
6. Softwarewerkzeuge für die Lineare Programmierung

- GNU Linear Programming Kit

GNU Linear Programming Kit

In erster Linie eine C-Bibliothek für die lineare Programmierung (LP) und gemischt-ganzzahlige Programmierung (MIP).

- Primaler und dualer Simplex-Algorithmus (LP)
- Primaler und dualer Innere-Punkte-Algorithmus (LP)
- Branch-and-Bound-Algorithmus (MIP, kombinatorische Optimierung)
- C API
- Modellierungssprache
- Stand-alone LP- bzw. MIP-Solver

Problemformen

GLPK kann LPs der folgenden Form verarbeiten:

Maximiere oder minimiere

$$z = c_0 + c_1x_1 + \cdots + c_nx_n$$

unter linearen Nebenbedingungen

$$\begin{array}{ccccccc} a_{11}x_1 & + & \cdots & + & a_{1n}x_n & \theta & b_1 \\ \vdots & & & & \vdots & \vdots & \vdots \\ a_{m1}x_1 & + & \cdots & + & a_{mn}x_n & \theta & b_m \end{array}$$

mit $\theta \in \{\leq, =, \geq\}$ und **Variablenbeschränkungen**

$$\begin{array}{ccccccc} l_1 & \leq & x_1 & \leq & u_1 \\ l_2 & \leq & x_2 & \leq & u_2 \\ \vdots & & \vdots & & \vdots \\ l_n & \leq & x_n & \leq & u_n \end{array}$$

Hierbei ist $l_i = -\infty$ und $u_i = \infty$ möglich.

Für **MIPs** oder **kombinatorische Probleme** können die Variablen x_i zusätzlich auf $x_i \in \mathbb{Z}$ oder $x_i \in \{0, 1\}$ eingeschränkt werden.

Eingabeformate

- Ein einfachsten ist es, ein LP in einem der beiden möglichen textuellen Eingabeformate zu formulieren und
- dieses mit dem Stand-alone Solver zu lösen.
- Mögliche Eingabeformate:
 - MPS-Format (IBM, 60er Jahre)
 - CPLEX-Format (CPLEX Optimization Inc., 80er Jahre)
- Wir nutzen das **CPLEX-Format!**

CPLEX-Format

- Plain Text
- zeilenorientiert
- formatfrei, d.h. Zwischenraum ist nicht relevant
- Syntaxelemente (Tokens):
 - Schlüsselwörter
 - Identifier
 - numerische Konstanten und Operatorsymbole
 - Trennzeichen

CPLEX-Format: Aufbau einer LP-Datei

1. Zielfunktion
2. Nebenbedingungen
3. Variablenbeschränkungen
4. Zuordnung der Variablen zu den Wertebereichen \mathbb{R} , \mathbb{Z} und $\{0, 1\}$
5. Schlüsselwort end

CPLEX-Format: Beschreibung der Zielfunktion

$$\left\{ \begin{array}{l} \text{maximize} \\ \text{minimize} \end{array} \right\} [\textit{identifizier} :] \textit{expression}$$

- *identifizier* ist eine optionale Bezeichnung für den Zielfunktionswert; folgt den übliche syntaktischen Regeln für Bezeichner.
- Standardbezeichner für Zielfunktionswert: obj
- *expression*:

$$s c x s c x \dots s c x$$

- s = Vorzeichen, c = numerische Konstante als Koeffizient, x = Bezeichner für eine Variable

- Das erste Vorzeichen (Standard: +) und die Koeffizienten (Standard: 1.0) sind optional.
- Koeffizienten dürfen in den für Programmiersprachen üblichen Notationen geschrieben werden.
- Variablen müssen nicht deklariert werden.

Beispiel:

Minimize Z : - x1 + 2 x2 - 3.5 x3 + 4.997e3x4 + x5 + x6 +
x7 - .01x8

CPLEX-Format: Nebenbedingungen

subject to
*nebenbedingung*₁
...
*nebenbedingung*_{*m*}

Jede *nebenbedingung*_{*i*} in der Form:

$$[r :] \textit{expression} \left\{ \begin{array}{l} \leq \\ \geq \\ = \end{array} \right\} b$$

- *r* ist optionaler Bezeichner für die Nebenbedingung; Standard: *r.iii*
- *b* ist numerische Konstante mit optionalem Vorzeichen

Beispiel für Produktionsproblem (siehe Beispiel 4.3):

maschine1: 40 a + 24 b ≤ 480

maschine2: 24 a + 48 b ≤ 480

maschine3: 60 b ≤ 480

CPLEX-Format: Variablenbeschränkungen

```
bounds  
bound1  
...  
boundp
```

- Dieser Abschnitt ist optional.

Standard: Für alle definierten Variablen x_i gilt $x_i \geq 0$.

- Auch wenn Abschnitt vorhanden, muss nicht für jede Variable eine Beschränkung $bound_j$ definiert werden.

Standard: Für alle Variablen x_i , für die keine Beschränkung definiert ist, gilt $x_i \geq 0$.

- Jede Definition einer Beschränkung $bound_j$ muss auf einer neuen Zeile erfolgen.

Die Beschränkungen $bound_j$ können wie folgt aufgebaut sein:

$x \geq l$	untere Schranke l für Variable x
$l \leq x$	untere Schranke l für Variable x
$x \leq u$	obere Schranke u für Variable x
$l \leq x \leq u$	untere Schranke l und obere Schranke u für Variable x
$x = t$	fixer Wert t für Variable x
x free	Variable x ist unbeschränkt

- Schlüsselwörter `inf` bzw. `infinity` stehen für ∞ .
- `-inf` entspricht somit $-\infty$.

