.89, Hillier/Lieberman, OR, S. 155.

eine Spalte verlängerte Basismatrix mittels einer Zerlegung invertieren und die zusätzliche Zeile von (4) direkt berechnen.[126] Hier soll jedoch ein anderer Weg beschritten werden, der meist schneller zum Ziel führt.[127]

Die neue Restriktion wird durch eine Schlupfvariable s_{m+1} in Gleichungsform überführt und dem bisherigen Optimaltableau als Zeile hinzugefügt. Für s_{m+1} ist eine Spalte einzuführen, die in allen übrigen Zeilen nur Nullen enthält, so daß die neue Schlupfvariable Basisvariable im erweiterten Tableau wird. Anschließend stellt man die kanonische Form wieder her, indem zur neuen Zeile geeignete Vielfache der anderen Tableauzeilen addiert werden.[128] Die Vorgehensweise entspricht dem am Ende von Abschnitt 2.3.2.2 beschriebenen Verfahren (Änderung der Zielfunktionszeile). Nun kann der Wert der Basisvariablen s_{m+1} abgelesen werden. Ist er nichtnegativ, bleibt das bisherige Tableau zulässig und optimal; die alte Basislösung erfüllt die neue Restriktion. Andernfalls wird der duale Simplexalgorithmus auf das vorliegende Tableau angewendet.

2.3.6.2 Streichen einer Restriktion

Abschließend soll untersucht werden, wie die optimale Lösung auf den Wegfall einer Restriktion im Problem (1) reagiert.

Im Optimaltableau (4) muß sowohl eine Zeile als auch die zur gestrichenen Restriktion gehörende Schlupfvariable s_r eliminiert werden. Dies ist leicht zu erreichen, wenn s_r in der bisherigen Lösung Basisvariable ist. Man streicht einfach die einzige s_r enthaltende Zeile sowie die s_r-Spalte im

126 Vgl. Dinkelbach, Analysen, S. 85-87.
127 Vgl. LinSen-Prozedur "RESTRIKTIONEN".
128 Vgl. Dürr/Kleibohm, OR, S. 90 f., Gal, LO, S. 183 ff., Hillier/Lieberman, OR, S. 155 ff., Stepan/Fischer, Optimierung, S. 100 f.

Endtableau. Das reduzierte Tableau bleibt zulässig, optimal und in kanonischer Form.[129] Wenn jedoch die Variable s_r Nichtbasisvariable ist, muß vorab ihr Basiseintritt erzwungen werden.[130] Erst danach ist s_r aus allen anderen Gleichungen eliminiert, so daß Zeile und Spalte entfallen können.[131] Um den Basistausch vornehmen zu können, wird ein von 0 verschiedener Koeffizient in der s_r-Spalte benötigt. Die Annahme, es gäbe kein Pivotelement, führt auf einen Widerspruch: Enthielte die Spalte einer Schlupfvariablen in (4) nur Nullen, wäre die Matrix D^{-1} singulär! Die Schlupfvariable s_r kann also stets zum Eintritt in die Basis gezwungen werden. Das nach dem Streichen von Zeile und Spalte verbleibende Tableau ist u.U. nicht optimal und/oder nicht zulässig. Daher wird die Rechnung mit dem Algorithmus aus Abb. 1 (vgl. 2.2.4) fortgesetzt, bis die Lösung des reduzierten Problems gefunden ist.

129 Ist s_r BV, so hat die wegfallende duale Variable y_r in der bisherigen dualen Optimallösung den Wert 0 (Satz 2). Dann bleibt die alte primale Basislösung optimal (Satz 1).

130 Vgl. LinSen-Prozedur "EINTRITT SCHLUPFVAR".

131 Vgl. LinSen-Prozedur "RESTRIKTIONEN".

2.4 Ergänzungen zur Sensibilitätsanalyse

2.4.1 Alternative Optimaltableaus

In den Abschnitten 2.3.2.1 und 2.3.3.1 ist deutlich geworden, daß bei primaler oder dualer Ausartung mehrere Optimaltableaus existieren können.
Betrachtet wird ein zulässiges und optimales Tableau (5). Besitzt das LO-Problem verschiedene optimale Lösungen, so muß (5) dual ausgeartet sein.[132] Nichtbasisvariable mit dem Zielfunktionskoeffizienten 0 können dann entweder in die Basis eintreten (wodurch man zu einem alternativen Optimaltableau gelangt) oder einen beliebigen positiven Wert annehmen (wenn es kein positives Pivotelement gibt).[133] Die LinSen-Prozedur "ALTERNENDTABLEAU" erfragt den Namen der Nichtbasisvariablen und führt - falls möglich - ihren Basiseintritt aus. Das erhaltene Tableau kann anschließend der Sensibilitätsanalyse unterzogen werden. Auf diesem Weg läßt sich jede existierende optimale Lösung finden und auf Stabilität untersuchen.
Bei primaler Ausartung kann zu ein und derselben optimalen Lösung x^* mehr als ein Optimaltableau gehören. LinSen erfragt den Namen der Basisvariablen mit dem Wert 0, die aus der Basis austreten soll. Wenn ein negatives Pivotelement vorliegt, wird der Basistausch durch einen dualen Simplexschritt vollzogen. Auf diese Weise lassen sich alle zu einer optimalen Lösung x^* gehörenden Optimaltableaus ermitteln. Die Kenntnis dieser Tableaus ist für die

132 Vgl. 2.3.2.1 und Krekó, Lehrbuch, S. 215. Liegt keine duale Ausartung vor, so ist die optimale *Lösung* x^* eindeutig. Die optimale *Basis* braucht dagegen nicht eindeutig zu sein, da möglicherweise primale Ausartung vorliegt und ein die Lösung x^* nicht verändernder Basistausch durchgeführt werden kann.
133 Vgl. van de Panne, LP, S. 98-101.

Berechnung isolierter Schwankungsbreiten zwingend erforderlich (vgl. 2.3.2.1, 2.3.3.1).

2.4.2 Ganzzahligkeit

Zwei Gründe sprechen dafür, in das Programm LinSen eine Prozedur "GANZZAHLIGKEIT" aufzunehmen. Einerseits kann das (nachträgliche) Hinzutreten der Ganzzahligkeitsforderung für einzelne Variable als Änderung der Ausgangsdaten interpretiert werden, deren Auswirkung auf die Lösung des LO-Problems zu untersuchen ist. Es liegt also eine der Sensibilitätsanalyse ähnliche Fragestellung vor. Andererseits besteht offenkundige rechentechnische Verwandtschaft zu bestimmten Verfahren der Sensibilitätsanalyse: Sowohl die Schnittebenen- als auch die Entscheidungsbaumverfahren der ganzzahligen Optimierung arbeiten mit dem nachträglichen Hinzufügen (und ggf. Streichen) von Restriktionen, so daß Vorgehensweisen und Ergebnisse des Abschnitts 2.3.6 anwendbar sind.[134]
LinSen enthält den Algorithmus von Gomory zur Lösung gemischt-ganzzahliger LO-Probleme.[135] Aus hier nicht darzustellenden Gründen verursacht dieser theoretisch elegante Algorithmus erhebliche numerische Schwierigkeiten und vermag im Rahmen von LinSen nur Probleme mit sehr wenigen ganzzahligen Variablen richtig zu lösen.[136] Er ist aber hervorragend geeignet, an einfachen Beispielen die Auswirkungen von Ganzzahligkeitsforderungen zu verdeutlichen, ohne den Programmumfang und den Speicherplatzbedarf wesentlich zu vergrößern.

134 Vgl. Müller-Merbach, OR, S. 368 f.
135 Der Beweis des Algorithmus findet sich z.B. in Burkard, Methoden, S. 158-160.
136 Vgl. etwa Burkard, Ganzzahlige Optimierung, S. 407 f., Schmitz/Schönlein, LO-Modelle, S. 96.

Nach Abschluß des Gomoryverfahrens ist eine Sensibilitätsanalyse im Sinne der eingangs getroffenen Definition nicht mehr sinnvoll, da das Optimaltableau durch die Schnittrestriktionen "verzerrt" wird: Zielfunktionskoeffizienten (Schattenpreise und wertmäßige Deckungsspannen) hängen von Anzahl und Verlauf der eingeführten Schnittebenen ab und geben nicht mehr die tatsächlichen Grenzgewinne wieder; Engpaßrestriktionen weisen plötzlich positive Schlupfvariablenwerte auf und erscheinen redundant.[137] Ein gemischt-ganzzahliges Problem ist kein LO-Problem im Sinne der Definition (1) mehr. Dies hat zur Folge, daß die Verfahren des Kapitels 2.3 nicht länger anwendbar sind. Im Programm LinSen wird deshalb die Hauptmenü-Option "Sensibilitätsanalyse" gesperrt, sobald der gemischt-ganzzahlige Algorithmus zum Einsatz kommt.

137 Vgl. v. Zwehl, Ganzzahlige Programmierung, S. 359.

3. Schlußbemerkungen

Kenntnis und Einsatz mathematischer Planungsverfahren werden durch die Verfügbarkeit standardisierter EDV-Programme entscheidend gefördert.[138] Eine sinnvolle Weiterentwicklung von LinSen könnte daher in der Verwendung des Standard-MPS-Formats für die Eingabedaten bestehen.[139] Darüber hinaus ist zur Behandlung größerer LO-Probleme sowie zur Erhöhung der Rechengenauigkeit die Einführung der revidierten Simplexmethode empfehlenswert.

Fast alle in dieser Arbeit dargestellten Algorithmen sind in das Programm LinSen aufgenommen worden. Damit ist es möglich, LO-Probleme einer umfassenden "postoptimalen" Analyse zu unterziehen und in vielfältiger Weise mit dem mathematischen Modell bzw. seinen Ausgangsdaten zu experimentieren. Die Sensibilitätsanalyse erweist sich als geeignetes Instrument, um die Starrheit exakter Optimierungsverfahren zu überwinden und Datenunsicherheit in der Planung zu berücksichtigen.[140] Sie sollte daher in keinem Standardprogramm zur linearen Optimierung fehlen.

138 Vgl. Scheer, EDV-orientierte BWL, S. 154.
139 Vgl. Schmitz/Schönlein, LO-Modelle, S. 55 ff.
140 Vgl. Dinkelbach, Analysen, S. VI.

ANHANG A

Anhang A: Betriebswirtschaftliches Anwendungsbeispiel zur Sensibilitätsanalyse

Die Möglichkeiten der Sensibilitätsanalyse sollen im folgenden anhand eines einfachen Zahlenbeispiels veranschaulicht werden, dem das betriebswirtschaftliche Problem der Produktionsprogrammplanung zugrunde liegt.[1] Dem Leser sei empfohlen, die einzelnen Schritte des Beispiels mit LinSen nachzuvollziehen.

Auf die ökonomische Interpretation des Simplexalgorithmus sowie einzelner Tableaus kann an dieser Stelle nicht eingegangen werden.[2] Das Beispiel soll ausschließlich die Verfahren des Kapitels 2.3 erläutern und betriebswirtschaftliche Anwendungsmöglichkeiten aufzeigen.

Ein Unternehmen produziert zwei Produkte P_1 und P_2 in den Mengen x_1 und x_2. Planungszeitraum sei jeweils ein Monat. P_1 und P_2 müssen die beiden Abteilungen A und B durchlaufen, die über eine monatliche Kapazität von 130 bzw. 110 Zeiteinheiten (ZE) verfügen. Der Zeitbedarf zur Bearbeitung einer Mengeneinheit (ME) des Produkts $P_1(P_2)$ beträgt in Abteilung A 2(3), in B 1(4) ZE. Von P_1 können monatlich maximal 50 ME abgesetzt werden. Da die Rohstoffpreise Schwankungen unterliegen, beruhen die Deckungsspannen der Produkte auf Schätzungen. Sie betragen 40 bzw. 30 DM/ME. Gesucht ist das gewinnmaximale Produktionsprogramm. Die Aufgabe lautet:

$$\begin{aligned}
&\max.\ G;\ G := 40x_1 + 30x_2 \\
&\qquad 2x_1 + 3x_2 \le 130 \\
&\qquad x_1 + 4x_2 \le 110 \\
&\qquad x_1 \le 50 \\
&\qquad x_1,\ x_2 \ge 0
\end{aligned} \tag{7}$$

1 Vgl. Adam, Produktionspolitik, S. 366 ff., Dinkelbach/Lorscheider, Entscheidungsmodelle, S. 24-32.
2 Vgl. hierzu Haupt/Wegener, Inhalt, S. 8-14.

Das Optimaltableau des LO-Problems (7) erhält man nach zwei primalen Simplexschritten:

BV	x_1	x_2	s_1	s_2	s_3	RS
G	0	0	10	0	20	2300
x_2	0	1	0,33	0	-0,67	10
s_2	0	0	-1,33	1	1,67	20
x_1	1	0	0	0	1	50

(8)

Die optimale Lösung ist $x^* = (50; 10)^T$.

Zu 2.3.2:

Ersetzt man c_1=40 und c_2=30 durch 40+t_1 bzw. 30+t_2, so lauten die Zielfunktionskoeffizienten der Strukturvariablen in (8) $-t_1$ bzw. $-t_2$. Zur Wiederherstellung der kanonischen Form ist die x_1-Zeile t_1-mal und die x_2-Zeile t_2-mal zur G-Zeile zu addieren. Die Zielfunktionskoeffizienten der Nichtbasisvariablen s_1 und s_3 bleiben nichtnegativ, wenn gilt: $10 + 0{,}33t_2 \geq 0$ und $20 + t_1 - 0{,}67t_2 \geq 0$. Der durch dieses Ungleichungssystem festgelegte kritische Bereich ist in Abb. 2 (S. 25) graphisch dargestellt. Die isolierten Schwankungsintervalle ergeben sich aus den Schnittmengen des kritischen Bereichs mit den Koordinatenachsen (jeweils ein t_j wird zu 0 vorgegeben): $t_1 \in [-20;+\infty[$, $t_2 \in [-30;30]$ bzw. $c_1 \in [20;+\infty[$, $c_2 \in [0;60]$. Man kann natürlich auch gleich die in 2.3.2.1 hergeleiteten Formeln heranziehen.

Die Geschäftsleitung geht davon aus, daß die Schwankungen der Deckungsspannen auf jeden Fall innerhalb des kritischen Bereichs liegen werden. Das optimale Programm wird daher verwirklicht.

Durch einen Vertrag mit dem Rohstofflieferanten lassen sich die c_j im Folgemonat präzise bestimmen: c_1=38, c_2=32. Die Basislösung bleibt optimal, da $(t_1, t_2)=(-2;2)$ im kritischen Bereich (Abb. 2) liegt. Man bestätigt dies durch Neuberechnung der Zielfunktionszeile in (8). Die neue G-Zeile lautet zunächst:

BV	x_1	x_2	s_1	s_2	s_3	RS	
G	-38	-32	0	0	0	0	(8)'

Um die kanonische Form wiederherzustellen, müssen die Koeffizienten der Basisvariablen x_1 und x_2 null werden. Man addiert zu (8)' 38-mal die x_1- und 32-mal die x_2-Zeile von (8) und erhält:

BV	x_1	x_2	s_1	s_2	s_3	RS	
G	0	0	10,67	0	16,67	2220	(9)

<u>Zu 2.3.3:</u>

Eine zusätzliche ZE der Abteilung A würde den Gewinn um 10,67 DM erhöhen. Die Kapazität ließe sich um 12 ZE aufstocken. Durch Berechnung der isolierten Schwankungsbreite von b_1 stellt man fest, daß die bisherige Basis mit $b_1^{neu}=130+12=142$ optimal bleibt: $b_1 \in [100;145]$ ($b_2 \in [90;+\infty[$, $b_3 \in [38;65]$). Der Gewinn steigt um $10,67 \cdot 12=128$ auf 2348 DM. Das Optimaltableau wird durch Neuberechnung der rechten Seite (RS) angepaßt:

$$D^{-1}b^{neu} = \begin{bmatrix} 0,33 & 0 & -0,67 \\ -1,33 & 1 & 1,67 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 142 \\ 110 \\ 50 \end{bmatrix} = \begin{bmatrix} 14 \\ 4 \\ 50 \end{bmatrix}$$

Das Optimaltableau lautet nun:

BV	x_1	x_2	s_1	s_2	s_3	RS	
G	0	0	10,67	0	16,67	2348	
x_2	0	1	0,33	0	-0,67	14	(10)
s_2	0	0	-1,33	1	1,67*	4	
x_1	1	0	0	0	1	50	

<u>Zu 2.3.6.2:</u>

Im Folgemonat wird mit einer Verbesserung der Absatzlage gerechnet. Wie ändert sich die optimale Lösung, wenn die Restriktion $x_1 \leq 50$ wegfällt ? s_3 ist NBV und muß zum Eintritt in die Basis gezwungen werden. Das Pivotelement ist in (10) mit einem Sternchen (*) gekennzeichnet. Man erhält:

BV	x_1	x_2	s_1	s_2	s_3	RS	
G	0	0	24	-10	0	2308	
x_2	0	1	-0,2	0,4*	0	15,6	(11)
s_3	0	0	-0,8	0,6	1	2,4	
x_1	1	0	0,8	-0,6	0	47,6	

Zeile und Spalte von s_3 werden gestrichen. Ein primaler Simplexschritt liefert das Optimaltableau:

BV	x_1	x_2	s_1	s_2	RS	
G	0	25	19	0	2698	(12)
s_2	0	2,5	-0,5	1	39	
x_1	1	1,5	0,5	0	71	

Das LO-Problem lautet nach diesen Veränderungen:

$$
\begin{aligned}
\text{max. } G;\ G &:= 38x_1 + 32x_2 \\
2x_1 + 3x_2 &\le 142 \\
x_1 + 4x_2 &\le 110 \\
x_1,\ x_2 &\ge 0
\end{aligned}
\qquad (13)
$$

Zu 2.3.5.1:

Da die Fertigung von P_2 sich nicht mehr lohnt, wird ein Ersatzprodukt P_3 entwickelt. Es verbraucht in beiden Abteilungen je 2 ZE/ME Kapazität und besitzt eine Deckungsspanne von 45 DM/ME. Ergänzt man (13) um die Variable x_3, so tritt im Dualproblem die Restriktion $2y_1 + 2y_2 \geq 45$ hinzu. Sie wird von der aus (12) ablesbaren bisherigen dualen Optimallösung $y^* = (19;0)^T$ nicht erfüllt. Mithin bleibt die alte primale Basislösung (12) nicht optimal, d.h. x_3 weist einen negativen Zielfunktionskoeffizienten auf: Der Wert der dualen Schlupfvariablen in der neuen dualen Restriktion $2y_1 + 2y_2 - z_3 = 45$ beträgt $z_3 = 2 \cdot 19 + 2 \cdot 0 - 45 = -7 < 0$. Man überzeugt sich davon durch direkte Berechnung des Zielfunktionskoeffizienten:

$$
D^{-1}a_3 = \begin{bmatrix} -0,5 & 1 \\ 0,5 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},
$$

$$d^T D^{-1} a_3 - c_3 = 0 \cdot 1 + 38 \cdot 1 - 45 = -7\ !$$

Tableau (12) ist um die Spalte $(x_3\ -7\ 1\ 1)^T$ zu ergänzen. Ein primaler Simplexschritt liefert das neue Optimaltableau:

BV	x_1	x_2	x_3	s_1	s_2	RS	
G	0	42,5	0	15,5	7	2971	(14)
x_3	0	2,5	1	-0,5	1	39	
x_1	1	-1	0	1	-1^*	32	

Zu 2.3.4.2, Fall "BV und Pivotelement 0":

Im nächsten Monat wird ein Teil der Produktion von P_1 an eine nicht ausgelastete Abteilung C abgegeben. Als Folge sinkt der Produktionskoeffizient von x_1 in Abteilung A von 2 auf 1 ZE/ME. Alle übrigen Koeffizienten bleiben unverändert. Da x_1 Basisvariable ist, muß geprüft werden, ob die Basismatrix nach der Änderung von a_{11} noch eine Inverse besitzt. Man sieht sofort, daß

$$D^{neu} = (a_3\ a_1^{neu}) = \begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix}$$

singulär ist (die Spalten sind linear abhängig). x_1 und x_3 können folglich nicht gemeinsam Basisvariable sein; a_3 und a_1^{neu} bilden keine Basis des R^2 mehr.

