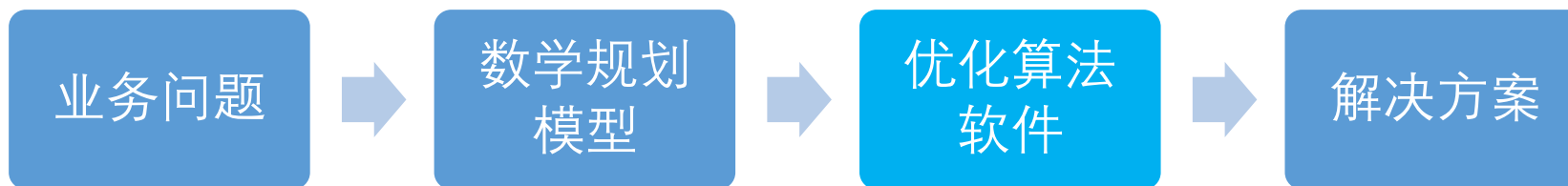


Introduction to CPLEX

问题求解



数学规划模型

- 线性规划 Linear Programming
- 整数规划 Integer Programming
- 网络流模型 Network flow
- 动态规划 Dynamic Programming
- 二次规划 Quadratic Programming
- 约束规划 Constraint Programming
- 凸优化 Convex Optimization
- 非线性规划 Nonlinear Programming
-

