Heuristics in the Longest Simple Path Problem

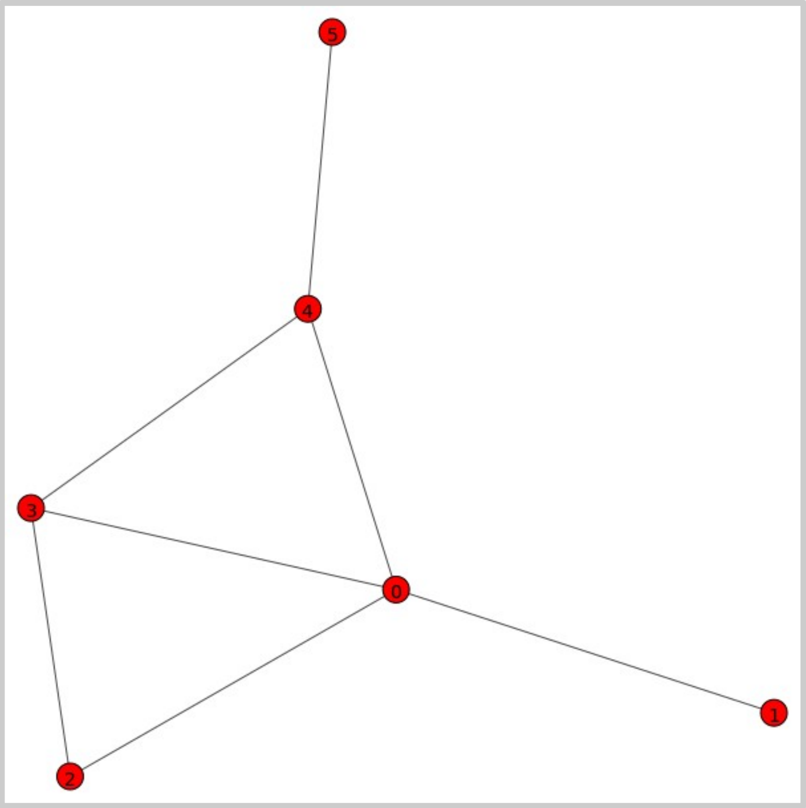
Justin Xie

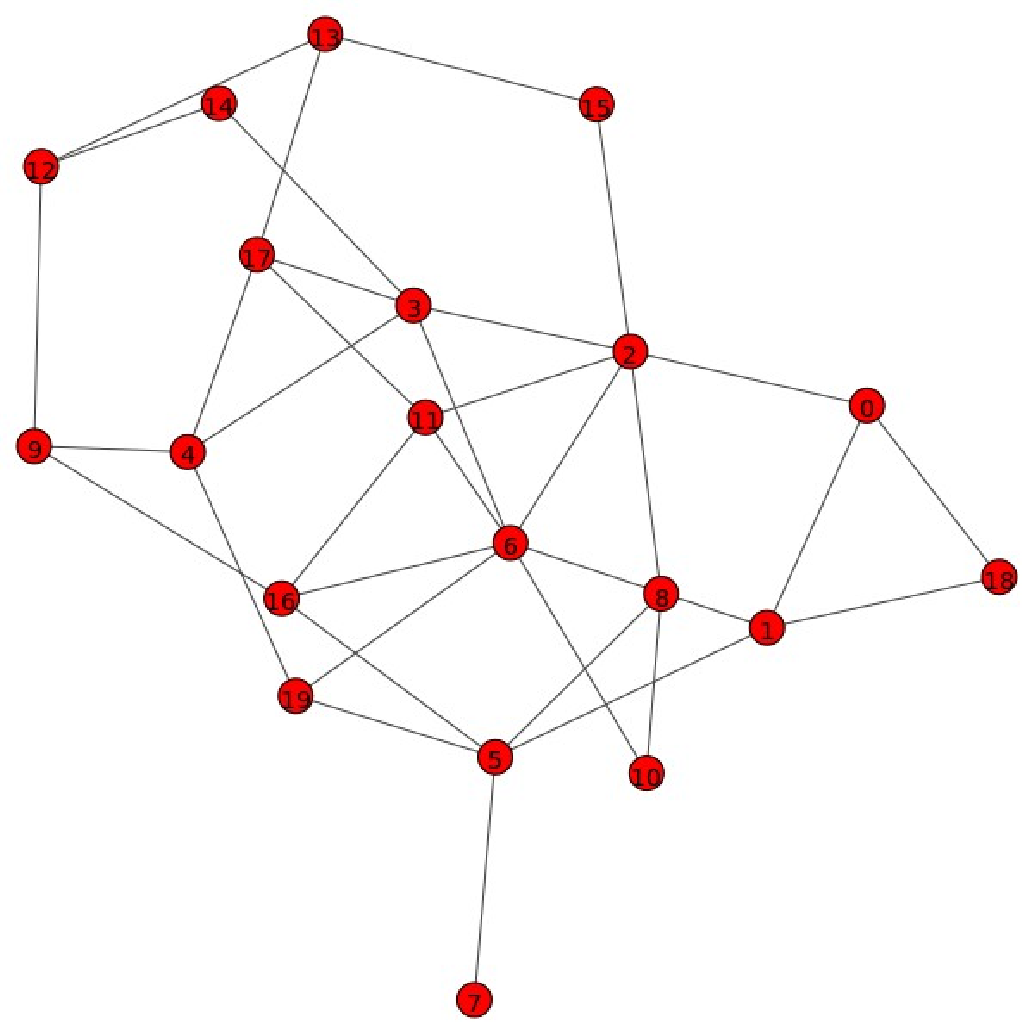
Mentors: Bart Massey and Cassaundra Smith

Agenda

1. Longest Simple Path Problem
2. Our Heuristics
   - Greedy Heuristics
   - Graph Pruning Heuristics
3. Comparison Results
4. Conclusions