LinSen erkennt diesen Fall daran, daß sich der in (14) zu x_1 gehörende Einheitsvektor nach der Neuberechnung der x_1-Spalte nicht wiederherstellen läßt: Wendet man auf a_1^{neu} dieselben Zeilentransformationen an, die das Tableau (14) erzeugten, so ergibt sich die neue x_1-Spalte in (14) zu

$$D^{-1}a_1^{neu} = \begin{bmatrix} -0,5 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0,5 \\ 0 \end{bmatrix},$$

$d^T D^{-1} a_1^{neu} - c_1 = (45;38) \cdot (0,5;0)^T - 38 = -15,5.$

Das zur Rekonstruktion der kanonischen Form erforderliche Pivotelement ist 0 ! Es kann nicht durch Multiplikation mit einer reellen Zahl wieder zu 1 werden. Der fehlende Einheitsvektor muß also in einer anderen Spalte des Tableaus rekonstruiert werden. Anders gesagt: x_1 verläßt die Basis (vgl. Abb. 3, S. 35). Ein Basistausch mit dem in (14) ausgewiesenen Pivotelement ergibt:

BV	x_1	x_2	x_3	s_1	s_2	RS
G	-15,5	35,5	0	22,5	0	3195
x_3	0,5	1,5	1	0,5	0	71
s_2	0	1	0	-1^*	1	-32

(15)

Das Tableau (15) ist weder zulässig noch optimal. Durch Basistausch von s_2 gegen s_1 wird die Zulässigkeit wiederhergestellt. Ein primaler Simplexschritt liefert dann das neue Optimaltableau:

BV	x_1	x_2	x_3	s_1	s_2	RS	
G	0	120	31	0	38	4180	(16)
x_1	1	4	2	0	1	110	
s_1	0	-1	0	1	-1	32	

Zu 2.3.5.2:

P_2 verbraucht mehr knappe Faktoren und erbringt weniger Deckungsbeitrag als P_1 und P_3. Die wertmäßige Deckungsspanne (Grenzgewinn von P_2) beträgt -120 DM/ME. Das Unternehmen beschließt daher, dieses Produkt zu streichen. Weil x_2 NBV ist, entfällt einfach die zugehörige Spalte in (16).

Zu 2.3.4.2, Fall "BV und Pivotelement $\neq 0$":

Im Folgemonat wird eine verbesserte Version von P_1 eingeführt. Sie besitzt eine um 2 DM/ME höhere Deckungsspanne und verbraucht in Abteilung A 1,2 ZE/ME. Das nun vorliegende LO-Problem lautet also:

$$\begin{aligned} &\max.\ G;\ G := 40x_1 + 45x_3 \\ &\quad 1{,}2x_1 + 2x_3 \le 142 \\ &\quad x_1 + 2x_3 \le 110 \\ &\quad x_1,\ x_3 \ge 0 \end{aligned} \qquad (17)$$

Tableau (16) ist zu revidieren. Man berechnet

$$D^{-1}a_1^{neu} = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1{,}2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0{,}2 \end{bmatrix},$$

$$d^T D^{-1} a_1^{neu} - c_1^{neu} = (38;0)\cdot(1;0{,}2)^T - 40 = -2.$$

Wichtig: d und D^{-1} sind gegenüber (16) noch unverändert. Die neue x_1-Spalte muß denselben Zeilentransformationen unterzogen werden, die das Tableau (16) erzeugten!

Man erhält:

BV	x_1	x_3	s_1	s_2	RS	
G	-2	31	0	38	4180	(18)
x_1	1^*	2	0	1	110	
s_1	0,2	0	1	-1	32	

Das Tableau (18) ist nicht in kanonischer Form. Im Gegensatz zur letzten Änderung der x_1-Spalte (vgl. S. 50 f.) läßt sich nun der Einheitsvektor der BV x_1 rekonstruieren: Die x_1-Zeile kann durch das Pivotelement (hier $1 \neq 0$) dividiert werden. Anschließend addiert man sie zweimal zur G-Zeile und subtrahiert sie 0,2-mal von der s_1-Zeile. Das neue Tableau in kanonischer Form ist (hier) bereits zulässig und optimal:[3]

BV	x_1	x_3	s_1	s_2	RS
G	0	35	0	40	4400
x_1	1	2	0	1	110
s_1	0	-0,4	1	-1,2	10

(19)

<u>Zu 2.3.4.2, Fall "NBV":</u>

Inzwischen ist es gelungen, den Produktionsprozeß von P_3 zu verbessern. Während der Zeitbedarf in Abteilung A geringfügig (von 2 auf 2,2 ZE/ME) ansteigt, kann der Kapazitätsbedarf in der Engpaßabteilung B halbiert werden (von 2 auf 1 ZE/ME). Allerdings verringert sich wegen des aufwendigeren Produktionsverfahrens die Deckungsspanne von P_3 um 5 DM/ME auf 40 DM/ME.

Der Änderung einer Spalte von (17) entspricht die Änderung einer Restriktion des zu (17) dualen Problems. Die modifizierte duale Nebenbedingung lautet $2{,}2y_1 + y_2 \geq 40$ und wird von der in (19) ablesbaren bisherigen optimalen Lösung $y^* = (0;40)^T$ gerade noch erfüllt. Dann bleibt die Basislösung von (19) optimal (Satz 1, S. 19), und es gilt weiterhin $x^* = (x_1^*, x_3^*)^T = (110;0)^T$.

Dieses unter Ausnutzung der Dualitätstheorie leicht gefundene Ergebnis läßt sich bestätigen, indem man die x_3-Spalte in (19) neu berechnet und feststellt, daß sich tatsächlich ein Zielfunktionskoeffizient

3 Die alte *Lösungsstruktur* (Menge der Basisvariablen) bleibt optimal, nicht aber die alte *Basis* (Matrix D wurde geändert!) oder die alte *Basislösung* (Variablenwerte änderten sich).

von 0 ergibt und das Tableau also nach wie vor optimal ist.

$$D^{-1}a_3^{neu} = \begin{bmatrix} 0 & 1 \\ 1 & -1,2 \end{bmatrix} \cdot \begin{bmatrix} 2,2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

$$d^T D^{-1} a_3^{neu} - c_3^{neu} = (40;0)\cdot(1;1)^T - 40 = 0 \geq 0.$$

<u>Zu 2.3.6.1:</u>

Zum Abschluß soll demonstriert werden, wie sich mit Hilfe der Sensibilitätsanalyse eine "sekundäre Zielfunktion" berücksichtigen läßt.[4]

Die optimale Lösung des vorliegenden Problems ist nicht eindeutig (duale Ausartung). Das Unternehmen möchte nun unter den gewinnmaximalen Lösungen diejenige ermitteln, die zum größtmöglichen Umsatz führt. $P_1(P_3)$ wird zu einem Preis von 90(100) DM/ME angeboten. Die Zielfunktion ist im bisherigen Endtableau ((19) mit geänderter x_3-Spalte) durch die Umsatzfunktion $U - 90x_1 - 100x_3 = 0$ zu ersetzen. Da der Zielfunktionskoeffizient der Basisvariablen x_1 null sein muß, addiert man zur Wiederherstellung der kanonischen Form die x_1-Zeile 90-mal zur U-Zeile und erhält:

BV	x_1	x_3	s_1	s_2	RS
U	0	-10	0	90	9900
x_1	1	1	0	1	110
s_1	0	1	1	-1,2	10

(20)

Der maximale Gewinn darf allerdings nicht unterschritten werden. Man fügt deshalb die Restriktion $G \geq 4400$ in folgender Form an das Tableau (20) an: $40x_1 + 40x_3 \geq 4400 \iff -40x_1 - 40x_3 + s_3 = -4400$. Zur Anpassung an die kanonische Form von (20) muß die x_1-Zeile 40-mal zu dieser Gleichung addiert werden:

BV	x_1	x_3	s_1	s_2	s_3	RS
U	0	-10	0	90	0	9900
x_1	1	1	0	1	0	110
s_1	0	1^*	1	-1,2	0	10
s_3	0	0	0	40	1	0

(21)

4 Vgl. <u>Müller-Merbach</u>, OR, S. 157.

Ein primaler Simplexschritt liefert das Optimaltableau:

BV	x_1	x_3	s_1	s_2	s_3	RS
U	0	0	10	78	0	10000
x_1	1	0	-1	2,2	0	100
x_3	0	1	1	-1,2	0	10
s_3	0	0	0	40	1	0

(22)

Sicherheitshalber sei abschließend das zu (22) gehörende LO-Problem angeschrieben:

$$\begin{aligned} &\max.\ U;\ U := 90x_1 + 100x_3 \\ &\quad 1{,}2x_1 + 2{,}2x_3 \le 142 \\ &\quad x_1 + x_3 \le 110 \\ &\quad 40x_1 + 40x_3 \ge 4400 \\ &\quad x_1,\ x_3 \ge 0 \end{aligned} \tag{23}$$

Auch wenn viele Ausgangsdaten nicht exakt quantifizierbar sind, läßt sich in der gezeigten Weise mit dem Modell experimentieren. Man kann dann nach einer Lösungsstruktur (Menge von Basisvariablen) suchen, die für möglichst viele wahrscheinliche Datenkonstellationen optimal ist, ohne nach jeder Abwandlung des Datensatzes das LO-Problem von Anfang an neu lösen zu müssen. Die damit verbundene erhebliche Zeitersparnis unterstreicht die Bedeutung der Sensibilitätsanalyse für den effizienten Einsatz der linearen Optimierung.

ANHANG B

Anhang B: Struktur des Programms LinSen

Die beigefügte Programmdiskette enthält neben den Pascal-Quelltextdateien LINSEN.PAS, AUSGABE.PAS, DEF.PAS sowie den kompilierten Bausteinen AUSGABE.TPU und DEF.TPU die ausführbare Programmdatei LINSEN.EXE. Sie wird durch Eingabe des Worts LinSen gestartet. Um mit dem Programm zu arbeiten, wird allein die EXE-Datei benötigt. Zugriffe auf externe Speicher erfolgen lediglich zum Sichern und Laden von LO-Problemen. Die vom Benutzer zu vergebenden Dateinamen müssen dem MS-DOS-Standard entsprechen und sollten mit der Endung ".LNS" versehen werden.[1]

Da LinSen die Größe von 64 KB überschreitet, ist eine Aufspaltung des Quelltextes in drei Teile vorgenommen worden, Abb. 4:[2]

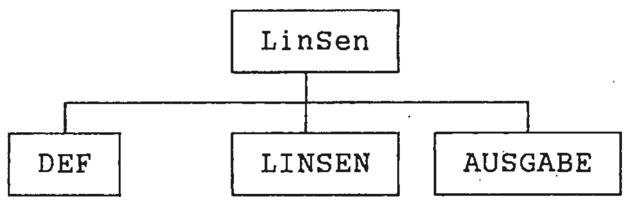

Abb. 4: Grobstruktur von LinSen

Die folgenden Abbildungen geben den Inhalt der wichtigsten Prozeduren wieder und sollen den Zugang zum Pascal-Quelltext (Anhang D) erleichtern.

1 Die Vorschriften für Dateinamen findet man z.B. in Schieb, DOS, S. 50 ff.
2 Zur Zerlegung in "Units" vgl. Herschel, Pascal, S. 247 ff.

DEF

- Definitionen: Konstante, Typen, Variable
- Prozeduren: Lies ganze Zahl
  PARAMETER
- Funktionen: SGN
  RUND

Abb. 5: Inhalt von DEF

AUSGABE

Prozeduren: DRUCKPROBE
AUFGABE AUSGEBEN
TABLEAU AUSGEBEN
BASISLSG AUSGEBEN
AUSGABEMENÜ
ERGEBNISAUSGABE
AUFGABE SPEICHERN
PROGRAMMSTART

Abb. 6: Inhalt von AUSGABE

LINSEN

Prozeduren: NEU
Initialisierung
Hauptmenü
Lies reelle Zahl
GLATT
Prüfe ZUL
Prüfe OPT
LÖSCH
EINGZLFKT
EINGABE
BASISTAUSCH
PRIMAL PIVOTZEILE
DUAL PIVOTSPALTE
BERECHNUNG
SENSIBILITÄTSANALYSE

Hauptprogramm

Abb. 7: Inhalt von LINSEN

BERECHNUNG

Prozeduren: Prüfe Ausartung
DUAL PIVOTZEILE
PHASE I
PHASE II
DUAL SIMPLEX
DUAL HILF

Berechnungsprogramm (Abb. 1)

Abb. 8: Inhalt von BERECHNUNG

SENSIBILITÄTSANALYSE

Prozeduren: BERECHNE **d**
AUSTRITT STRUKTURVAR
EINTRITT SCHLUPFVAR
LIES SPALTE
NEUBERECHNUNG ZLFKT
ZIELFKTKOEFF
RECHTE SEITEN
TABLEAUKOEFF
STRUKTURVAR
RESTRIKTIONEN
GANZZAHLIGKEIT
ALTERNENDTABLEAU

Menü

Abb. 9: Inhalt von SENSIBILITÄTSANALYSE

ANHANG C

Anhang C: Variablenverzeichnis von LinSen

Das nachstehende Verzeichnis enthält alle im Programm vorkommenden Variablennamen in alphabetischer Reihenfolge. Stichwortartige Angaben kennzeichnen den Ort der Definition und die Bedeutung jeder Variablen. Konstante, Typen, Übergabeparameter, Prozeduren und Funktionen sind nicht aufgeführt, können aber mit Hilfe des Anhangs B leicht gefunden werden.

