

Modellierung und Optimierung mit OPL

2 Einführung in OPL

Andreas Popp



- ▶ seit 1988 vertriebener Solver (Software in die Lösungsverfahren implementiert sind) für lineare Optimierungsmodelle
- ▶ zuerst von CPLEX Optimization Inc., dann ILOG, dann IBM verkauft
- ▶ starke Verbreitung in Wissenschaft und Industrie
- ▶ Schnittstellen für bekannte Programmiersprachen wie C++, Java oder C#
- ▶ eigene Modellierungssprache: OPL (Optimization Programming Language)
- ▶ im Rahmen der IBM Academic Initiative kostenfrei für akademische Anwendungen

2.1 Aufbau eines OPL-Projekts

Modelldateien Beschreibung des allgemeinen
Optimierungsmodells (Endung: .mod)

Datendateien Daten zur Instanziierung eines OPL-Modells
(Endung: .dat)

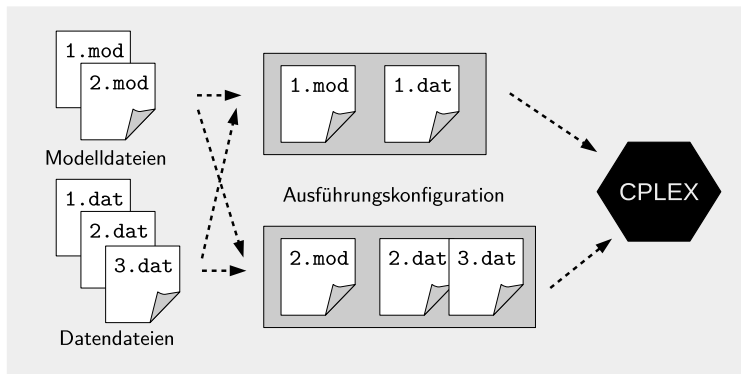
Einstellungsdateien Einstellungen für den Solver (Endung:
.ops)

Modelldateien Beschreibung des allgemeinen
Optimierungsmodells (Endung: .mod)

Datendateien Daten zur Instanziierung eines OPL-Modells
(Endung: .dat)

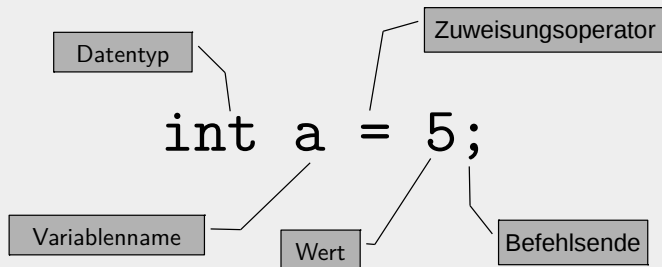
Einstellungsdateien Einstellungen für den Solver (Endung:
.ops)

Aufbau eines OPL-Projekts



2.2 Grundlegende Datentypen und Operatoren

Aufbau einer einfachen Zuweisungsanweisung



`int` (kurz für: „Integer“); ein ganzzahliger Wert mit beliebigem Vorzeichen. Beispielliterale: `0`, `1`, `-2`, `-786`

`float` Gleitkommazahl mit beliebigem Vorzeichen. Beispielliterale: `0.0`, `1.0`, `3.14`, `-7.86`

`boolean` eigentlich ein logischer Wahrheitswert; bei Entscheidungsvariablen eine 0-1-Variable.

`string` eine Zeichenkette. Beispielliterale: `"1"`, `"B"`, `"Berlin"`

Set eine geordnete Menge von Elemente von (u.a)
primitiven Datentypen, z.B.

```
{string} Standorte =  
    {"Ansbach", "Berlin", "Cottbus"};
```

Array ein über ein Set indiziertes Tupel von (u.a.) primitiven Datentypen, Sets oder anderen Arrays, z.B.

```
float Fixkosten[Standorte] =  
    [27.4, 58.3, 30.0];
```

