

# COMPLEJIDAD Y OPTIMIZACIÓN

PROYECTO FINAL - PRIMERA PARTE

## **Autores:**

Juan David Pino - 1610460  
*juan.david.pino@correounivalle.edu.co*

Julian Andrés Castaño - 1625743  
*castano.julian@correounivalle.edu.co*

Bryan Steven Biojó - 1629366  
*bryan.biojo@correounivalle.edu.co*

## **Docente:**

Robinson Andrey Duque, Ph.D

UNIVERSIDAD DEL VALLE  
FACULTAD DE INGENIERÍA  
CALI - VALLE  
OCTUBRE 23 DE 2020

# 1. Informe de Rendimiento

## 1.1. Descripción del solver SAT utilizado

Para la verificación de las satisfactibilidad de las diferentes instancias SAT, utilizamos el solver **Glucose 3**, ya que es uno de los más estables y viene incluido en la librería **PySAT** de Python 3

**Sobre PySAT:** PySAT es un conjunto de herramientas de Python que tiene como objetivo proporcionar una interfaz simple y unificada para una serie de solucionadores para el **Problema de Satisfactibilidad Booleana (SAT)** de última generación, así como para una variedad de codificaciones de cardinalidad y pseudobooleanas. PySAT puede ser útil para resolver problemas en NP pero también más allá de NP [1].

## 1.2. Gráfico comparativo SAT vs X-SAT

Para la realización de este gráfico se tomaron 20 instancias SAT y se redujeron a 14-SAT. Los datos de los tiempos (en segundos) se tomaron con la librería **Time** de Python 3, mientras que los datos de los tiempos se mapearon en un archivo CSV para ser utilizados como un dataframe a través de las librerías **NumPy** y **Pandas** (también de Python 3). Se distingue entre instancias satisfactibles *azul* e insatisfactibles *rojo*.

### 1.2.1. Tiempos de las instancias en SAT y 14-SAT

	instance	sat_time	xsat_time	clause_result	color
0	0	0.005330	0.006259	unsatisfiable	red
1	1	0.004765	0.004722	unsatisfiable	red
2	2	0.002558	0.002436	satisfiable	blue
3	3	0.002428	0.003036	satisfiable	blue
4	4	0.002508	0.002452	satisfiable	blue

Figura 1: Tiempos de SAT y 14-SAT (CSV)

### 1.2.2. Gráfica de tiempos de las instancias en SAT y X-SAT (14-SAT)

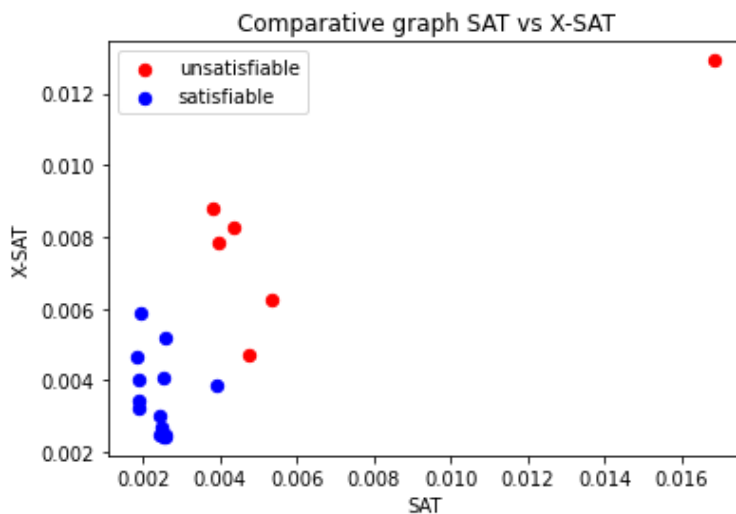


Figura 2: Gráfico de dispersión de SAT y X-SAT (14-SAT)

## 2. Conclusiones

Para el conjunto de 20 instancias en 14-SAT (14 satisfactibles y 6 insatisfactibles) se puede observar que no hay una diferencia tan marcada en los tiempos de comprobación con el solver Glucose 3, ni siquiera se tarda 1 segundo. También, se observa que en la mayoría de los casos las instancias se resolvieron más rápido en SAT que en 14-SAT, salvo una instancia insatisfactible que tardó demasiado tiempos en ambos (la que está más a la derecha en la gráfica).

### 3. Anexos

El código fuente y los demás archivos utilizados para la realización de este proyecto, se pueden encontrar en el siguiente repositorio de **GitHub** (<https://github.com/AndresDFX/XSAT-Red>)

## Referencias

- [1] J. M. Alexey Ignatiev, Antonio Morgado, “PySAT: A Python toolkit for prototyping with SAT oracles,” in *SAT*, pp. 428–437, 2018.