Variable	Herkunft	Bedeutung
A	DEF	Matrix des Simplextableaus
Aktuelle_Sp.	Lies ...	aktuelle Spalte
Aktuelle_Z.	Lies ...	aktuelle Zeile
Alle_Schritte	DEF	Ausgabe aller Tableaus
AltT	ALTERN.	Menüpunkt ALTERNENDTABLEAU
Antwort	DEF	Tastatureingabe
Art	DEF	Typ der Restriktion
Ausgabeart	DEF	Bildschirm oder Drucker
ausgeartet	DEF	primal ausgeartet
b	DEF	Tableauspalte RS
bc	DEF	Vektor der rechten Seiten
BV	DEF	Vektor der Basisvariablen
BZ	DEF	Basiszeile einer Variablen
C	DEF	Matrix der Ausgangsdaten
d	SENSIB.	Vektor **d**
Datei	DEF	Dateivariable
Dateiname	DEF	DOS-Dateiname
Druck	ZIELFKT.	≈ Ausgabeart
Drucker_ber.	DEF	Druckerstatus (bereit)
DUAL	DEF	dualer Simplexalgorithmus
EAFehler	DEF	Ein-/Ausgabe-Fehler
eindeutig	DEF	optimale Lösung eindeutig
Eingabeart	EINGABE	Tastatur oder Laufwerk
Eingabezahl	Lies ...	Eingabezahl
EingName	EINGABE	eingegebener Name
FERTIG	GANZZAHL.	ganzzahlige Optimallösung
g	GANZZAHL.	Ganzzahligkeitsforderung
GANZ	DEF	Gomoryverfahren
Ganze_Zahl	DEF	ganze Zahl
GG	DEF	Anzahl ≥-Restriktionen
GL	DEF	Anzahl =-Restriktionen

GrGl	DEF	1, falls ≥-Restriktion(en)
H	DEF	Hilfszielfunktionszeile
h1	DEF	≈ STNAMEN
h2	DEF	≈ VARNAMEN
h3	DEF	≈ HILF
h4	DEF	≈ Keine_Restr
Haupt	DEF	Hauptmenüpunkt
Hb0	DEF	Hilfszielfunktionswert
HILF	DEF	Hilfszielfunktion aktiv
i	DEF	Index (meist Zeile)
IT	DEF	Anzahl Iterationen
j	DEF	Index (meist Spalte)
k	DEF	Index
Keine_Restr	DEF	keine Restriktionen
KG	DEF	Anzahl ≤-Restriktionen
Kode	DEF	Eingabe-Fehlerkode
KV	DEF	Anzahl künstliche Variable
KV_in_Basis	PHASE I	künstliche Basisvariable
l	DEF	Index
LEERLSG	DEF	keine zulässige Lösung
M	AUSGABE	Ausgabemenüpunkt
m	DEF	Anzahl Restriktionen
Max	DEF	Maximum
Max	EINGABE	Zeilenindex
MaxMin	DEF	+(-)1 für min.(max.)
Min	DEF	Minimum
n	DEF	Anzahl Variable insgesamt
Name	SENSIB.	eingegebener Variablenname
Namensvektor	DEF	Vektor der Variablennamen
Neue_Aufgabe	DEF	neue Aufgabe
Nr	SENSIB.	Variablenindex
Num	EINGABE	Nummer
Num	GANZZAHL.	Nummer
Nummer	Hauptm.	Zähler
nur_Nullen	PHASE I	Zeile enthält nur Nullen
OGexist	ZIELFKT.	Obergrenze existiert
OPT	DEF	Tableau optimal
OPTLSG	DEF	optimale Lösung existiert
Phase1	DEF	Phase I läuft
Phase2	DEF	Phase II läuft
Piv	BASIST.	Pivotelement
Piv	SENSIB.	Pivotelement o.ä.
Position	AUSGABE	Schreibposition
r	DEF	Index (meist Pivotzeile)
R1	DEF	Stellen feine Rundung
R2	DEF	Stellen grobe Rundung
RechteS	RECHTE S.	Menüpunkt RS

Reelle_Zahl	DEF	reelle Zahl
Restr	RESTRIKT.	Menüpunkt RESTRIKTIONEN
s	DEF	Index (meist Pivotspalte)
Schnitte	GANZZAHL.	Anzahl Gomory-Schnitte
Sensib	SENSIB.	Sensibilitäts-Menüpunkt
ST	DEF	Anzahl Strukturvariable
STNAMEN	DEF	Namen für Strukturvariable
StV	STRUKTUR.	Menüpunkt STRUKTURVAR
Toleranz	DEF	Rundungsfehlertoleranz
UGexist	ZIELFKT.	Untergrenze existiert
UNBESCHR	DEF	nicht nach oben beschränkt
VARNAMEN	DEF	Namen für alle Variable
VOLL	GANZZAHL.	Speicherplatz erschöpft
Wahl	GANZZAHL.	Menüpunkt GANZZAHLIGKEIT
Z	DEF	Zähler Zyklusschutz
Zeigervar.	Initial.	Zeigervariable
Zeilenende	AUSGABE	Zeilenende erreicht
Zielvar	DEF	gekürzter Name von x_0
Zlfkt	ZIELFKT.	Zielfunktions-Menüpunkt
ZS	DEF	Zyklusschutz-Vektor
ZUL	DEF	Tableau zulässig
ZYKLUS	DEF	Zyklusschutz aktiviert

ANHANG D

Anhang D: Quelltext von LinSen (Programmliste / TURBO-Pascal 5.5)

