OPL CPLEX tutorial: How to solve an optimization problem in IBM ILOG CPLEX Optimization studio

Masoud Madani

Master Student of Industrial Engineering, Concordia University, Montreal, Canada

What is “CPLEX”, “IDE” and “OPL”

- **Optimization Programming Language (OPL)**
  - A simple and influential modeling language for optimization problems
- **CPLEX**
  - A powerful solver for LP/QP/CP, consequently, MIP, MIQP, etc. from IBM.
- **CPLEX Studio IDE**
  - A user-friendly interface for the CPLEX solver and the CP optimizer
  - Other platforms capable of extracting the CPLEX solver includes;
    - Microsoft Solver Excel, Matlab, CPLEX Interactive Optimizer, Python API, web (online servers such as www.neos-server.org)
    - Concert Technology, Callable Library
- **Integrated Development Environment (IDE)**
  - An IBM product to edit OPL models, solve problems, display results, organize models, data and settings into configurations
- **How to download IBM ILOG CPLEX Optimization Studio (Version 12.6.3)** **free for students**
  - http://www-01.ibm.com/support/docview.wss?uid=swg24041278

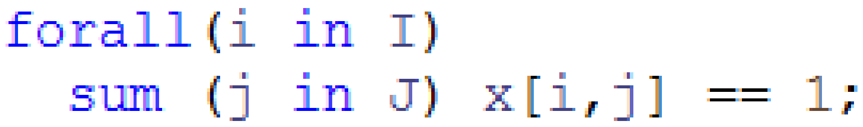
How to code an optimization problem in CPLEX Studio IDE

- **Creating a new project:** to associate .mod, .dat and .ops files
- **Creating a new .mod file** (tutorial-production-1)

  1. Ranges
  2. Decision variables (⤦)
  3. Input values (⤣)
  4. Objective function
  5. Constraints

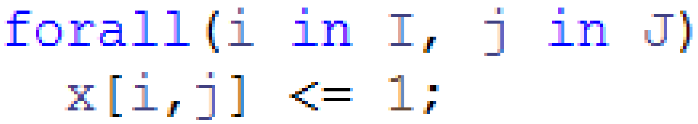
Tricks for Frequently Used Arguments in .mod

- $\sum_{j \in J} x_{ij} = 1 \quad ; \quad \forall\, i \in I$



```
forall(i in I)
  sum (j in J) x[i,j] == 1;
```

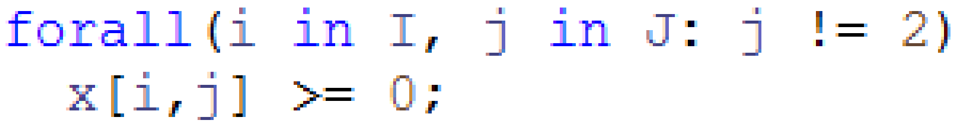
- $x_{ij} \leq 1 \quad ; \quad \forall\, i \in I \;, j \in J$



```
forall(i in I, j in J)
  x[i,j] <= 1;
```

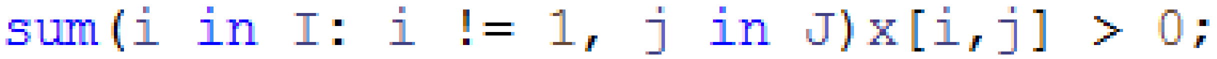
Tricks for Frequently Used Arguments in .mod

- $x_{ij} \geq 0 \quad ; \quad \forall\, i \in I \;, j \in J\backslash\{2\}$



```
forall(i in I, j in J: j != 2)
  x[i,j] >= 0;
```

- $\sum_{i \in I\backslash\{1\}} \sum_{j \in J} x_{ij} > 0$



```
sum(i in I: i != 1, j in J)x[i,j] > 0;
```

How to code an optimization problem

- **Creating a new project:** to associate .mod, .dat and .ops files
- **Creating a new .mod file**
  1. Ranges
  2. Decision variables (⤦)
  3. Input values (⤣)
  4. Objective function
  5. Constraints
  6. Execution (optional) (tutorial-production-3)
- For more flexibility ⟶ **Creating a new .dat file**
  - Be careful about configuration (tutorial-production-2)
- For more flexibility ⟶ **Having several configurations** (tutorial-production-5)

