Lehrstuhl für Informatik I
Algorithmen und Komplexität
Universität Würzburg

Würzburg, den 30. April 2025

Prof. Dr. Alexander Wolff
Samuel Wolf, M. Sc.

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

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

- auf einen CIP-Pool-Rechner des Instituts von zuhause aus zugreifen.
- 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 (ganzzahlige) lineare Programme zu lösen. Das Programm kann von zuhause aus auf einem CIP-Pool-Rechner, mit dem man per ssh verbunden ist, ausgeführt werden.

Dieses Paket enthält die Programmiersprache OPL, mit der auf intuitive Weise lineare Programme formuliert und gelöst werden können.

2 Alternativen

Es ist auch eine Entwicklungsumgebung (IDE) enthalten, die auf diese Sprache zugeschnitten ist und auf Eclipse basiert. Sie kann vor Ort an den CIP-Pool-Rechnern der Informatik genutzt werden – 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 OPL-Programme intuitiv formuliert werden können. Es gibt aber auch freie LP-Solver, die sie ebenfalls zum Lösen von linearen Programmen nutzen können.

3 Verbinden mit einem CIP-Pool-Rechner

Im Folgenden wird beschrieben, wie Sie sich auf der Kommandozeile mit einem CIP-Pool-Rechner verbinden. Das funktioniert sowohl auf der Linux-Kommandozeile (bash) als auch auf der Windows-Kommandozeile (cmd), sofern dort der ssh-Befehl verfügbar ist. Dies sollte ab Windows 10 standardmäßig der Fall sein. Sobald Sie über ssh mit einem Linux-Rechner des Instituts verbunden sind, sollte dort der standardmäßige Linux-Befehlsatz für die Kommandozeile verfügbar sein.

3.1 VPN-Verbindung ins Universitäts-Netzwerk

Zunächst müssen Sie sich über SSH mit dem Sprungbrettrechner des Computerpools verbinden. Wenn Sie sich nicht im Uni-Netzwerk befinden, müssen Sie sich zunächst über VPN mit dem Uni-Netzwerk verbinden. Wenn Sie hierzu mehr Informationen benötigen, besuchen Sie `https://www.rz.uni-wuerzburg.de/dienste/it-sicherheit/vpn/` .

3.2 SSH-Verbindung zum Sprungbrettrechner

Da Sie sich nun im Uni-Netzwerk befinden, können Sie den folgenden Befehl ausführen, wobei Sie `s000000` durch Ihren eigenen JMU-Account („s-Nummer“) ersetzen. Geben Sie anschließend Ihr zugehöriges Passwort ein.

```
ssh s000000@cipgate.informatik.uni-wuerzburg.de -X -Y
```

Sie sind nun mit dem Sprungbrettrechner des Computerpools verbunden. Für rechenintensive Aufgaben sollten Sie sich jedoch mit einem Rechner der CIP-Pools verbinden.

3.3 SSH-Verbindung zu einem CIP-Pool-Rechner

Wenn Sie sich mit dem Sprungbrettrechner des Computerpools verbunden haben, wird Ihnen eine Liste von eingeschalteten Rechnern der CIP-Pools (und wie viele Nutzer dort angemeldet sind) angezeigt. Sollte die Liste leer sein, ist gerade kein Rechner eingeschaltet. Sie können selbst einen Rechner von außen „aufwecken“, also einschalten. Hierfür müssen Sie den folgenden Befehl abschicken, wenn Sie mit dem Sprungbrettrechner verbunden sind:

```
wakeAP rechnername
```

Die Rechnernamen sind benannt als CIP-Pool-Name minus Rechnernummer. Verwenden Sie die Rechner im CIP-Pool `a001`, hier gibt es die Rechner 01 bis 20, also z. B.:

```
wakeAP a001-01
```

Achtung – da montags Updates auf Windows eingespielt werden, sind an diesem Tag die Rechner in `a001` nicht verfügbar. Weichen Sie auf den CIP-Pool `bibsem` oder einen

anderen Tag aus. Wenn Sie einen Befehl zum „Aufwecken“ ausgeführt haben, warten Sie einen Moment bis der Rechner gestartet ist. Sie können sich nun mittels SSH mit einem CIP-Pool-Rechner verbinden, wobei Sie `rechnername` durch den Namen des Rechners, z. B. `a001-01`, ersetzen. Geben Sie anschließend erneut Ihr Passwort ein.

```
ssh rechnername
```

3.4 Setzen der Umgebungsvariablen (inzwischen nicht mehr nötig)

Sie sind nun über Ihren Rechenzentrums-Account mit einem CIP-Pool-Rechner verbunden und können dort auch auf Ihr Dateisystem zugreifen. In diesem Dateisystem werden Sie die OPL-Dateien ablegen und von dort aus lösen lassen. IBM ILOG CPLEX ist bereits auf den CIP-Pool-Rechnern installiert, um es jedoch von überall in Ihrem Dateisystem ausführen zu können, müssen Sie ggf. noch die zugehörigen Umgebungsvariablen setzen. Da Ihre Umgebungsvariablen auf dem CIP-Pool-Rechner nicht dauerhaft gespeichert werden, müssen Sie diese bei jeder Sitzung zu Beginn neu setzen. Verwenden Sie hierfür die folgenden zwei Befehle (ggf. ohne den Backslash (\) am Zeilenende, wenn Sie keinen Zeilenumbruch verwenden):

```
export PATH=\
"$PATH:/opt/cplex/cplex/bin/x86-64_linux:/opt/cplex/opl/bin/x86-64_linux"

export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:\
/opt/cplex/opl/lib/x86-64_linux/static_pic:/opt/cplex/cplex/lib/x86-64_linux/sta
```

Sie können diese Befehle auch in die Datei `.bashrc` in Ihrem Heimatverzeichnis übernehmen. Dann werden sie jedes Mal ausgeführt, wenn Sie sich einloggen.

