

OR Software

LINGO

Rezensenten: K. Haase, R. Kolisch, Kiel

Programm: LINGO INDUSTRIAL/PC

Version: 3.0

Im Lieferumfang enthalten: Modelle aus den Bereichen Investition und Finanzierung, Produktionswirtschaft, Statistik, Marketing und Operations Research

Hardware: PC mit 386 Prozessor und 8 MB

Betriebssystem: DOS / WINDOWS

Disketten: Zwei 3.5" Disketten für DOS und eine Diskette für WINDOWS

Dokumentation: Handbuch mit ausführlicher Einführung und vielen Anwendungsbeispielen

Import/Export: ASCII-, MPS-, LINDO-, WKS-, WK1- und XLS-Format

Bezugsquelle: ADDITIVE Soft- und Hardware für Technik und Wissenschaft GmbH, Rohrwiesenstr. 2, 61381 Friedrichsdorf/Ts., Tel.: 06172/5905-0, Fax: 06172/77613, Email: info@additive-net.de, Homepage: <http://www.additive-net.de>

1 Kurzbeschreibung

LINGO (Language for Interactive General Optimization) ist eine algebraische Modellierungssprache, mit der lineare und nichtlineare Optimierungsprobleme formuliert und gelöst werden können. Sowohl die Zielfunktion als auch die Nebenbedingungen können dabei nichtlinear sein; LINGO erkennt automatisch, ob es sich um ein lineares oder nichtlineares Modell handelt. Unterschiedliche Versionen des Programms sind für Problemgrößen (Anzahl der Nebenbedingungen \times Anzahl der Variablen) von 200×400 bis 32.000

× 100.000, verschiedene Hardwareplattformen (PC, Macintosh, Workstation) und Betriebssysteme (DOS, WINDOWS, UNIX) erhältlich. Das von uns getestete LINGO INDUSTRIAL/PC wird im Paket als DOS- und als WINDOWS-Version ausgeliefert und kann Probleme bis zu 8.000 Nebenbedingungen und 16.000 Variablen bearbeiten. Wir gehen in den folgenden Ausführungen in erster Linie auf die in der DOS-Version vorhandene (Kern-)Funktionalität des Programms ein. Die WINDOWS-Version zeichnet sich darüber hinaus durch eine benutzerfreundliche Oberfläche aus; auf diese weisen wir in den entsprechenden Abschnitten hin.

2 Installation

Das Programm und die Anwendungsbeispiele werden in der DOS-Version automatisch durch Eingabe von `A:\INSTALL` installiert; für die WINDOWS-Version ist der Befehl `A:\SETUP` im Windows Program Manager einzugeben. Die gesamte Installation verläuft unproblematisch und fehlerfrei.

3 Bedienung

LINGO wird durch die Eingabe `lingo` gestartet. Das Programm besitzt drei verschiedene Bearbeitungsmodi: Kommando-Modus, Modelleingabe-Modus und Editier-Modus. Nach dem Aufruf befindet sich das Programm im Kommando-Modus, in dem Dateien eingelesen und ausgegeben, Modelle gelöst sowie Einstellungen in der Programmumgebung vorgenommen werden können. Die Eingabe des Befehls `model` wechselt in den Modelleingabe-Modus, der es erlaubt, zeilenweise das Modell einzugeben. Diese Form der Eingabe ist insbesondere für größere Probleme recht mühsam und fehleranfällig. In der DOS-Version läßt sich durch den Befehl `edit` in den Editier-Modus wechseln. Hier steht ein Ganzseiten-Editor zur Bearbeitung der Problemformulierung zur Verfügung. In der WINDOWS-Version wird dem Anwender eine sehr übersichtliche Entwicklungsoberfläche mit einem komfortablen Editor, der auch Such- und Ersetzfunktionen sowie Sprungbefehle beinhaltet, präsentiert.

Im Kommando-Modus stellt LINGO Hilfen zur Verfügung. Die Eingabe des Befehls `com` (command) zeigt alle verfügbaren Befehle an; Informationen zu einem spezifischen Befehl können mit `help 'Name des Befehls'` ange-

fordert werden. Bspw. gibt der Befehl `help model` eine Übersicht der zur Modellierung verfügbaren Operatoren und Funktionen.

4 Anwendungsbeispiel

Um die Arbeitsweise von LINGO darzustellen, verwenden wir das in Abbildung 1 gegebene Beispiel eines Transportproblems mit drei Anbietern und vier Nachfragern.