Zugriff mittels Index, z.B.:

```
Fixkosten["Cottbus"] → 30.0
```

11/1 ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ↺ 🔍 ↻

- ▶ Zuweisungsoperator =
- ▶ Arithmetische Operatoren
 - + Addition
 - Subtraktion
 - * Multiplikation
 - / Division (selten in linearen Modellen)
- ▶ Vergleichsoperatoren (für lineare Modelle)
 - == gleich
 - <= kleiner-gleich
 - >= größer-gleich

- $$\forall i \in I \rightarrow \text{forall}(i \text{ in } I)$$

2.3 Mathematische Modelle in OPL-Syntax

Mathematisches Modell

Parameter:

p_i Preis von Produkt $i \in I$

c_r Kapazität von Ressource $r \in R$

v_{ri} Kapazitätsverbrauch von Produkt $i \in I$ auf
Ressource $r \in R$

Datendatei

```
//Parameter  
p = [2.9, 3.3, 2.2];  
c = [64.0, 48.0];  
v = [  
    [5.3, 2.9, 2.5],  
    [3.9, 4.8, 3.1]  
];
```


Mathematisches Modell

$$\max \sum_{i \in I} p_i \cdot x_i$$

Modelldatei

```
//Zielfunktion  
maximize sum(i in I)(p[i]*x[i]);
```


Beispiel: Produktionsproblem.mod

```

1 //Indexmengen
2 {string} I = ...; //Produkte
3 {string} R = ...; //Ressourcen
4
5 //Parameter
6 float p[I] = ...; //Preis
7 float c[R] = ...; //Kapazität
8 float v[R][I] = ...; //Kapazitätsverbrauch
9
10 //Entscheidungsvariablen
11 dvar float+ x[I]; //Produktionsmenge
12
13 //Zielfunktion
14 maximize sum(i in I)(p[i] * x[i]);
15
16 //Nebenbedingungen
17 subject to{
18
19     //Kapazitätsrestriktion
20     forall(r in R)
21         sum(i in I)(v[r][i]*x[i]) <= c[r];
22
23 }

```

```
1 //Indextmengen
2 I = {"Produkt_1", "Produkt_2", "Produkt_3"};
3 R = {"Maschine_A", "Maschine_B"};
4
5 //Parameter
6 p = [2.9, 3.3, 2.2];
7 c = [64.0, 48.0];
8 v = [
9     [5.3, 2.9, 2.5],
10    [3.9, 4.8, 3.1]
11 ];
```

• • •

← Optimalwert

• • •

← Optimallösung

2.4 Die CPLEX Studio IDE

Die CPLEX Studio IDE

2 Einführung in
OPL

CC-BY-SA
A. Popp

The screenshot displays the IBM ILOG CPLEX Optimization Studio IDE. The main window shows the 'Produktionsproblem.mod' file with the following code:

```
1 //Indexmengen
2 {string} I = ...; //Produkte
3 {string} R = ...; //Ressourcen
4
5 //Parameter
6 float p[I] = ...; //Preis
7 float c[R] = ...; //Kapazität
8 float v[R][I] = ...; //Kapazitätsverbrauch
9
10 //Entscheidungsvariablen
11 dvar float+ x[I]; //Produktionsmenge
12
13 //Zielfunktion
14 maximize sum(i in I)(p[i] * x[i]);
15
16 //Nebenbedingungen
17 subject to
18
19 //Kapazitätsrestriktion
20 forall(r in R)
21   sum(i in I)(v[r][i]*x[i]) <= c[r];
22
23 }
```

The left sidebar shows the 'OPL-Projekte' tree with 'Produktionsproblem (Beispiel 1.1)' selected. Below it, the 'Problem' tab shows the 'Lösung mit Zielsatzung 35,617' and a table of variable values:

Name	Wert
Externe Daten (5)	
c	[64 48]
p	("Produkt_1" "Produkt_2" "Prod
v	[2.9 3.3 2.2]
R	("Maschine_A" "Maschine_B")
v	[5.3 2.9 2.5] [3.9 4.8 3.1]

The right sidebar shows the 'Gliederung' (Outline) view with the following structure:

- unter Verwendung von CPLEX
- Externe Daten (5)
 - c: float[R]
 - I: (string)
 - p: float[I]
 - R: (string)
 - v: float[R][I]
- Entscheidungsvariablen (1)
 - x: dvar float+[I]
- Zielfunktion: einfach
- Nebenbedingungen (1)

The bottom status bar indicates '1 Elemente ausgewählt' and a timer at '00:00:00:45'.

