June 2019 DMC Challenge

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)

May 2019 DMC challenge was exactly a coloring problem that is part of the OPL CPLEX examples :

https://www.ibm.com/support/knowledgecenter/SSSA5P_12.7.0/ilog.odms.ide.help/examples/html/opl/color/color.mod.html

The June 2019 DMC challenge is a variation so let me adapt to the new constraints.

The new requirement is:

> So, some neighboring counties may have the same colors but there is a relative cost for such violations:
>
> France – Luxembourg: $257
> Luxembourg – Germany: $904
> Luxembourg – Belgium: $568
>
> We need to find a solution that minimizes the total violation cost.

Let me offer 2 methods:

Method 1: Keep using CPO (constraint programming) and add an objective which is minimize relaxation

```
using CP;

range r = 0..2; // 3 colors insread of 4

string Names[r] = ["blue", "white", "yellow"];

dvar int Belgium in r;
dvar int Denmark in r;
dvar int France in r;
dvar int Germany  in r;
dvar int Luxembourg in r;
dvar int Netherlands in r;

// minimize relaxation
minimize 257*(France == Luxembourg)+904*(Germany == Luxembourg)+568*(Belgium ==
Luxembourg);

subject to {
   Belgium != France;
   Belgium != Germany;
   Belgium != Netherlands;
   //Belgium != Luxembourg;
   Denmark != Germany;
   France != Germany;
   //France != Luxembourg;
   //Germany != Luxembourg;
   Germany != Netherlands;
}

execute {
   writeln("Belgium:     ", Names[Belgium]);
   writeln("Denmark:     ", Names[Denmark]);
   writeln("France:      ", Names[France]);
   writeln("Germany:     ", Names[Germany]);
   writeln("Luxembourg:  ", Names[Luxembourg]);
   writeln("Netherlands: ", Names[Netherlands]);
}

tuple resultT {
       string name;
       string value;
};
{resultT} solution = {};
execute{
   solution.add("Belgium", Names[Belgium]);
   solution.add("Denmark", Names[Denmark]);
   solution.add("France", Names[France]);
   solution.add("Germany", Names[Germany]);
   solution.add("Luxembourg", Names[Luxembourg]);
   solution.add("Netherlands", Names[Netherlands]);
   writeln(solution);
}
```

Which gives

```
// solution with objective 257
Belgium:     yellow
```

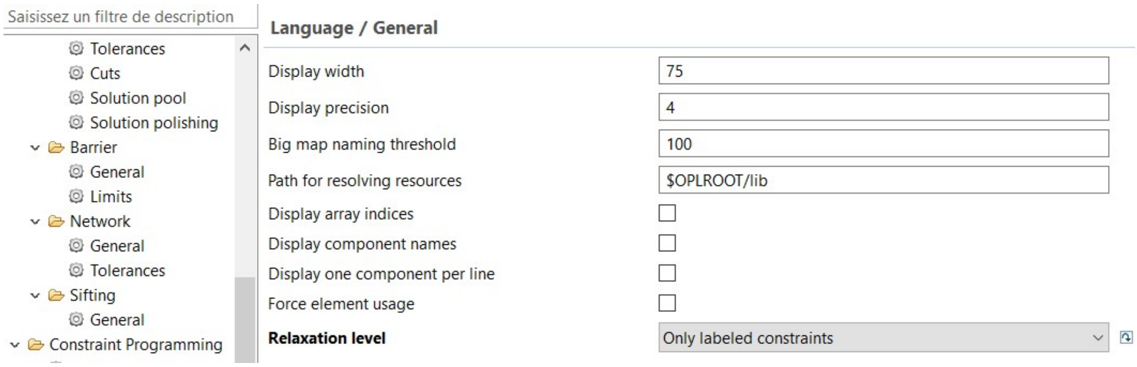
```
Denmark:     blue
France:      blue
Germany:     white
Luxembourg:  blue
Netherlands: blue
 {<"Belgium" "yellow"> <"Denmark" "blue"> <"France" "blue"> <"Germany" "white">
     <"Luxembourg" "blue"> <"Netherlands" "blue">}
```

Method 2: use CPLEX MIP relaxation tool

```
range r = 0..2; // 3 colors instead of 4

string Names[r] = ["blue", "white", "yellow"];

dvar int Belgium in r;
dvar int Denmark in r;
dvar int France in r;
dvar int Germany  in r;
dvar int Luxembourg in r;
dvar int Netherlands in r;


subject to {
   Belgium != France;
   Belgium != Germany;
   Belgium != Netherlands;
   ct1:568*(Belgium==Luxembourg)<=0;;
   Denmark != Germany;
   France != Germany;
   ct2:257*(France==Luxembourg)<=0;
   ct3:904*(Germany==Luxembourg)<=0;;
   Germany != Netherlands;
}

execute {
   writeln("Belgium:     ", Names[Belgium]);
   writeln("Denmark:     ", Names[Denmark]);
   writeln("France:      ", Names[France]);
   writeln("Germany:     ", Names[Germany]);
   writeln("Luxembourg:  ", Names[Luxembourg]);
   writeln("Netherlands: ", Names[Netherlands]);
}

tuple resultT {
      string name;
      string value;
};
{resultT} solution = {};
execute{
   solution.add("Belgium", Names[Belgium]);
   solution.add("Denmark", Names[Denmark]);
   solution.add("France", Names[France]);
   solution.add("Germany", Names[Germany]);
   solution.add("Luxembourg", Names[Luxembourg]);
   solution.add("Netherlands", Names[Netherlands]);
   writeln(solution);
}
```

Labeling a constraint means CPLEX can relax it.

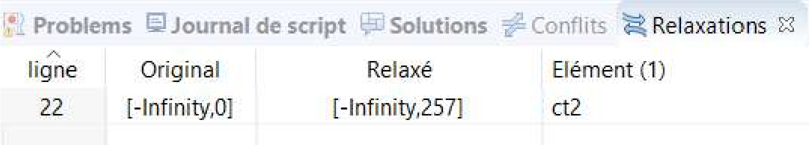
And then we tell CPLEX to relax only constraints and not decision variables:



Language / General

Display width	75
Display precision	4
Big map naming threshold	100
Path for resolving resources	$OPLROOT/lib
Display array indices	□
Display component names	□
Display one component per line	□
Force element usage	□
Relaxation level	Only labeled constraints

And then we get in the relaxation tab



ligne	Original	Relaxé	Elément (1)
22	[-Infinity,0]	[-Infinity,257]	ct2

Which means ct2 was relaxed (France != Luxembourg)

And in the scripting tab we get:

```
// solution (feasible relaxed sum of infeasibilities) with objective 257

Belgium:     blue
Denmark:     blue
France:      yellow
Germany:     white
Luxembourg:  yellow
Netherlands: yellow
 {<"Belgium" "blue"> <"Denmark" "blue"> <"France" "yellow"> <"Germany" "white">
     <"Luxembourg" "yellow"> <"Netherlands" "yellow">}
```