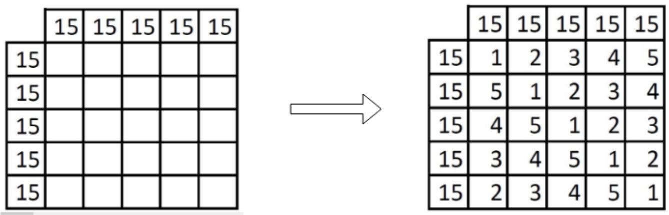
C++ Project - Solving Kakuro

Elie ABI HANNA DAHER and Bilal EL CHAMI
GitHub repository : github.com/elieahd/kakuro

***Abstract*—This document illustrate the work done for an academic project that consists of solving a Kakuro Grid as a contraint satisfaction problem (CSP) in C++.**

I. Introduction

Kakuro consists in filling a grid with numbers that sum up to a certain values for each column and row. Each cell needs to be filled with a value between 1 and 9.
The following example represents a 5x5 grid where the sum of each row has to be 15 and the sum of each column has to be 15.



	15	15	15	15	15
15					
15					
15					
15					
15					

	15	15	15	15	15
15	1	2	3	4	5
15	5	1	2	3	4
15	4	5	1	2	3
15	3	4	5	1	2
15	2	3	4	5	1

Fig. 1. Example of solving a 5x5 kakuro gird

II. Implementation

The following diagram reperesents the class diagrams of our project

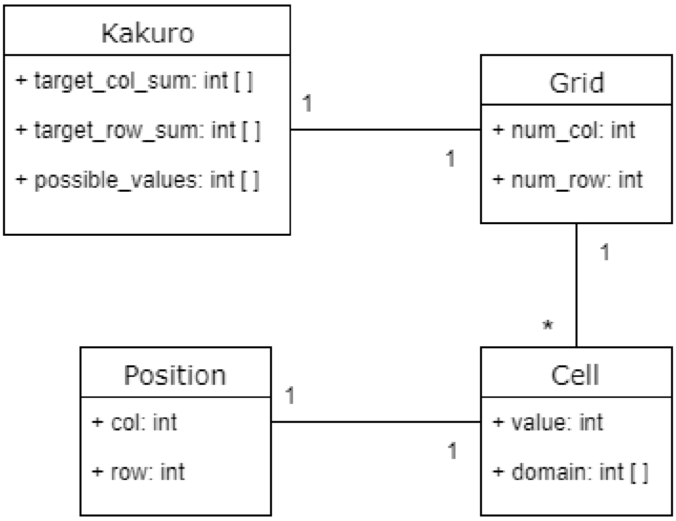
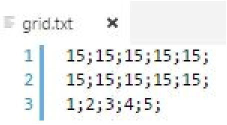


Fig. 2. Class diagram

The initial grid of the kakuro will be present in a file, which will contain the possible values of a cell and the target sum of each column and row. So for the grid in the figure **??** we will have the following file



```
15;15;15;15;15;
15;15;15;15;15;
1;2;3;4;5;
```

Fig. 3. File input

III. Solving - Forward Checking

Algorithm 1 Forward checking

```
procedure ForwardChecking
    if no free variable then
        return true
    end if
    choose a free variable var
    for each value $v$ in $var.domain$ do
        assign value to var
        update the domains of the free variable
        if no domain is empty or inconsistent then
            if ForwardChecking() then
                return true
            end if
        end if
    end for
    return false
end procedure
```

IV. Monte Carlo Kakuro

V. Experimentals - Simulation

TABLE I
Simulation Parameters

3x3 grid	0.3 s
5x5 grid	0.3 s
10x10 grid	0.3 s

References

[1] T. Cazenave. *Monte-Carlo Kakuro*