CIVIL-557

Decision Aid Methodologies In Transportation

Lab II: Using Mathematical Solver (CPLEX)

Virginie Lurkin

Transport and Mobility Laboratory (TRANSP-OR)
École Polytechnique Fédérale de Lausanne EPFL

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Overview

- Using mathematical solver
  - What is a mathematical solver
  - What is the aim of this course
  - What is CPLEX
  - CPLEX for academics
  - Installing CPLEX
  - Using CPLEX
  - First Example to solve
- Container storage problem
- Transportation problem

Using Mathematical Solver

What is a mathematical solver?

- It is a piece of **software** that takes as an input the mathematical formulation of an optimization problem.
- In this course: formulation needs to be **linear**
- The solver applies several **optimization techniques**, in order to find the optimal solution(s).
- These techniques include the simplex method, but also many other methods (branch and bound, cutting planes, probing, pre-processing, heuristics, etc.)

Hillier, F. S., & Lieberman, G. J. (2001). Introduction to Operations Research, McGraw Hill. New York, page 23.

What is the aim of this course?

- To learn mathematical modeling and linearization
- To solve mathematical models using CPLEX
- Overall – solving of decision problems arising in transportation

Hillier, F. S., & Lieberman, G. J. (2001). Introduction to Operations Research, McGraw Hill. New York, page 23.

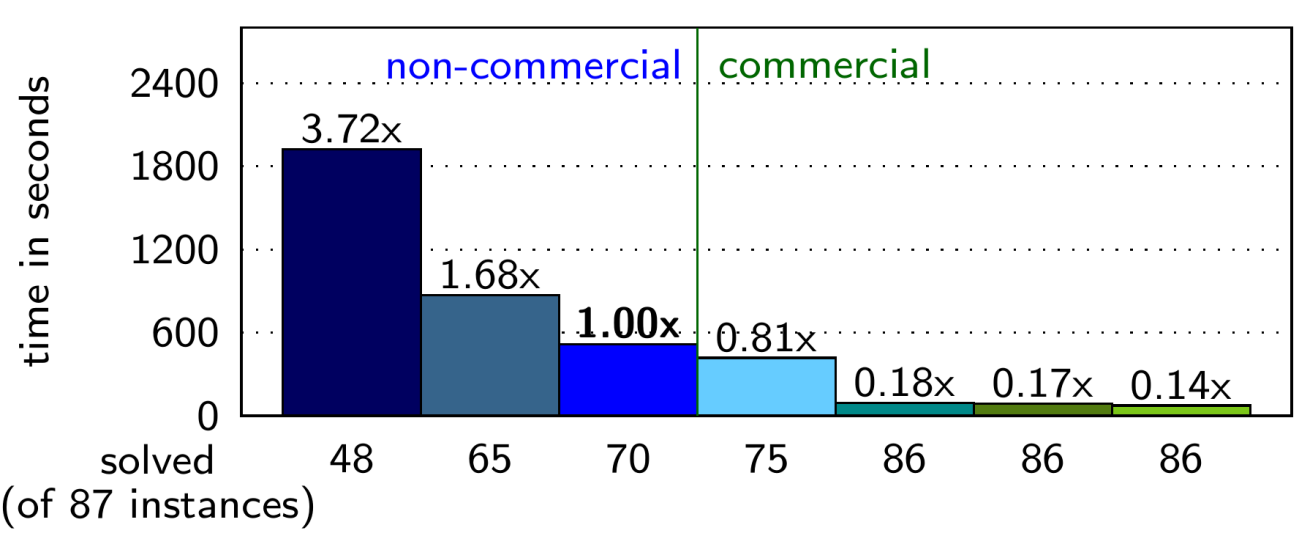
What is CPLEX?

Model business issues mathematically and solve them with IBM ILOG CPLEX Optimizer's powerful algorithms to produce precise and logical decisions.

IBM ILOG CPLEX Optimizer's mathematical programming technology enables decision optimization for improving efficiency, reducing costs, and increasing profitability.

- Fundamental algorithms: IBM ILOG CPLEX Optimizer provides flexible, high-performance mathematical programming solvers for linear programming, mixed integer programming, quadratic programming, and quadratically constrained programming problems. These include a distributed parallel algorithm for mixed integer programming to leverage multiple computers to solve difficult problems.
- Robust algorithms for demanding problems: IBM ILOG CPLEX Optimizer has solved problems with millions of constraints and variables.

What is CPLEX?