In den Zeilen 2] bis 6] werden die Parameter und Variablen des Modells definiert. LINGO unterscheidet dabei zunächst nicht zwischen Variablen und Parametern, sondern definiert alle Angaben als Variablen. Erst durch das Gleichsetzen von Variablen mit Konstanten werden diese zu Parametern. Es besteht die Möglichkeit, einfache Variablen sowie (mehrfach) indizierte Variablen zu definieren. Werden, wie im vorliegenden Beispiel, einfach indizierte Variablen verwendet, so ist zunächst ein Bezeichner der Menge, bspw. `ANBIETER` in Zeile 3], voranzustellen. Anschließend folgt die Angabe der in der Menge befindlichen Elemente, hier die Elemente 1 bis 3. Für jedes Element können schließlich ein oder mehrere Attribute definiert werden; so besitzt bspw. jedes Element der Menge `ANBIETER` das Attribut `KAPAZITAET`. Analog zu Zeile 3] folgt in Zeile 4 die Definition der 4-elementigen Menge `NACHFRAGER` mit dem Attribut `BEDARF`. In Zeile 5] wird das sehr hilfreiche Konstrukt der sogenannten „derived sets“ verwendet, um eine zweifach indizierte Variable zu definieren. Die Elemente der Menge `VERBINDUNGEN` sind als das Kreuzprodukt der Mengen `ANBIETER` und `NACHFRAGER` definiert; jedes Element besitzt zwei Attribute: die Transportmenge (`TMENGE`) und die Kosten des Transports einer Mengeneinheit (`KOSTEN`). Die Definition der Variablen muß mit dem Befehl `SETS:` (in Zeile 2]) begonnen und mit dem Befehl `ENDSETS` (in Zeile 6]) abgeschlossen werden.

Die Zeile 9] enthält die Formulierung der Zielfunktion. Dabei wird von dem Operator `@SUM` Gebrauch gemacht. Dieser summiert das Produkt aus `KOSTEN` und `TMENGE` über alle 12 Elemente der Menge `VERBINDUNGEN` auf.

Der erste Nebenbedingungstyp (in den Zeilen 12] und 13]) fordert, daß für jeden Nachfrager die Summe der von den vier Anbietern gelieferten Mengen mindestens der Bedarfshöhe entspricht. Zur kompakten Formulierung wird dabei der `@FOR`-Operator verwendet. Dieser generiert für jedes Element der `NACHFRAGER`-Menge die in der Folgezeile definierte Nebenbedingung. Bei

der eigentlichen Nebenbedingung (in Zeile 13]) wird wiederum von dem @SUM-Operator zur Aufsummierung der an den Nachfrager gelieferten Transportmengen Gebrauch gemacht. Die Erzeugung des zweiten Nebenbedingungstyps (Zeilen 15] und 16]), der für jeden Anbieter garantiert, daß die ausgelieferten Mengen die Kapazität nicht übersteigen, erfolgt analog.

In den Zeilen 19] bis 25] werden die Daten definiert. Beispielsweise betragen die Kosten für den Transport einer Mengeneinheit von Anbieter eins zu Nachfrager vier sieben Geldeinheiten. Der Angabe der Daten ist ein DATA:-Befehl voran und ein ENDDATA-Befehl hinten an zu stellen.

Mit dem **gen**-Befehl (generate) kann aus der algebraischen Formulierung das vollständig ausgeschriebene Optimierungsproblem generiert werden. Abbildung 2 gibt das lineare Optimierungsproblem (LOP) für unser Beispiel wieder. Über die Generierung des Optimierungsmodells hinaus besteht die Möglichkeit, eine MPS- oder LINDO-Formulierung des Modells zu erzeugen sowie abzuspeichern.