Die CPLEX Studio IDE

2 Einführung in
OPL

CC-BY-SA
A. Popp

IBM ILOG CPLEX Optimization Studio

Projektbrowser: Übersicht über die Inhalte des aktuellen Projekts

```
//Indexmengen
2 {string} I = ...; //Produkte
3 {string} R = ...; //Ressourcen
4
5 //Parameter
6
17
18
19 //Kapazitätsrestriktion
20 forall(r in R)
21   sum(i in I) (v[r][i]*x[i]) <= c[r];
22
23 }
```

Name	Wert
Lösung mit Zielsetzung 35,617	
Externe Daten (5)	
c	[64 48]
rI	("Produkt_1" "Produkt_2" "Prod
p	[2.9 3.3 2.2]
rR	("Maschine_A" "Maschine_B")
v	[[5.3 2.9 2.5] [3.9 4.8 3.1]]

0 Elemente

00:00:00:45

Die CPLEX Studio IDE

2 Einführung in
OPL

CC-BY-SA
A. Popp

The screenshot shows the IBM ILOG CPLEX Optimization Studio IDE interface. The main window displays a CPLEX model file named `Produktionsproblem.mod`. The model code includes comments for products, resources, and parameters, and defines decision variables `x[i]` and `v[r]`. The objective function is to maximize the sum of `x[i]` weighted by `p[i]`. Constraints include capacity restrictions for resources `r` based on `v[r]` and `x[i]`.

Annotations with red callout boxes identify the following components:

- Projekt**: Points to the `Produktionsproblem (Beispiel 1.1)` project in the OPL-Projekte tree.
- Ausführungskonfiguration**: Points to the `Lewig_Sanstenen (Standard)` execution configuration in the tree.
- Modelldatei**: Points to the `Produktionsproblem.mod` model file.
- Datendatei**: Points to the `Lewig_Sanstenen.dat` data file.

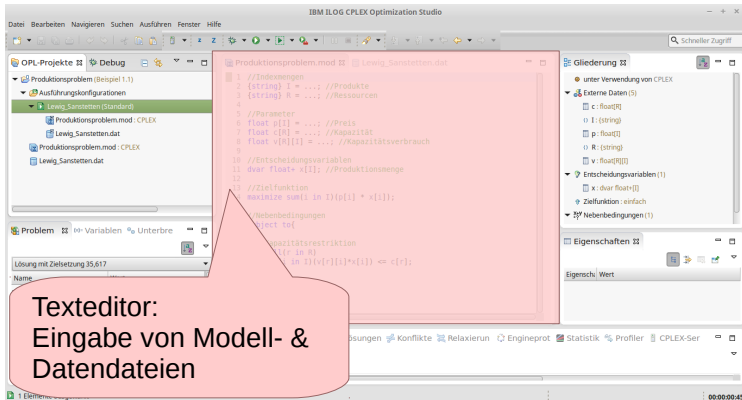
The bottom-left pane shows the solution results for the objective function, with a value of 35,617. The bottom-right pane shows the 'Gliederung' (Structure) view, which lists the model components: External Data (5), Decision Variables (1), Objective Function (1), and Side Conditions (1).

Name	Wert
Externe Daten (5)	
c	[64 48]
r	["Produkt_1" "Produkt_2" "Produkt_3"]
p	[2.9 3.3 2.2]
R	["Maschine_A" "Maschine_B"]
v	[[5.3 2.9 2.5] [3.9 4.8 3.1]]