Auf den nachfolgenden Seiten sind die drei Dateien DEF.PAS, AUSGABE.PAS und LINSEN.PAS abgedruckt. Zur Orientierung sei auf Anhang B und C verwiesen.

```
Unit Def;
(*$N+*)
interface
uses crt;

const   maxzeil     =125;  (* max. Anzahl Restriktionen/Strukturvar.  *)
        maxspal     =251;  (* => max. Anzahl Spalten der Matrix A     *)
                           (* =  2 * maxzeil + 1                      *)
        stschlupf   =250;  (* max. Anzahl nichtkünstlicher Variablen  *)

type    zeile_A     =array[1..maxspal] of double;
        zeile_C     =array[1..maxzeil] of double;
        zeiger_A    =^zeile_A;
        zeiger_C    =^zeile_C;
        Matrix_A    =array[0..maxzeil] of zeiger_A;
        Matrix_C    =array[0..maxzeil] of zeiger_C;

var     A                                :Matrix_A;
        C                                :Matrix_C;

        b,bc                             :array[0..maxzeil] of double;
        H                                :array[1..maxspal] of double;

        BV,Art                           :array[0..maxzeil] of integer;
        ZS,BZ                            :array[1..maxspal] of integer;

        Reelle_Zahl,Hb0,Toleranz,Max,Min :double;

        R1,R2,r,s,Z,h1,h2,h3,h4,
        ST,KG,GG,GL,KV,m,n,i,j,k,l,
        MaxMin,Kode,Ganze_Zahl,GrGl      :integer;

        IT                               :longint;

        Datei                            :text;

        Namensvektor                     :array[0..maxspal] of string[5];

        Dateiname                        :string[55];

        Zielvar                          :string[4];

        Haupt,Antwort,Ausgabeart         :char;
```

```
        Drucker_bereit,EAFehler,
        Alle_Schritte,ZUL,OPT,eindeutig,
        Phase1,Phase2,ZYKLUS,ausgeartet,
        Neue_Aufgabe,Keine_Restr,
        STNAMEN,VARNAMEN,DUAL,HILF,
        OPTLSG,LEERLSG,UNBESCHR,GANZ        :boolean;


procedure Lies_ganze_Zahl(UG,OG:integer);
function SGN(Z:double):integer;
function RUND(Z:double;Stellen:integer):double;
procedure PARAMETER;

implementation

procedure Lies_ganze_Zahl(UG,OG:integer);
var Aktuelle_Zeile, Aktuelle_Spalte:integer;
    Eingabezahl                     :string[25];
begin
Aktuelle_Spalte:=wherex;Aktuelle_Zeile:=wherey;
repeat
   gotoxy(Aktuelle_Spalte,Aktuelle_Zeile);clreol;
   readln(Eingabezahl);
   val(Eingabezahl,Ganze_Zahl,Kode);
   if (Kode=0) and ((Ganze_Zahl<UG) or (Ganze_Zahl>OG)) then Kode:=1
 until (Kode=0) or ( Eingabezahl='')
end;

function SGN(Z:double):integer;
begin
if z>0 then SGN:=1 else if z<0 then SGN:=-1 else SGN:=0
end;

function RUND(Z:double;Stellen:integer):double;
begin
RUND:=SGN(Z)*INT(abs(Z)*exp(Stellen*ln(10))+0.5)/exp(Stellen*ln(10))
end;

procedure PARAMETER;
 procedure paramtitel;
 begin
 clrscr;gotoxy(4,5);highvideo;write('* *   P A R A M E T E R   * *');
 lowvideo
 end;
begin
repeat
paramtitel;
gotoxy(4,8);write('1. Elimination  von Rundungsfehlern');
gotoxy(4,10);write('Feine Rundung auf wieviele Stellen (z.B. 6) ? ');
Lies_ganze_Zahl(4,16);if Kode=0 then R1:=ganze_Zahl
                      else begin gotoxy(55,10);write(R1) end;
gotoxy(4,11);write('Grobe Rundung auf wieviele Stellen (z.B. 2) ? ');
Lies_ganze_Zahl(1,R1-1);if Kode=0 then R2:=ganze_Zahl
                      else begin gotoxy(55,11);write(R2) end;
gotoxy(4,12);write('Toleranz für die Differenz        (z.B. 4) ? ');
```

```
Lies_ganze_Zahl(R2+1,18);if Kode=0 then
                      Toleranz:=exp(-ganze_Zahl*ln(10))
                      else begin gotoxy(55,12);
                                 write(-ln(Toleranz)/ln(10):2:0) end;
gotoxy(4,14);write('Wenn sich die grobe und die feine Rundung um weniger ');
gotoxy(4,15);write('als ',Toleranz:20:18,' unterscheiden,');
gotoxy(4,16);write('dann wird die Zahl durch die grobe Rundung ersetzt.');
antwort:=readkey; paramtitel;gotoxy(4,8);write('2. Sicherung gegen Zyklen');
gotoxy(4,10);write('Zyklusschutz aktivieren ? ');
antwort:=upcase(readkey);if antwort='J'then begin
                            gotoxy(16,10);clreol;highvideo;write(' aktiviert');
                            lowvideo;write(' !');ZYKLUS:=true;
                            end
                           else begin gotoxy(16,10);clreol;
                                write(' abgeschaltet.');ZYKLUS:=false;
                                end;
antwort:=readkey;paramtitel;gotoxy(4,8);write('3. Bildschirmausgabe');
gotoxy(4,10);write('Ausgabe aller Rechenschritte ? ');
antwort:=upcase(readkey);if antwort='J' then begin
                            write(' JA');alle_Schritte:=true end
                           else begin
                            write(' NEIN');alle_Schritte:=false end;
antwort:=readkey;paramtitel;gotoxy(4,8);write('Eingabe korrekt ? ');
antwort:=upcase(readkey);
until antwort<>'N';
clrscr;
end;

begin
end.
```

```
Unit Ausgabe;
(*$N+*)
interface
uses Def,crt,printer;

procedure DRUCKPROBE;
procedure AUFGABE_AUSGEBEN;
procedure TABLEAU_AUSGEBEN;
procedure BASISLSG_AUSGEBEN;
procedure AUSGABEMENUE;
procedure ERGEBNISAUSGABE;
procedure AUFGABE_SPEICHERN;
procedure PROGRAMMSTART;

implementation

procedure DRUCKPROBE;
begin
(*$I-*) writeln(lst);writeln(lst,'LinSen');writeln(lst) (*$I+*);
Drucker_bereit:=IOResult=0;
end;

procedure AUFGABE_AUSGEBEN;
var Position     :integer;
    Zeilenende   :boolean;
procedure WARTE;
begin
antwort:=readkey;clrscr;gotoxy(20,1)
end;
function Vorzeichen(Koeffizient:double):string;
begin
Vorzeichen:=' ';
if (RUND(Koeffizient,12)>0) and (Position >0) then Vorzeichen:='+';
if (RUND(Koeffizient,12)<0) then Vorzeichen:='-'
end;
begin
repeat
clrscr;gotoxy(4,5);highvideo;
write('* *  A U F G A B E   A U S G E B E N   * *');gotoxy(4,8);
write('B');gotoxy(4,10);write('D');lowvideo;gotoxy(5,8);
write('ildschirm');gotoxy(5,10);write('rucker');gotoxy(4,12);write('');
Ausgabeart:=upcase(readkey);
if Ausgabeart<>'D' then begin
    clrscr;write('Vorliegendes LO-Problem :');writeln;writeln;
    if Maxmin=1 then write('min.') else write ('max.');
    write(Namensvektor[0],':',Namensvektor[0],':= ');Position:=0;
    gotoxy(20,3);
    for j:=1 to ST do begin
                      if RUND(c[0]^[j],12)<>0 then begin
                         write (Vorzeichen(Maxmin*c[0]^[j]));
                         write (abs(RUND(Maxmin*c[0]^[j],2)):10:2);
                         write (Namensvektor[j]);Position:=Position+1;
                         if (Position/3=int(Position/3)) and (Position>0)
                             then gotoxy(20,wherey+1);
                         if wherey>=22 then warte;
                         end;
```

```
                  end;
 if RUND(bc[0],12)<>0 then begin
                write(Vorzeichen(-Maxmin*bc[0]));
                write(abs(Maxmin*bc[0]):10:2);
                end
          else if Position=0 then write('0');
 writeln;if wherey>=22 then warte;
 if not Keine_Restr then for i:=1 to KG+GG+GL do begin
  Position:=0;writeln;gotoxy(20,wherey);if wherey>=22 then warte;
  for j:=1 to ST do begin
      if RUND(c[i]^[j],12)<>0 then begin
      write(Vorzeichen(c[i]^[j]),abs(RUND(c[i]^[j],2)):10:2);
      write(Namensvektor[j]);Position:=Position+1;
      if(Position/3=int(Position/3)) and (Position>0) then
         gotoxy(20,wherey+1);if wherey>=22 then warte;
      end;
  end;
  if Position=0 then write('0');
  if (Position/3=int(Position/3)) and (Position>0) then
     gotoxy(69,wherey-1) else gotoxy(69,wherey);
  case Art[i] of
       1:write('≤');
      -1:write('≥');
       0:write('=');
    end;
  gotoxy(71,wherey);write(RUND(bc[i],2):10:2);
 end;
 writeln;writeln;if wherey>=22 then warte;
 gotoxy(20,wherey);write(Namensvektor[1]);
 if ST>1 then if ST=2 then write(',',Namensvektor[2])
    else if ST=3 then write(',',Namensvektor[2],',',Namensvektor[3])
         else write(', .... ,',Namensvektor[ST]);
 write(' ≥ 0   ');
 warte;
 gotoxy(8,5);write('Noch einmal ? ');antwort:=upcase(readkey);
 end
else begin
 DRUCKPROBE;
 if Drucker_bereit then begin
    (*$I-*)writeln(lst,'Vorliegendes LO-Problem :');writeln(lst);
    if Maxmin=1 then write(lst,'min.') else write (lst,'max.');
    write(lst,Namensvektor[0],';',Namensvektor[0],':= ');Position:=0;
    write(lst,' ');
    for j:=1 to ST do begin
                  if RUND(c[0]^[j],12)<>0 then begin
                     write (lst,Vorzeichen(Maxmin*c[0]^[j]));
                     write (lst,abs(RUND(Maxmin*c[0]^[j],2)):10:2);
                     write (lst,Namensvektor[j]);Position:=Position+1;
                     if (Position/3=int(Position/3)) and (Position>0)
                        then begin
                             writeln(lst);for k:=1 to 19 do write(lst,' ');
                             end;
                     end;
                  end;
 if RUND(bc[0],12)<>0 then begin
              write(lst,Vorzeichen(-Maxmin*bc[0]));
```

```
                write(lst,abs(Maxmin*bc[0]):10:2);
                end
            else if Position=0 then write(lst,'0');
    writeln(lst);
    if not Keine_Restr then for i:=1 to KG+GG+GL do begin
     Position:=0;writeln(lst);writeln(lst);Zeilenende:=false;
     for k:=1 to 19 do write(lst,' ');
     for j:=1 to ST do begin
         if RUND(c[i]^[j],12)<>0 then begin
         write(lst,Vorzeichen(c[i]^[j]),abs(RUND(c[i]^[j],2)):10:2);
         write(lst,Namensvektor[j]);Position:=Position+1;
         if j<ST then begin
                   Zeilenende:=true;
                   for k:=j+1 to ST do
                       if RUND(c[i]^[k],2)<>0 then Zeilenende:=false
                   end;
         if(Position/3=int(Position/3)) and (Position>0) and (j<ST)
             and not Zeilenende then begin
             writeln(lst);for k:=1 to 19 do write(lst,' ')    end;
         end;
     end;
     if Position=0 then begin
                  write(lst,'0');for k:=1 to 48 do write(lst,' ') end;
     if (Position=3*int(Position/3)) and (Position>0) then write(lst,' ');
     if (Position=3*int(Position/3)+1) then for k:=1 to 33 do write(lst,' ');
     if (Position=3*int(Position/3)+2) then for k:=1 to 17 do write(lst,' ');
     case Art[i] of
          1:write(lst,'≤ ');
         -1:write(lst,'≥ ');
          0:write(lst,'= ');
       end;
     write(lst,RUND(bc[i],2):10:2);
    end;
    writeln(lst);writeln(lst);for k:=1 to 19 do write(lst,' ');
    write(lst,Namensvektor[1]);
    if ST>1 then if ST=2 then write(lst,',',Namensvektor[2])
       else if ST=3 then write(lst,',',Namensvektor[2],',',Namensvektor[3])
            else write(lst,', ... ,',Namensvektor[ST]);
     writeln(lst,' ≥ 0   '); (*$I+*)
       end
    else begin
            gotoxy(4,12);highvideo;write('Drucker ist nicht bereit !  ');
            lowvideo;antwort:=readkey;
            end;
    antwort:='N';clrscr;
    end;
until antwort<>'J'
end;


procedure TABLEAU_AUSGEBEN;
var Position:integer;
begin
clrscr;gotoxy(4,5);highvideo;
write('* *  T A B L E A U    A U S G E B E N  * *');gotoxy(4,8);
write('B');gotoxy(4,10);write('D');lowvideo;gotoxy(5,8);
```

```
write('ildschirm');gotoxy(5,10);write('rucker');gotoxy(4,12);write('');
Ausgabeart:=upcase(readkey);
if Ausgabeart<>'D'then begin
 clrscr;gotoxy(4,1);highvideo;
 write('* *  TABLEAU NACH ',IT,' SIMPLEXSCHRITT');if IT<>1 then write('EN');
 if KV>0 then write(' - Phase I');if Phase2 then write(' - Phase II');
 write('  * *');lowvideo;gotoxy(1,3);write('BV   ');
 if (n-m<6) and (m<19) then begin
   for j:=1 to n do if BZ[j]=0 then write('     ',Namensvektor[j],'   ');
   writeln('     RS     ');
   for i:=0 to m do begin
           if (i=0) and (Maxmin=1) then write('-',Zielvar)
              else write(Namensvektor[BV[i]]);
           for j:=1 to n do if BZ[j]=0 then write(RUND(a[i]^[j],4):12:4);
           writeln(RUND(b[i],4):12:4);
           end;
   if KV>0 then begin
            write('  H  ');
            for j:=1 to n do if BZ[j]=0 then write(RUND(H[j],4):12:4);
            writeln(RUND(Hb0,4):12:4);
            end;
   end
   else begin
        gotoxy(1,3);
        writeln('Zeilenvektoren ohne die Koeffizienten der Basisvariablen ! ');
        writeln('Folge der Spalten:');
        for j:=1 to n do begin
                      if BZ[j]=0 then write(Namensvektor[j],',');
                      if wherex>74 then writeln end;
        writeln('RS');writeln;
        for i:=0 to m do begin
            if (i=0) and (Maxmin=1) then write('-',Zielvar)
               else write(Namensvektor[BV[i]]);writeln('-Zeile: ');
            for j:=1 to n do begin
                       if BZ[j]=0 then write(RUND(a[i]^[j],4):12:4,',');
                       if wherex>66 then writeln;
                       if wherey>22 then begin antwort:=readkey;clrscr end;
                  end;
            writeln(RUND(b[i],4):12:4);writeln;
            if wherey>22 then begin antwort:=readkey;clrscr end;
        end;
        if KV>0 then begin
           writeln('Hilfszielfunktionszeile: ');
           for j:=1 to n do begin
               if BZ[j]=0 then write(RUND(H[j],4):12:4,',');
               if wherex>66 then writeln;
               if wherey>22 then begin antwort:=readkey;clrscr end;
               end;
           writeln(RUND(Hb0,4):12:4);writeln;
           if wherey>22 then begin antwort:=readkey;clrscr end;
          end;
      end;  (* Vektorform *)
  writeln;highvideo;write('D');lowvideo;write('rucken ');
  Ausgabeart:=upcase(readkey);
 clrscr;
 end;  (* Bildschirmausgabe *)
```

```
if Ausgabeart='D' then begin
   DRUCKPROBE;
   if Drucker_bereit then begin
      (*$I-*)
      write(lst,'* *  TABLEAU NACH ',IT,' SIMPLEXSCHRITT');
      if IT<>1 then write(lst,'EN');
      if KV>0 then write(lst,' - Phase I');
      if Phase2 then write(lst,' - Phase II');
      writeln(lst,'  * *');writeln(lst);
      if (n-m<6) then begin
      write(lst,'BV   ');
      for j:=1 to n do if BZ[j]=0 then write(lst,'     ',Namensvektor[j],'   ');
      writeln(lst,'      RS      ');
      for i:=0 to m do begin
            if (i=0) and (Maxmin=1) then write(lst,'-',Zielvar)
               else write(lst,Namensvektor[BV[i]]);
            for j:=1 to n do if BZ[j]=0 then write(lst,RUND(a[i]^[j],4):12:4);
            writeln(lst,RUND(b[i],4):12:4);
            end;
      if KV>0 then begin
             write(lst,'  H  ');
             for j:=1 to n do if BZ[j]=0 then write(lst,RUND(H[j],4):12:4);
             writeln(lst,RUND(Hb0,4):12:4);
             end;
   end
   else begin
    writeln(lst,'Zeilenvektoren ohne die Koeffizienten der Basisvariablen ! ');
        writeln(lst,'Folge der Spalten:');Position:=0;
        for j:=1 to n do begin
                        if BZ[j]=0 then begin
                           write(lst,Namensvektor[j],',');
                           Position:=Position+1;
                           if Position=6 then begin
                              writeln(lst);Position:=0 end;
                         end;
                     end;
        writeln(lst,'RS');writeln(lst);
        for i:=0 to m do begin
            Position:=0;
            if (i=0) and (Maxmin=1) then write(lst,'-',Zielvar)
               else write(lst,Namensvektor[BV[i]]);writeln(lst,'-Zeile: ');
            for j:=1 to n do begin
                          if BZ[j]=0 then begin
                             write(lst,RUND(a[i]^[j],4):12:4,',');
                             Position:=Position+1;
                             if Position=6 then begin
                                writeln(lst);Position:=0 end;
                             end;
                     end;
            writeln(lst,RUND(b[i],4):12:4);writeln(lst);
        end;
        if KV>0 then begin
           writeln(lst,'Hilfszielfunktionszeile: ');Position:=0;
           for j:=1 to n do begin
               if BZ[j]=0 then begin
                  write(lst,RUND(H[j],4):12:4,',');
```

```
                    Position:=Position+1;
                    if Position=6 then begin
                       writeln(lst);Position:=0 end;
                    end;
                 end;
              writeln(lst,RUND(HbO,4):12:4);writeln(lst);
             end;
        end;   (* Vektorform *)
       (*$I+*)
       end
       else begin
           highvideo;write(' Drucker ist nicht bereit ! ');antwort:=readkey;
           lowvideo;
           end;
   end; (* Druckerausgabe *)
end;    (* Proz. Ausgabe *)

procedure BASISLSG_AUSGEBEN;
begin
clrscr;gotoxy(4,5);highvideo;write('* *  B A S I S L Ö S U N G  * *');
lowvideo;gotoxy(1,8);
for i:=0 to m do begin
    if (i=0) and (Maxmin=1) then write('-',Zielvar)
       else write(Namensvektor[BV[i]]);
    write(' = ',RUND(b[i],8):16:8,' ≈ ',RUND(b[i],4):12:4);writeln;
    if wherey>24 then begin antwort:=readkey;clrscr;writeln;writeln end;
    end;
if Phasel then writeln('  H   = ',RUND(HbO,8):16:8,' ≈ ',RUND(HbO,4):12:4);
writeln;write('Alle anderen Variablen haben den Wert 0.          ');
highvideo;write('D');lowvideo;writeln('rucken ');antwort:=upcase(readkey);
if antwort='D' then begin
   DRUCKPROBE;
   if Drucker_bereit then begin
   (*$I-*)
      writeln(lst,'* *  B A S I S L Ö S U N G  * *');writeln(lst);
      for i:=0 to m do begin
          if (i=0) and (Maxmin=1) then write(lst,'-',Zielvar)
             else write(lst,Namensvektor[BV[i]]);
          write(lst,' = ',RUND(b[i],8):16:8,' ≈ ',RUND(b[i],4):12:4);
          writeln(lst);
      end;
      if Phasel then
       writeln(lst,'  H   = ',RUND(HbO,8):16:8,' ≈ ',RUND(HbO,4):12:4);
      writeln(lst);writeln(lst,'Alle anderen Variablen haben den Wert 0. ');
    (*$I+*)
   end
    else begin
           highvideo;write(' Drucker ist nicht bereit ! ');
           lowvideo;antwort:=readkey;
           end;
   end;
clrscr;
end;
```

```
procedure AUSGABEMENUE;
var M:char;