Die Lösung des aktuellen Problems erfolgt mit dem Befehl **go**. LINGO generiert zunächst intern die vollständige Problemformulierung und versucht dann alle Variablen zu instanzieren, die entweder bereits im Datenteil mit Konstanten gleichgesetzt wurden oder deren Wert sich aus dem Nebenbedingungssystem ergibt. In unserem Beispiel werden **KAPAZITAET**, **BEDARF** und **KOSTEN** zu Parametern, und es verbleibt nur die Entscheidungsvariable **TMENGE**. Anschließend wird das Problem auf Linearität überprüft, und in Abhängigkeit von der Problemstellung wird ein Verfahren zur Lösung linearer (Simplexverfahren, Branch-and-Bound) bzw. nichtlinearer Probleme aufgerufen. Während der Berechnung informiert eine PopUp-Box über den Zielfunktionswert und das Ausmaß der Unzulässigkeit der derzeit besten Lösung sowie über die Anzahl der bisher durchgeführten (Pivot-)schritte. Abbildung 3 zeigt für das optimale Ergebnis die Instanzierung aller von Null verschiedenen Variablen und Parameter. Der Zielfunktionswert beträgt 161. Ferner sind die Schattenpreise und die „Reduced Cost“ angegeben. Mit dem Befehl **range** kann im Anschluß an die optimale Lösung des Problems eine Sensitivitätsanalyse durchgeführt werden.

```

1] ! Transportproblem mit 3 Anbietern und 4 Nachfragern;
2] SETS:
3] ANBIETER /1..3/ : KAPAZITAET;
4] NACHFRAGER /1..4/ : BEDARF;
5] VERBINDUNGEN (ANBIETER, NACHFRAGER): KOSTEN, TMENGE;
6] ENDSETS
7] !;
8] ! Zielfunktion;
9] MIN = @SUM( VERBINDUNGEN: KOSTEN * TMENGE);
10] !;
11] ! Nebenbedingungen;
12] @FOR( NACHFRAGER (J):
13] @SUM( ANBIETER (I): TMENGE( I, J)) >= BEDARF( J));
14] !;
15] @FOR( ANBIETER( I):
16] @SUM( NACHFRAGER( J): TMENGE( I, J)) <= KAPAZITAET( I));
17] !;
18] ! Daten;
19] DATA:
20] KAPAZITAET = 30, 25, 21;
21] BEDARF = 15, 17, 22, 12;
22] KOSTEN = 6, 2, 6, 7,
23] 4, 9, 5, 3,
24] 8, 8, 1, 5;
25] ENDDATA

```

Abbildung 1: LINGO-Formulierung des Transportproblems

```

MIN 5 TMENGE(3,4) + TMENGE(3,3) + 8 TMENGE(3,2) + 8 TMENGE(3,1)
+ 3 TMENGE(2,4) + 5 TMENGE(2,3) + 9 TMENGE(2,2) + 4 TMENGE(2,1)
+ 7 TMENGE(1,4) + 6 TMENGE(1,3) + 2 TMENGE(1,2) + 6 TMENGE(1,1)
SUBJECT TO
2] TMENGE(3,1) + TMENGE(2,1) + TMENGE(1,1) >= 15
3] TMENGE(3,2) + TMENGE(2,2) + TMENGE(1,2) >= 17
4] TMENGE(3,3) + TMENGE(2,3) + TMENGE(1,3) >= 22
5] TMENGE(3,4) + TMENGE(2,4) + TMENGE(1,4) >= 12
6] TMENGE(1,4) + TMENGE(1,3) + TMENGE(1,2) + TMENGE(1,1) <= 30
7] TMENGE(2,4) + TMENGE(2,3) + TMENGE(2,2) + TMENGE(2,1) <= 25
8] TMENGE(3,4) + TMENGE(3,3) + TMENGE(3,2) + TMENGE(3,1) <= 21
END

```

Abbildung 2: Lineares Programm des Transportproblems

5 Weitere Möglichkeiten

Das dargestellte Beispiel zeigt nur einen kleinen Teil der mit LINGO gebotenen Möglichkeiten. Weitere Optionen sind folgende:

- Parameter können im Datenteil mit einem ? instanziiert werden. LINGO fragt dann nach Start des Lösungsverfahrens automatisch nach den aktuellen Werten.
- Die @-Operatoren können mit einer oder mehreren Bedingungen versehen werden, so daß die Summe über eine Teilmenge gebildet wird.
- Dünn besetzte Matrizen können so definiert werden, daß nur die von Null verschiedenen Eintragungen gespeichert werden.
- Teile der Modellformulierung können mit einem @FILE-Operator aus separaten Dateien eingebunden werden. Damit eröffnet sich die Möglichkeit, den Modellteil mit verschiedenen a priori definierten Daten zu lösen.
- Mit der @IMPORT-Funktion lassen sich Daten aus Tabellenkalkulationsprogrammen laden.
- LINGO läßt sich in Batch-Programme einbinden oder als Subroutine in C-Programmen aufrufen.

