

Aidan and Zander - Auto ILP and Pareto-Optimization Project

Aidan

- General logic
 - Constraints: Execution, resource, dependency
 - ASAP/ALAP
 - Variable generation and creation for each possible step
 - Pathfinding algorithm - DFS
- Command line argument parsing
 - -l for latency
 - -a for memory
 - -g for edgelist file
- Graph analyzation
 - NetworkX - read_weighted_edgelist()
- Logic steps for scheduling algorithms
 - What steps to take in which order depending on what minimization you want

Zander

- ILP file writing
- Constraint reading
- GLPK integration
 - Sending ILP to GLPK
 - Getting printouts from GLPK results
- Looping logic for the different scheduling/minimization types
 - Figuring out how many times to loop through each minimization for the best performance in the shortest possible time
- Graph building for Pareto-Optimal
 - Maps all feasible latency and memory constraints

Both

- Debugging
 - Went through the entire code together to work out any problems
 - Primarily utilized print statements for debugging
- Slideshow presentation
- Testing of code with different edge list graphs