How to have connection with Excel

- For more flexibility ⟶ **Linkage with an Excel file** (tutorial-production-4)
  - Input
    1. SheetConnection functionName ("excel file name");
    2. Data from SheetRead (functionName, "sheetname!cells");
  - Output
    1. SheetConnection functionName ("excel file name");
    2. Data to SheetWrite (functionName, "sheetname!cells");
- **Generalized Programming** (tutorial-production-6)

Tricks:

- **3-dimensional table in Excel**

(tutorial-production-multi period)

- ➢ create a 2-dimensional sub-table and input the data to it
- ➢ create a new range for the sub-table
- ➢ create the main table by enumerating sub-table

How CPLEX solves problems

- Solves linear programming (LP) problems using simplex/dual simplex/interior point methods
- Solves network flow problems using the network simplex method
- Solves convex quadratic programming (QP) using interior point methods
- Solves pure integer (IP) and mixed integer programming problems by using branch and cut (B&C) methods

How CPLEX solves a MIP problem

- **Branch and Cut (B&C)**
  - B&B combined with cutting planes methods
  - Dynamic programming

- **CP Optimizer**
  - Constraints Propagation
  - Constructive Search techniques

Constraint Propagation

- Tries to reduce the domain of decision variables, which leads in reduction in search space;

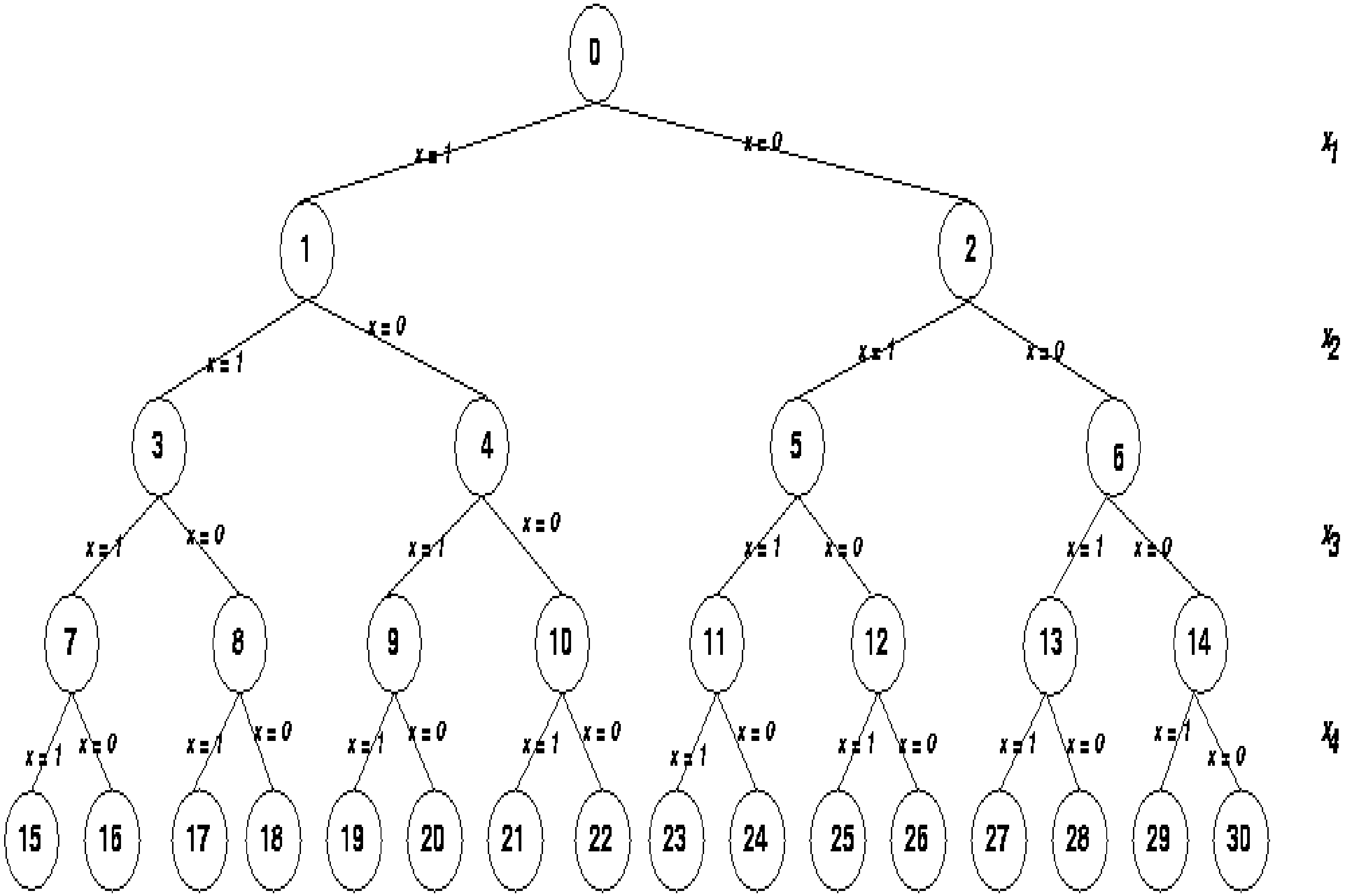
$$x + y \leq 8$$
$$x, y \in \{1,2,3,4,5,6,7,8,9,10\}$$