常用优化软件

- Matlab
- Mathematica
- ILOG CPLEX
- Gurobi
-

IBM ILOG CPLEX Optimizer

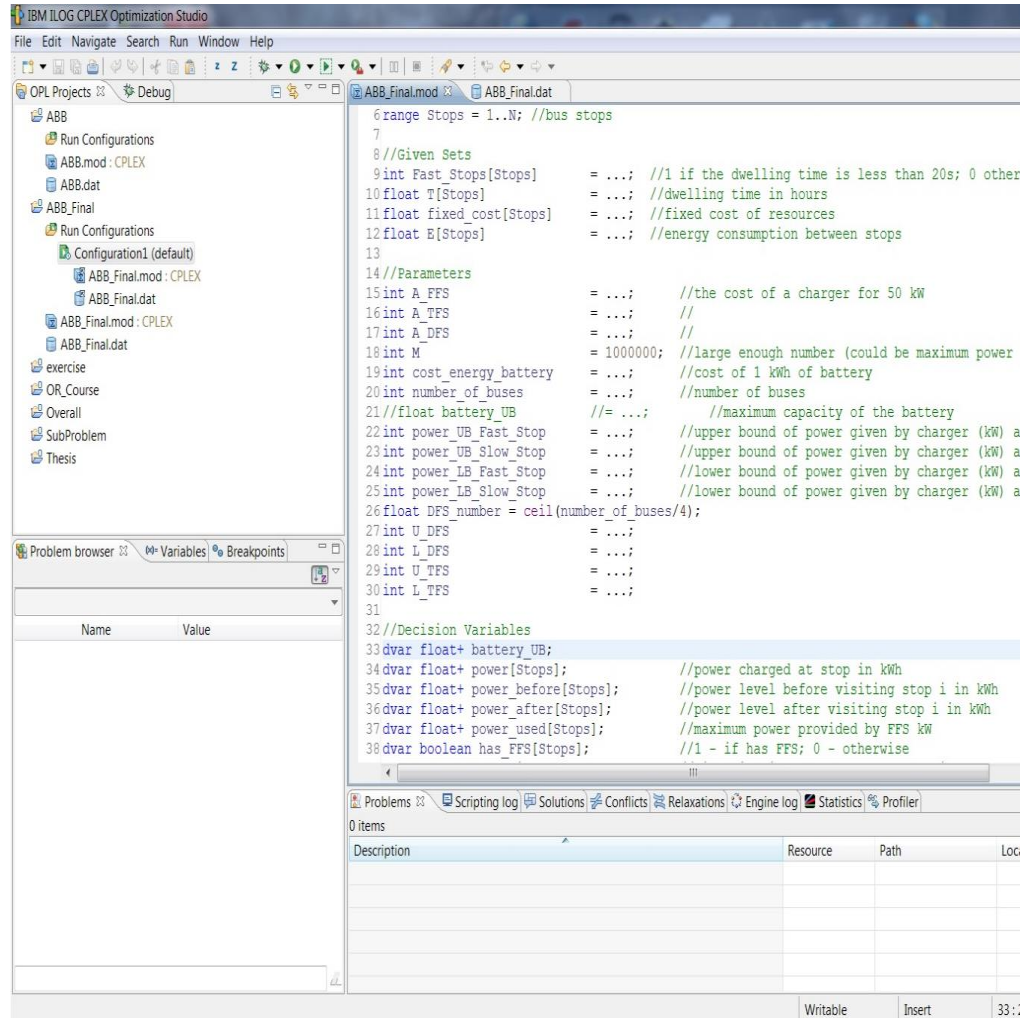


- 数学优化领域最著名软件之一
- 历史
 - 2009年IBM公司买下ILOG CPLEX
 - 1997年被ILOG公司买下
 - 1988年由CPLEX Optimization Inc公司商业化
 - 最早由 Robert E. Bixby用C语言开发, **C** Programming **Simplex** Method -- CPLEX
- 适用场景
 - 线性规划 Linear Programming
 - 整数规划 Integer Programming
 - 网络流模型 Network flow
 - 二次规划 Quadratic Programming
 - 约束规划 Constraint Programming

CPLEX

- IBM ILOG CPLEX Optimization Studio
 - IDE 集成开发环境
 - OPL 建模语言
- CPLEX Callable library C语言
- CPLEX Concert Technology
 - 提供编程语言调用接口
 - 支持语言包括 C++、C#、Java、Excel、Matlab
- 最新版本12.8.0
- 适用场景
 - 线性规划 Linear Programming
 - 整数规划 Integer Programming
 - 网络流模型 Network flow
 - 二次规划 Quadratic Programming
 - 约束规划 Constraint Programming

IBM ILOG CPLEX Optimization Studio



```
// declare data
...
// declare decision variables
...
// declare objective function and constraints
maximize
    sum (p in Products) profit[p] * production[p];
subject to {
    forall (p in Products)
        production[p] >= minProd[p];
    sum (p in Products) Atime[p] * production[p] <= Aavail;
    sum (p in Products) Ptime[p] * production[p] <= Pavail;
}
```

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

$$\text{s.t. } x_i \geq m_i, \quad \forall i \in I$$

$$\sum_{i \in I} a_i x_i \leq b$$

$$\sum_{i \in I} c_i x_i \leq d$$

OPL 建模语言

CPELX Concert Technology

Maximize $x_1 + 2x_2 + 3x_3$

subject to $-x_1 + x_2 + x_3 \leq 20$
 $x_1 - 3x_2 + x_3 \leq 30$

with these bounds $0 \leq x_1 \leq 40$
 $0 \leq x_2 \leq +\infty$
 $0 \leq x_3 \leq +\infty$

