Challenge June – 2021

Where is gold ?

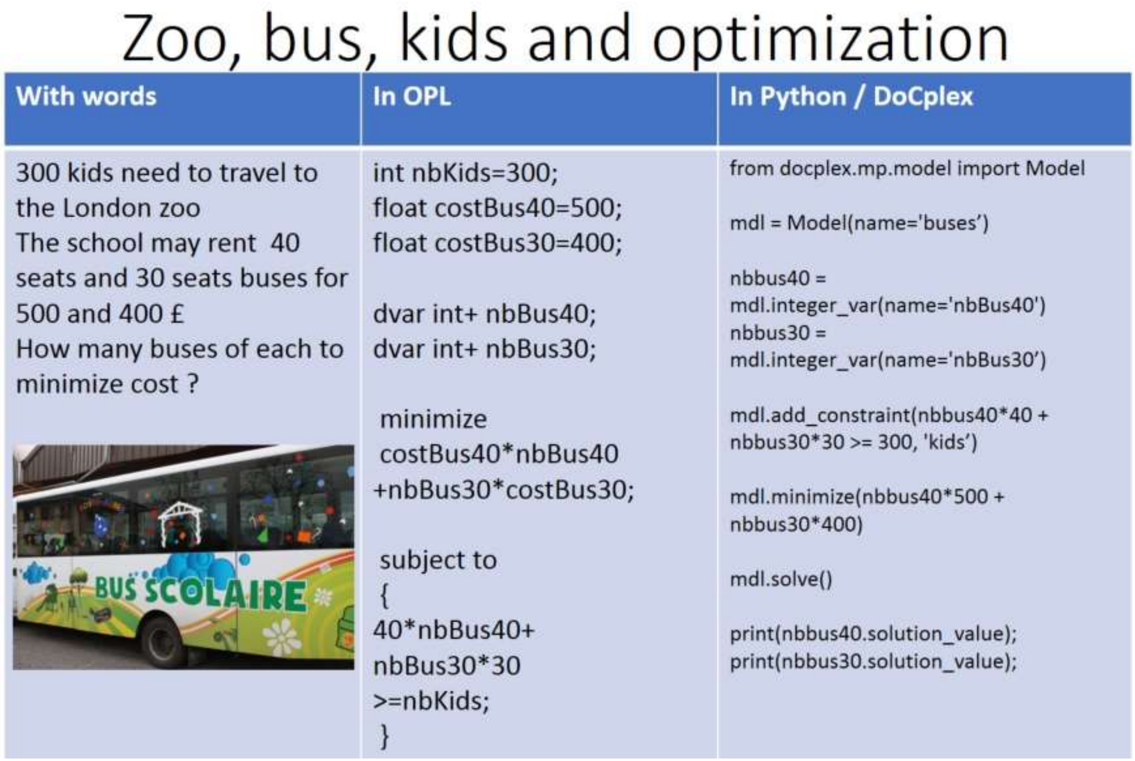
A solution with OPL CPLEX
by Alex Fleischer afleischer@fr.ibm.com

OPL (Optimization Programming Language) is an abstract modeling language that helps model easily optimization problems that can be solved both with IBM CPLEX linear programming and IBM CPLEX constraint programming CPOptimizer (CPO)

Let us remember that with ILOG (French company bought by IBM in 2009) we had two kind of decision engines:

A) Rule based (JRules, ODM)
B) Optimization based (CPLEX)

Here a small example of a tiny optimization model, in English, OPL and Python



Zoo, bus, kids and optimization

With words	In OPL	In Python / DoCplex

300 kids need to travel to the London zoo
The school may rent 40 seats and 30 seats buses for 500 and 400 £
How many buses of each to minimize cost ?
	int nbKids=300;
	float costBus40=500;
	float costBus30=400;
	
	dvar int+ nbBus40;
	dvar int+ nbBus30;
	
	minimize
	costBus40*nbBus40
	+nbBus30*costBus30;
	
	subject to
	{
	40*nbBus40+
	nbBus30*30
	>=nbKids;
	}
		from docplex.mp.model import Model
		
		mdl = Model(name='buses')
		
		nbbus40 =
		mdl.integer_var(name='nbBus40')
		nbbus30 =
		mdl.integer_var(name='nbBus30')
		
		mdl.add_constraint(nbbus40*40 +
		nbbus30*30 >= 300, 'kids')
		
		mdl.minimize(nbbus40*500 +
		nbbus30*400)
		
		mdl.solve()
		
		print(nbbus40.solution_value);
		print(nbbus30.solution_value);

We can call CPLEX from many languages (C,C++,.NET,Java,Python …) but using OPL leads to a clear frontier between the model and the code that will embed the model. (Not far from Decision Model and Notation (DMN) principle: “The notation is designed to be readable by business and IT users alike. This enables various groups to effectively collaborate in defining a decision model”)

Now let’s move to the June 2021 DMC challenge:

There are three boxes, but only one of them has gold inside. Additionally, each box has a message printed on it. One of these messages is true and the other two are lies.
The first box says, “Gold is in this box”.
The second box says, “Gold is not in this box”.
The third box says, “Gold is in not in Box 1”.

Which box contains the gold?

In OPL CPLEX no need to be very clever, we need to translate the constraints.

```
{string} boxes={"box1","box2","box3"};

dvar boolean gold[boxes];

subject to
{
  // gold is in one box only
  sum( b in boxes) gold[b]==1;

  // One of these messages is true and the other two are lies.
  // The first box says, “Gold is in this box”.
  // The second box says, “Gold is not in this box”.
  // The third box says, “Gold is in not in Box 1”.
  (gold["box1"]==1) + !(gold["box2"]==1) + (gold["box1"]!=1)==1;


}

string whereIsGold=first({b | b in boxes:gold[b]==1});
execute
{
  writeln("where is gold ? ",whereIsGold);
}
```

Which gives

```
where is gold ? box2
```

So gold is in box 2.

And then to prove that gold cannot be anywhere else, you may add

```
gold["box2"]!=1;
```

And check that then there is no solution.

Within CPLEX we have 2 tools:

Linear programming and constraint programming. Here we used Linear Programming. But if we add “using CP;” in the model then constraint programming will be used and that works fine too.

[Making Decision Optimization Simple](https://www.linkedin.com/pulse/making-decision-optimization-simple-alex-fleischer/) : https://www.linkedin.com/pulse/making-decision-optimization-simple-alex-fleischer/