begin
repeat
 clrscr;gotoxy(4,5);highvideo;write('*  I T E R A T I O N    ',IT,' *');
 gotoxy(4,8);write('T');gotoxy(4,10);write('B');gotoxy(4,12);write('P');
 gotoxy(4,14);write('W');lowvideo;gotoxy(5,8);write('ableau ausgeben');
 gotoxy(5,10);write('asislösung ausgeben');gotoxy(5,12);
 write('arameter ändern');gotoxy(5,14);writeln('eiterrechnen');
 M:=upcase(readkey);
 case M of
      'T': TABLEAU_AUSGEBEN;
      'B': BASISLSG_AUSGEBEN;
      'P': PARAMETER;
      end;
until (M<>'T') and (M<>'B') and (M<>'P');
end;

procedure ERGEBNISAUSGABE;
  begin
  clrscr;gotoxy(4,5);highvideo;write('*  E R G E B N I S S E  *');
  gotoxy(4,8);
  if LEERLSG then write('Das LO-Problem besitzt keine zulässige Lösung !!!');
  if UNBESCHR then begin
     write('Die Zielfunktion ist nach ');
     if Maxmin=1 then write('unten ') else write('oben ');
     write('nicht beschränkt !!');
     end;
  if OPTLSG then begin
     write('Optimale Lösung gefunden !');lowvideo;gotoxy(4,10);
     if Maxmin=1 then write('MINIMUM = ') else write('MAXIMUM = ');
     write(-Maxmin*RUND(b[0],4):12:4);gotoxy(4,12);
     if eindeutig then write('Eindeutige Lösung.')
        else write('(Basis-) Lösung ist nicht eindeutig !');gotoxy(4,14);
     if ausgeartet then write('Optimale Lösung ist (primal) ausgeartet.');
     highvideo;
     end;
  gotoxy(4,18);write('D');lowvideo;write('rucken   ');
  antwort:=upcase(readkey);
  if antwort='D' then begin
     DRUCKPROBE;
     if Drucker_bereit then begin
        (*$I-*)
        writeln(lst,'*  E R G E B N I S S E  *');writeln(lst);
        if LEERLSG then
           writeln(lst,'Das LO-Problem besitzt keine zulässige Lösung !!!');
        if UNBESCHR then begin
           write(lst,'Die Zielfunktion ist nach ');
           if Maxmin=1 then write(lst,'unten ') else write(lst,'oben ');
           writeln(lst,'nicht beschränkt !!');
           end;
  if OPTLSG then begin
     writeln(lst,'Optimale Lösung gefunden !');writeln(lst);
     if Maxmin=1 then write(lst,'MINIMUM = ') else write(lst,'MAXIMUM = ');
     writeln(lst,-Maxmin*RUND(b[0],4):12:4);writeln(lst);
     if eindeutig then writeln(lst,'Eindeutige Lösung.')
```

```
         else writeln(lst,'(Basis-) Lösung ist nicht eindeutig !');
      if ausgeartet then
         writeln(lst,'Optimale Lösung ist (primal) ausgeartet.');
      end;
         (*$I+*)
         end
         else begin
         gotoxy(4,18);highvideo;write('Drucker ist nicht bereit !    ');
         lowvideo;antwort:=readkey;
         end;
      end;
end;

procedure AUFGABE_SPEICHERN;
begin
if STNAMEN then h1:=1 else h1:=0;
if VARNAMEN then h2:=1 else h2:=0;
if HILF then h3:=1 else h3:=0;
if Keine_Restr then h4:=1 else h4:=0;
clrscr;gotoxy(4,5);highvideo;
write('* *  A U F G A B E   S P E I C H E R N  * *');lowvideo;
gotoxy(4,8);
write('Unter welchem Namen (ggf. mit Laufwerkangabe und Endung ".LNS") ? ');
gotoxy(4,10);readln(Dateiname);(*$I-*)
assign(Datei,Dateiname);rewrite(Datei);
write(Datei,n:4,m:4,ST:4,KG:4,GG:4,GL:4,KV:4,Maxmin:3,Hb0:12:4);
writeln(Datei);write(Datei,Zielvar:4);writeln(Datei);
for i:=0 to KG+GG+GL do begin
    writeln(Datei,b[i]:16:8,bc[i]:16:8);writeln(Datei);
    for j:=1 to n do writeln(Datei,a[i]^[j]:16:8);
    for j:=1 to ST do writeln(Datei,c[i]^[j]:16:8);
    end;
for i:=1 to KG+GG+GL do writeln(Datei,BV[i]:4,Art[i]:3);
for j:=1 to n do writeln(Datei,H[j]:12:4,BZ[j]:4);
for j:=0 to n do writeln(Datei,Namensvektor[j]);
write(Datei,h1:2,h2:2,h3:2,h4:2);
close(Datei);(*$I+*)
EAFehler:=ioresult<>0;
if EAFehler then begin
   gotoxy(4,12);highvideo;write('E/A - Fehler !!!  ');lowvideo;
   antwort:=readkey;
   end;
end;

procedure PROGRAMMSTART;
begin
clrscr;lowvideo;gotoxy(1,1);write('╔');for j:=1 to 78 do write('=');write('╗');
for i:=2 to 23 do begin gotoxy(1,i);write('║');gotoxy(80,i);write('║') end;
gotoxy(1,24);write('╚');for j:=1 to 78 do write('=');write('╝');highvideo;
gotoxy(22,5); write('L i n e a r e    O p t i m i e r u n g');
gotoxy(22,7); write('                 u n d');
gotoxy(22,9); write('S e n s i b i l i t ä t s a n a l y s e');
gotoxy(22,11);write('         ( L i n S e n )');lowvideo;
gotoxy(22,16);write('von Thomas H e r i n g , 1990.');
gotoxy(22,18);write('85 KB, TURBO-Pascal 5.5 . ');
gotoxy(22,20);write('Alle Rechte vorbehalten.');
```

```
gotoxy(22,22);write('');
for j:=1 to 100 do begin delay(150);if keypressed then j:=100 end;
if keypressed then antwort:=readkey;
clrscr;gotoxy(4,3);highvideo;write('Das Programm LinSen');lowvideo;
gotoxy(1,6);
write('   Dieses Programm löst beliebige lineare Optimierungsprobleme mit');
writeln(' bis zu ');
writeln('    ',maxzeil:3,' Variablen und Nebenbedingungen.');
writeln('    Seine wichtigsten Eigenschaften:');writeln;
writeln('     * Primaler Simplexalgorithmus mit Zweiphasenmethode;');
writeln('     * Dualer Simplexalgorithmus;');
writeln('     * Umfassende Sensibilitätsanalyse;');
writeln('     * Gemischt-ganzzahliger Algorithmus (nur für sehr kleine Probleme);');
writeln('     * Zyklusschutz zuschaltbar;');
writeln('     * Nur eine künstliche Variable für alle ≥ -Bedingungen ;');
writeln('     * Unterstützung eines mathematischen Koprozessors;');
writeln('     * Speicherung von 16 Dezimalstellen;');
writeln('     * Rundungsparameter variierbar;');
writeln('     * Variablennamen frei wählbar;');
writeln('     * Zahlreiche Ausgabeoptionen;');
writeln('     * Durchgehende Menüsteuerung mit interaktiver Eingabe;');
writeln('     * Eingabekontrolle und gezielte Korrektur von Eingabedaten.');
writeln;write('    Beliebige Taste drücken ! ');antwort:=readkey;
clrscr;gotoxy(4,5);highvideo;
write('Bedienungsgrundsätze von LinSen');lowvideo;gotoxy(1,8);
writeln('    1. LinSen spricht Deutsch: J für "ja", N für "nein" tippen,');
writeln('       sonst die angegebene Taste drücken.');writeln;
writeln('    2. Durch Drücken der Eingabetaste wird die Voreinstellung (z.B.');
writeln('       eine bereits im Speicher befindliche Zahl) bestätigt, so daß');
writeln('       sich eine (erneute) Eingabe erübrigt.');writeln;
writeln('    3. Vor der Fortsetzung von Bildschirmausgaben erwartet LinSen');
writeln('       einen beliebigen Tastenanschlag.');writeln;
writeln('    4. Bei der Tastatureingabe einer neuen Aufgabe geht LinSen immer');
writeln('       davon aus, daß zuerst die ≤-, dann die ≥- und zuletzt die');
writeln('       =-Bedingungen folgen.');writeln;write('  ');antwort:=readkey;
clrscr;gotoxy(4,5);highvideo;write('Spezielle Hinweise');lowvideo;gotoxy(1,8);
writeln('    1. Die Sensibilitätsanalyse erlaubt keine =-Restriktionen. ');
writeln('       Ggf. wähle man Punkt 1 des Hauptmenüs, verneine die Abfrage');
writeln('       "Urdaten löschen", definiere die =- nun als ≥-Bedingungen und');
writeln('       löse die Aufgabe erneut. Mit der Sensibilitätsanalyse lassen');
writeln('       sich dann die noch fehlenden ≤-Bedingungen hinzufügen.');
writeln;
writeln('    2. Die ganzzahlige Optimierung mit Schnittrestriktionen ist');
writeln('       stark anfällig gegen kumulierte Rundungsfehler. Bei falschen');
writeln('       Ergebnissen läßt sich manchmal durch Experimentieren mit den');
writeln('       Rundungsparametern Abhilfe schaffen.  ');writeln;
writeln('    3. Einzulesende Dateien müssen im LinSen-Format vorliegen;');
writeln('       andernfalls ist ein Programmabsturz nicht auszuschließen.');
writeln;write('   ');
antwort:=readkey;
end;

begin
end.
```

```
Program LinSen;
(*$N+*)
uses Def,Ausgabe,crt,printer;


procedure NEU;
begin
ST:=1;KG:=0;GG:=0;GL:=0;KV:=0;MaxMin:=-1;STNamen:=false;VarNamen:=false;
Neue_Aufgabe:=true;GANZ:=false;
end;

procedure Initialisierung;

var      zeigervariable_A        :zeiger_A;
         zeigervariable_C        :zeiger_C;
begin
NEU;
optlsg:=false;leerlsg:=false;unbeschr:=false;zyklus:=false;
alle_Schritte:=false;BV[0]:=0;Art[0]:=2;Phase1:=false;Phase2:=false;
R1:=6;R2:=2;Toleranz:=1E-4;n:=0;HILF:=false;Keine_Restr:=false;
for i:=0 to maxzeil do begin
     new(zeigervariable_A);new(zeigervariable_C);
    A[i]:=zeigervariable_A;C[i]:=zeigervariable_C;
 end;
end;

procedure Hauptmenue;
var Nummer        :shortint;
begin
clrscr;
highvideo;gotoxy(4,4);
write('* * *   L i n S e n  -  H a u p t m e n ü   * * *');
for nummer:=1 to 8 do begin
    gotoxy(4,6+2*nummer);write(nummer)
   end;
lowvideo;gotoxy(8,8);write('Neue Aufgabe');gotoxy(8,10);
write('Aufgabe ausgeben');gotoxy(8,12);write('Endtableau ausgeben');
gotoxy(8,14);write('Basislösung ausgeben');gotoxy(8,16);
write('Sensibilitätsanalyse');gotoxy(8,18);write('Aufgabe abspeichern');
gotoxy(8,20);write('Parameter');gotoxy(8,22);write('LinSen  beenden');
gotoxy(4,24);write('');
repeat
 Haupt:=readkey;
 if (Haupt='1') and (OPTLSG or LEERLSG or UNBESCHR) then begin
      gotoxy(4,24);highvideo;write('Alte Urdaten löschen ?');lowvideo;
      antwort:=upcase(readkey);if (antwort<>'N') or GANZ
      then Neue_Aufgabe:=true
      end;
 if (Haupt='8') and (OPTLSG or LEERLSG or UNBESCHR) then begin
      gotoxy(4,24);highvideo;
      write('Bitte mit J bestätigen !');antwort:=upcase(readkey);
      if antwort<>'J' then begin
               Haupt:='f';gotoxy(4,24);clreol;lowvideo;
               end;
      end;
 if (Haupt in ['2','3','4','5','6']) and (n=0) then Haupt:='f';
```

```
 if (Haupt in ['5','6']) and GANZ then Haupt:='f';
 if (Haupt='5') and (GL>0) then begin
    gotoxy(4,24);highvideo;
    write('Gleichungsrestriktionen sind nicht erlaubt ! ');lowvideo;
    Haupt:='f';antwort:=readkey;gotoxy(4,24);clreol;
    end;
 until Haupt in ['1','2','3','4','5','6','7','8'] ;
 end;


procedure Lies_reelle_Zahl;
var Aktuelle_Zeile, Aktuelle_Spalte:integer;
    Eingabezahl                   :string[25];
begin
Aktuelle_Spalte:=wherex;Aktuelle_Zeile:=wherey;
repeat
   gotoxy(Aktuelle_Spalte,Aktuelle_Zeile);clreol;
   readln(Eingabezahl);
   val(Eingabezahl,Reelle_Zahl,Kode)
 until (Kode=0) or ( Eingabezahl='')
end;

(* Elimination von Rundungsfehlern  *)
procedure GLATT(var G:double);
begin
if abs(RUND(G,R1)-RUND(G,R2)) < Toleranz then G:=RUND(G,R2)
end;

procedure Pruefe_ZUL;
begin
ZUL:=true;
for i:=1 to m do begin
             GLATT(b[i]);
             if b[i]<0 then begin ZUL:=false;i:=m end
             end
end;

procedure Pruefe_OPT;
begin
OPT:=true;
for j:=1 to n do begin
             GLATT(a[0]^[j]);
             if a[0]^[j]<0 then begin OPT:=false;j:=n end;
             end;
end;

procedure LOESCH;
begin
for i:=0 to maxzeil do begin
    b[i]:=0;bC[i]:=0;
    for j:=1 to maxspal do a[i]^[j]:=0;
    for j:=1 to maxzeil do c[i]^[j]:=0;
    end;
for j:=1 to maxspal do h[j]:=0;
for j:=0 to maxspal do Namensvektor[j]:='*****';
Hb0:=0;
```

```
end;

procedure EINGZLFKT;
begin
repeat
      clrscr;highvideo;gotoxy(4,5);
      write('* Eingabe der Zielfunktionsbeiträge *');lowvideo;
      writeln;writeln;writeln;
      for j:=1 to ST do begin
          write (Namensvektor[j],'  ?      ');
          write(RUND(Maxmin*c[0]^[j],4):12:4,'     ');
          Lies_reelle_Zahl;
          if Kode=0 then C[0]^[j]:=MaxMin*Reelle_Zahl
                     else begin
                          gotoxy(32,wherey-1);writeln
                          (RUND(MaxMin*C[0]^[j],4):12:4);
                          end;
          end;
      writeln;writeln;write('Eingabe korrekt ? ');antwort:=upcase(readkey);
   until antwort<>'N';
end;

procedure EINGABE;
var Eingabeart       :char;
    EingName,Num     :string[5];
    Max              :integer;
begin
repeat
if neue_Aufgabe then NEU;
gotoxy(4,24);clreol;write('Tastatur (');highvideo;write('T');lowvideo;write
      (') oder Laufwerk (');highvideo;write('L');lowvideo;write(') ?  ');
Eingabeart:=upcase(readkey);
if Eingabeart <> 'L' then begin
EAFehler:=false;
repeat
 repeat
  clrscr;
  gotoxy(4,5);highvideo;write('* * L I N E A R E   O P T I M I E R U N G * *');
  gotoxy(4,8);lowvideo;write('Maximieren oder minimieren ?');
  gotoxy(4,10);write('MAX --->');highvideo;write(' 1');lowvideo;
  gotoxy(4,12);write('MIN --->');highvideo;write(' 2');lowvideo;
  gotoxy(4,24);write('');Antwort:=readkey;
  if (antwort='1') or ((antwort<>'2') and (maxmin=-1)) then begin
       maxmin:=-1;gotoxy(4,10);highvideo;write('MAX');
       end;
  if (antwort='2') or ((antwort<>'1') and (maxmin=1))  then begin
       maxmin:=1;gotoxy(4,12);highvideo;write('MIN');
       end;
  lowvideo;gotoxy(4,14);write('Wieviele Strukturvariable (max.',maxzeil,') ?  ');
  Lies_ganze_Zahl(1,maxzeil);
  if Kode=0 then ST:=Ganze_Zahl
          else begin gotoxy(43,14);write(ST) end;
  gotoxy(4,16);write('Wieviele  ≤ -Restriktionen ?  ');
  Lies_ganze_Zahl(0,maxzeil);
  if Kode=0 then KG:=Ganze_Zahl
          else begin gotoxy(34,16);write(KG) end;
```

```
 gotoxy(4,18);write('Wieviele  ≥ -Restriktionen ?  ');
 Lies_ganze_Zahl(0,maxzeil-KG);
 if Kode=0 then GG:=Ganze_Zahl
          else begin gotoxy(34,18);write(GG) end;
 gotoxy(4,20);write('Wieviele  = -Restriktionen ?  ');
 Lies_ganze_Zahl(0,maxzeil-KG-GG);
 if Kode=0 then GL:=Ganze_Zahl
          else begin gotoxy(34,20);write(GL) end;
 gotoxy(4,24);write('Eingabe korrekt ? ');antwort:=upcase(readkey)
until antwort<>'N';write(' Bitte warten ! ');
 Keine_Restr:=false;if KG+GG+GL=0 then begin
                          KG:=1;Art[1]:=1;Keine_Restr:=true end;
 n:=ST+KG+GG+GL; if GG>0 then begin n:=n+1;GrGl:=1 end else GrGl:=0;
 KV:=n-ST-KG-GG ; m:=KG+GG+GL;
 if neue_Aufgabe then begin LOESCH;Neue_Aufgabe:=false end;
repeat
 clrscr;
 gotoxy(4,5);highvideo;write('* Individuelle Variablenbezeichnung *');
 for i:=1 to 3 do begin gotoxy(4,6+2*i);write(i) end;lowvideo;
 gotoxy(8,8);write('Alle Variablen');gotoxy(8,10);write('Strukturvariable');
 gotoxy(8,12);write('Nein');gotoxy(4,14);antwort:=readkey;
 if (antwort='1') or ((antwort<>'2') and (antwort<>'3') and varnamen)
   then begin
    gotoxy(2,8);write('*');varnamen:=true;stnamen:=false end;
 if (antwort='2') or ((antwort<>'1') and (antwort<>'3') and stnamen)
   then begin
    gotoxy(2,10);write('*');stnamen:=true;varnamen:=false end;
 if ((antwort<>'1') and (antwort<>'2') and not varnamen and not stnamen)
    or (antwort='3') then begin
    gotoxy(2,12);write('*');varnamen:=false;stnamen:=false end;
 if varnamen or stnamen then begin
          gotoxy(4,16);write('Maximal fünf Zeichen Länge !');
          gotoxy(4,18);writeln('Name der Variablen');writeln;
          for j:=0 to n-KV do begin
              write('   x',j,' ? ');gotoxy(16,wherey);readln(EingName);
              if EingName='' then  begin
                     gotoxy(16,wherey-1);writeln(Namensvektor[j]) end
                 else Namensvektor[j]:=concat(EingName,'     ');
             if (j=ST) and stnamen then j:=n-KV;
             end;
       end;
  for j:=0 to n do begin
  if (not varnamen and not stnamen ) or (stnamen and (j > ST)) or
     (varnamen and (j>ST+KG+GG)) then begin
          str(j,Num);
          if j >ST+KG+GG then Namensvektor[j]:=concat('x',Num,'k      ')
                         else Namensvektor[j]:=concat('x',Num,'      ')
          end
  end;
  Zielvar:=Namensvektor[0];
 writeln;writeln;write('   Eingabe korrekt ? ');antwort:=upcase(readkey);
until antwort<>'N';
EINGZLFKT;
if not Keine_Restr then begin
 repeat
    clrscr;highvideo;gotoxy(4,5);
```

```
    write('* Eingabe der rechten Seiten *');lowvideo;gotoxy(1,9);
    write('Nichtnegativ für alle Restriktionen !');gotoxy(1,11);
    for i:=0 to m do begin
      if i=0 then writeln('Absolutglied der Zielfunktion ?   ')
              else writeln('Rechte Seite der Restriktion ',i,' ?  ');
    if (i>0) and (i<=KG) then Art[i]:=1
             else if i<=KG+GG then Art[i]:=-1
                  else Art[i]:=0;
      repeat
           gotoxy(40,wherey-1);
           if i>0 then writeln(RUND(bc[i],4):12:4)
                  else writeln(RUND(-Maxmin*bc[i],4):12:4);
           gotoxy(60,wherey-1);clreol;
           Lies_reelle_Zahl;
           if Kode=0 then begin
                         bc[i]:=Reelle_Zahl;
                         if i=0 then bc[i]:=-Maxmin*bc[i]
                         end
                else begin
                     gotoxy(60,wherey-1);
                     if i>0 then writeln(RUND(bc[i],4):12:4)
                            else writeln(RUND(-Maxmin*bc[i],4):12:4)
                     end;
         until (i=0) or (bc[i]>=0);
    end;
    writeln;writeln;write('Eingabe korrekt ? ');antwort:=upcase(readkey);
  until antwort<>'N'
 end;
if not Keine_Restr then begin
     repeat
       clrscr;highvideo;gotoxy(4,5);
       write('* Eingabe der Koeffizientenmatrix der Nebenbedingungen *');
       lowvideo;gotoxy(1,8);
       for i:=1 to m do begin
                    repeat
                    writeln('Restriktion ',i);
                    for j:=1 to ST do begin
                        write(Namensvektor[j],' ? ');gotoxy(16,wherey);
                        write(RUND(c[i]^[j],4):12:4,'    ');
                        Lies_reelle_Zahl;
                        if Kode=0 then C[i]^[j]:=Reelle_Zahl
                           else begin
                                gotoxy(32,wherey-1);writeln
                                (RUND(C[i]^[j],4):12:4)
                                end;
                        end;
                   writeln;writeln('Restriktion korrekt erfaßt ? ');
                   antwort:=upcase(readkey);writeln;
                   until antwort<>'N';
                   end;
      writeln;write('Eingabe korrekt ? ');antwort:=upcase(readkey);
     until antwort<>'N'
  end;
writeln;writeln;writeln;
writeln('Ausgabe des LO-Problems gewünscht ? ');antwort:=upcase(readkey);
if antwort='J' then AUFGABE_AUSGEBEN;
```

```
 writeln;writeln;writeln('Soll der ganze Eingabeteil wiederholt werden ? ');
 antwort:=upcase(readkey);
