Lehrstuhl für Informatik I
Algorithmen & Komplexität
Universität Würzburg

Würzburg, den 5. Mai 2020

Prof. Dr. Alexander Wolff
Diana Sieper, B. Sc.

Einführung in IBM ILOG CPLEX Optimization Studio – Teil 1

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

Diese Software ist nicht frei verfügbar. Wir haben uns aber trotzdem für sie entschieden, da sie durch die mitgelieferte IDE einfach zu bedienen ist. Es gibt aber auch freie LP-Solver, die sie ebenfalls für die Lösung der Übungsaufgaben nutzen können.

1 Start des Programms

Die Software ist im CIP-Pool E40 im Rechenzentrums-Gebäude und in den CIP-Pools im Informatikgebäude installiert. Sie können sie starten, indem Sie im Verzeichnis `/opt/ibm/ILOG/CPLEX_Studio1262/opl/oplide/` das Programm `oplide` ausführen. Von der Kommandozeile kann die IDE mittels

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

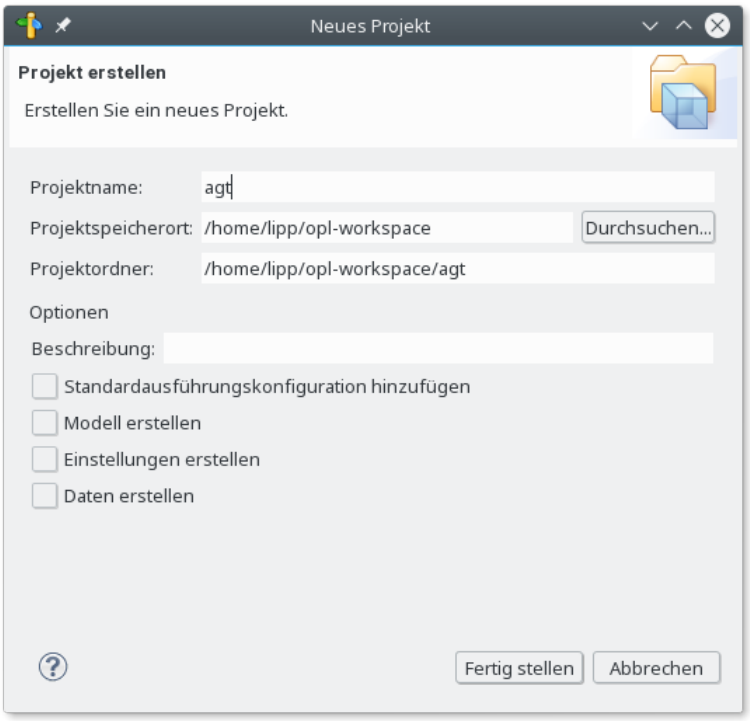
gestartet werden.

Der Zugriff auf die in den CIP-Pools installierte Software ist auch weiterhin per ssh möglich, aber gegebenenfalls umständlich und langsam. Alternativ können Sie sich 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 oben angegebenen Verzeichnispfad, um das Programm zu starten.

Aufgrund der Situation haben wir uns entschlossen, alle Übungsaufgaben, in denen CPLEX verwendet wird, zu Bonusaufgaben zu machen. Wir empfehlen dennoch, die Aufgaben sorgfältig zu bearbeiten, um sich mit dem wichtigen Konzept der Linearen Programmierung vertraut zu machen!

2 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.

Der folgende Quelltext entspricht dem linearen Programm von den Vorlesungsfolien:

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

maximize 30 * x1 + 50 * x2;
subject to {
      4 * x1 + 11 * x2 <= 880;
      x1 + x2 <= 150;
      x2 <= 60;
};
```

Kurze Erklärung dieses Quelltextes:

- `dvar`: deklariert eine Variable deren Wert optimiert werden soll
- `float+`: Datentyp: nicht-negative Gleitkommazahl
- `maximize`: gibt die Zielfunktion an (Alternative: `minimize`)
- `subject to`: in diesem Block werden die linearen Constraints definiert

3 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 ganz oben 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, `x1`: 110, `x2`: 40

4 Ausführung auf der Kommandozeile

Sie können OPL-Programme alternativ auch auf der Kommandozeile ausführen und so die Installation im CIP-Pool von zu Hause über SSH nutzen.

Dazu loggen Sie sich zunächst auf einem beliebigen Arbeitsplatzrechner in einem CIP-Pool ein. Um OPL aufrufen zu können, müssen Sie zwei Umgebungsvariablen anpassen; in der bash beispielsweise mit den folgenden Befehlen:

```
export PATH="$PATH:/opt/ibm/ILOG/CPLEX_Studio1262/opl/bin/x86-64_linux"
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:\
/opt/ibm/ILOG/CPLEX_Studio1262/opl/bin/x86-64_linux"
```

Anschließend können Sie ein Modell mit `oplrun mein-modell.mod` ausführen. Allerdings wird dabei standardmäßig nur der Zielfunktionswert und nicht die Werte der einzelnen Variablen ausgegeben. Das können Sie dadurch ändern, dass sie eine kurze Ausgabefunktion am Ende Ihres Modells einfügen. Im oben angegebenen Beispiel könnte das so aussehen:

```
execute {
      writeln("x1: ", x1);
      writeln("x2: ", x2);
}
```