Sie haben nun alles vorbereitet, um lineare Programme durch die IBM-ILOG-CPLEX-Installation der Universität Würzburg lösen zu lassen.

4 Anlegen eines linearen Programms im OPL-Format

Wenn Sie die ssh-Session (z.B.) von einem Ubuntu-System gestartet haben können Sie jetzt remote die IDE starten und entsprechend in die GUI-Anleitung wechseln:

```
oplide
```

Sollte dies von ihrem (z.B. Windows-) System nicht unterstützt werden, wird im folgenden die Verwendung von OPL in der Konsole erklärt.

Die *Optimization Programming Language* (OPL) ist eine Programmiersprache, die für IBM ILOG CPLEX zur Beschreibung mathematischer Optimierungsmodelle geschaffen wurde. Dateien im OPL-Format haben die Endung `.mod`, was für Modell steht.

4.1 Anlegen einer OPL-Modell-Datei

Wir werden nun als Beispiel das lineare Programm aus der Vorlesung ins OPL-Format übertragen. Hierfür legen wir uns einen Ordner und eine `.mod`-Datei an, welche wir direkt auf dem CIP-Pool-Rechner bearbeiten werden. Alternativ können Sie diese Datei natürlich auch zunächst lokal auf ihrem Rechner erstellen und anschließend auf den CIP-Pool-Rechner laden und ausführen. In diesem Fall können Sie direkt zum nächsten Abschnitt springen.

Wenn Sie sich im Home-Verzeichnis Ihres Rechenzentrums-Accounts befinden, legen Sie beispielsweise den Ordner `agt22` und darin den Ordner `test_lp` an, in den wir das lineare Programm legen werden.

```
mkdir agt22
cd agt22
mkdir test_lp
cd test_lp
```

Jetzt legen wir eine leere Datei an, in die wir unser lineares Programm schreiben werden.

```
touch test.mod
```

Sie können diese Datei mit einem Textbearbeitungsprogramm Ihrer Wahl editieren. Auf der Kommandozeile bietet sich hierfür `nano` oder `vi` an. Öffnen Sie die Datei also beispielsweise mit:

```
nano test.mod
```

4.2 Syntax des linearen Programms

Variable 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 Programm 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;
}
```

Wir wollen, dass neben dem Zielfunktionswert auch der zugehörige Wert der Variablen ausgegeben wird. Hierfür ergänzen wir den folgenden Befehlsblock am Ende der Datei. Wie in C oder Java können ein- bzw. mehrzeilige Kommentare mit `//` bzw. `/* */` gesetzt werden.

```
execute {
   //Gib Werte der Variablen x1 und x2 aus
   writeln("x1: ", x1);
   writeln("x2: ", x2);
}
```

Insgesamt sieht die Datei also wie folgt aus:

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

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

execute {
   //Gib Werte der Variablen x1 und x2 aus
   writeln("x1: ", x1);
   writeln("x2: ", x2);
}
```

Sie können die Datei nun speichern und das Textbearbeitungsprogramm verlassen.

5 Ausführen des Programms

Nachdem Sie, wie in Abschnitt 3.4 beschrieben, die Umgebungsvariablen für IBM ILOG CPLEX gesetzt haben und das lineare Programm in einer .mod-Datei im gerade ausgewählten Verzeichnis vorliegen haben, können Sie dieses nun mit dem folgenden Befehl lösen lassen.

```
oplrun test.mod
```

Nun sollte IBM ILOG CPLEX Ihr lineares Programm lösen und die folgende Ausgabe erzeugen:

```
<<< setup


<<< generate

Tried aggregator 1 time.
LP Presolve eliminated 1 rows and 0 columns.
Reduced LP has 2 rows, 2 columns, and 4 nonzeros.
Presolve time = 0,00 sec. (0,00 ticks)

Iteration log . . .
Iteration:     1   Scaled dual infeas =             0,000000
Iteration:     2   Dual objective     =          5300,000000

<<< solve


OBJECTIVE: 5300
x1: 110
x2: 40

<<< post process


<<< done
```

Uns interessiert die Ausgabe nach <<< solve, also:

```
OBJECTIVE: 5300
x1: 110
x2: 40
```

Die Ausgabe beschreibt eine optimale Lösung, die von IBM ILOG CPLEX für unser lineares Programm gefunden wurde. Wir erzielen den größten Zielwert, nämlich 5300, wenn wir x_1 auf 110 und x_2 auf 40 setzen.

Prüfen Sie gerne per Hand nach, dass diese Werte mit Zielfunktion und Nebenbedingungen übereinstimmen, und dass sie für keine anderes Paar von Werten für x_1 und x_2 einen größeren Zielwert erreichen, das ebenfalls die Nebenbedingungen einhält.

Sie wissen nun, wie man ein lineares Programm in OPL formuliert und dieses vom Computer lösen lässt. Nun sind Sie bereit im Rahmen der Übungsaufgaben Ihr eigenes lineares Programm zu erstellen und lösen zu lassen. Viel Erfolg!

Sie beenden Ihre ssh-Sessions mit zwei Mal:

```
exit
```