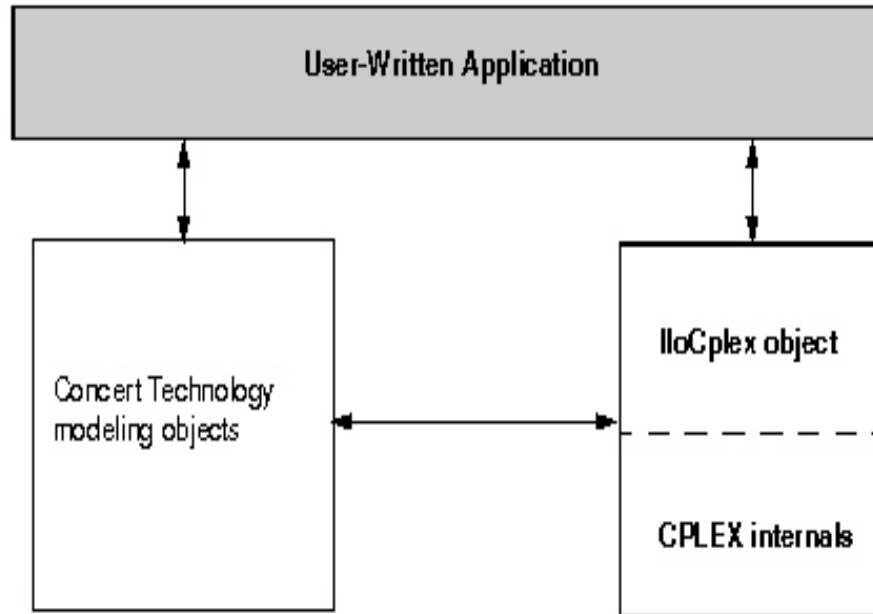
```
3 // Model
4 IloCplex cplex = new IloCplex();
5 double[] lb = {0.0, 0.0, 0.0};
6 double[] ub = {40.0, Double.MAX_VALUE, Double.MAX_VALUE};
7 IloNumVar[] x = cplex.numVarArray(3, lb, ub);
8 double[] objvals = {1.0, 2.0, 3.0};
9 cplex.addMaximize(cplex.scalProd(x, objvals));
10 cplex.addLe(cplex.sum(cplex.prod(-1.0, x[0]),
11 cplex.prod( 1.0, x[1]),
12 cplex.prod( 1.0, x[2])), 20.0);
13 cplex.addLe(cplex.sum(cplex.prod( 1.0, x[0]),
14 cplex.prod(-3.0, x[1]),
15 cplex.prod( 1.0, x[2])), 30.0);
16 // Optimize
17 cplex.solve();
```

通过Java调用CPLEX接口，建立数学模型，以及求解。

CPLEX Toolkit

- CPLEX allows one to work in several ways.
- An IDE that uses the OPL modeling language
- An interactive optimizer that reads MPS/LP input
- A callable library in several languages
 - C (Callable library)
 - C++ (Concert Technology)
 - **Java (Concert Technology)**

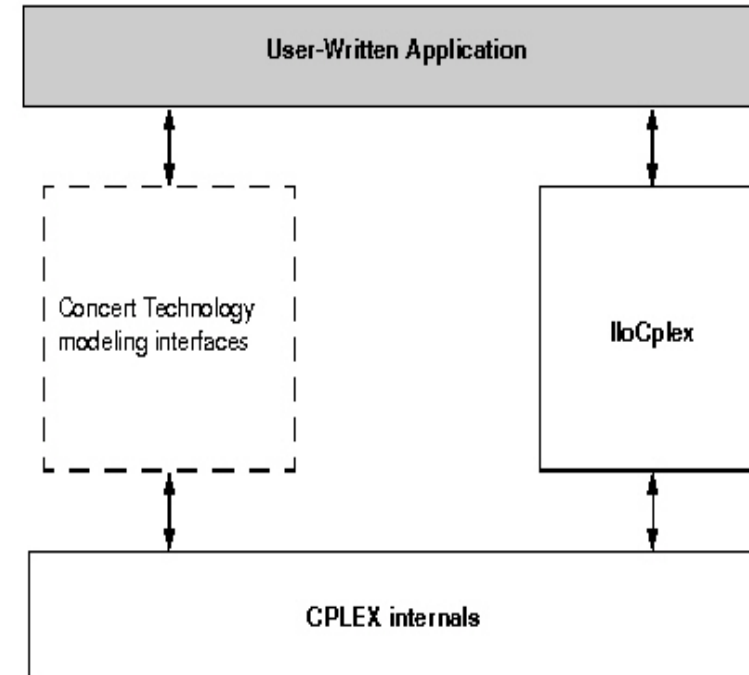
Concert Technology



C++

An `IloModel` object defines the optimization problem (constraints, objective function, etc.)