Initial Propagation:

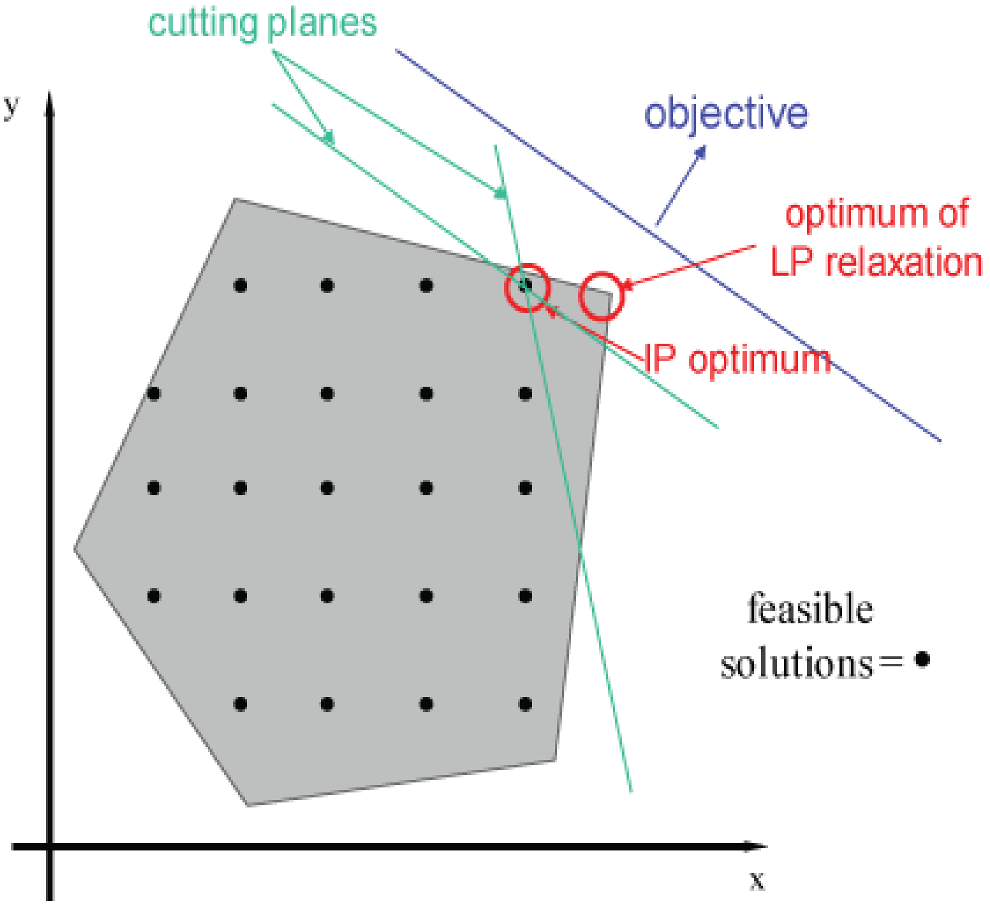
$$x \in \{1,2,3,4,5,6,7,\cancel{8,9,10}\}$$
$$y \in \{1,2,3,4,5,6,7,\cancel{8,9,10}\}$$