Longest Simple Path:
The longest path in a graph that does not repeat vertices

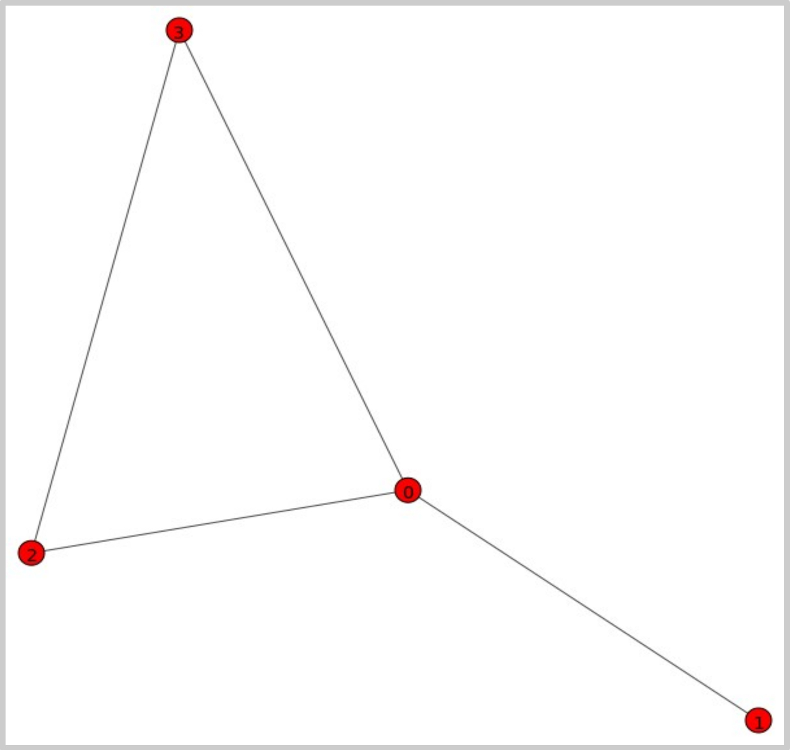


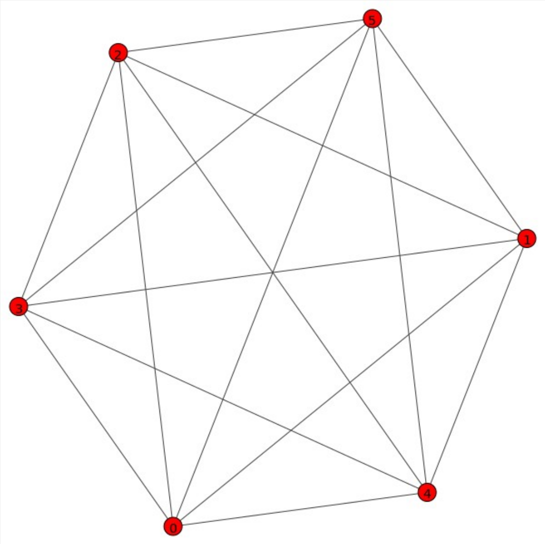

13
14
15
12
17
3
2
0
11
9
4
6
18
8
16
1
19
5
10
7

The Brute-Force Solution

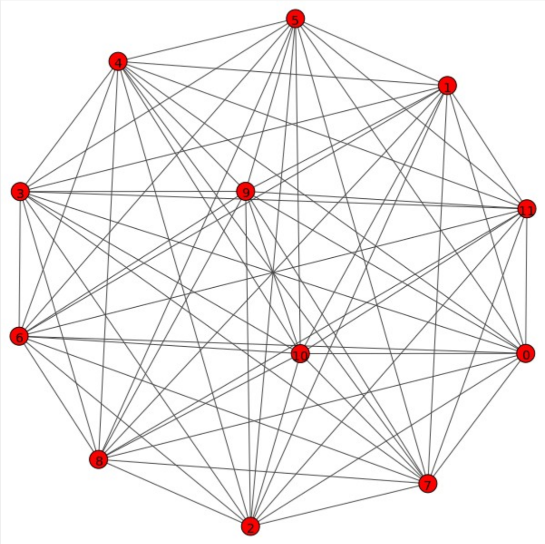
1. Find all simple paths

2. Find the longest of those simple paths

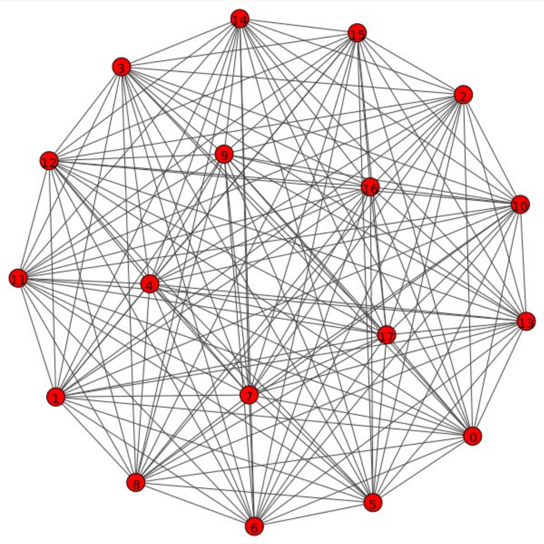




120 Simple Paths



479,001,600 Simple Paths



Over 6.4 Quintillion
Simple Paths

Number of paths increases in exponential behavior.

The Longest Simple Path Problem is NP-Hard.

Heuristics

- Shortcuts for speeds
- Trade-offs for accuracy

Agenda

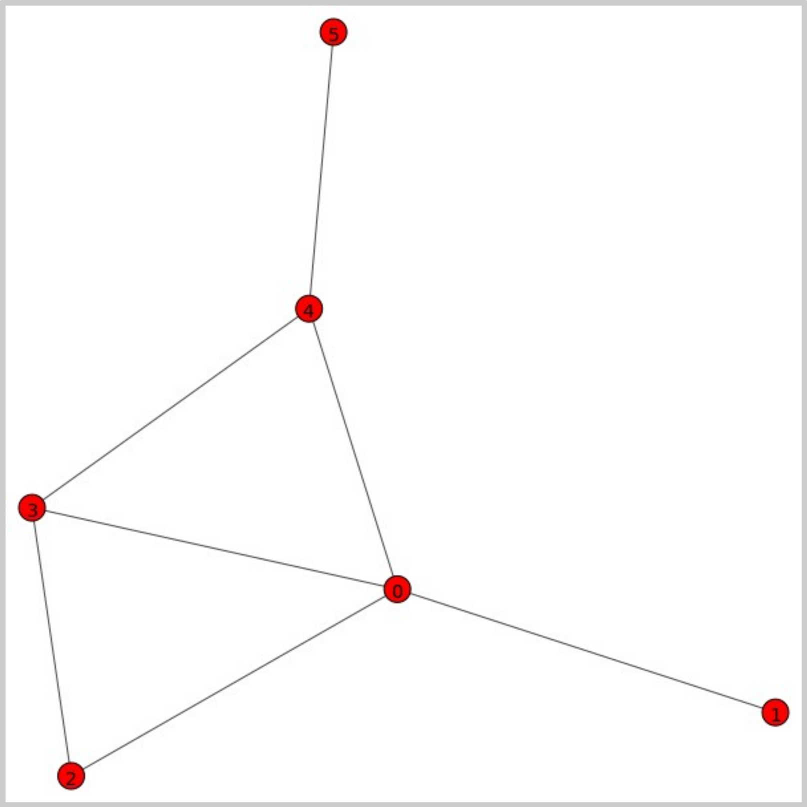
1. Longest Simple Path Problem
2. Our Heuristics
   - **Greedy Heuristics**
   - Graph Pruning Heuristics
3. Performance Comparison
4. Conclusions

