Auto ILP Scheduling

Zander and Aidan

Start of Demonstration

- ChatGPT

Can you generate a random data flow graph for me in weighted edgelist format with no cycles and 25 nodes, where the weights are integers?

```
python3.11 autoILP.py -l 20 -a 50 -g 25node.edgelist
```

Big Idea

- Input: DFG in weighted edge-list format
- Output: Output from GLPK with automatic scheduling results based on latency minimization, memory minimization, or Pareto-Optimal analysis
- Command-line Argument:
    - Latency Minimization:
        - python autoILP.py -a (memory constraint) -g (filename)
    - Memory Minimization:
        - python autoILP.py -l (latency constraint) -g (filename)
    - Pareto-Optimal Analysis:
        - python autoILP.py -l (latency constraint) -a (memory constraint) -g (filename)
        - Will run a series of loops over all possible latency and memory constraints below the given constraints and display a graph of all feasible solutions

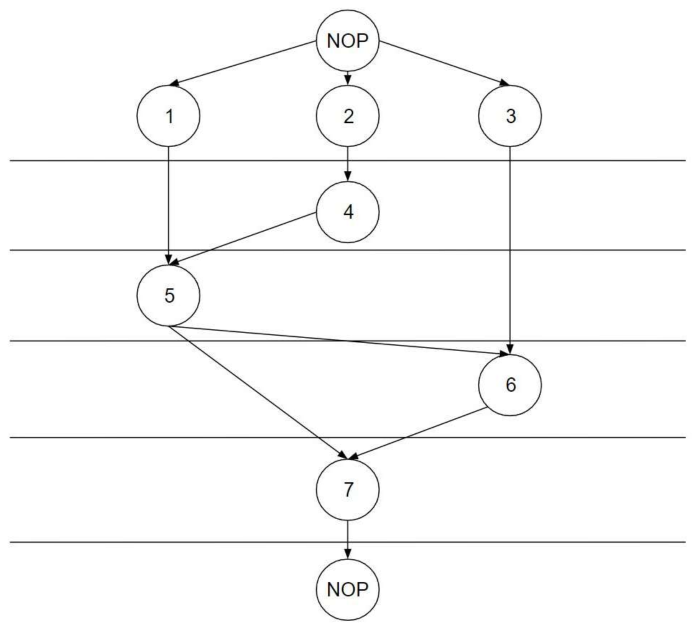
Design Flow

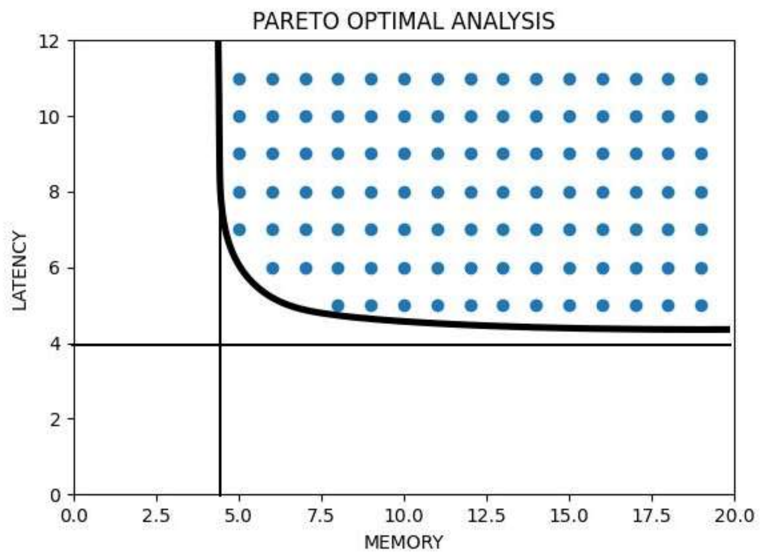
1) Determine what type of minimization is requested
2) Create a directed graph from the weighted edge-list
3) Find ASAP and ALAP representations of each node
4) Create the ILP variables
5) Generate execution, dependency, and resource constraints
6) Create ILP file and use GLPK to perform minimization
7) Loop as needed to generate new constraints based on memory and/or latency optimization

Detailed Implementation

- Utilized NetworkX For Graph Generation and Organization
- ASAP, ALAP, and Paths Generated from DFS
- Variables and Step Locations
- Constraints
    - Execution Constraints
    - Dependency Constraints
    - Resource Constraints
- ILP and GLPK

Results


NOP
1
2
3
4
5
6
7
NOP


PARETO OPTIMAL ANALYSIS
LATENCY
MEMORY
0
2
4
6
8
10
12
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

Team Roles

- Aidan
  - Execution Logic
  - Graph Analyzation
  - ILP Formulations
- Zander
  - General Output Statements
  - ILP Printing
  - GLPK Integration
  - Pareto-Optimal Graphing
  - Memory/Latency Minimization
- Both
  - Pair-programming
  - General layout and logic framework
  - Debugging

End of Demonstration