- CBC 2.9.4
- MIPCL 1.1.2
- **SCIP 3.2.1 – SoPlex 2.2.1**
- SCIP 3.2.1 – CPLEX 12.6.3
- Xpress 8.0.0
- CPLEX 12.6.3
- Gurobi 6.5.0

data: Hans Mittelmann
graphics: ZIB

CPLEX for academics

- Academics can get free version of CPLEX: https://www.ibm.com/developerworks/community/blogs/jfp/entry/CPLEX_Is_Free_For_Students?lang=en

Installing CPLEX

- CPLEX is already installed on the lab's pcs
- If you want to use CPLEX on your own personal laptop follow the manual at: https://www.ibm.com/support/knowledgecenter/SSSA5P_12.7.1/ilog.odms.studio.help/pdf/gscplex.pdf
- Works on both Windows and Mac iOS

Using CPLEX Studio IDE

- We will use the **CPLEX Studio IDE** and the modeling language **OPL**
- OPL allows you to write a mathematical representation of a problem that is separate from your data.
- In OPL, a project is defined as one or more model files and one or more data files.
- Within a project, a run configuration represents a problem instance with a model and a data file, and you can have multiple run configurations within an OPL project.

Using CPLEX Studio IDE

- Each project contains two main files:

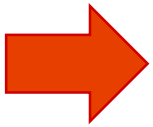
  1. A file **.mod** that contains the model to solve
  2. A file **.dat** that contains the data of the model

Using CPLEX Studio IDE

- The decision variables are defined with the key word `dvar` followed by their type `(int, float, boolean,…)`
- The objective function follows the key word `minimize` or `maximize`
- The constraints are defined in a block using {} and follows the key word `subject to`
- The main operators are:
  - `+ - * /`
  - `<= >= ==`

First example to solve

$$\begin{aligned} \min \quad & 3x + 2y \\ \text{subject to} \quad & x - y \geq 5 \\ & 3x + 2y \geq 10 \end{aligned}$$



```
dvar float x;
dvar float y;

minimize 3*x + 2*y;

subject to{
   x - y >= 5;
   3*x + 2*y >= 10;
}
```

Container storage problem

Exercise

- Solve the container storage problem for the large instance provided in the file `Labo2.dat`

Solution

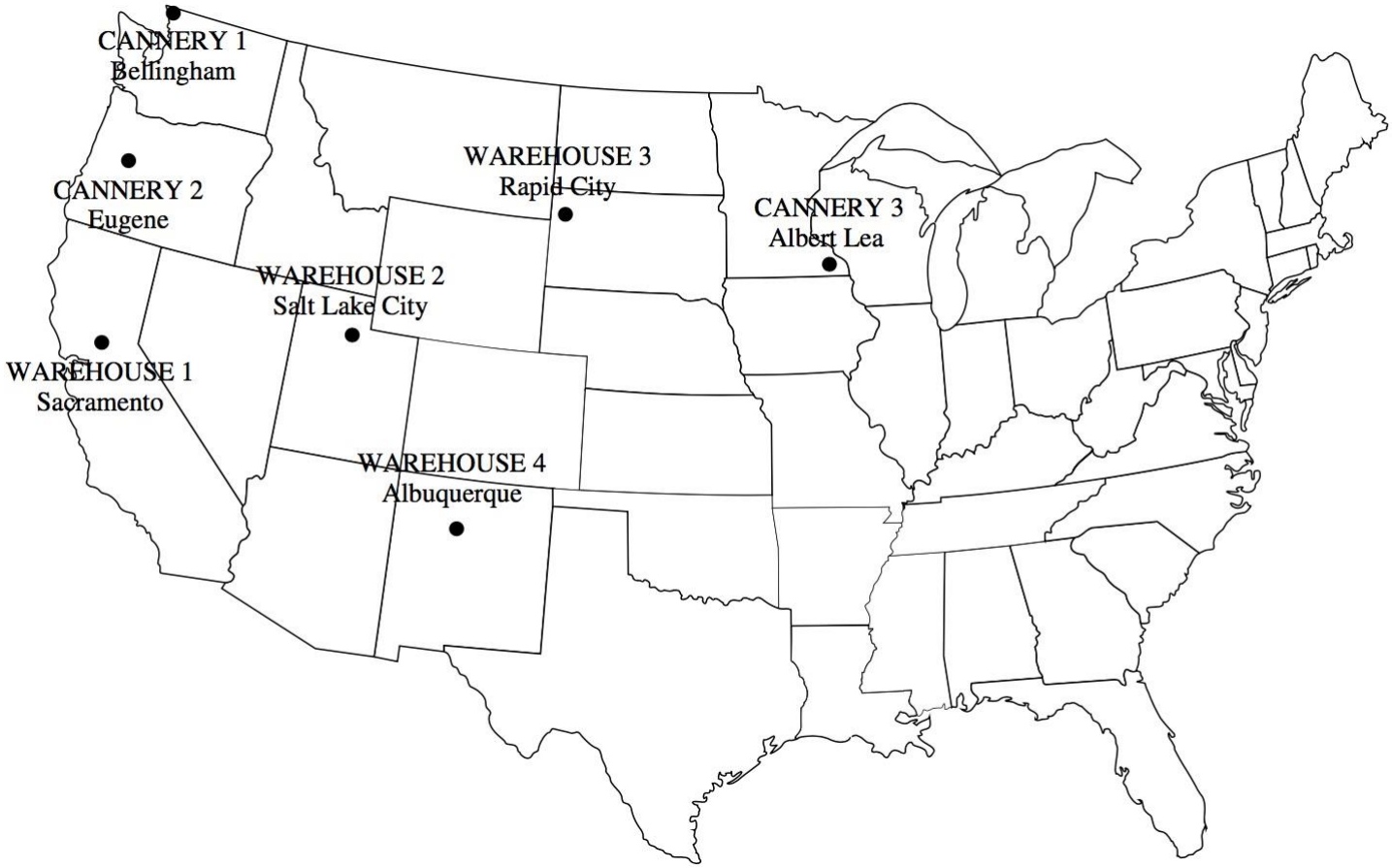
```
int N=...;
int B=...;
int A=...;
int a[1..100]=...;
range block = 1..B;

float F = (N+sum (i in block) a[i])/(A*B);

dvar int x[block];
dvar float uplus[block];
dvar float uminus[block];

minimize sum (i in block) (uplus[i]+uminus[i]);

subject to {
    sum (i in block) x[i] == N;

    forall (i in block)
      a[i] + x[i] - A*F == uplus[i] - uminus[i];

    forall (i in block){
        x[i]>=0;
        uplus[i]>=0;
        uminus[i]>=0;
    }
}
```