VARIABLE	VALUE	REDUCED COST
KAPAZITAET(1)	30.00000	0.000000E+00
KAPAZITAET(2)	25.00000	0.000000E+00
KAPAZITAET(3)	21.00000	0.000000E+00
BEDARF(1)	15.00000	0.000000E+00
BEDARF(2)	17.00000	0.000000E+00
BEDARF(3)	22.00000	0.000000E+00
BEDARF(4)	12.00000	0.000000E+00
KOSTEN(1, 1)	6.000000	0.000000E+00
KOSTEN(1, 2)	2.000000	0.000000E+00
KOSTEN(1, 3)	6.000000	0.000000E+00
KOSTEN(1, 4)	7.000000	0.000000E+00
KOSTEN(2, 1)	4.000000	0.000000E+00
KOSTEN(2, 2)	9.000000	0.000000E+00
KOSTEN(2, 3)	5.000000	0.000000E+00
KOSTEN(2, 4)	3.000000	0.000000E+00
KOSTEN(3, 1)	8.000000	0.000000E+00
KOSTEN(3, 2)	8.000000	0.000000E+00
KOSTEN(3, 3)	1.000000	0.000000E+00
KOSTEN(3, 4)	5.000000	0.000000E+00
TMENGE(1, 1)	2.000000	0.000000E+00
TMENGE(1, 2)	17.00000	0.000000E+00
TMENGE(1, 3)	1.000000	0.000000E+00
TMENGE(2, 1)	13.00000	0.000000E+00
TMENGE(2, 4)	12.00000	0.000000E+00
TMENGE(3, 3)	21.00000	0.000000E+00
ROW	SLACK OR SURPLUS	DUAL PRICE
1	161.0000	1.000000
2	0.000000E+00	-6.000000
3	0.000000E+00	-2.000000
4	0.000000E+00	-6.000000
5	0.000000E+00	-5.000000
7	0.000000E+00	2.000000
8	0.000000E+00	5.000000

Abbildung 3: Ausgabe des optimalen Ergebnis für das Transportproblem

6 Stärken und Schwächen des Programms

Folgende Stärken und Schwächen des Programms sind zu konstatieren:

Stärken

- LINGO ist sehr leicht zu erlernen. Erfahrungen bei Seminar- und Diplomarbeiten haben gezeigt, daß Studenten ohne Vorkenntnisse in (algebraischen) Programmiersprachen LINGO innerhalb weniger Stunden mit Erfolg anwenden konnten.
- Im Rahmen der Forschung eignet sich LINGO zur schnellen Validierung von Modellen und zur Bestimmung von Referenzergebnissen. Die Möglichkeit, unkompliziert einzelne Nebenbedingungen hinzuzufügen und ausblenden zu können, macht LINGO zu einem hilfreichen Instrument bei der Generierung von Schranken und Schnittebenen.
- Mit Hilfe der Funktion @RAND lassen sich gleichverteilte Zufallszahlen generieren. Dadurch kann LINGO als flexibler Datengenerator eingesetzt werden, der zudem die Möglichkeit bietet, die erzeugten Problem instanzen in unterschiedlichen Formaten abzuspeichern.

Schwächen

- Der in der DOS-Version integrierte Editor ist nicht sehr bedienungsfreundlich. Bspw. fehlt die Möglichkeit, bestimmte Zeichenstrings zu suchen oder zu ersetzen sowie in eine bestimmte Zeile zu springen.
- Die Fehlermeldungen erleichtern nur bedingt die Suche nach syntaktischen Fehlern. Bei der Modellierung empfiehlt es sich daher, die Restriktionen sukzessive einzugeben und zu überprüfen.
- In der DOS-Version können Daten sehr leicht verloren gehen, weil LINGO sowohl beim Einlesen eines neuen Modells als auch bei Beendigung des Programms das aktuell im Arbeitsspeicher befindliche Modell ohne vorherigen Hinweis überschreibt. Ebenso werden beim Abspeichern von Modellen bereits existierende Dateien ohne Vorwarnung überschrieben.
- Die Verwendung von Indizes ≤ 0 ist nicht möglich.
- Eine parametrische Sensitivitätsanalyse ist nicht durchführbar.

7 Resümee

LINGO ist ein ausgereiftes und leicht bedienbares Programm zur Modellierung und Lösung linearer und nichtlinearer Optimierungsprobleme. Die Robustheit des Programms, die schnell erlernbare Modellierungssprache und das didaktisch hervorragend aufgebaute Handbuch machen LINGO zu einem wertvollen Hilfsmittel in der quantitativ orientierten Betriebswirtschaftslehre. Während die (Kern-)Funktionalität in der DOS- und in der WINDOWS-Version identisch ist, sei dem Anwender jedoch die WINDOWS-Version empfohlen, da diese eine sichere Handhabung von Dateien erlaubt und zudem eine Entwicklungsoberfläche bietet.

Dr. Knut Haase, Dr. Rainer Kolisch
Christian-Albrechts-Universität zu Kiel
Institut für Betriebswirtschaftslehre
Lehrstuhl für Produktion und Logistik
Olshausenstr. 40, D-24098 Kiel
Tel.: 0431/880-4581, Fax: 0431/880-1531, e-mail: kolisch@bwl.uni-kiel.de