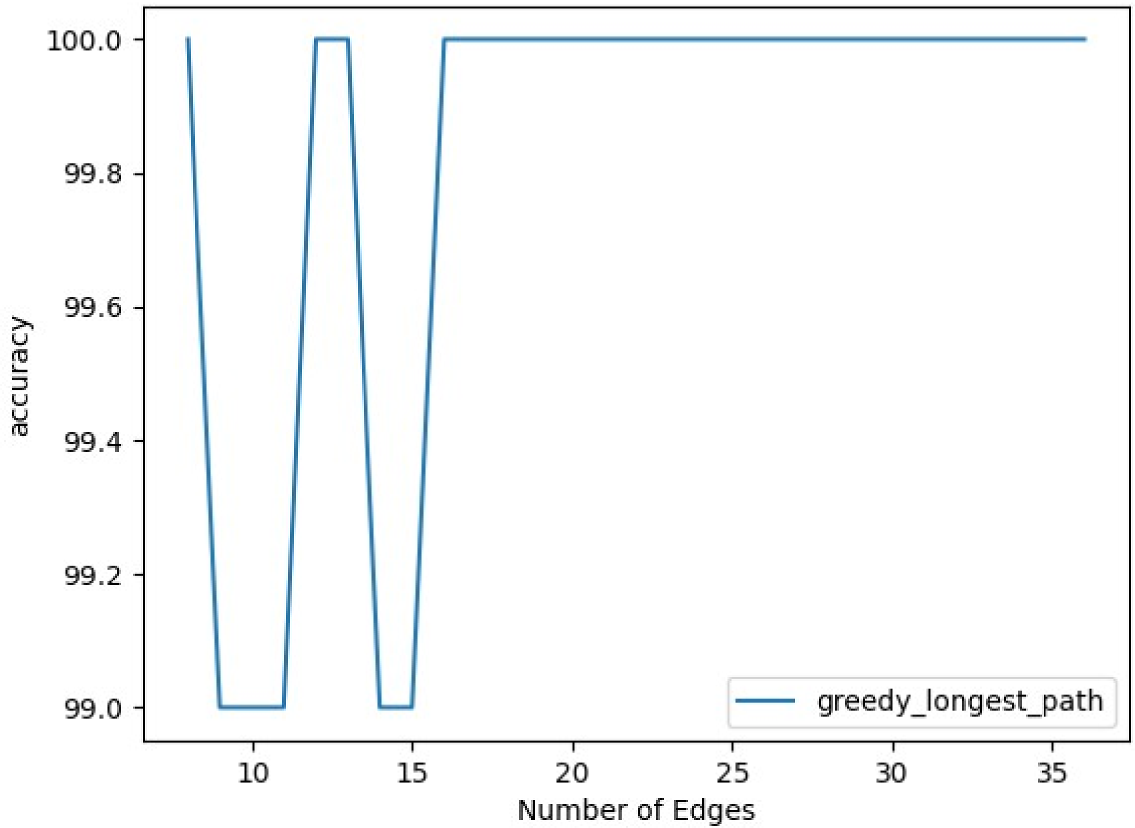
The Greedy Heuristics

- Best choice in the moment
- Doesn’t look at bigger picture

Basic Greedy Heuristic

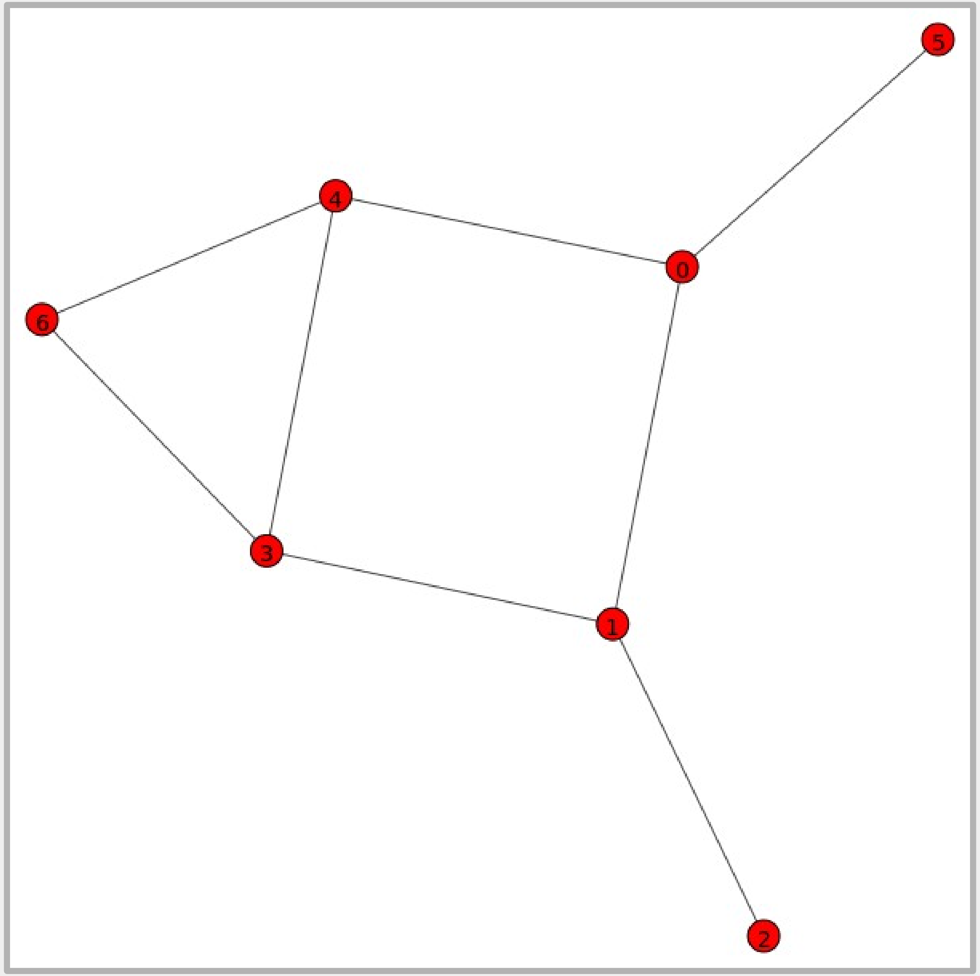
- Starts at each vertex
- Chooses neighbor with least next options
- Knocks out vertices with least options first




100.0
99.8
99.6
99.4
99.2
99.0
accuracy
10
15
20
25
30
35
Number of Edges
greedy_longest_path

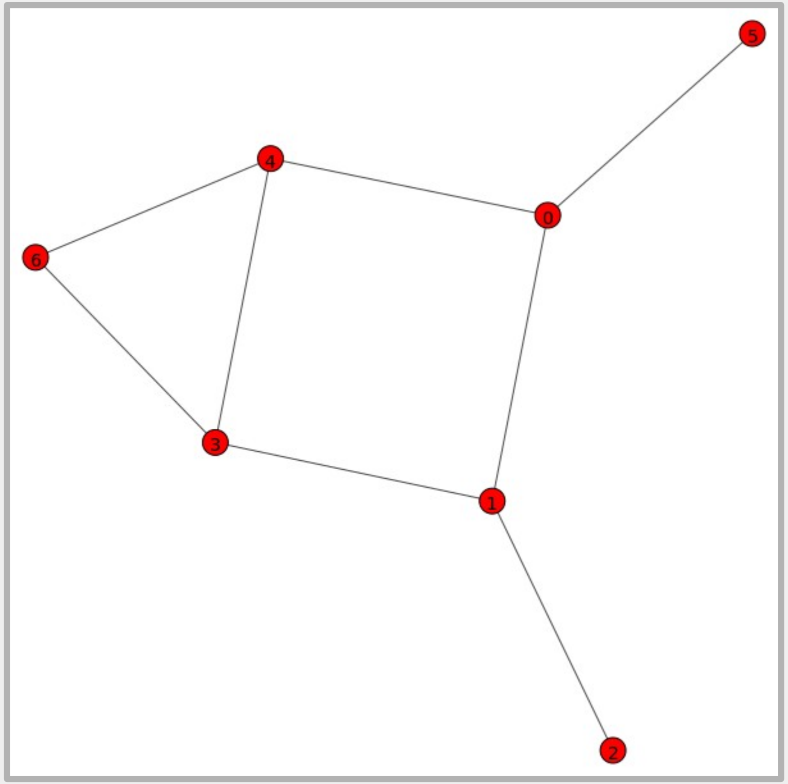
Greedy heuristic takes path:

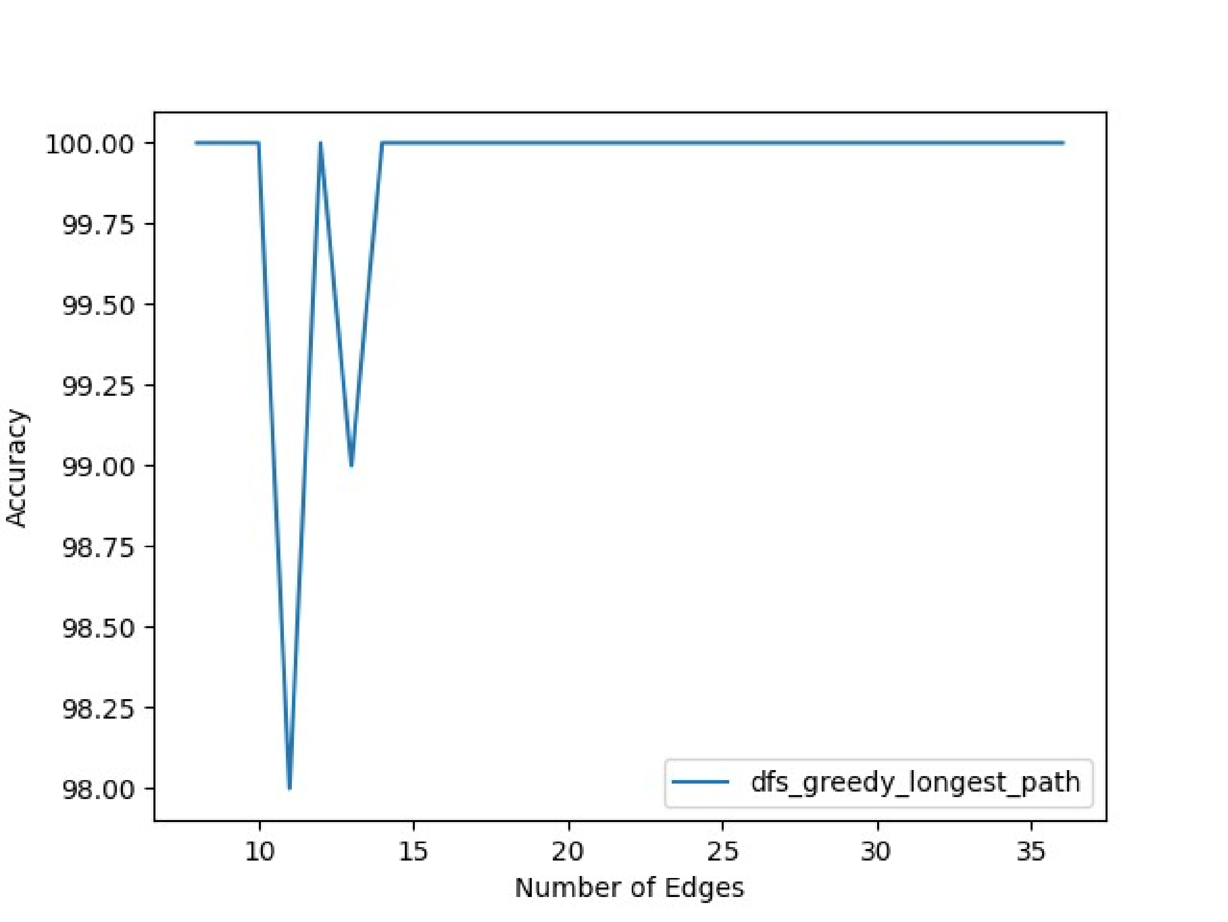
2, 1, 0, 4, 6, 3



Depth-First Search Greedy Heuristic

- Basic greedy heuristic sometimes has ties between neighbors
- DFS added to tie-break
- Compares trees created by DFS and calculates internal path length (IPL)
  - Highest IPL is next vertex





100.00
99.75
99.50
99.25
99.00
98.75
98.50
98.25
98.00
Accuracy
10
15
20
25
30
35
Number of Edges
dfs_greedy_longest_path

Agenda

1. Longest Simple Path Problem
2. Our Heuristics
    - Greedy Heuristics
    - **Graph Pruning Heuristics**
3. Performance Comparison
4. Conclusions

Graph Pruning and Stretching Heuristics

Cut graphs into a trees in order to use Dijkstra's Dangle Algorithm

Graph and Vertex Characteristics

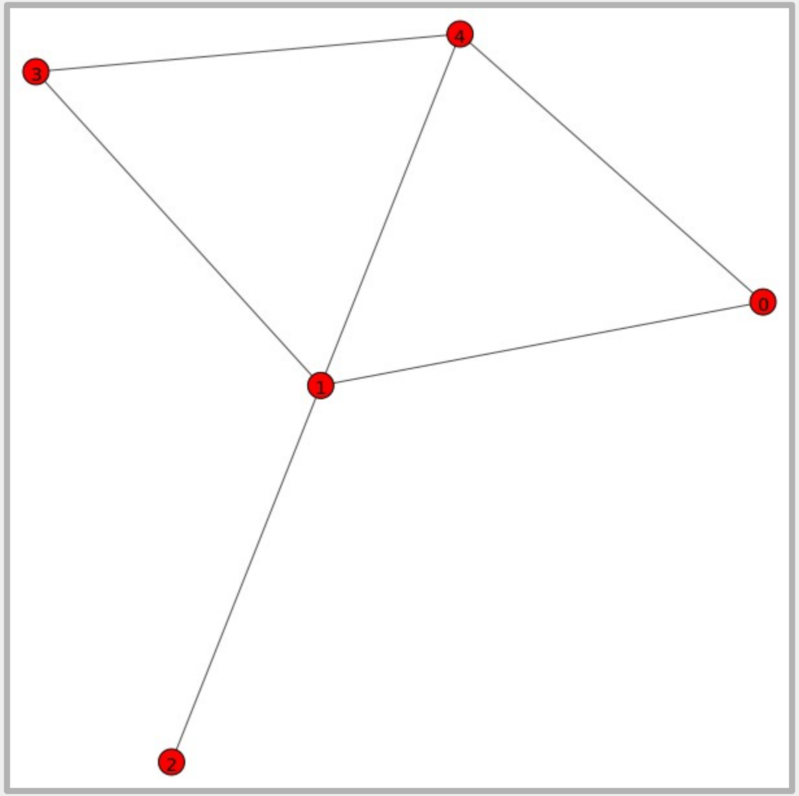
Given a vertex *V*

Vertex Periphery: Longest shortest path from *V* to any other vertex

Total Vertex Periphery (TVP): Total of shortest paths from *V* to all other vertices

- Lower periphery = More central

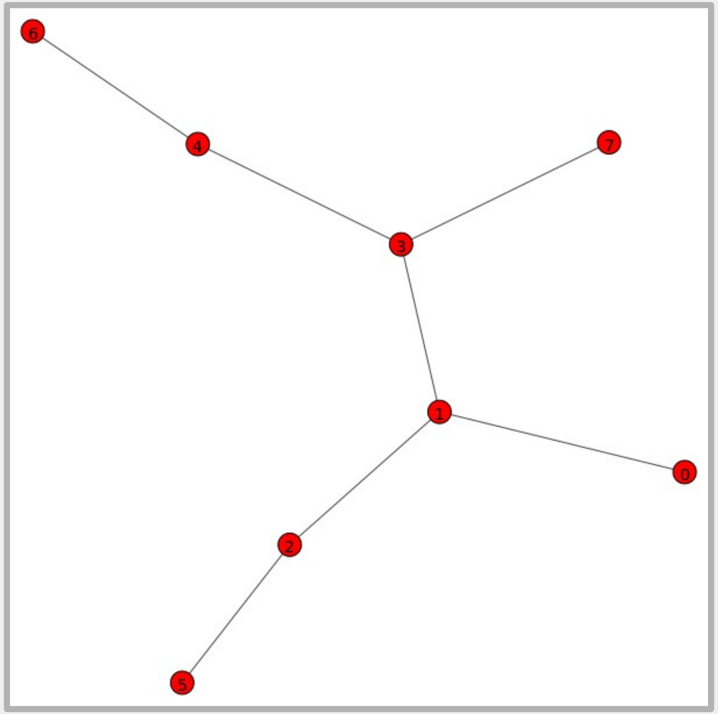
Graph Periphery: Sum of TVP of all vertices



Dijkstra’s Dangle Algorithm

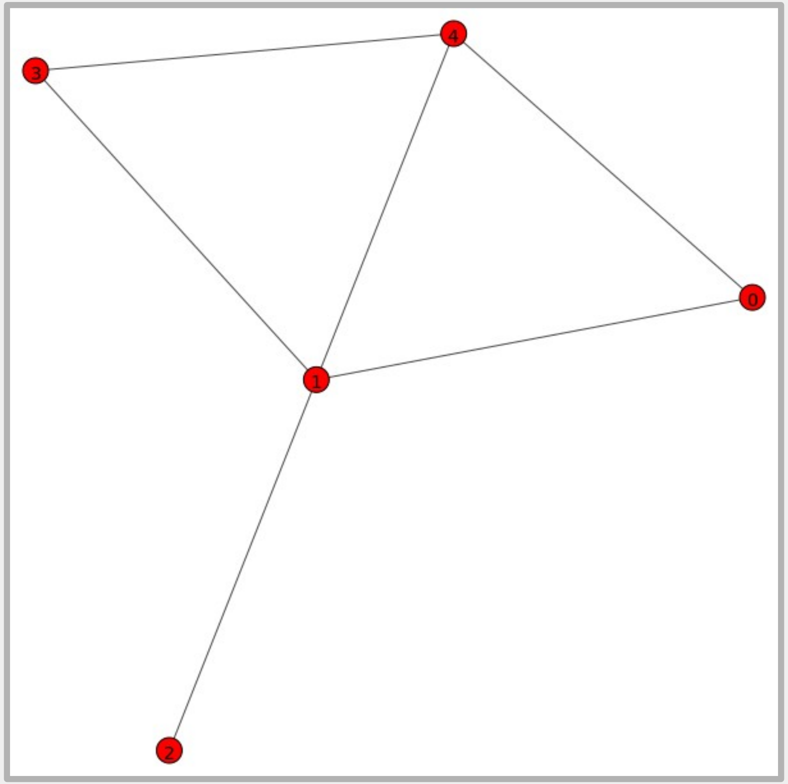
A tree with *n* vertices always has *n-1* edges