An `IloCplex` object reads a model, extracts its data, solves the problem and answers queries on solution.



Java

An `IloCplex` object not only defines the optimization problem but also solves the problem and answers queries on solution

No `IloModel`

Structure of an application

- CPLEX Java Packages
 - `import`
`ilog.concert.*`
 - `import`
`ilog.cplex.*`
- an [IloCplex](#) object
- Java error handling in CPLEX uses exceptions
 - Try/Catch
 - Throw exceptions

```
import ilog.concert.*;
import ilog.cplex.*;
static public class Application {
    static public main(String[] args) {
        try {
            IloCplex cplex = new IloCplex();
            // create model and solve it
        } catch (IloException e) {
            System.err.println("Concert exception caught: " + e);
        }
    }
}
```

Create the model

- An `IloCplex` Object implements
 - Variables
 - Objective
 - Constraints

Modeling classes	Description
<code>IloNumVar</code>	modeling variables
<code>IloRange</code>	ranged constraints of the type $lb \leq \text{expr} \leq ub$
<code>IloObjective</code>	optimization objective
<code>IloNumExpr</code>	expression using variables

Define variables

```
// e.g. define continuous variables x0 x1 x2
IloNumVar[] x = new IloNumVar[3];
x[0] = cplex.numVar(0, 40, "x0");
x[1] = cplex.numVar(0, Double.MAX_VALUE, "x1");
x[2] = cplex.numVar(0, Double.MAX_VALUE, "x2");
```

$$\begin{aligned} 0 \leq x_0 &\leq 40 \\ 0 \leq x_1 &\leq \infty \\ 0 \leq x_2 &\leq \infty \\ x_i &\in \mathbb{R} \end{aligned}$$

```
// e.g. create four continuous variables, all with
bounds 0 and 100:
```

```
IloNumVar[] x = cplex.numVarArray(4, 0.0,
100.0);
```

```
// 整数变量 e.g. x0 是0-40间的整数变量
```

```
• x[0] = cplex.intVar(0, 40, "x0");
```

Build Expressions

$$x[0] + 2*x[1] + 3*x[2]$$

- // use type [IloNumExpr](#)

```
IloNumExpr expr = cplex.sum(x[0], cplex.prod(2.0, x[1]), cplex.prod(3.0, x[2]));
```

- // use type [IloLinearNumExpr](#)

```
IloLinearNumExpr expr = cplex.linearNumExpr();  
expr.addTerm(1.0, x[0]);  
expr.addTerm(2.0, x[1]);  
expr.addTerm(3.0, x[2]);
```

Objective

- `cplex.addMinimize(expr)`
- `cplex.addMaximize(expr)`

Constraints

- `cplex.addLe(expr, ub) // expr <= ub`
- `cplex.addGe(expr, lb) // expr >= lb`
- `cplex.addEq(expr, q) // expr = q`

输出模型

- `cplex.exportModel("example_1.lp");`

Solve the Model

- `cplex.solve()`

Query the results

Query the objective value of that solution

- `double objval = cplex.getObjValue();`

Query solution values for all the variables

- `double[] xval = cplex.getValues(x);`

Write solution into files

- `cplex.writeSolution("example_1.xml");`

Example 1

- A linear program

$$\begin{aligned} \max \quad & x_0 + 2x_1 + 3x_2 \\ & -x_0 + x_1 + x_2 \leq 20 \\ & x_0 - 3x_1 + x_2 \leq 30 \\ & 0 \leq x_0 \leq 40 \\ & 0 \leq x_1 \leq \infty \\ & 0 \leq x_2 \leq \infty \\ & x_i \in \mathbb{R} \end{aligned}$$