Die CPLEX Studio IDE

2 Einführung in
OPL

CC-BY-SA
A. Popp



The screenshot displays the IBM ILOG CPLEX Optimization Studio interface. The main window shows the 'Produktionsproblem.mod' file with the following code:

```
1 //Indexmengen
2 {string} I = ...; //Produkte
3 {string} R = ...; //Ressourcen
4
5 //Daten
6
7 //Zielfunktion
8
9 //Nebenbedingungen
10
11 //Variablen
12
13 //Optimalitätsrestriktion
14 forall(r in R)
15     sum(i in I) (v[r][i]*x[i]) <= c[r];
16
17 subject to
18
19
20
21
22
23 }
```

A red callout box with the text "Problembrowser: Hier erscheint nach erfolgreicher Lösung des Problems das Ergebnis" points to the 'Problem' tab in the bottom-left pane. This pane shows the solution results for the 'Produktionsproblem.mod' file, with a total value of 35.617. The results are displayed in a table:

Name	Wert
Externe Daten (5)	
c	[54.48]
I	("Produkt_1" "Produkt_2" "Produkt_3")
p	[2.9 3.3 2.2]
R	("Maschine_A" "Maschine_B")
v	[5.3 2.9 2.5] [3.9 4.8 3.1]

The bottom status bar indicates '1 Elemente ausgewählt' and a timer showing '00:00:00:45'.

Die CPLEX Studio IDE

2 Einführung in
OPL

CC-BY-SA
A. Popp

IBM ILOG CPLEX Optimization Studio

Produktionsproblem.mod

```
1 //Indexmengen
2 {string} I = ...; //Produkte
3 {string} R = ...; //Ressourcen
4
5 //Parameter
6
7 //Kapazitätsrestriktion
8 forall(r in R)
9   sum(i in I)(v[r][i]*x[i]) <= c[r];
10
11 //Zielfunktion
12 maximize sum(i in I)(p[i]*x[i]);
13
14 //Nebenbedingungen
15 subject to
16
17 //Kapazitätsrestriktion
18 forall(r in R)
19   sum(i in I)(v[r][i]*x[i]) <= c[r];
20
21
22
23 }
```

Gliederungsansicht zur leichteren Navigation im Editor.

Gliederung

- unter Verwendung von CPLEX
- Externe Daten (5)
 - c: float[]
 - f: float[]
 - p: float[]
 - R: (string)
 - v: float[][]
- Entscheidungsvariablen (1)
 - x: float[]
- Zielfunktion: einfach
- Nebenbedingungen (1)

Eigenschaften

Eigenschaft: Wert

Problem

Lösung mit Zielsatzung 35,617

Name	Wert
c	[64 48]
f	("Produkt_1" "Produkt_2" "Prod
p	[2.9 3.3 2.2]
R	("Maschine_A" "Maschine_B")
v	[[5.3 2.9 2.5] [3.9 4.8 3.1]]

1 Elemente ausgewählt

0 Elemente

00:00:00:45

2.5 Fehler und Warnungen in OPL

Arten von Fehlern

Einteilung nach Schwere

Fehler Verhindert das erfolgreiche Abschließen eines Lösungslaufs

Warnung Behindert den Lösungslauf nicht, es kann aber zu unerwünschten Ergebnissen kommen.
Manchmal Hinweis auf Fehler im Code.

Einteilung nach Zeitpunkt des Auftretens

Compilerfehler Treten bei Übersetzung des Problems für den Solver auf. Werden von der IDE erkannt.

Laufzeitfehler Treten erst zur Laufzeit des Solvers auf.
Werden nicht von der IDE erkannt, aber nach Lösungslauf angezeigt.

- ▶ `syntax error, unexpected ...` (Compilerfehler)
 - ▶ Compiler versteht die Anweisung nach „unexpected“ an dieser Stelle nicht
 - ▶ fehlender Strichpunkt?
- ▶ `syntax error, unexpected =` (Compilerfehler)
 - ▶ Spezialfall zu oben
 - ▶ meist Verwechslung von Zuweisungsoperator `=` und Vergleichsoperator `==`
- ▶ Der Typ `...` kann nicht für `...` verwendet werden (Compilerfehler)
 - ▶ Datentypen durcheinander gebracht
- ▶ Der Index für den Array `...` liegt außerhalb des gültigen Bereichs (Laufzeitfehler)
 - ▶ ein Array wurde mit einem ungültigen Index angesprochen