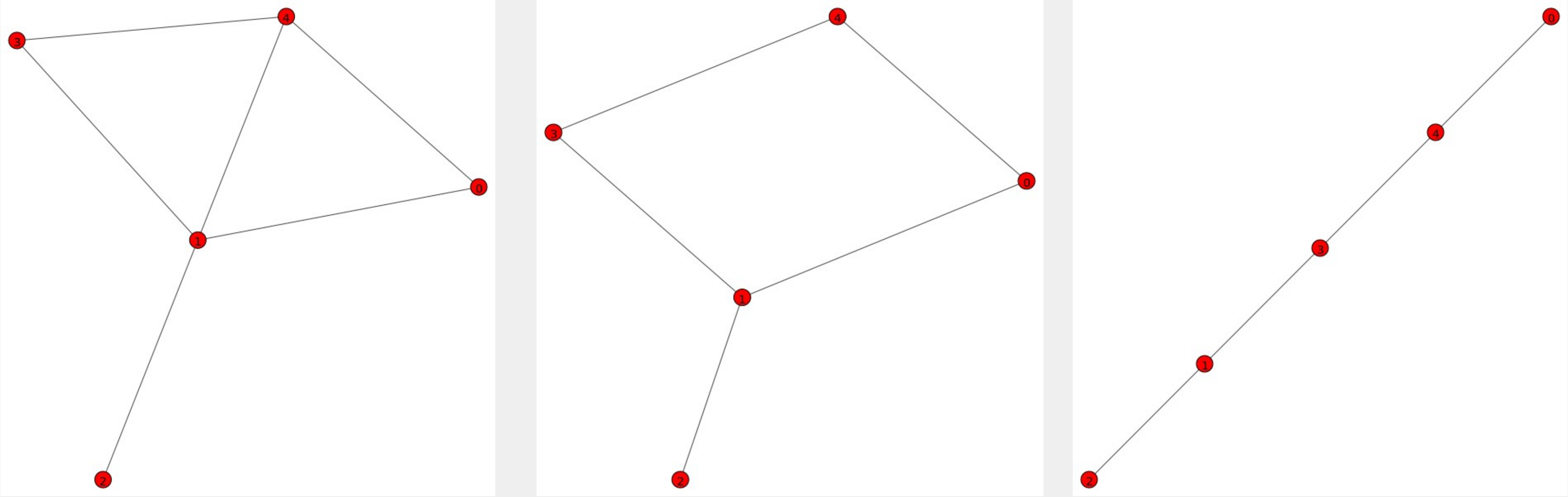
Find “extremes” of the tree and the length of path between them to get longest path in a tree

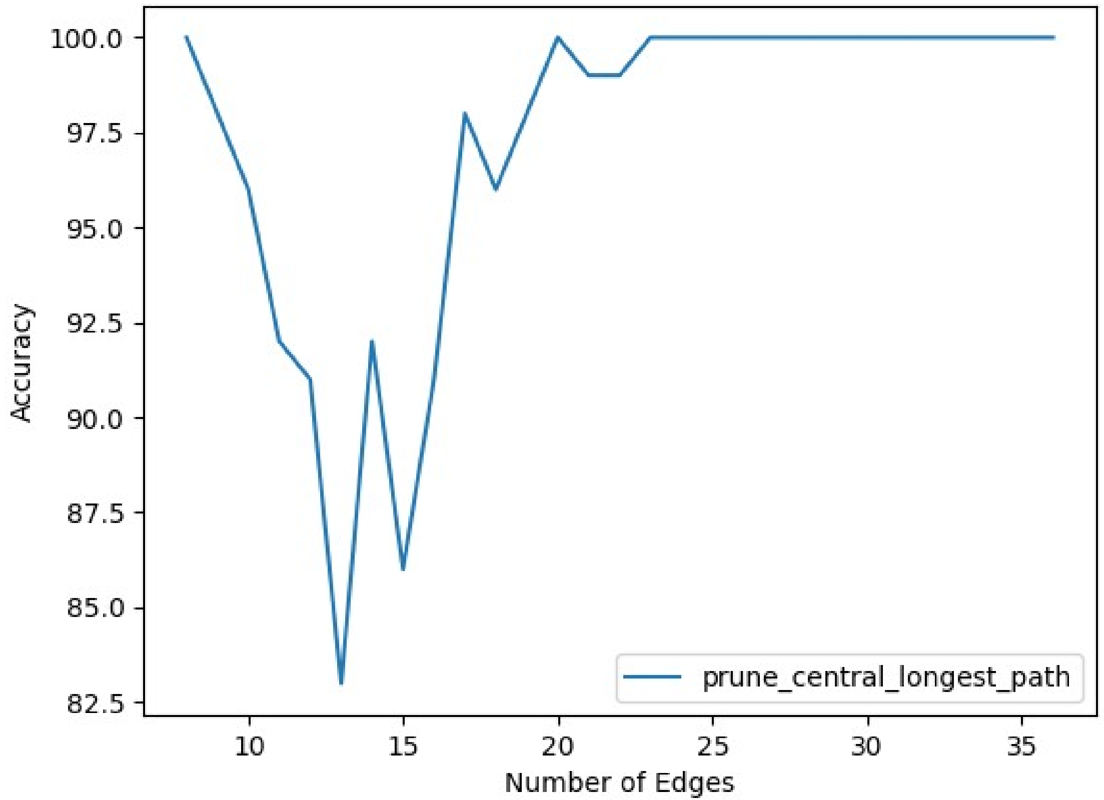


Pruning Most Central Edges

- Chooses two vertices with lowest Total Vertex Periphery
    - Remove edge between them if it exists and can be removed without discontinuity