Transportation Problem

Transportation Problem

One of the main products of the P & T COMPANY is canned peas. The peas are prepared at three canneries (near Bellingham, Washington; Eugene, Oregon; and Albert Lea, Minnesota) and then shipped by truck to four distributing warehouses in the western United States (Sacramento, California; Salt Lake City, Utah; Rapid City, South Dakota; and Albuquerque, New Mexico). Because the shipping costs are a major expense, management is initiating a study to reduce them as much as possible. For the upcoming season, an estimate has been made of the output from each cannery, and each warehouse has been allocated a certain amount from the total supply of peas. This information (in units of truckloads), along with the shipping cost per truckload for each cannery-warehouse combination, is given in Table 8.2. Thus, there are a total of 300 truck- loads to be shipped. The problem now is to determine which plan for assigning these shipments to the various cannery-warehouse combinations would *minimize the total shipping cost.*

Transportation Problem – Layout



Transportation Problem – Table 8.2

TABLE 8.2 Shipping data for P & T Co.

		Shipping Cost ($) per Truckload				
		Warehouse				
		1	**2**	**3**	**4**	**Output**
Cannery	1	464	513	654	867	75
	2	352	416	690	791	125
	3	995	682	388	685	100
Allocation		80	65	70	85	

Assignment #2

- Draw a network representation of the problem (hint: canneries are nodes on the left, warehouses are nodes on the right, arcs represent routing, each node has a demand).
- Formulate the problem mathematically.
- Implement the problem and report the optimal solution.
- Update your model and code to be generic (i.e. the model can solve the transportation problem for any size).
- Solve the problem for a large instance (file `Large_Instance.txt`).
- What methodology will be used by CPLEX to find the optimal solution?

Assignment #2

- Write a short report containing your answers
- Individual work or in group (max 3 students)
- Send your reports to virginie.lurkin(at)epfl.ch
- Send your reports by 8:00PM next Monday

References

- https://www.ibm.com/developerworks/community/blogs/jfp/entry/CPLEX_Is_Free_For_Students?lang=en
- http://www.chkwon.net/julia/book/juliabook-preview.pdf
- http://jump.readthedocs.io/en/latest/refmodel.html
- http://jump.readthedocs.io/en/latest/refvariable.html
- http://jump.readthedocs.io/en/latest/quickstart.html#objective-and-constraints