- Initial Constraint Propagation
- Constraint Propagation during Search

B&B


0
x = 1
x = 0
x1
1
2
x = 1
x = 0
x = 1
x = 0
x2
3
4
5
6
x = 1
x = 0
x = 1
x = 0
x = 1
x = 0
x = 1
x = 0
x3
7
8
9
10
11
12
13
14
x = 1
x = 0
x = 1
x = 0
x = 1
x = 0
x = 1
x = 0
x = 1
x = 0
x = 1
x = 0
x = 1
x = 0
x = 1
x = 0
x4
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

Cutting Planes



Cutting Planes

Setting configuration

- Set LP format to gain B&C information
- Decide about relaxed variables/constraints
- Decide about B&C tree:
  - MIP dynamic search switch
  - Node selection
  - Branching direction

How to interpret "Engine log" and "Statistics" taps

- Integrality relaxed problems are usually solved by dual simplex
  - How to control it
- How to control MIP tolerances
- How to control MIP limits
- How to control MIP Strategy
  - How to set B&C tree structure
  - How to obtain Upper Bound (UP)
- How to control MIP cuts

Formulation Tightness and CPLEX Approach

- A Capacitated Facility Location Problem (CFLP)

$$Min\ z = \sum_{j \in J} f_j\, y_j + \sum_{i \in I} \sum_{j \in J} c_{ij}\, x_{ij}$$

$s.t.:$

$$\sum_{i \in I} d_i\, x_{ij} \leq \mu_j\, y_j \quad ; \qquad \forall j \in J$$

$$\sum_{j \in J} x_{ij} \geq d_i; \qquad \forall i \in I$$

Formulation Tightness and CPLEX Approach

- Valid inequalities for CFLP;

$$\sum_{j\in J} x_{ij} \geq d_i \qquad \longrightarrow \qquad \sum_{j\in J} x_{ij} = d_i$$

$$\sum_{i\in I} x_{ij} \leq y_j \quad ; \quad \forall j \in J$$

$$\sum_{j\in J} \mu_j \, y_j \geq \sum_{i\in I} d_i \, x_{ij}$$

Advanced Programming Techniques

- Pre-solving
- Parallel Optimizers

Why using CPLEX Studio IDE

- A user-friendly interface for the CPLEX solver and the CP optimizer
- No need to programming languages
- No need to define directories, linkers, preprocessors, etc.

Drawback of CPLEX Studio IDE

We have to use Concert Technology/Callable Library to have;

- Advanced MIP Control Interface (CallBack)
  - Customized solution method (Solve / Incumbent CallBack)
  - Full control of solution method (MIP Control CallBack)
  - Stopping procedure for any inspection (Selection CallBacks)
  - Adding user-cuts / lazy-constraints (Cut CallBack)
  - …

The most powerful interfaces for CPLEX solver and CP optimizer with full control;

- Concert Technology: for C++/Java/.NET users
- Callable Library: for C users

Suggested References

- **Implementation**
  - Help – IBM ILOG CPLEX Optimization Studio
  - IBM ILOG CPLEX Optimization Studio OPL Language User's Manual, IBM
  - https://www.ibm.com/support/knowledgecenter/

- **Theoretical Concepts**
  - Wolsey, Integer Programming
  - 50 Years of Integer Programming 1958-2008
  - Fischetti, Glover, Lodi, The Feasibility Pump
  - Nemhauser, Integer Programming and Combinatorial Optimization

Thank you for attention