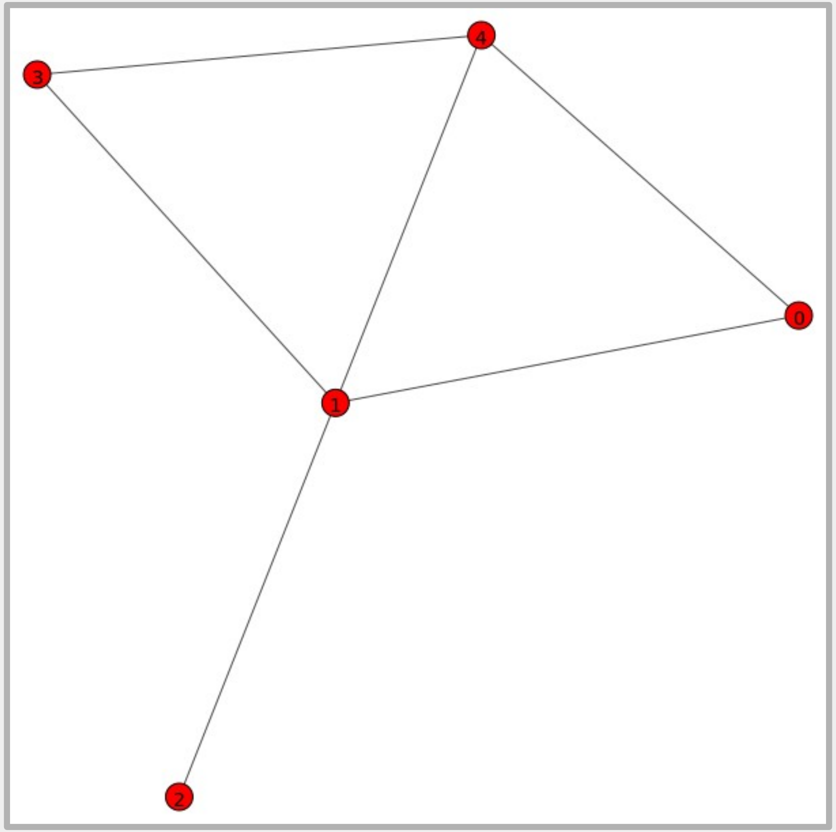

3
4
0
1
2
4
3
0
1
2
0
4
3
1
2

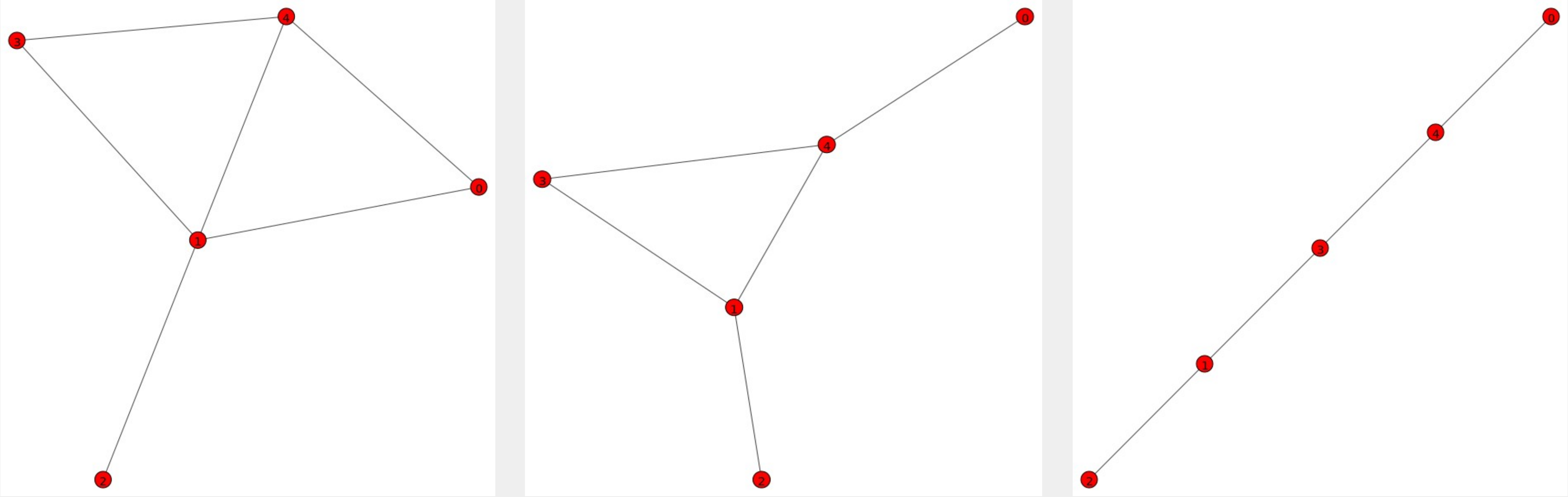


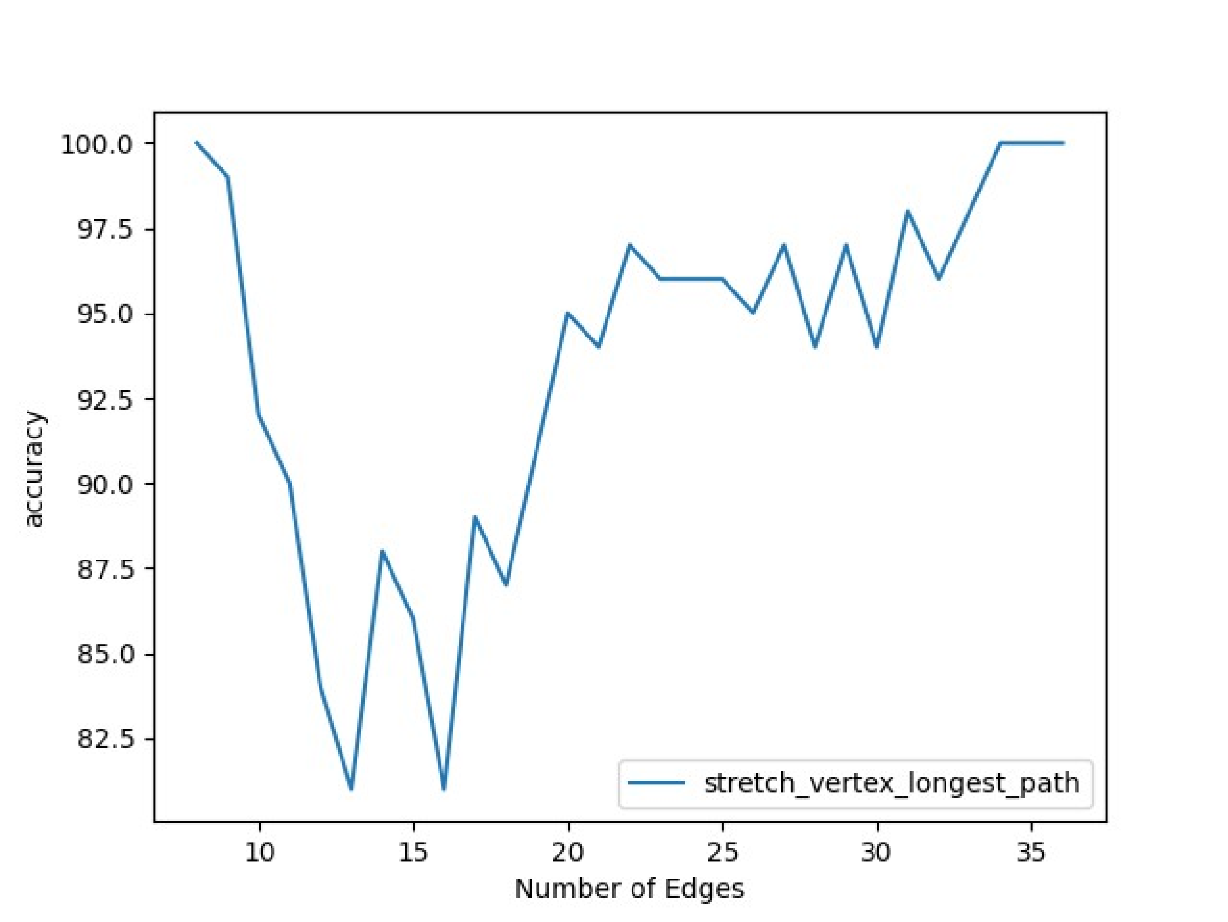
100.0
97.5
95.0
92.5
90.0
87.5
85.0
82.5
Accuracy
10
15
20
25
30
35
Number of Edges
prune_central_longest_path

Stretch Vertex With Lowest Total Periphery

- Selects vertex with lowest Total Vertex Periphery
- Removes edge to maximize the TVP of that vertex



