1. Introduction:

A python code is written to extract the gates and information about the interconnects. The gates are considered as nodes and interconnects as edges.

2. Inputs:

The file path of a benchmark file which is a netlist file is given as input. Nodes and edges will be extracted from the benchmark file and the graph will be plotted. Then KL algorithm will be performed.

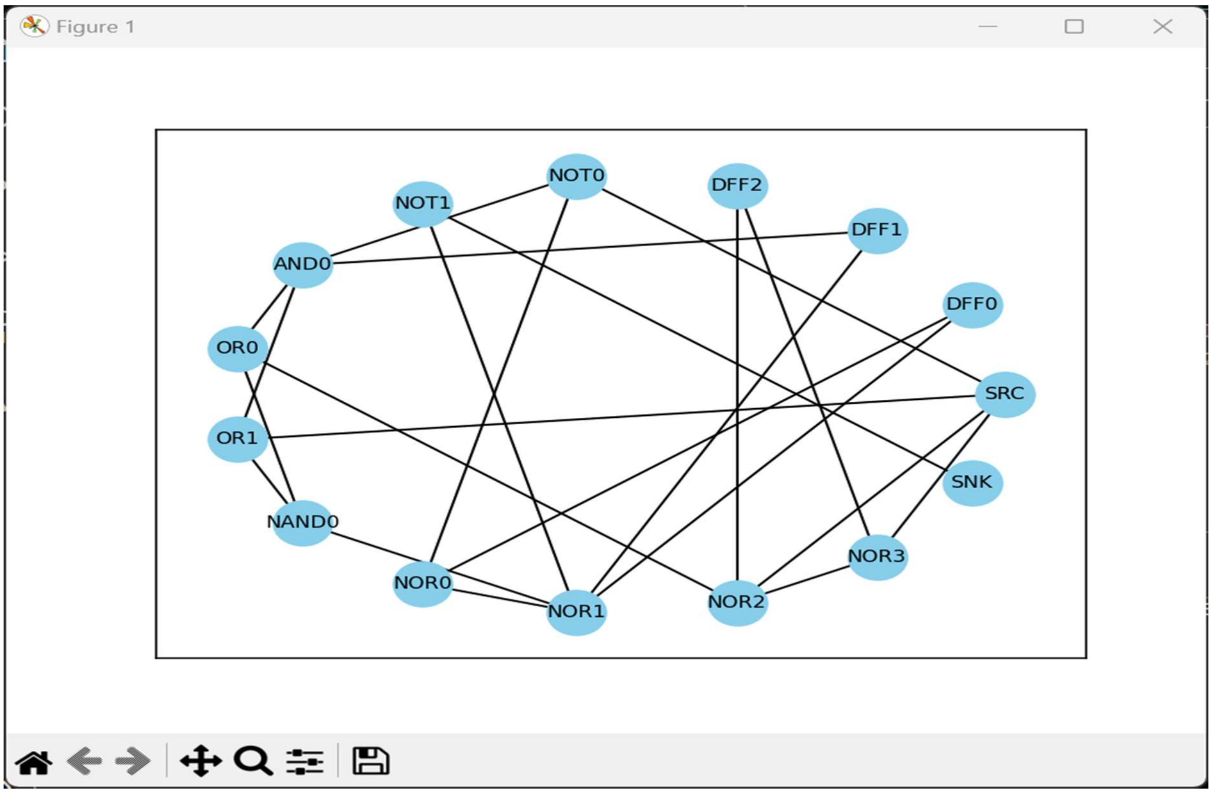
E.g.

```
inputs, outputs, gate_io =
netlist_graph.extract_gate_io(r"C:\Users\akhil\OneDrive\Documents\SEMESTER_2\VDA\Assignment_1\
ISCAS89\s27.bench")
```

If the total number of nodes is odd, a dummy node is added as KL algorithm works on partitions of equal sizes.

