

# Einführung in IBM ILOG CPLEX Optimization Studio – Teil 1 (unter Verwendung der GUI)

In diesem Dokument werden Sie lernen, wie Sie ...

- die IDE mit grafischer Oberfläche (GUI) des LP-Solvers IBM ILOG CPLEX auf einem CIP-Pool-Rechner unter Linux öffnen können.
- ein einfaches lineares Programm in der Programmiersprache OPL schreiben.
- lineare Programme im OPL-Format auf einem CIP-Pool-Rechner mithilfe der Software IBM ILOG CPLEX lösen lassen können.

## 1 Überblick

Wir werden Ihnen im Rahmen der Vorlesung mit dem IBM ILOG CPLEX Optimization Studio ein Tool vorstellen, das genutzt werden kann um (ganzahlige) lineare Programme zu lösen.

Dieses Paket enthält die Programmiersprache OPL, mit der auf intuitive Weise lineare Programme formuliert und gelöst werden können. Es ist auch eine Entwicklungsumgebung (IDE) enthalten, die auf diese Sprache zugeschnitten ist und auf Eclipse basiert.

## 2 Alternativen

Der Zugriff auf die in den CIP-Pools installierte Software ist auch per ssh möglich – wie das funktioniert wird in einer separaten Anleitung erklärt. Alternativ können Sie sich auch die kostenlose Demoversion hier <https://www.ibm.com/products/ilog-cplex-optimization-studio> herunterladen. Diese ist für die zu bearbeitenden Übungsaufgaben ausreichend. Orientieren Sie sich am besten am Rest der Anleitung, um das Programm zu starten und zu bedienen.

Die reguläre Version von IBM ILOG CPLEX ist nicht frei verfügbar. Wir haben uns aber trotzdem für sie entschieden, da sie gut funktioniert und sie durch die mitgelieferte IDE einfach zu bedienen ist. Es gibt aber auch freie LP-Solver, die sie ebenfalls zum Lösen von linearen Programmen nutzen können.

### 3 Start des Programms

Die Software ist in allen CIP-Pools im Informatikgebäude unter Linux installiert. Unter Windows ist sie nicht vorinstalliert. Wenn Ihrer CIP-Pool-Rechner unter Windows gestartet ist, fahren Sie diesen herunter, starten Sie ihn neu und achten Sie darauf, dass er unter Linux (aktuell Ubuntu) startet.

Das IBM ILOG CPLEX Optimization Studio können Sie öffnen, indem Sie im Verzeichnis `/opt/ibm/ILOG/CPLEX_Studio1262/opl/oplide/` das Programm `oplide` ausführen. Von der Kommandozeile aus kann die IDE mittels

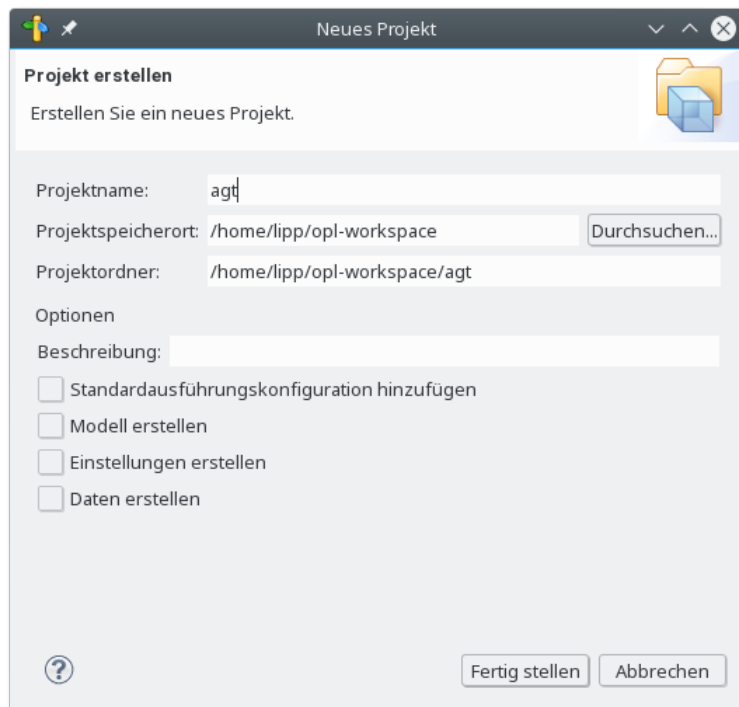
```
/opt/ibm/ILOG/CPLEX_Studio1262/opl/oplide/oplide
```

gestartet werden.

