

TP 3-4

Problem. The goal of this assignment is to use integer solvers to solve the puzzle game Sudoku. We will be using **Google OR-Tools** open source suite.

1. Download, Compile and Run Sample Program

1. From the course website, download the file `ortools-Debian.zip` and unzip it.
2. The source file for this project is `ortools-Debian/sudoku.cc`. This is the file in which you have to write all your code. For the moment, it contains the following sample integer program:

```

maximize           $2x_1 + 3x_2 + 4x_3 + 2x_4$ 

with constraints

                 $x_1 + 2x_2 + 4x_3 + 2x_4 = 20$ 

                 $x_1 - 3x_2 + x_3 - x_4 \geq 10$ 

                 $0 \leq x_1, x_2, x_3, x_4 \leq 10$  is an integer.

```

3. Go through the code **carefully** to see the syntax for how to use the solver for specifying and solving the above linear program. Here is a summary of the correspondence between linear program and its implementation using **orTools**.

LINEAR PROGRAM	ORTOOLS IMPLEMENTATION
∞	<code>const double infinity = solver.infinity();</code>
variable <code>float</code> $-2.0 \leq x \leq 8.0$	<code>MPVariable *x = solver.MakeNumVar(-2.0f, 8.0f, "");</code>
variable <code>float</code> $-6.0 \leq y$	<code>MPVariable *y = solver.MakeNumVar(-6.0f, solver.infinity(), "");</code>
variable <code>int</code> $z \leq 4$	<code>MPVariable *z = solver.MakeIntVar(-solver.infinity(), 4, "");</code>
O.F.: Maximize $5x - 4y + 3z$	<code>MPObjecive* const objective = solver.MutableObjective();</code> <code>objective->SetCoefficient(x, 5.0f);</code> <code>objective->SetCoefficient(y, -4.0f);</code> <code>objective->SetCoefficient(z, 3.0f);</code> <code>objective->SetMaximization();</code>

Solve the LP	<code>const MPSolver::ResultStatus result_status = solver.Solve();</code>
Value of the O.F.	<code>objective->Value()</code>

- Go to the `ortools-Debian/` directory, and then compile this file by typing

```
make run SOURCE=sudoku.cc
```

If it compiles properly¹, then the executable will be created at the location `ortools-Debian/bin/sudoku`.

- The program takes in a text file Sudoku puzzle. Right now it does not matter which file you give it, as the main code is empty and you have to write it. Run the executable as

```
bin/sudoku input1.txt
```

It should output the following solution to the above integer program:

```
x1 = 10
x2 = 0
x3 = 2
x1 = 1
```

- Your goal is to modify this file to write the integer program to solve any Sudoku puzzle. You can find out about the rules of this game [here](#).
- The Sudoku puzzle will be given as an input file. For example, below left shows an order 3 Sudoku puzzle (so with $3 \times 3 = 9$ rows and columns), while the right side shows the corresponding input to the program (this is `input2.txt` puzzle).

7						4		
	2			7			8	
		3			8			9
			5			3		
	6			2			9	
		1			7			6
			3			9		
	3			4			6	
		9			1			5

```
3
7 0 0 0 0 0 4 0 0
0 2 0 0 7 0 0 8 0
0 0 3 0 0 8 0 0 9
0 0 0 5 0 0 3 0 0
0 6 0 0 2 0 0 9 0
0 0 1 0 0 7 0 0 6
0 0 0 3 0 0 9 0 0
0 3 0 0 4 0 0 6 0
0 0 9 0 0 1 0 0 5
```

- You are also given `input1.txt`, `input2.txt`, `input3.txt`, `input4.txt`, `input5.txt`—five Sudoku puzzle text files to try your program on. You can also try many other Sudoku puzzles from the web once you have completed your program.

¹If you are using Debian on your own computer, you will need to install additional packages with `sudo apt-get -y install build-essential zlib1g-dev`.