 if antwort='J' then begin
         writeln;writeln;writeln('Alles neu    ---> 1');writeln;writeln
         ('Abänderung   ---> 2');antwort:=readkey;
         if antwort='1' then NEU;antwort:='J';
         end;
until antwort<>'J';
for i:=0 to m do begin
               b[i]:=bc[i];
               for j:=1 to ST do a[i]^[j]:=c[i]^[j];
               for j:=ST+1 to n do a[i]^[j]:=0;
               end;
for j:=1 to n do begin h[j]:=0;BZ[j]:=0 end;Hb0:=0;
Pruefe_OPT;if OPT and (GL=0) then DUAL:=true else DUAL:=false;
if DUAL then begin
        writeln;writeln;
        write('Soll die duale Simplexmethode eingesetzt werden ? ');
        antwort:=upcase(readkey);
        if antwort='N'then begin DUAL:=false;write(' NEIN') end
                      else write(' JA');
       end;

if KG>0 then for i:=1 to KG do begin
             a[i]^[ST+i]:=1;BV[i]:=ST+i;BZ[ST+i]:=i;Art[i]:=1 end;
if GG>0 then for i:=KG+1 to KG+GG do begin
             a[i]^[ST+i]:=-1;Art[i]:=-1 end;
if DUAL then begin
             n:=n-KV;KV:=0;
             if GG>0 then for i:=KG+1 to m do begin
                    b[i]:=-b[i];
                    for j:=1 to n do a[i]^[j]:=-a[i]^[j];
                    BV[i]:=ST+i;BZ[ST+i]:=i;
                    end;
             end
        else begin
             if GG>0 then begin
                     Max:=KG+1;
                     for i:=KG+1 to KG+GG do if b[i]>b[Max] then Max:=i;
                     for i:=KG+1 to KG+GG do begin
                            if i<>Max then begin
                                  b[i]:=b[Max]-b[i];
                                  for j:=1 to ST+KG+GG do
                                      a[i]^[j]:=a[Max]^[j]-a[i]^[j];
                                      BV[i]:=ST+i;BZ[ST+i]:=i;
                                  end;
                            end;
                     a[Max]^[ST+KG+GG+1]:=1;
                     BV[Max]:=ST+KG+GG+1;BZ[ST+KG+GG+1]:=Max;
                     end;
             if GL>0 then begin
                     for i:=KG+GG+1 to m do begin
                            Art[i]:=0;
                            a[i]^[ST+i+GrGl]:=1;
                            BV[i]:=ST+i+GrGl;BZ[ST+i+GrGl]:=i;
                            end;
```

```
                    end;
            if KV>0 then begin
                    Phase1:=true;
                    for j:=ST+KG+GG+1 to n do h[j]:=1;
                    for k:=ST+KG+GG+1 to n do begin
                          for l:=1 to m do begin
                                 if BV[l]=k then begin i:=l;l:=m end;
                                 end;
                          for j:=1 to n do h[j]:=h[j]-a[i]^[j];
                          Hb0:=Hb0-b[i];
                          end;
                    end;
            end;
writeln;writeln;writeln('Sollen Parameter geändert werden ? ');
antwort:=upcase(readkey);if antwort='J'then PARAMETER;
end
else begin
     write(' Bitte warten ! ');LOESCH;Neue_Aufgabe:=false;
     clrscr;gotoxy(4,5);highvideo;
     write('* *  E I N G A B E   Ü B E R   L A U F W E R K   * *');
     lowvideo;gotoxy(4,8);write('Dateiname ?    ');readln(Dateiname);
     (* Warnung: Absturz möglich,falls keine LinSen-Datei ! *)
     (*$I-*)assign(Datei,Dateiname);reset(Datei);
     read(Datei,n,m,ST,KG,GG,GL,KV,Maxmin,Hb0);readln(Datei);
     read(Datei,Zielvar);readln(Datei);
     for i:=0 to KG+GG+GL do begin
         readln(Datei,b[i],bc[i]);readln(Datei);
         for j:=1 to n do readln(Datei,a[i]^[j]);
         for j:=1 to ST do readln(Datei,c[i]^[j]);
         end;
         for i:=1 to KG+GG+GL do readln(Datei,BV[i],Art[i]);
         for j:=1 to n do readln(Datei,H[j],BZ[j]);
         for j:=0 to n do readln(Datei,Namensvektor[j]);
         read(Datei,h1,h2,h3,h4);
     if h1=1 then STNAMEN:=true else STNAMEN:=false;
     if h2=1 then VARNAMEN:=true else VARNAMEN:=false;
     if h3=1 then HILF:=true else HILF:=false;
     if h4=1 then Keine_Restr:=true else Keine_Restr:=false;
     close(Datei);(*$I+*)
     EAFehler:=not(ioresult=0);
     if EAFehler then begin
        gotoxy(4,10);highvideo;write('E/A - Fehler !  ');lowvideo;
        antwort:=readkey;Neue_Aufgabe:=true;
        end;
     end;
until not EAFehler;
end;        (* Ende Eingabeproz. *)

procedure BASISTAUSCH(r,s:integer);
var Piv:double;
begin
if alle_Schritte then AUSGABEMENUE else begin gotoxy(46,11);write(IT:5);end;
(* SIMPLEX-PROZEDUR *)
BZ[BV[r]]:=0;BV[r]:=s;BZ[s]:=r;IT:=IT+1;
Piv:=a[r]^[s];b[r]:=b[r]/Piv;
for j:=1 to n do a[r]^[j]:=a[r]^[j]/Piv;
```

```
for i:=0 to m do begin
     if i<>r then begin
        Piv:=a[i]^[s];b[i]:=b[i]-Piv*b[r];
        for j:=1 to n do a[i]^[j]:=a[i]^[j]-Piv*a[r]^[j];
        end;
    end;
if Phase1 or HILF then begin
   Piv:=H[s];Hb0:=Hb0-Piv*b[r];for j:=1 to n do H[j]:=H[j]-Piv*a[r]^[j];
   end;
end;

procedure PRIMAL_PIVOTZEILE;
begin
if ZYKLUS then for i:=1 to m do ZS[i]:=0;r:=0;
for i:=1 to m do if RUND(a[i]^[s],R1)>0 then begin
    Min:=b[i]/a[i]^[s];r:=i;i:=m end;
if r=0 then UNBESCHR:=true;
if not UNBESCHR then begin
   for i:=1 to m do if RUND(a[i]^[s],R1)>0 then
       if b[i]/a[i]^[s]<Min then begin Min:=b[i]/a[i]^[s];r:=i end;
   if ZYKLUS then begin
      Z:=1;ZS[1]:=r;randomize;
      for i:=1 to m do
          if (i<>r) and (RUND(a[i]^[s],R1)>0) then begin
          if b[i]/a[i]^[s]=Min then begin Z:=Z+1;ZS[Z]:=i end;
          end;
      if Z>1 then r:=ZS[random(Z)+1];
      end;
   end;
end;

procedure DUAL_PIVOTSPALTE;
begin
if ZYKLUS then for j:=1 to n do ZS[j]:=0;s:=0;
for j:=1 to n do if RUND(a[r]^[j],R1)<0 then begin
    Max:=a[0]^[j]/a[r]^[j];s:=j;j:=n end;
if s=0 then LEERLSG:=true;
if not LEERLSG then begin
   for j:=1 to n do if RUND(a[r]^[j],R1)<0 then
       if a[0]^[j]/a[r]^[j]>Max then begin Max:=a[0]^[j]/a[r]^[j];s:=j end;
   if ZYKLUS then begin
      Z:=1;ZS[1]:=s;randomize;
      for j:=1 to n do
          if (j<>s) and (RUND(a[r]^[j],R1)<0) then begin
          if a[0]^[j]/a[r]^[j]=Max then begin Z:=Z+1;ZS[Z]:=j end;
          end;
      if Z>1 then s:=ZS[random(Z)+1];
      end;
   end;
end;

procedure BERECHNUNG;

  procedure Pruefe_Ausartung;
  begin
  eindeutig:=true;
```

```
for j:=1 to n do if (BZ[j]=0) and (RUND(a[0]^[j],R1)=0) then begin
        eindeutig:=false;j:=n end;
ausgeartet:=false;
for i:=1 to m do if RUND(b[i],R1)=0 then begin ausgeartet:=true;i:=m end;
end;

procedure DUAL_PIVOTZEILE;
begin
Min:=0;r:=0;
for i:=1 to m do begin
    GLATT(b[i]);if b[i]<Min then begin Min:=b[i];r:=i end;
    end;
end;

procedure PHASE_I;
var KV_in_Basis,nur_Nullen :boolean;
begin
repeat
 Min:=0;s:=0;
 for j:=1 to n do begin
     GLATT(H[j]);if H[j]<Min then begin Min:=H[j];s:=j end;
     end;
 if s=0 then OPTLSG:=true else PRIMAL_PIVOTZEILE;
 if not OPTLSG then BASISTAUSCH(r,s);
until OPTLSG;OPTLSG:=false;
for i:=1 to m do if BV[i]>ST+KG+GG then begin
                GLATT(b[i]);if b[i]>0 then begin
                                             LEERLSG:=true;i:=m end;
                end;
if alle_Schritte then AUSGABEMENUE;Phase1:=false;
if not LEERLSG then begin
   Phase2:=true;
   for i:=0 to m do for j:=ST+KG+GG+1 to n do a[i]^[j]:=0;
   n:=ST+KG+GG;KV:=0;
   repeat
   KV_in_Basis:=false;
   for i:=1 to m do if BV[i]>ST+KG+GG then begin
                            KV_in_Basis:=true;l:=i;i:=m end;
   if KV_in_Basis then begin
      nur_Nullen:=true;
      for j:=1 to n do begin
          GLATT(a[l]^[j]);if a[l]^[j]<>0 then begin
                           nur_Nullen:=false;k:=j;j:=n end;
          end;
      if nur_Nullen then begin
         (* Zeile l streichen ! *)
         if alle_Schritte then AUSGABEMENUE;
         for j:=1 to n do if BZ[j]>l then BZ[j]:=BZ[j]-1;
         if l<m then for i:=l to m-1 do begin
            BV[i]:=BV[i+1];b[i]:=b[i+1];
            for j:=1 to n do a[i]^[j]:=a[i+1]^[j];
            end;
         for j:=1 to n do a[m]^[j]:=0;b[m]:=0;m:=m-1;
         end
         else BASISTAUSCH(l,k);
      end;
```

```
   until not KV_in_Basis;
   for j:=ST+KG+GG+1 to ST+KG+GG+GrGl+GL do Namensvektor[j]:='*****';
   end;
end;

procedure PHASE_II;
begin
repeat
 Min:=0;s:=0;
 for j:=1 to n do begin
    GLATT(a[0]^[j]);if a[0]^[j]<Min then begin Min:=a[0]^[j];s:=j end;
    end;
 if s=0 then begin OPTLSG:=true;Pruefe_Ausartung end
        else PRIMAL_PIVOTZEILE;
 if not(OPTLSG or UNBESCHR) then BASISTAUSCH(r,s);
until OPTLSG or UNBESCHR;
end;

procedure DUAL_SIMPLEX;
begin
repeat
 DUAL_PIVOTZEILE;
 if r=0 then begin OPTLSG:=true;Pruefe_Ausartung end
    else DUAL_PIVOTSPALTE;
 if not(OPTLSG or LEERLSG) then BASISTAUSCH(r,s);
until OPTLSG or LEERLSG;
end;

procedure DUAL_HILF;
begin
(* Heuristische Hilfszielfunktion *)
HILF:=true;Min:=0;Hb0:=0;
for j:=1 to n do begin
    GLATT(a[0]^[j]);if a[0]^[j]<Min then Min:=a[0]^[j];
    end;
for j:=1 to n do
    if BZ[j]>0 then H[j]:=0 else H[j]:=a[0]^[j]-Min+1; (* H[j] ≥ 0 *)
repeat
 DUAL_PIVOTZEILE;
 if r=0 then OPTLSG:=true
    else begin  (* Auswahl Pivotspalte bei Hilfszielfkt. *)
         if ZYKLUS then for j:=1 to n do ZS[j]:=0;s:=0;
         for j:=1 to n do if RUND(a[r]^[j],R1)<0 then begin
             Max:=H[j]/a[r]^[j];s:=j;j:=n end;
         if s=0 then LEERLSG:=true;
         if not LEERLSG then begin
            for j:=1 to n do if RUND(a[r]^[j],R1)<0 then
                if H[j]/a[r]^[j]>Max then begin Max:=H[j]/a[r]^[j];s:=j end;
            if ZYKLUS then begin
               Z:=1;ZS[1]:=s;randomize;
               for j:=1 to n do
                   if (j<>s) and (RUND(a[r]^[j],R1)<0) then begin
                   if H[j]/a[r]^[j]=Max then begin Z:=Z+1;ZS[Z]:=j end;
                   end;
               if Z>1 then s:=ZS[random(Z)+1];
               end;
```

```
              end;
            end;
    if not(OPTLSG or LEERLSG) then BASISTAUSCH(r,s);
   until OPTLSG or LEERLSG;OPTLSG:=false;
   HILF:=false;
   end;

begin
Phase2:=false;OPTLSG:=false;LEERLSG:=false;UNBESCHR:=false;IT:=0;
clrscr;gotoxy(28,11);highvideo;write('║  Iteration Nr.        ║');
gotoxy(28,10);write('╔══════════════════════╗');
gotoxy(28,12);write('╚══════════════════════╝');lowvideo;
Pruefe_ZUL;
if ZUL then begin
 if KV>0 then begin
 PHASE_I;if not LEERLSG then PHASE_II;
 end
 else PHASE_II;
end
else begin
 Pruefe_OPT;
 if OPT then DUAL_SIMPLEX
 else begin
 DUAL_HILF;if not LEERLSG then PHASE_II;
 end;
end;
if not GANZ then ERGEBNISAUSGABE;
end; (* Berechnung *)

procedure SENSIBILITAETSANALYSE;
var Sensib  :char;
    Piv     :double;
    Nr      :integer;
    Name    :string[5];
    d       :array[1..maxzeil] of double;

procedure BERECHNE_d;
begin
for i:=1 to m do if BV[i]<=ST then d[i]:=-c[0]^[BV[i]] else d[i]:=0;
end;

procedure AUSTRITT_STRUKTURVAR(Nr:integer);
begin
r:=BZ[Nr];DUAL_PIVOTSPALTE;
if not(OPTLSG and (s>0)) then begin
   (* erstbestes Pivotelement, aber nicht die 1 in der Spalte Nr !!!  *)
   for j:=1 to n do if (RUND(a[r]^[j],R1)<>0) and (j<>Nr)
       then begin s:=j;j:=n end;
   end;
BASISTAUSCH(r,s);
end;

procedure EINTRITT_SCHLUPFVAR(Nr:integer);
begin
s:=Nr;PRIMAL_PIVOTZEILE;
if not(OPTLSG and (r>0)) then begin
```

```
   (* erstbestes Pivotelement *)
   for i:=1 to m do if RUND(a[i]^[s],R1)<>0 then begin r:=i;i:=m end;
   end;
BASISTAUSCH(r,s);
end;

procedure LIES_SPALTE(Nr:integer);
begin
repeat
    write('  Zielfunktionsbeitrag ?          ',Maxmin*c[0]^[Nr]:12:4,'     ');
    Lies_reelle_Zahl;
    if Kode=0 then c[0]^[Nr]:=Maxmin*reelle_Zahl
       else begin gotoxy(55,wherey-1);writeln(Maxmin*c[0]^[Nr]:12:4) end;
    for i:=1 to m do begin
        write('   Koeffizient ',i:3,' ?                    ',c[i]^[Nr]:12:4,'     ');
        Lies_reelle_Zahl;
        if Kode=0 then c[i]^[Nr]:=reelle_Zahl
        else begin gotoxy(55,wherey-1);writeln(c[i]^[Nr]:12:4) end;
        end;
   writeln;write('   Eingabe korrekt ? ');antwort:=upcase(readkey);
   writeln;writeln;
   until antwort<>'N';
end;

procedure NEUBERECHNUNG_ZLFKT;
(* wird von mehreren Prozeduren benötigt, da Zielfunktionszeile *)
(* anfällig gegenüber kumulierten Rundungsfehlern !             *)
begin
for j:=1 to ST do a[0]^[j]:=c[0]^[j];
   for j:=ST+1 to n do a[0]^[j]:=0;b[0]:=bc[0];
   (* Wiederherstellung der kanonischen Form *)
   for k:=1 to n do if (BZ[k]>0) and (RUND(a[0]^[k],R1)<>0) then begin
          PIV:=a[0]^[k];
          for j:=1 to n do a[0]^[j]:=a[0]^[j]-PIV*a[BZ[k]]^[j];
          b[0]:=b[0]-PIV*b[BZ[k]]
          end;
end;

procedure ZIELFKTKOEFF;
var Zlfkt,Ausgabeart       :char;
    UGexist,OGexist,Druck :boolean;
begin
clrscr;highvideo;gotoxy(4,5);
write('* Änderung der Zielfunktionsbeiträge *');gotoxy(4,8);write('1');
gotoxy(4,10);write('2');lowvideo;gotoxy(8,8);writeln('isoliert');gotoxy(8,10);
write('simultan');gotoxy(4,12);write('');zlfkt:=readkey;
if not optlsg then zlfkt:='2';
if zlfkt='2'then begin
   EINGZLFKT;
   NEUBERECHNUNG_ZLFKT;
   BERECHNUNG;
   end
   else begin
   highvideo;write('D');lowvideo;write('rucken  ');
   Ausgabeart:=upcase(readkey);
   if Ausgabeart='D' then begin
```

```
   DRUCKPROBE;if not Drucker_bereit then begin
              write('Nicht bereit ! ');Ausgabeart:=readkey;Ausgabeart:='B'
              end;
   end;
if Ausgabeart='D' then Druck:=true else Druck:=false;
(*$I-*)clrscr;gotoxy(4,5);highvideo;
write('* Isolierte Schwankungsbreiten der Zielfunktionsbeiträge *');
lowvideo;gotoxy(1,8);
if Druck then begin
   writeln(lst,
      '* Isolierte Schwankungsbreiten der Zielfunktionsbeiträge *');
   writeln(lst);writeln(lst);
   end;
for j:=1 to ST do begin
    if wherey>23 then begin antwort:=readkey;clrscr end;
    write('    ',Namensvektor[j],'  :   ');
    if Druck then write(lst,Namensvektor[j],'  :   ');
    if BZ[j]=0 then begin
       (* falls xj NBV *)
       if Maxmin=1 then begin
          (* Min-Problem *)
          writeln('[ ',c[0]^[j]-a[0]^[j]:12:4,' ; +∞            [');
          if Druck then
           writeln(lst,'[ ',c[0]^[j]-a[0]^[j]:12:4,' ; +∞            [');
          end
          else begin
          (* Max-Problem *)
          writeln(']            -∞ ; ',-c[0]^[j]+a[0]^[j]:12:4,' ]');
          if Druck then
             writeln(lst,']            -∞ ; ',-c[0]^[j]+a[0]^[j]:12:4,' ]');
          end;
       end
       else begin
       (* falls xj BV *)
       UGexist:=false;OGexist:=false;
       (* Untergrenze für Änderung *)
       for k:=1 to n do if k<>j then
           if RUND(Maxmin*a[BZ[j]]^[k],R1)<0 then begin
              MAX:=Maxmin*a[0]^[k]/a[BZ[j]]^[k];k:=n;UGexist:=true
              end;
       if UGexist then begin
          for k:=1 to n do if k<>j then
              if RUND(Maxmin*a[BZ[j]]^[k],R1)<0 then
                 if Maxmin*a[0]^[k]/a[BZ[j]]^[k]>MAX then
                    MAX:=Maxmin*a[0]^[k]/a[BZ[j]]^[k];
          write('[ ',Maxmin*c[0]^[j]+MAX:12:4,' ; '); if Druck then
           write(lst,'[ ',Maxmin*c[0]^[j]+MAX:12:4,' ; ');
          end
          else begin
          write(']           -∞ ; ');if Druck then
              write(lst,']           -∞ ; ');
          end;
       (* Obergrenze für Änderung *)
       for k:=1 to n do if k<>j then
           if RUND(Maxmin*a[BZ[j]]^[k],R1)>0 then begin
              MIN:=Maxmin*a[0]^[k]/a[BZ[j]]^[k];k:=n;OGexist:=true
```

```
              end;
          if OGexist then begin
             for k:=1 to n do if k<>j then
                 if RUND(Maxmin*a[BZ[j]]^[k],R1)>0 then
                    if Maxmin*a[0]^[k]/a[BZ[j]]^[k]<MIN then
                       MIN:=Maxmin*a[0]^[k]/a[BZ[j]]^[k];
             writeln(Maxmin*c[0]^[j]+MIN:12:4,' ]'); if Druck then
              writeln(lst,Maxmin*c[0]^[j]+MIN:12:4,' ]');
             end
             else begin
             writeln('+∞            [');if Druck then
                     writeln(lst,'+∞            [');
             end;
          end;  (* xj BV *)
       end;(*$I+*)
   gotoxy(4,wherey);antwort:=readkey;
   end;
end;

procedure RECHTE_SEITEN;
var UGexist,OGexist,Druck   :boolean;
    RechteS,Ausgabeart      :char;
begin
clrscr;gotoxy(4,5);highvideo;
write('* Änderung der rechten Seiten *');gotoxy(4,8);write('1');gotoxy(4,10);
write('2');lowvideo;gotoxy(8,8);write('isoliert');gotoxy(8,10);
write('simultan');gotoxy(4,12);write('');RechteS:=readkey;
if not optlsg then RechteS:='2';
if RechteS='2' then begin
   repeat (* Eingabe RS *)
    clrscr;highvideo;gotoxy(4,5);
    write('* Eingabe der rechten Seiten *');lowvideo;gotoxy(1,9);
    for i:=0 to m do begin
        if i=0 then writeln('Absolutglied der Zielfunktion ?  ')
           else writeln('Rechte Seite der Restriktion ',i,' ?  ');
        gotoxy(40,wherey-1);
        if i>0 then writeln(RUND(bc[i],4):12:4)
              else writeln(RUND(-Maxmin*bc[i],4):12:4);
        gotoxy(60,wherey-1);clreol;Lies_reelle_Zahl;
        if Kode=0 then begin
           bc[i]:=Reelle_Zahl;if i=0 then bc[i]:=-Maxmin*bc[i];
           end
           else begin
           gotoxy(60,wherey-1);
           if i>0 then writeln(RUND(bc[i],4):12:4)
                 else writeln(RUND(-Maxmin*bc[i],4):12:4)
           end;
        end;
    writeln;writeln;write('Eingabe korrekt ? ');antwort:=upcase(readkey);
   until antwort<>'N';
   (* Neuberechnung der rechten Seite *)
   for i:=1 to m do begin
       b[i]:=0;
       for j:=ST+1 to n do b[i]:=b[i]+a[i]^[j]*bc[j-ST]*Art[j-ST];
       GLATT(b[i]);
       end;
```

```
b[0]:=bc[0];BERECHNE_d;
for i:=1 to m do b[0]:=b[0]+d[i]*b[i];GLATT(b[0]);
BERECHNUNG;NEUBERECHNUNG_ZLFKT;
end  (* simultan *)
else begin
highvideo;write('D');lowvideo;write('rucken  ');
Ausgabeart:=upcase(readkey);
if Ausgabeart='D' then begin
   DRUCKPROBE;if not Drucker_bereit then begin
             write('Nicht bereit ! ');Ausgabeart:=readkey;Ausgabeart:='B';
             end;
   end;