Nachdem das Programm gestartet ist, werden Sie aufgefordert, einen Arbeitsbereich auszuwählen. Geben Sie einen Pfad Ihrer Wahl an oder bestätigen Sie den vorgeschlagenen Pfad.

### 4 Anlegen eines Projekts und Modells

- Legen Sie zunächst mittels Datei → Neu → OPL-Projekt ein neues Projekt an und vergeben Sie einen Namen dafür:



- Ein Modell entspricht der Beschreibung eines linearen Programms. Fügen Sie dem Projekt ein Modell hinzu mittels Datei → Neu → Modell und vergeben Sie einen Dateinamen. Modelldateien enden auf „.mod“.
- Öffnen Sie das neue Modell und fügen Sie den Quellcode für das lineare Programm hinzu.

## 5 Syntax des linearen Programms

In diesem Kapitel beschreiben wir schrittweise den Quellcode für das lineare Programm aus den Vorlesungsfolien, welchen Sie in Ihre „.mod“-Datei schreiben.

Variablen haben die Bezeichnung `dvar` gefolgt vom Typ und Namen der Variablen. Befehle und Zuweisungen werden wie in Java oder C mit Semikolon beendet. Der Typ ist z. B. `float+` für nicht-negative Gleitkommazahlen oder `int+` für nicht-negative Ganzzahlen. Bedenken Sie, dass ganzzahlige lineare Programme i. A. schwieriger zu lösen sind als lineare Programme ohne die Beschränkung auf Ganzzahlen.

Wir legen die Variablen  $x_1$  und  $x_2$  an.

```
dvar float+ x1;  
dvar float+ x2;
```

Als nächstes beschreiben wir unsere Zielfunktion. Wir wollen die Funktion  $30x_1 + 50x_2$  maximieren. Das tun wir mittels:

```
maximize 30 * x1 + 50 * x2;
```

Die Funktion soll unter den folgenden 3 Nebenbedingungen maximiert werden:

$$\begin{aligned}4x_1 + 11x_2 &\leq 880 \\x_1 + x_2 &\leq 150 \\x_2 &\leq 60\end{aligned}$$

Das erreichen wir in OPL, indem wir einen `subject to`-Block ergänzen:

```
subject to {  
    4 * x1 + 11 * x2 <= 880;  
    x1 + x2 <= 150;  
    x2 <= 60;  
}
```

Wie in C oder Java können ein- bzw. mehrzeilige Kommentare mit `//` bzw. `/* */` gesetzt werden.

Insgesamt sieht die Datei also wie folgt aus:

```
//zwei positive nicht-ganzzahlige Variablen  
dvar float+ x1;  
dvar float+ x2;  
  
maximize 30 * x1 + 50 * x2; //Zielfunktion  
subject to {  
    //Nebenbedingungen  
    4 * x1 + 11 * x2 <= 880;  
    x1 + x2 <= 150;  
    x2 <= 60;  
}
```

## 6 Ausführen des Programms

- Zunächst müssen Sie eine Ausführungskonfiguration für das Programm anlegen: Rechter Mausklick auf das Modell → Zur Ausführungskonfiguration hinzufügen → Neue Ausführungskonfiguration. Den Namen können Sie beliebig wählen.
- Die Ausführungskonfigurationen werden oben links im Projektordner angezeigt. Wählen Sie die neue Ausführungskonfiguration mit der rechten Maustaste aus und klicken Sie auf *Ausführen*.
- Im *Problembrowser* (standardmäßig unten links) wird das Ergebnis angezeigt. In diesem Beispiel: Zielwert: 5300,  $x_1$ : 110,  $x_2$ : 40



Name	Wert
—  Entscheidungsvariablen (2)	
.o x1	110
.o x2	40