CPLEX-Format: Wertebereiche von Variablen

- optional; Standard: für alle Variablen $x_i : x_i \in \mathbb{R}$
- notwendig für ganzzahlige oder kombinatorische Probleme
- Alle ganzzahligen bzw. kombinatorischen Variablen werden jeweils in einem eigenen Abschnitt aufgelistet.
- In jeder Zeile nur eine Variable!

ganzzahle Variablen ($x_i \in \mathbb{Z}$):

Integer

x_1

...

x_q

binäre Variablen ($x_i \in \{0, 1\}$):

Binary

x_1

...

x_q

CPLEX-Format: Einfaches LP-Beispiel

Beispiel 6.1. Produktionsplanung von Beispiel 4.3:

```
Maximize
    10 a + 40 b
subject to
    maschine1: 40 a + 24 b <= 480
    maschine2: 24 a + 48 b <= 480
    maschine3:          60 b <= 480
end
```

CPLEX-Format: IP-Beispiel

Beispiel 6.2. Verschnittproblem von Beispiel 1.6:

Minimize

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9$$

subject to

$$3 x_1 + x_2 + 2 x_5 + x_6 + x_8 \geq 10$$

$$x_2 + x_4 \geq 45$$

$$2 x_3 + x_8 + x_9 \geq 21$$

$$x_4 + x_5 + 2 x_6 + 3 x_7 + x_9 \geq 42$$

integer

x1

x2

x3

x4

x5

```
x6  
x7  
x8  
x9  
end
```

Stand-alone Solver

Kommando:

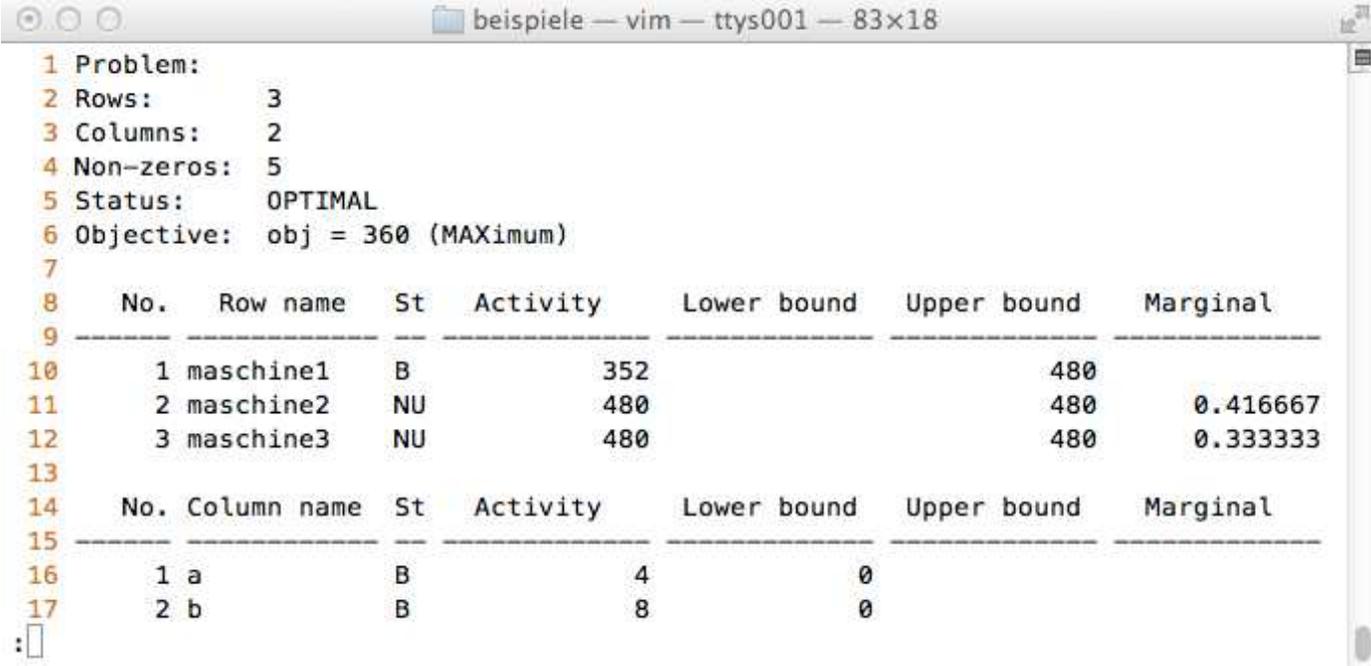
```
glpsol [options...] filename
```

- Löst LP aus *filename*
- Für LP/MIP im CPLEX-Format: Option `--lp`
- Ausgabe der optimalen Lösung in Datei: Option `-o filename`
- In Ausgabedatei: optimale Basislösung

Aufruf:

```
glpsol --lp production.lp -o production.sol
```

Inhalt von production.sol:



```
beispiele — vim — ttys001 — 83x18
1 Problem:
2 Rows:      3
3 Columns:   2
4 Non-zeros: 5
5 Status:    OPTIMAL
6 Objective: obj = 360 (MAXimum)
7
8  No.  Row name  St  Activity  Lower bound  Upper bound  Marginal
9 -----
10  1  maschine1  B   352      480          480
11  2  maschine2  NU  480      480          480  0.416667
12  3  maschine3  NU  480      480          480  0.333333
13
14  No.  Column name  St  Activity  Lower bound  Upper bound  Marginal
15 -----
16  1  a           B   4         0
17  2  b           B   8         0
18
```