3. Outputs:

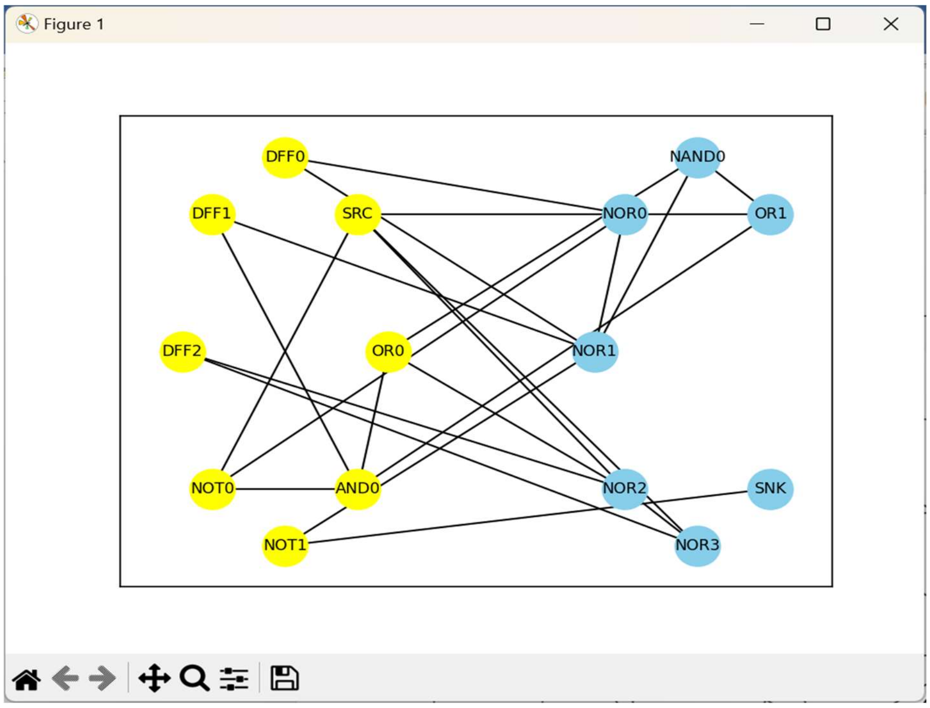
Undirected Graph: -



Partition List and Cut Size before applying KL Algorithm: -

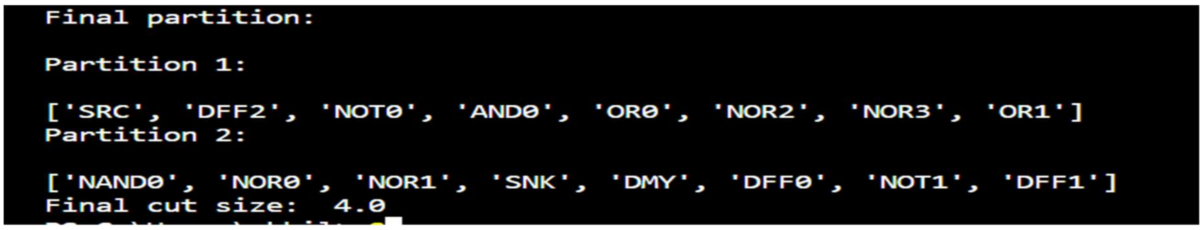
```
Starting partitionin of the circuit:
Initial partition

Partition 1:

['SRC', 'DFF0', 'DFF1', 'DFF2', 'NOT0', 'NOT1', 'AND0', 'OR0']
Partition 2:

['OR1', 'NAND0', 'NOR0', 'NOR1', 'NOR2', 'NOR3', 'SNK', 'DMY']
Initial cut size:  14.0
```

Partitioned Graph before applying KL Algorithm: -



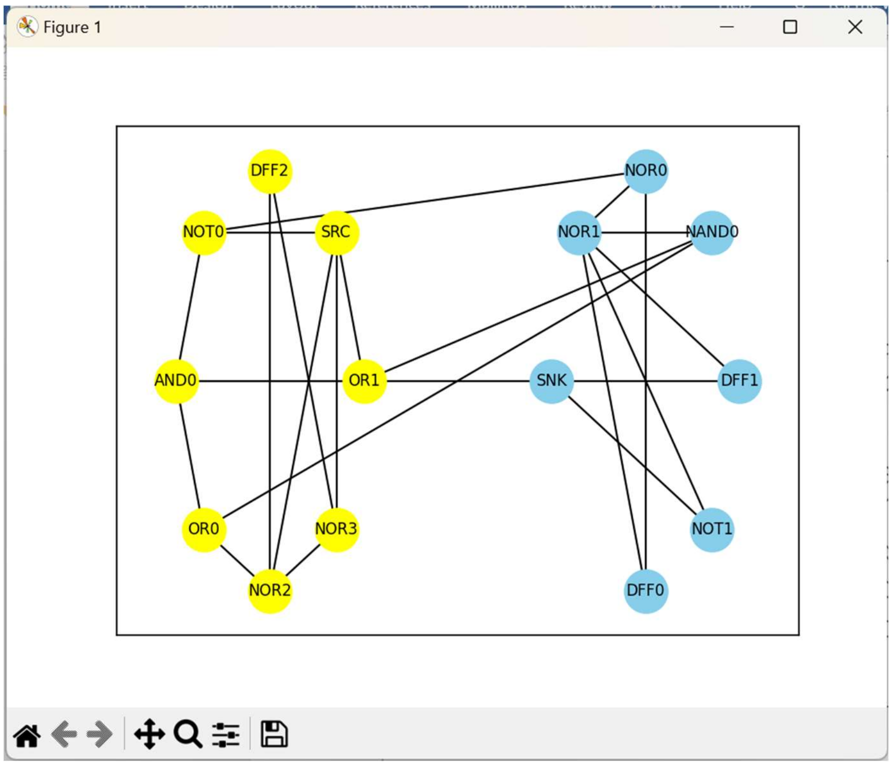
Note: Dummy node is not plotted in the respective partition.

Partition List and Cut Size after applying KL Algorithm: -



```
  Final partition:

  Partition 1:

  ['SRC', 'DFF2', 'NOT0', 'AND0', 'OR0', 'NOR2', 'NOR3', 'OR1']
  Partition 2:

  ['NAND0', 'NOR0', 'NOR1', 'SNK', 'DMY', 'DFF0', 'NOT1', 'DFF1']
  Final cut size:  4.0
```

Partitioned Graph after applying KL Algorithm: -



Note: Dummy node is not plotted in the respective partition.