if Ausgabeart='D' then Druck:=true else Druck:=false;
(*$I-*)clrscr;gotoxy(4,5);highvideo;
write('* Isolierte Schwankungsbreiten der rechten Seiten *');
lowvideo;gotoxy(1,8);
if Druck then begin
   writeln(lst,'* Isolierte Schwankungsbreiten der rechten Seiten *');
   writeln(lst);writeln(lst);
   end;
for i:=1 to m do begin
    if wherey>23 then begin antwort:=readkey;clrscr end;
    write('   Restriktion ',i:3,'  :   ');
    if Druck then write(lst,'Restriktion ',i:3,'  :   ');
    UGexist:=false;OGexist:=false;
    (* Untergrenze für Änderung *)
    for l:=1 to m do if RUND(a[l]^[ST+i],R1)>0 then begin
        MAX:=-b[l]/a[l]^[ST+i];l:=m;UGexist:=true;
        end;
    if UGexist then
       for l:=1 to m do if RUND(a[l]^[ST+i],R1)>0 then
           if -b[l]/a[l]^[ST+i]>MAX then MAX:=-b[l]/a[l]^[ST+i];
    (* Obergrenze für Änderung *)
    for l:=1 to m do if RUND(a[l]^[ST+i],R1)<0 then begin
        MIN:=-b[l]/a[l]^[ST+i];l:=m;OGexist:=true;
        end;
    if OGexist then
       for l:=1 to m do if RUND(a[l]^[ST+i],R1)<0 then
           if -b[l]/a[l]^[ST+i]<MIN then MIN:=-b[l]/a[l]^[ST+i];
    (* Ausgabe *)
    if Art[i]=1 then begin
       if UGexist then begin
          write('[ ',bc[i]+MAX:12:4,' ; ');if Druck then
          write(lst,'[ ',bc[i]+MAX:12:4,' ; ');
          end
          else begin
          write(']          -∞ ; ');if Druck then
          write(lst,']          -∞ ; ');
          end;
       if OGexist then begin
          writeln(bc[i]+MIN:12:4,' ] ');if Druck then
          writeln(lst,bc[i]+MIN:12:4,' ] ');
          end
          else begin
          writeln('+∞          [ ');if Druck then
          writeln(lst,'+∞          [ ');
```

```
            end;
         end (* ≤ *)
         else begin
         if OGexist then begin
            write('[ ',bc[i]-MIN:12:4,' ; ');if Druck then
            write(lst,'[ ',bc[i]-MIN:12:4,' ; ');
            end
            else begin
            write(']          -∞ ; ');if Druck then
            write(lst,']          -∞ ; ');
            end;
         if UGexist then begin
            writeln(bc[i]-MAX:12:4,' ] ');if Druck then
            writeln(lst,bc[i]-MAX:12:4,' ] ');
            end
            else begin
            writeln('+∞          [ ');if Druck then
            writeln(lst,'+∞          [ ');
            end;
         end;(* ≥ *)
      end;
   gotoxy(4,wherey);antwort:=readkey;
   end; (*$I+*)(* isoliert *)
end;

procedure TABLEAUKOEFF;
begin
clrscr;highvideo;gotoxy(4,5);
write('* Änderung der Koeffizientenspalte einer Strukturvariablen *');
gotoxy(4,8);lowvideo;write('Welche Strukturvariable ?  ');readln(Name);
Name:=concat(Name,'      ');writeln;writeln;
Nr:=0;for j:=1 to ST do if Namensvektor[j]=Name then begin Nr:=j;j:=ST end;
if Nr=0 then begin
   write('  Eine Strukturvariable "',Name,'" ist nicht definiert ! ');
   antwort:=readkey;
   end
 else begin
   BERECHNE_d;
   LIES_SPALTE(Nr);
   (* Neuberechnung der Spalte im Optimaltableau *)
   for i:=1 to m do begin
      a[i]^[Nr]:=0;
      for j:=ST+1 to n do a[i]^[Nr]:=a[i]^[Nr]+a[i]^[j]*c[j-ST]^[Nr]*Art[j-ST];
      GLATT(a[i]^[Nr]);
      end;
   a[0]^[Nr]:=c[0]^[Nr];
   for i:=1 to m do a[0]^[Nr]:=a[0]^[Nr]+d[i]*a[i]^[Nr];  (* altes d !! *)
   GLATT(a[0]^[Nr]);
   (* Falls BV: Fallunterscheidung Pivotelement *)
   if BZ[Nr]>0 then begin
      if RUND(a[BZ[Nr]]^[Nr],R1)=0 then
        AUSTRITT_STRUKTURVAR(Nr)
        else begin
        (* Wiederherstellung der kanonischen Form *)
        Piv:=a[BZ[Nr]]^[Nr];for j:=1 to n do a[BZ[Nr]]^[j]:=a[BZ[Nr]]^[j]/Piv;
        b[BZ[Nr]]:=b[BZ[Nr]]/Piv;
```

```
        for i:=0 to m do if i<>BZ[Nr] then if RUND(a[i]^[Nr],R1)<>0 then begin
                Piv:=a[i]^[Nr];
                for j:=1 to n do a[i]^[j]:=a[i]^[j]-Piv*a[BZ[Nr]]^[j];
                b[i]:=b[i]-Piv*b[BZ[Nr]];
                end;
        end;
      end;
   BERECHNUNG;
   NEUBERECHNUNG_ZLFKT;
 end;
end;

procedure STRUKTURVAR;
var StV   :char;
begin
clrscr;gotoxy(4,5);highvideo;write('* Strukturvariable *');gotoxy(4,8);
write('1');gotoxy(4,10);write('2');lowvideo;gotoxy(8,8);if ST<maxzeil then
write('hinzufügen');gotoxy(8,10);if ST>1 then write('streichen');gotoxy(4,12);
write('');StV:=readkey;
if ((StV='1') and (ST=maxzeil)) or ((StV='2') and (ST=1)) then StV:='f';
if StV='1' then begin
   (* Rechtsverschiebung des Tableaus *)
   for i:=0 to m do begin
       for j:=ST+KG+GG+1 downto ST+2 do a[i]^[j]:=a[i]^[j-1];a[i]^[ST+1]:=0;
       c[i]^[ST+1]:=0;if BV[i]>=ST+1 then BV[i]:=BV[i]+1;
       end;
       for j:=ST+KG+GG+1 downto ST+2 do begin
           BZ[j]:=BZ[j-1];Namensvektor[j]:=Namensvektor[j-1];
           end;
   BZ[ST+1]:=0;
   gotoxy(4,12);write('Name der neuen Strukturvariablen ?   ');
   readln(Namensvektor[ST+1]);
   Namensvektor[ST+1]:=concat(Namensvektor[ST+1],'      ');
   ST:=ST+1;n:=n+1;writeln;writeln;LIES_SPALTE(ST);
   (* Berechnung der Spalte im Optimaltableau *)
   for i:=1 to m do begin
       for j:=ST+1 to n do
        a[i]^[ST]:=a[i]^[ST]+a[i]^[j]*c[j-ST]^[ST]*Art[j-ST];
       GLATT(a[i]^[ST]);
       end;
   a[0]^[ST]:=c[0]^[ST];BERECHNE_d;
   for i:=1 to m do a[0]^[ST]:=a[0]^[ST]+d[i]*a[i]^[ST];
   BERECHNUNG;
   end;(* einfügen *)
if StV='2' then begin
   gotoxy(4,12);write('Welche Strukturvariable soll wegfallen ?    ');
   readln(Name);Name:=concat(Name,'      ');Nr:=0;
   for j:=1 to ST do if Namensvektor[j]=Name then begin Nr:=j;j:=ST end;
   if Nr=0 then begin
      gotoxy(4,14);
      write('Eine Strukturvariable "',Name,'" ist nicht bekannt !  ');
      antwort:=readkey;
      end
      else begin
      if BZ[Nr]>0 then AUSTRITT_STRUKTURVAR(Nr);
      (* Linksverschiebung des Tableaus *)
```

```
        for i:=0 to m do begin
            for j:=Nr to n-1 do a[i]^[j]:=a[i]^[j+1];a[i]^[n]:=0;
            if Nr<ST then for j:=Nr to ST-1 do c[i]^[j]:=c[i]^[j+1];
            c[i]^[ST]:=0;if BV[i]>Nr then BV[i]:=BV[i]-1;
            end;
        for j:=Nr to n-1 do begin
            BZ[j]:=BZ[j+1];Namensvektor[j]:=Namensvektor[j+1];
            end;
        BZ[n]:=0;Namensvektor[n]:='*****';
        ST:=ST-1;n:=n-1;
        BERECHNUNG;
        end;
   end;(* streichen *)
NEUBERECHNUNG_ZLFKT;
end;

procedure RESTRIKTIONEN;
var Restr  :char;
begin
clrscr;gotoxy(4,5);highvideo;write('* Restriktion *');gotoxy(4,8);write('1');
gotoxy(4,10);write('2');lowvideo;gotoxy(8,8);if m<maxzeil then
write('hinzufügen');gotoxy(8,10);if m>1 then write('streichen');gotoxy(4,12);
write('');Restr:=readkey;
if ((Restr='1') and (m=maxzeil)) or ((Restr='2') and (m=1)) then Restr:='f';
if Restr='1' then begin
   (* Einlesen *)
   gotoxy(4,12);write('Von welcher Art ist die neue Restriktion ? ');
   highvideo;gotoxy(4,14);write('1');gotoxy(4,16);write('2');lowvideo;
   gotoxy(8,14);write('≤');gotoxy(8,16);write('≥');gotoxy(4,18);write('');
   antwort:=readkey;
   if antwort='2' then Art[m+1]:=-1 else Art[m+1]:=1;writeln;writeln;
   repeat
    write('    Rechte Seite ?                    ',bc[m+1]:12:4,'     ');
    Lies_reelle_Zahl;
    if Kode=0 then bc[m+1]:=reelle_Zahl
       else begin gotoxy(55,wherey-1);writeln(bc[m+1]:12:4) end;
    for j:=1 to ST do begin
        write('    Koeffizient von ',Namensvektor[j]);
        write(' ?         ',c[m+1]^[j]:12:4,'     ');
        Lies_reelle_Zahl;
        if Kode=0 then c[m+1]^[j]:=reelle_Zahl
        else begin gotoxy(55,wherey-1);writeln(c[m+1]^[j]:12:4) end;
        end;
    writeln;write('    Eingabe korrekt ? ');antwort:=upcase(readkey);
    writeln;writeln;
   until antwort<>'N';
   write('    Name der zugehörigen Schlupfvariablen ?    ');
   readln(Namensvektor[n+1]);
   Namensvektor[n+1]:=concat(Namensvektor[n+1],'       ');
   (* Anfügen an Endtableau *)
   BV[m+1]:=n+1;BZ[n+1]:=m+1;b[m+1]:=bc[m+1]*Art[m+1];a[m+1]^[n+1]:=1;
   for j:=1 to ST do a[m+1]^[j]:=c[m+1]^[j]*Art[m+1];
   for j:=ST+1 to n do a[m+1]^[j]:=0;for i:=0 to m do a[i]^[n+1]:=0;
   if Art[m+1]=1 then KG:=KG+1 else GG:=GG+1;m:=m+1;n:=n+1;
   (* Wiederherstellung der kanonischen Form *)
   for k:=1 to n-1 do if (BZ[k]>0) and (RUND(a[m]^[k],R1)<>0) then begin
```

```
        Piv:=a[m]^[k];
        for j:=1 to n-1 do a[m]^[j]:=a[m]^[j]-Piv*a[BZ[k]]^[j];
        b[m]:=b[m]-Piv*b[BZ[k]];
        for j:=1 to n-1 do GLATT(a[m]^[j]);GLATT(b[m]);
        end;
   BERECHNUNG;
   end;(* einfügen *)
if Restr='2' then begin
   gotoxy(4,12);write('Nummer der wegfallenden Restriktion ?  ');
   repeat Lies_ganze_Zahl(1,m) until Kode=0;Nr:=ST+ganze_Zahl;
   if BZ[Nr]=0 then EINTRITT_SCHLUPFVAR(Nr);r:=BZ[Nr];
   if Art[Nr-ST]=1 then KG:=KG-1 else GG:=GG-1;
   (* Spalte der in Basis befindlichen Schlupfvar. streichen *)
   for i:=0 to m do begin
       if Nr<n then for j:=Nr to n-1 do a[i]^[j]:=a[i]^[j+1];a[i]^[n]:=0;
       if BV[i]>Nr then BV[i]:=BV[i]-1;
       end;
   if Nr<n then for j:=Nr to n-1 do begin
       BZ[j]:=BZ[j+1];Namensvektor[j]:=Namensvektor[j+1];
       end;
   BZ[n]:=0;Namensvektor[n]:='*****';n:=n-1;
   (* Zeile der in Basis befindlichen Schlupfvariable streichen *)
   for j:=1 to n do begin
       if r<m then for i:=r to m-1 do a[i]^[j]:=a[i+1]^[j];a[m]^[j]:=0;
       if BZ[j]>r then BZ[j]:=BZ[j]-1;
       end;
   for j:=1 to ST do begin
       if Nr-ST<m then for i:=Nr-ST to m-1 do c[i]^[j]:=c[i+1]^[j];
       c[m]^[j]:=0;
       end;
   if r<m then for i:=r to m-1 do begin
      b[i]:=b[i+1];BV[i]:=BV[i+1]
      end;
   b[m]:=0;BV[m]:=0;
   if Nr-ST<m then for i:=Nr-ST to m-1 do begin
      bc[i]:=bc[i+1];Art[i]:=Art[i+1]
      end;
   bc[m]:=0;Art[m]:=0;
   m:=m-1;
   BERECHNUNG;
   end;(* streichen *);
NEUBERECHNUNG_ZLFKT;
end;

procedure GANZZAHLIGKEIT;
var g                    :array[1..maxspal] of char;
    FERTIG,VOLL          :boolean;
    Wahl                 :char;
    Schnitte             :longint;
    Num                  :string[5];

function GAUSS(x:double):double;  (* nicht ":integer;" wegen Wertebereich ! *)
begin
if (x=int(x)) or (x>0) then GAUSS:=int(x) else GAUSS:=int(x-1)
end;
```

```
function BRUCH(x:double):double;
begin
BRUCH:=x-GAUSS(x)
end;

begin
FERTIG:=false;VOLL:=false;Schnitte:=0;clrscr;gotoxy(4,3);highvideo;
write('* Ganzzahligkeit von Strukturvariablen *');lowvideo;gotoxy(4,6);
writeln('Als Schlußpunkt der Sensibilitätsanalyse kann überprüft werden,');
writeln('   welche Auswirkung Ganzzahligkeitsforderungen auf die Lösung der');
writeln('   Aufgabe haben. ');
writeln('   Verwendet wird Gomorys Schnittebenenverfahren für gemischt-ganz-');
writeln('   zahlige Probleme. LinSen erlaubt daher i.a. nur eine sehr kleine');
writeln('   Anzahl ganzzahliger Variabler.');writeln;
writeln('   Wegen der hinzutretenden Schnittebenen sind anschließende Sensi-');
writeln('   bilitätsbetrachtungen nicht mehr sinnvoll; es empfiehlt sich');
writeln('   also, die Aufgabe ggf. vorher abzuspeichern.');writeln;writeln;
gotoxy(4,18);highvideo;write('1');gotoxy(4,20);write('2');gotoxy(4,22);
write('3');lowvideo;gotoxy(8,18);write('Zurück zum Hauptmenü');gotoxy(8,20);
write('Weiter ( ohne Speicherung der Aufgabe )');gotoxy(8,22);
write('Weiter ( mit  Speicherung der Aufgabe )');gotoxy(4,24);write('');
Wahl:=readkey;
if Wahl in ['2','3'] then begin
   GANZ:=true;
   if Wahl='3' then AUFGABE_SPEICHERN;
   clrscr;gotoxy(4,5);
   write('Bitte "G" eingeben, wenn die Variable ganzzahlig werden soll !');
   repeat
   writeln;writeln;
   for j:=1 to ST do begin
       write('   ',Namensvektor[j],'   ');antwort:=upcase(readkey);
       if antwort='G' then g[j]:='G' else g[j]:='R';
       if g[j]='G' then writeln('ganzzahlig') else writeln('reell');
       end;
   writeln;write('   Eingabe korrekt ?   ');antwort:=upcase(readkey);
   until antwort<>'N';
   for j:=1 to ST do if g[j]='G'
       then insert('G',Namensvektor[j],5);
   for j:=ST+1 to maxspal do g[j]:='R';
   (* GOMORY II - Algorithmus *)
   repeat
   if Schnitte>0 then BERECHNUNG;
   if not LEERLSG then begin
      (* Streichen unwirksamer Schnittebenen *)
      for l:=1 to m do if BV[l]>ST+KG+GG then begin
          (* Spalte der in Basis befindlichen Schlupfvar. streichen *)
          Nr:=BV[l];r:=BZ[Nr];
          for i:=0 to m do begin
             if Nr<n then for j:=Nr to n-1 do a[i]^[j]:=a[i]^[j+1];a[i]^[n]:=0;
             if BV[i]>Nr then BV[i]:=BV[i]-1;
             end;
          if Nr<n then for j:=Nr to n-1 do begin
             BZ[j]:=BZ[j+1];Namensvektor[j]:=Namensvektor[j+1];
             end;
          BZ[n]:=0;Namensvektor[n]:='*****';n:=n-1;
          (* Zeile der in Basis befindlichen Schlupfvariable streichen *)
```

```
            for j:=1 to n do begin
                if r<m then for i:=r to m-1 do a[i]^[j]:=a[i+1]^[j];a[m]^[j]:=0;
                if BZ[j]>r then BZ[j]:=BZ[j]-1;
                end;
            if r<m then for i:=r to m-1 do begin
              b[i]:=b[i+1];BV[i]:=BV[i+1]
              end;
            b[m]:=0;BV[m]:=0;
            m:=m-1;
            l:=1;
            end;
        FERTIG:=true;
        for i:=1 to m do begin
            GLATT(b[i]);
            if (b[i]<>int(b[i])) and (g[BV[i]]='G') then begin
                                        FERTIG:=false;r:=i;i:=m
                                        end;
            end;
        if (not FERTIG) and (m=maxzeil) then VOLL:=true;
        if (not FERTIG) and (not VOLL) then begin
           (* Gomory-Schnitt *)
           Schnitte:=Schnitte+1;str(Schnitte,Num);
           Namensvektor[n+1]:=concat(Num,'g    ');
           BV[m+1]:=n+1;BZ[n+1]:=m+1;a[m+1]^[n+1]:=1;
           for i:=0 to m do a[i]^[n+1]:=0;
           b[m+1]:=-BRUCH(b[r]);
           for j:=1 to n do begin
               GLATT(a[r]^[j]);
               if g[j]='G' then begin
                  (* xj soll ganzz. werden *)
                  if BRUCH(a[r]^[j])<=BRUCH(b[r]) then
                     a[m+1]^[j]:=-BRUCH(a[r]^[j]) else
                     a[m+1]^[j]:=BRUCH(b[r])/(1-BRUCH(b[r]))*(BRUCH(a[r]^[j])-1);
                  end
                  else begin
                  (* xj braucht nicht ganzz. zu werden *)
                  if a[r]^[j]>=0 then a[m+1]^[j]:=-a[r]^[j] else
                     a[m+1]^[j]:=BRUCH(b[r])/(1-BRUCH(b[r]))*a[r]^[j];
                  end;
               end;
           m:=m+1;n:=n+1;
           end;
        end;
    until FERTIG or LEERLSG or VOLL;
    if FERTIG then NEUBERECHNUNG_ZLFKT;
    clrscr;gotoxy(4,5);highvideo;
    if FERTIG or LEERLSG then
       write('Schnittebenenverfahren abgeschlossen ! ') else
       write('Speicherplatz erschöpft !!! ');antwort:=readkey;
    end;
end;
```

```
procedure ALTERNENDTABLEAU;
var AltT  :char;

 procedure Lies_Var;
 begin
 writeln;write('   Welche Variable ?  ');readln(Name);
 Name:=concat(Name,'      ');Nr:=0;
 for j:=1 to n do if Namensvektor[j]=Name then begin Nr:=j;j:=n end;
 if Nr=0 then begin writeln;write('   Unbekannt !  ');antwort:=readkey end;
 end;

begin
clrscr;gotoxy(4,5);highvideo;write('* Alternatives Optimaltableau *');
gotoxy(4,8);write('1');gotoxy(4,10);write('2');lowvideo;gotoxy(8,8);
if not eindeutig then
   write('Eintritt einer Variable in die Basis');gotoxy(8,10);
if ausgeartet then write('Austritt einer Basisvariable');gotoxy(4,12);
write('');AltT:=readkey;
if ((AltT='1') and eindeutig) or ((AltT='2') and (not ausgeartet))
   then AltT:='f';
if AltT='1' then begin
   writeln;Lies_Var;
   if (Nr>0) and (RUND(a[0]^[Nr],R1)=0) then if BZ[Nr]=0 then begin
      s:=Nr;PRIMAL_PIVOTZEILE;
      if r>0 then BASISTAUSCH(r,s)
         else begin
              writeln;write('   Eintritt in Basis nicht möglich !  ');
              antwort:=readkey;
              end;
      end;
   end;
if AltT='2' then begin
   writeln;Lies_Var;
   if (Nr>0) then if BZ[Nr]>0 then if RUND(b[BZ[Nr]],R1)=0 then begin
      r:=BZ[Nr];DUAL_PIVOTSPALTE;
      if s>0 then BASISTAUSCH(r,s)
         else begin
              writeln;write('   Austritt aus Basis nicht möglich !  ');
              end;
      end;
   end;
BERECHNUNG;
end;

begin
Keine_Restr:=false;
if KV>0 then begin
   (* Künstliche Variable aus der Basis treiben *)
   for i:=0 to m do a[i]^[ST+KG+GG+1]:=0;
   n:=ST+KG+GG;KV:=0;r:=BZ[ST+KG+GG+1];
   for j:=1 to n do begin
       GLATT(a[r]^[j]);
       if a[r]^[j]<>0 then begin s:=j;j:=n end;
       end;
   BASISTAUSCH(r,s);BERECHNUNG;
   end; (* KV aus Basis *)
```

```
clrscr;highvideo;gotoxy(4,4);
write('* *  SENSIBILITÄTSANALYSE  * *');
for i:=1 to 7 do begin gotoxy(4,6+2*i);write(i) end;lowvideo;
gotoxy(8,8);write('Zielfunktionsbeiträge ändern');
gotoxy(8,10);write('Rechte Seiten ändern');gotoxy(8,12);
write('Tableaukoeffizienten ändern');gotoxy(8,14);
write('Strukturvariable hinzufügen/streichen');gotoxy(8,16);
write('Restriktion hinzufügen/streichen');gotoxy(8,18);
if OPTLSG then write('Ganzzahligkeit');gotoxy(8,20);
if (OPTLSG and (ausgeartet or not eindeutig)) then
write('Alternatives Optimaltableau');gotoxy(4,22);write('');
repeat
 Sensib:=readkey;
 if (Sensib='6') and (not optlsg) then Sensib:='f';
 if (Sensib='7') then if (not optlsg) or (optlsg and not(ausgeartet or
                                              not eindeutig)) then Sensib:='f';
until Sensib in ['1','2','3','4','5','6','7'];
Case Sensib of

    '1':ZIELFKTKOEFF;
    '2':RECHTE_SEITEN;
    '3':TABLEAUKOEFF;
    '4':STRUKTURVAR;
    '5':RESTRIKTIONEN;
    '6':GANZZAHLIGKEIT;
    '7':ALTERNENDTABLEAU;

   end;
end;

Begin            (*   H A U P T P R O G R A M M   *)
Initialisierung;
Programmstart;
Repeat
  Hauptmenue;
  Case Haupt of

     '1':begin EINGABE;BERECHNUNG end;
     '2':AUFGABE_AUSGEBEN;
     '3':TABLEAU_AUSGEBEN;
     '4':BASISLSG_AUSGEBEN;
     '5':SENSIBILITAETSANALYSE;
     '6':AUFGABE_SPEICHERN;
     '7':PARAMETER;

   end
until Haupt='8';
clrscr
End.
```