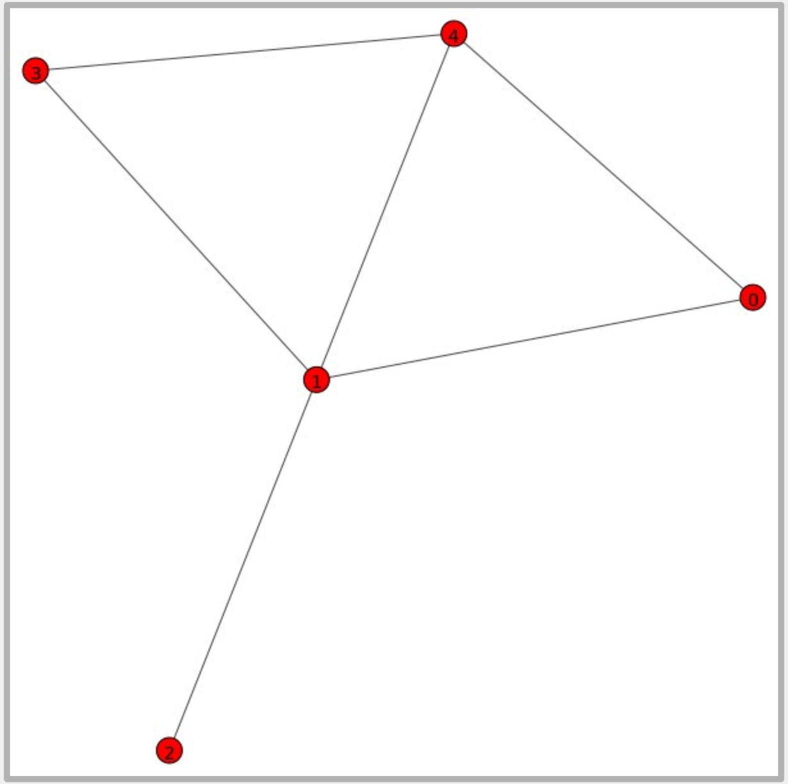

4
3
0
1
2
0
4
3
1
2
0
4
3
1
2

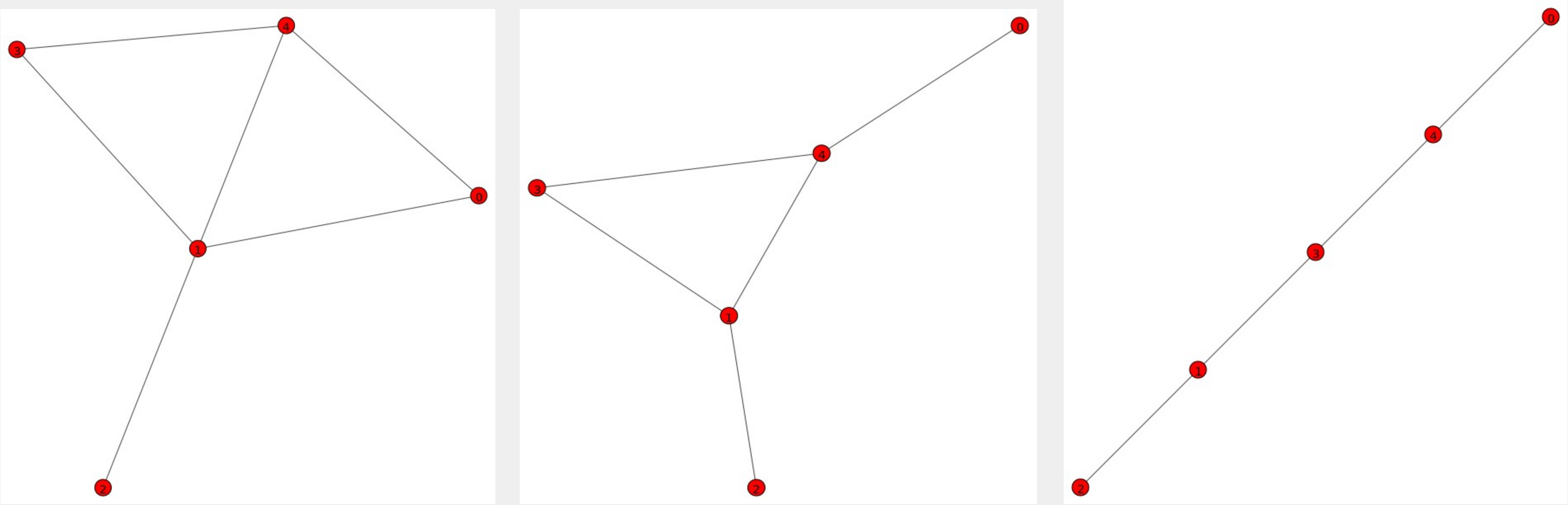


100.0
97.5
95.0
92.5
90.0
87.5
85.0
82.5
accuracy
10
15
20
25
30
35
Number of Edges
stretch_vertex_longest_path