Example 1

$$\begin{aligned} \max \quad & x_0 + 2x_1 + 3x_2 \\ & -x_0 + x_1 + x_2 \leq 20 \\ & x_0 - 3x_1 + x_2 \leq 30 \\ & 0 \leq x_0 \leq 40 \\ & 0 \leq x_1 \leq \infty \\ & 0 \leq x_2 \leq \infty \\ & x_i \in \mathbb{R} \end{aligned}$$

```
public static void main(String[] args) throws IOException {
    IloCplex cplex = new IloCplex();

    // define variables x0 x1 x2
    IloNumVar[] x = new IloNumVar[3];
    x[0] = cplex.numVar(0, 40, "x0");
    x[1] = cplex.numVar(0, Integer.MAX_VALUE, "x1");
    x[2] = cplex.numVar(0, Integer.MAX_VALUE, "x2");

    // objective
    IloLinearNumExpr obj = cplex.linearNumExpr();
    obj.addTerm(1, x[0]);
    obj.addTerm(2, x[1]);
    obj.addTerm(3, x[2]);
    cplex.addMaximize(obj);

    // constraints
    IloLinearNumExpr c1 = cplex.linearNumExpr();
    c1.addTerm(-1, x[0]);
    c1.addTerm(1, x[1]);
    c1.addTerm(1, x[2]);
    cplex.addLe(c1, 20, "c1");

    IloLinearNumExpr c2 = cplex.linearNumExpr();
    double[] a2 = {1, -3, 1};
    c2.addTerms(a2, x);
    cplex.addLe(c2, 30, "c2");
}
```


Example 1

```
public static void main(String[] args) throws IloException {
    IloCplex cplex = new IloCplex();

    // define variables x0 x1 x2
    IloNumVar[] x = new IloNumVar[3];
    x[0] = cplex.numVar(0, 40, "x0");
    x[1] = cplex.numVar(0, Integer.MAX_VALUE, "x1");
    x[2] = cplex.numVar(0, Integer.MAX_VALUE, "x2");

    // objective
    IloLinearNumExpr obj = cplex.linearNumExpr();
    obj.addTerm(1, x[0]);
    obj.addTerm(2, x[1]);
    obj.addTerm(3, x[2]);
    cplex.addMaximize(obj);

    // constraints
    IloLinearNumExpr c1 = cplex.linearNumExpr();
    c1.addTerm(-1, x[0]);
    c1.addTerm(1, x[1]);
    c1.addTerm(1, x[2]);
    cplex.addLe(c1, 20, "c1");

    IloLinearNumExpr c2 = cplex.linearNumExpr();
    double[] a2 = {1, -3, 1};
    c2.addTerms(a2, x);
    cplex.addLe(c2, 30, "c2");
}
```

```
// output model
cplex.exportModel("example_1.lp");

if(cplex.solve()){
    System.out.println("Status: "+cplex.getStatus());
    // get optimal objective value
    double objVal = cplex.getObjValue();
    System.out.println("objValue: "+objVal);
    // get optimal solution value
    double[] _x = cplex.getValues(x);
    System.out.println("x = "+Arrays.toString(_x));

    // write solution to file
    cplex.writeSolution("example_1.xml");
}else{
    System.err.println("Infeasible");
}
```

Java 编译与运行

```
MacBook-2:example naiquh$ ls
example1.java  example2.java
MacBook-2:example naiquh$ javac -classpath /Users/naiquh/Applications/Cplex127/cplex/lib/cplex.jar example1.java
MacBook-2:example naiquh$ ls
example1.class example1.java  example2.java
MacBook-2:example naiquh$ java -Djava.library.path=/Users/naiquh/Applications/Cplex127/cplex/bin/x86-64_osx -classpath /Users/naiquh/Applications/Cplex127/cplex/lib/cplex.jar:. example1
Tried aggregator 1 time.
No LP presolve or aggregator reductions.
Presolve time = 0.00 sec. (0.00 ticks)

Iteration log . . .
Iteration:      1    Dual objective      =      220.000000
Status: Optimal
objValue: 202.5
x = [40.0, 17.5, 42.5]
MacBook-2:example naiquh$
```

- 编译

- `javac -classpath path_to_cplex.jar Test.java`

- 运行

- `java -classpath path_to_cplex.jar:. -Djava.library.path=path_to_shared_library Test`