Literaturverzeichnis

Adam, D. (Produktionspolitik) Produktionspolitik, 5. Aufl., Wiesbaden 1988.

Barnett, S. (Stability) Stability of the Solution to a Linear Programming Problem, in: Operational Research Quarterly, Bd. 13 (1962), S. 219-228.

Beale, E.M.L. (Cycling) Cycling in the Dual Simplex Algorithm, in: Naval Research Logistics Quarterly, Bd. 2 (1955), S. 269-275.

Beale, E.M.L. (Mathematical Programming) Mathematical Programming in Practice, Belfast 1968.

Biethahn, J. (EDV) Einführung in die EDV für Wirtschaftswissenschaftler, 6. Aufl., München/Wien 1989.

Bloech, J. (Änderung) Zum Problem der nachträglichen Änderung industrieller Produktionsprogramme, in: ZfB, 36. Jg. (1966), S. 186-197.

Burkard, R.E. (Methoden) Methoden der ganzzahligen Optimierung, Wien/New York 1972.

Burkard, R.E. (Ganzzahlige Optimierung) Ganzzahlige Optimierung, in: Grundlagen des Operations Research, Hrsg. T. Gal, Bd. 2, Berlin u.a. 1987, S. 361-444.

Churchman, C.W.
Ackoff, R.L.
Arnoff, E.L. (OR) Operations Research, 3. Aufl., Wien/München 1966.

Collatz, L.
Wetterling, W. (Optimierungsaufgaben) Optimierungsaufgaben, 2. Aufl., Berlin/Heidelberg/New York 1971.

Dantzig, G.B. (LP und Erweiterungen) Lineare Programmierung und Erweiterungen, Berlin/Heidelberg/New York 1966.

Dinkelbach, W. (Analysen) Sensitivitätsanalysen und parametrische Programmierung, Berlin/Heidelberg/New York 1969.

Dinkelbach, W. (Sensitivitätsanalysen) Sensitivitätsanalysen, in: Handwörterbuch der mathematischen Wirtschaftswissenschaften, Hrsg. J. Beckmann, G. Menges, R. Selten, Bd. 3, Wiesbaden 1979.

Dinkelbach, W.
Lorscheider, U. (Entscheidungsmodelle) Entscheidungsmodelle und lineare Programmierung, München/Wien 1986.

Dürr, W.
Kleibohm, K. (OR) Operations Research, 2. Aufl., München/Wien 1988.

Ellinger, T. (OR) Operations Research, 2. Aufl., Berlin u.a. 1985.

Fischer, G. (Lineare Algebra) Lineare Algebra, 9. Aufl., Braunschweig/Wiesbaden 1989.

Gal, T. (Sensitivitätsanalyse) Betriebliche Entscheidungsprobleme, Sensitivitätsanalyse und parametrische Programmierung, Berlin/New York 1973.

Gal, T. (Shadow Prices) Shadow Prices and Sensitivity Analysis in Linear Programming under Degeneracy, in: OR Spektrum, Bd. 8 (1986), S. 59-71.

Gal, T. (LO) Lineare Optimierung, in: Grundlagen des Operations Research, Hrsg. T. Gal, Bd. 1, Berlin u.a. 1987, S. 56-254.

Gal, T.
Gehring, H. (Planungstechniken) Betriebswirtschaftliche Planungs- und Entscheidungstechniken, Berlin/New York 1981.

Garvin, W.W. (Introduction) Introduction to Linear Programming, New York/Toronto/London 1960.

Gass, S.I. (LP) Linear Programming, 3. Aufl., New York u.a. 1969.

Grob, H.L.
Reepmeyer, J.-A. (EDV) Einführung in die EDV, 3. Aufl., München 1990.

Hadley, G. (LP) Linear Programming, Reading/London 1962.

Haupt, P.
Wegener, H. (Inhalt) Wirtschaftlicher Inhalt eines ausgewählten Optimierungsverfahrens, in: WiSt, 2. Jg. (1973), S. 8-14.

Herschel, R. (Pascal) Turbo Pascal 4.0/5.0, München/Wien 1989.

Hillier, F.S.
Lieberman, G.J. (OR) Operations Research, 4. Aufl., München/Wien 1988.

Hu, T.C. (Ganzzahlige Programmierung) Ganzzahlige Programmierung und Netzwerkflüsse, München/Wien 1972.

Joksch, H.C. (LP) Lineares Programmieren, 2. Aufl., Tübingen 1965.

Kern, W. (Empfindlichkeit) Die Empfindlichkeit linear geplanter Programme, in: Betriebsführung und Operations Research, Hrsg. A. Angermann, Frankfurt am Main 1963, S. 49-79.

Krekó, B. (Lehrbuch) Lehrbuch der linearen Optimierung, 6. Aufl., Berlin 1973.

Künzi, H.P., Tzschach, H.G., Zehnder, C.A. (Mathematische Optimierung) Numerische Methoden der mathematischen Optimierung, Stuttgart 1967.

Müller-Merbach, H. (Round-Off Errors) On Round-Off Errors in Linear Programming, Lecture Notes in Operations Research and Mathematical Systems, Heft 37, Berlin 1970.

Müller-Merbach, H. (OR) Operations Research, 3. Aufl., München 1973.

Neumann, K. (OR) Operations Research Verfahren, Bd. 1, München/Wien 1975.

Panne, C. van de (LP) Linear Programming and Related Techniques, 2. Aufl., Amsterdam/New York/Oxford 1976.

Ravi, N., Wendell, R.E. (Tolerance) The Tolerance Approach to Sensitivity Analysis of Matrix Coefficients in Linear Programming, in: Management Science, Bd. 35 (1989), S. 1106-1119.

Runzheimer, B. (OR) Operations Research I, 3. Aufl., Wiesbaden 1986.

Sandor, P.E. (Ranging) Some Problems of Ranging in Linear Programming, in: Journal of the Canadian Operational Research Society, Bd. 2 (1964), S. 26-31.

Scheer, A.-W. (EDV-orientierte BWL) EDV-orientierte Betriebswirtschaftslehre, 4. Aufl., Berlin u.a. 1990.

Schieb, J. (DOS) Das große Buch zu MS-DOS/PC-DOS bis einschließlich Version 3.3, 8. Aufl., Düsseldorf 1989.

Schmitz, P., Schönlein, A. (LO-Modelle) Lineare und linearisierbare Optimierungsmodelle sowie ihre ADV-gestützte Lösung, Braunschweig 1978.

Shetty, C.M. (Analyses) On Analyses of the Solution to a Linear Programming Problem, in: Operational Research Quarterly, Bd. 12 (1961), S. 89-104.

Stahlknecht, P., Ohmann, R. (LP auf dem PC) Lineare Programmierung auf dem PC, München/Wien 1987.

Steiner, F.-J. Czerwinski, M.	(Lexikon) Turbo-Pascal-Lexikon, Haar bei München 1989.
Stepan, A. Fischer, E.O.	(Optimierung) Betriebswirtschaftliche Optimierung, 2. Aufl., München/Wien 1989.
Vajda, S.	(Mathematical Programming) Mathematical Programming, Reading/London 1961.
Wendell, R.E.	(Sensitivity Analysis) The Tolerance Approach to Sensitivity Analysis in Linear Programming, in: Management Science, Bd. 31 (1985), S. 564-578.
Witte, T. Deppe, J.-F. Born, A.	(LP) Lineare Programmierung, Wiesbaden 1975.
Wöhe, G.	(Allgemeine BWL) Einführung in die Allgemeine Betriebswirtschaftslehre, 16. Aufl., München 1986.
Zimmermann, H.J. Gal, T.	(Redundanz) Redundanz und ihre Bedeutung für betriebliche Optimierungsentscheidungen, in: ZfB, 45. Jg. (1975), S. 221-236.
Zimmermann, W.	(OR) Operations Research, 4. Aufl., München/Wien 1989.
Zwehl, W. von	(Ganzzahlige Programmierung) Programmierung, ganzzahlige, in: Handwörterbuch der Wirtschaftswissenschaft, Hrsg. W. Albers u.a., Bd. 6, Stuttgart u.a. 1981, S. 349-360.
Zwehl, W. von	(LP) Programmierung, lineare, in: Handwörterbuch der Wirtschaftswissenschaft, Hrsg. W. Albers u.a., Bd. 6, Stuttgart u.a. 1981, S. 360-369.

Veröffentlichungen des Lehrstuhls für Betriebswirtschaftslehre, insbesondere Unternehmensgründung und Unternehmensnachfolge

Hrsg.: Professor Dr. habil. Thomas Hering, Fern-Universität Hagen

Nr. 1 Hering, Th.: Der Entscheidungswert bei der Fusion, 2002.

Nr. 2 Hering, Th.: Fünf Jahre Lehrstuhl für Betriebswirtschaftslehre, insbesondere Unternehmensgründung und Unternehmensnachfolge – Bericht über den Stiftungszeitraum, 2005.

Nr. 3 Hering, Th.: Leitfaden für die Erstellung von Seminar-, Bakkalaureus-, Magister- und Diplomarbeiten, 2007.

Nr. 4 Hering, Th.: Optimale Seminarthemenvergabe als klassisches Transportproblem der Unternehmensforschung, 2009.

Nr. 5 Vincenti, A.J.F.: Einheits- oder klassische Sozialversicherung, 2009.

Nr. 6 Hering, Th., Toll, Ch.: Zur Fusionsgrenzquote bei Vermögensmaximierung, 2009.

Nr. 7 Hering, Th.: Zur Bestimmung der Grenzeinlagequote eines Investors, 2009.

Nr. 8 Hering, Th., Schneider, J., Ostmeyer, J.: Die approximative Dekomposition als heuristische Vorgehensweise zur Investitionsrechnung divisionalisierter Unternehmen auf dem unvollkommenen Kapitalmarkt unter Unsicherheit – Eine beispielhafte Darstellung, 2010.

Nr. 9 Hering, Th., Schneider, J., Toll, Ch.: Die simultane Investitions- und Finanzierungsplanungssimulation als heuristische Vorgehensweise auf dem unvollkommenen Kapitalmarkt unter Unsicherheit, 2010.

Nr. 10 Hering, Th.: Zehn Jahre Lehrstuhl für Betriebswirtschaftslehre, insbesondere Unternehmensgründung und Unternehmensnachfolge, 2010.

Nr. 11 Hering, Th., Schneider, J., Toll, Ch.: Zur Unternehmensbewertung im Rahmen der Konfliktsituation vom Typ Kauf auf dem unvollkommenen Kapitalmarkt unter Unsicherheit, 2011.

20 Mark aus der deutschen Gründerzeit

Veröffentlichungen des Lehrstuhls für Betriebswirtschaftslehre, insbesondere Investitionstheorie und Unternehmensbewertung

Hrsg.: Professor Dr. habil. Thomas Hering, Fern-Universität Hagen

Nr. 12 Hering, Th.: Fünfzehn Jahre Lehrstuhl für Betriebswirtschaftslehre, insbesondere Investitionstheorie und Unternehmensbewertung, 2015.

Nr. 13 Hering, Th.: Leitfaden für die Erstellung von Seminar-, Bakkalaureus-, Magister- und Diplomarbeiten, 2017.

Nr. 14 Hering, Th.: Grundbegriffe des Rechnungswesens und der Kostenrechnung, 2018.

Nr. 15 Hering, Th.: Lineare Optimierung und Sensitivitätsanalyse mit LinSen, 2018.

Nr. 16 Hering, Th.: Quelltext von LinSen 1.5 (1994), 2018.

1 Mark aus der Glanzzeit der deutschen Wissenschaft