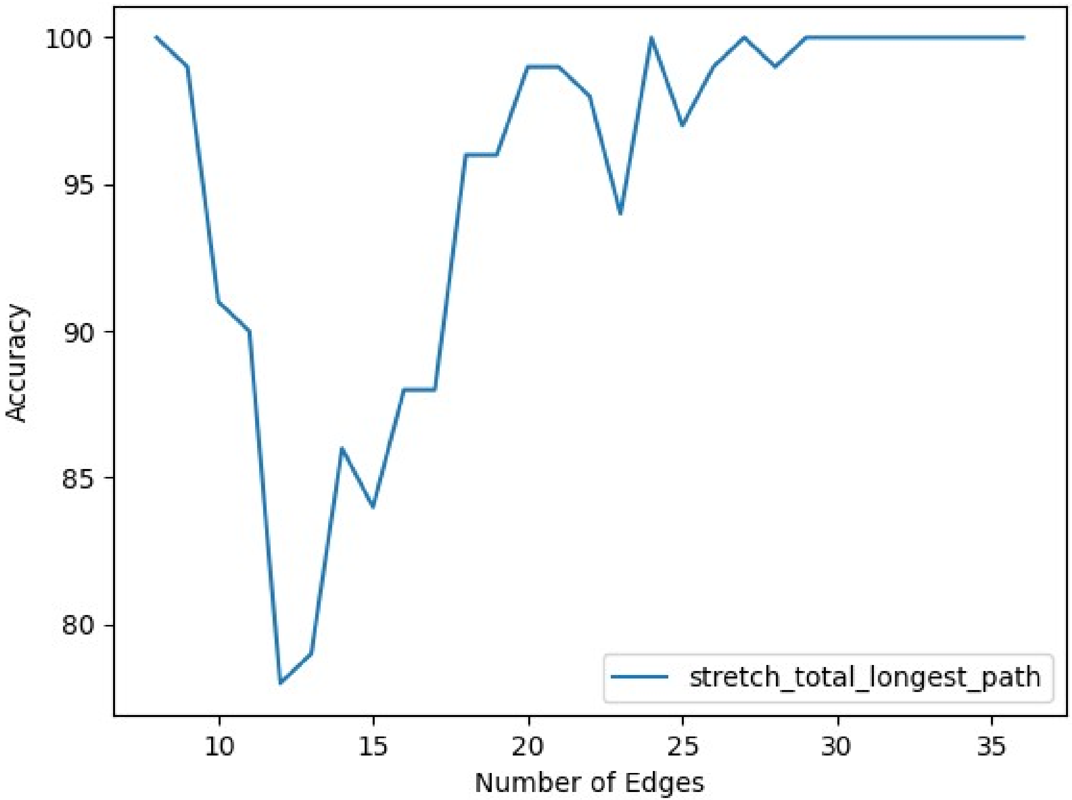
Stretching Graph Periphery

Removes edge in graph that will maximize the graph periphery



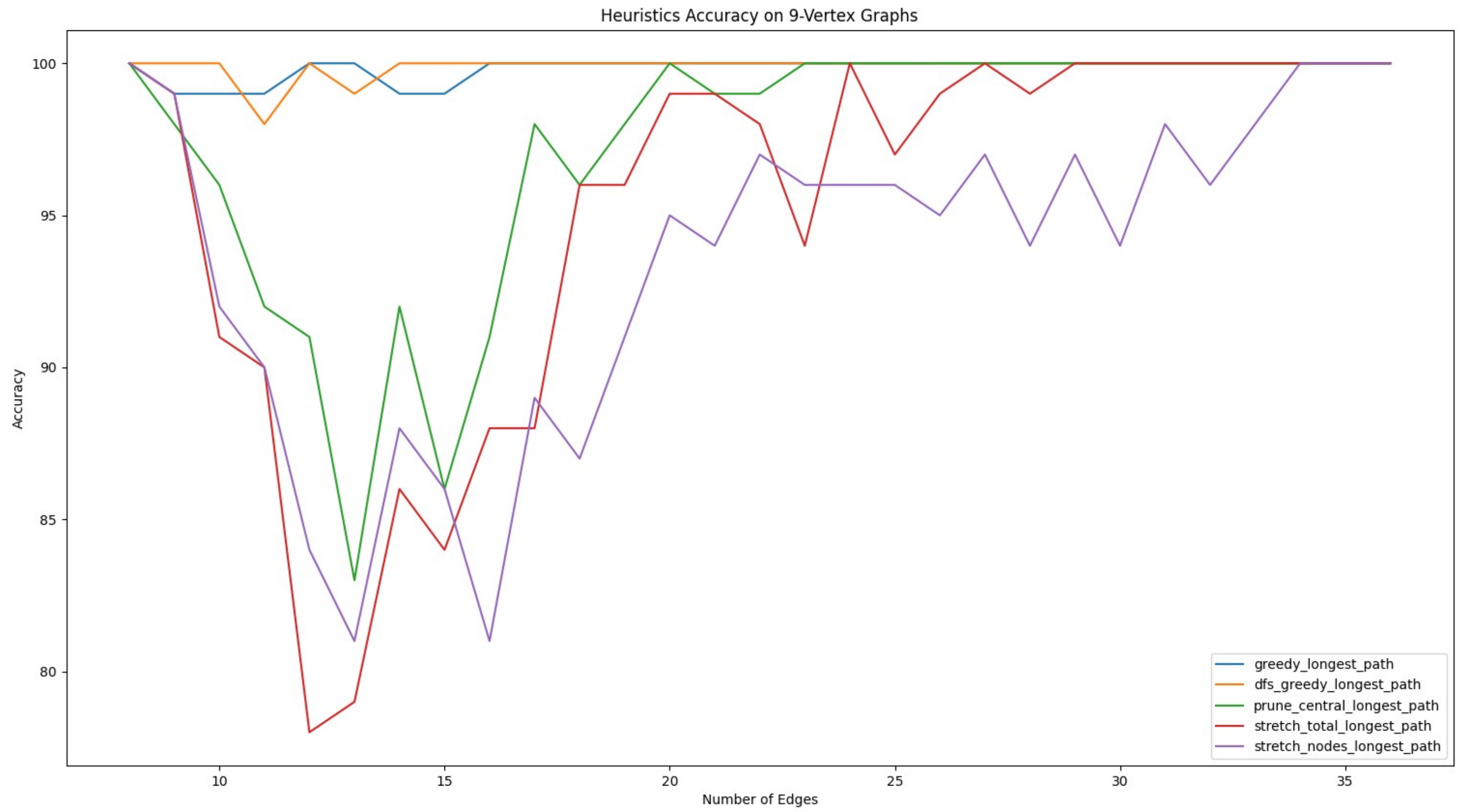


3
4
0
1
2
0
4
3
1
2
0
4
3
1
2

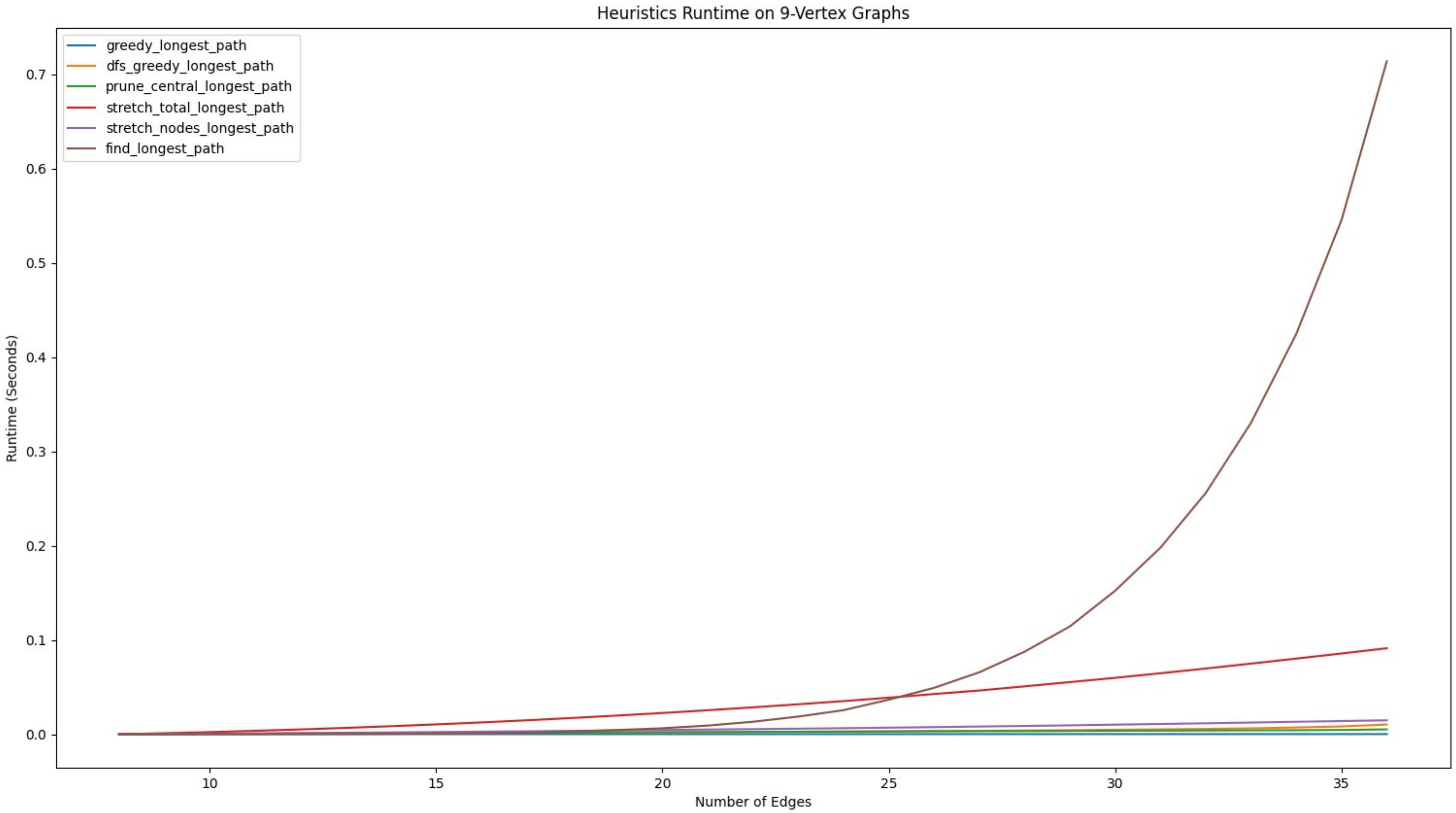

100
95
90
85
80
Accuracy
10
15
20
25
30
35
Number of Edges
stretch_total_longest_path

Agenda

1. Longest Simple Path Problem
2. Our Heuristics
   - Greedy Heuristics
   - Graph Pruning Heuristics
3. **Performance Comparison**
4. Conclusions



Heuristics Accuracy on 9-Vertex Graphs
Accuracy
Number of Edges
100
95
90
85
80
10
15
20
25
30
35
greedy_longest_path
dfs_greedy_longest_path
prune_central_longest_path
stretch_total_longest_path
stretch_nodes_longest_path



Heuristics Runtime on 9-Vertex Graphs
greedy_longest_path
dfs_greedy_longest_path
prune_central_longest_path
stretch_total_longest_path
stretch_nodes_longest_path
find_longest_path
Runtime (Seconds)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
10
15
20
25
30
35
Number of Edges

Conclusions

1. The Longest Simple Path Problem is NP-Hard
2. We implemented 5 heuristics as alternative methods
3. Heuristics are faster, but less accurate
4. Greedy heuristics are capable on dense graphs
   - DFS did not improve greedy heuristic too much
5. Pruning heuristics show promise for use with other well-known search heuristics like A* and local-search
6. Sparse graphs are the most difficult to solve