

Programación con restricciones

Gonzalo Martínez, Miguel González, Ignacio Pedrero

1 Programación con restricciones

La programación con restricciones es una técnica general para resolver problemas de restricciones. Se basa en la idea de que un problema se puede describir mediante un conjunto de restricciones y un dominio para las variables, y luego se busca encontrar una asignación de valores a las variables que cumpla con todas las restricciones. Estos problemas constan de variables y restricciones, y una vez estas han sido definidas, el objetivo es buscar soluciones que cumplan tanto con las variables como con las restricciones.

2 Satisfabilidad y Optimización

En este artículo, veremos cómo usar CP de dos maneras diferentes: -Satisfacibilidad: el objetivo es encontrar una o varias soluciones factibles (es decir , soluciones que respeten nuestras restricciones) al reducir un gran conjunto de soluciones potenciales; -Optimización : el objetivo es encontrar la mejor solución factible de acuerdo con una función objetivo, al igual que la Programación Lineal (PL).

3 Ejemplo de satisfabilidad

Hemos ido a visitar a 5 clarividentes, o adivinos, para que intentaran predecir el número de la lotería. Pero no nos lo van a poner tan fácil, ya que cada uno sólo nos dice una característica de este número.

- Clarividente 1: El número de la lotería es múltiplo de 7 y de 13.
- Clarividente 2: El primer dígito del número de la lotería es 5.
- Clarividente 3: El segundo dígito del número de la lotería es menor que 6 y mayor que 1.
- Clarividente 4: La suma del número de la lotería y de mi dni es menor que el dni de mi hermano pero mayor que el dni de mi primo.

- Clarividente 5: La raíz del número de la lotería es menor que mi código postal

Sabiendo todo esto, ahora nos toca crear un código para intentar sacar el número de la lotería.

Lo primero que debemos hacer es importar y crear el modelo y el solucionador de CP-SAT.

```
from ortools.sat.python import cp_model

model = cp_model.CpModel()
solver = cp_model.CpSolver()
```

A continuación, definimos la variable, y como sabemos que el número de la lotería tiene 5 cifras, el límite inferior será 0, y el superior será 99999.

```
numero = model.NewIntVar(0, 99999, 'numero')
```

El siguiente paso es declarar las restricciones:

1. Es múltiplo de 7 y de 13

```
# es múltiplo de 7 y de 13
model.AddModuloEquality(0, numero, 7)
model.AddModuloEquality(0, numero, 13)
```

2. El primer dígito es 5

```
# el primer dígito es 5
model.AddDivisionEquality(5, numero, 10000)
```

3. El segundo dígito es menor que 6 y mayor que 1

```
# el segundo dígito es menor que 6 y mayor que 1
model.Add(numero < 56000)
model.Add(numero > 51000)
```

4. La suma del número de la lotería y mi dni es menor que el dni de mi hermano menor pero mayor que el dni de mi primo. Sabemos que nuestro dni es 03876543, el de nuestro hermano es 03930051 y el de nuestro primo es 03928595.

```
# la suma del numero y mi dni es menor que el dni de
# mi hermano menor pero mayor que el dni de mi primo
# nuestro dni: 03876543
# dni hermano menor: 03930051
# dni primo: 03928595
model.Add(numero+3876543 < 3930146)
model.Add(numero+3876543 > 3928595)
```

5. la raíz del número de la lotería es menor que mi código postal. Sabemos que nuestro código postal es 229.

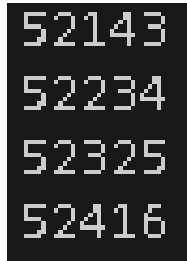
```
# la raíz del numero es menor que mi código postal
# código postal: 229
model.Add(numero <= 229*229)
```

Ahora que nuestro modelo ya está completo, podemos resolverlo con OR-Tools, y lo haremos de forma que nos imprima todas las soluciones posibles.

```
class PrintSolutions(cp_model.CpSolverSolutionCallback):
    """Callback to print every solution."""
    def __init__(self, variable):
        cp_model.CpSolverSolutionCallback.__init__(self)
        self.__variable = variable
    def on_solution_callback(self):
        print(self.Value(self.__variable))

solution_printer = PrintSolutions(numero)
solver.parameters.enumerate_all_solutions = True
status = solver.Solve(model, solution_printer)
```

Este es el resultado final:



52143
52234
52325
52416

Ahora, ya sabemos que el número de la lotería va a ser uno de esos 4 números.

4 Diferencia entre CP y LP

LP se utiliza para optimizar funciones lineales sujetas a restricciones lineales, y su objetivo es maximizar o minimizar la función objetivo. Mientras que, CP se utiliza para resolver problemas de decisión sujetos a restricciones más generales, que no necesariamente son lineales, es más utilizado para problemas de decisiones secuenciales o combinatorias.