

The optimization of a Boolean functions is very important to save the hardware required to implement a design. The main Logic Synthesis challenges are [1]:

1. Deal with large input functions and large set of Don't Cares (DC).
2. Optimize dedicated parts of large circuits.

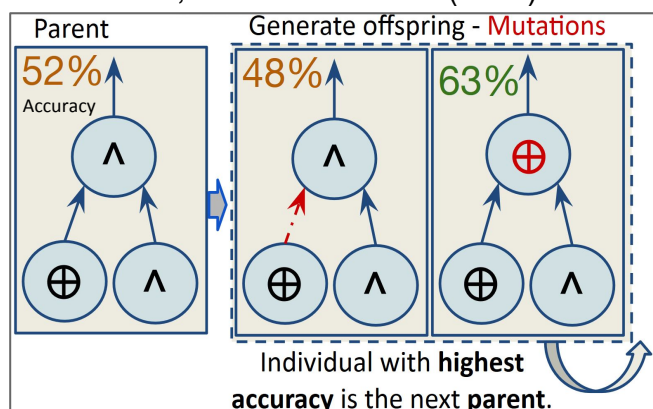
[1] Chang, Kai-Hui, et al. (TODAES) 2010: 1-24.

**Goal: Explore Machine Learning techniques to perform Logic Optimization with acceptable accuracy.**

## Cartesian Genetic Programming (CGP) Optimization Flow

Cartesian Genetic Programming (CGP) evolves chromosomes which represent functioning programs [2], comprising of **three function types**:

AND, XOR and Inverter (XAIg).



[2] Manazir, Abdul, and Khalid Raza. " (CSUR) 51.6 (2019).

❖ For an **incomplete Truth Table** provide a **circuit with high accuracy** with relation to the incomplete Boolean Function.

❖ **C++** Implementation (github available - <https://github.com/gudeh/IWLS2020Final>).

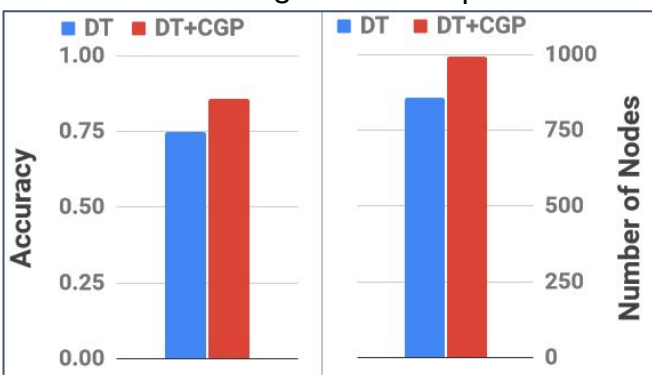
❖ **Benchmarks**: provided by the IWLS 2020 Contest.

### Results:

❖ CGP was able to improve some DT's solutions.

❖ **85.65%** accuracy achieved.

➤ Main challenge: runtime optimization

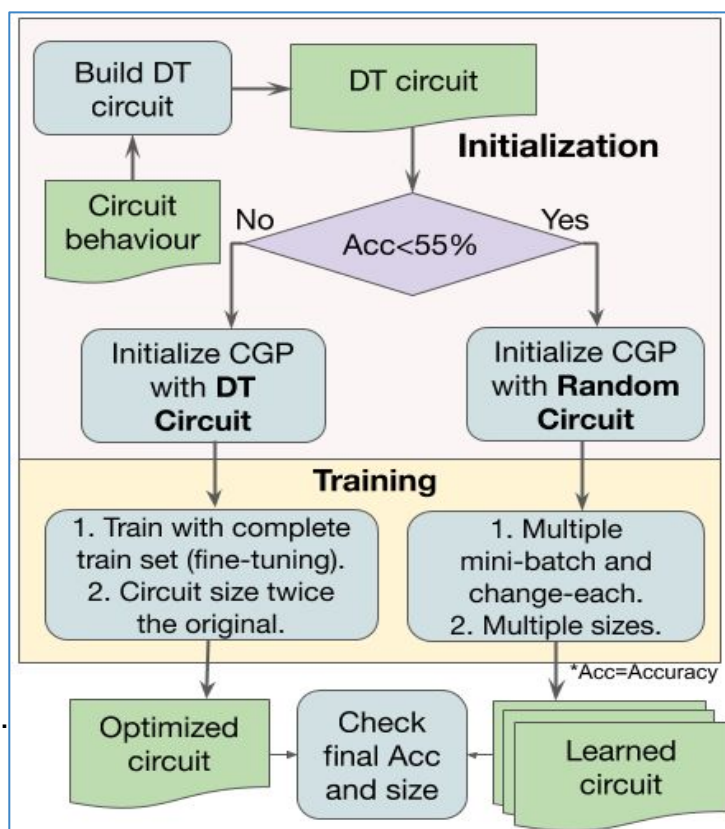


The CGP flow may be initialized with:

❖ **Random circuit**: Learning with 80% data set.

❖ **Decision-Tree (DT) circuit**: Fine-tuning the solution accuracy.

**Output**: AIG (And-Inverter Graph) format.



### Next Steps:

Adopt CGP flow to area and energy optimization.

Explore CGP flow to automatic generation of Approximate Circuits.