

# Faster Shortest Path Computation for Traffic Assignment

Boshen Chen

Supervised by: Dr. Andrea Raith, Olga Perederieieva

Department of Engineering Science  
University of Auckland

July 17, 2013

- transportation network with supply and demand nodes

- transportation network with supply and demand nodes
- arcs have non-linear costs for capturing congestion effects

- transportation network with supply and demand nodes
- arcs have non-linear costs for capturing congestion effects
- all-or-nothing assignment

- transportation network with supply and demand nodes
- arcs have non-linear costs for capturing congestion effects
- all-or-nothing assignment
  - send traffic all at once to a node via the shortest path

- transportation network with supply and demand nodes
- arcs have non-linear costs for capturing congestion effects
- all-or-nothing assignment
  - send traffic all at once to a node via the shortest path

- transportation network with supply and demand nodes
- arcs have non-linear costs for capturing congestion effects
- all-or-nothing assignment
  - send traffic all at once to a node via the shortest path

Initialise: all-or-nothing assignment on  
shortest path with free flow travel times

- transportation network with supply and demand nodes
- arcs have non-linear costs for capturing congestion effects
- all-or-nothing assignment
  - send traffic all at once to a node via the shortest path

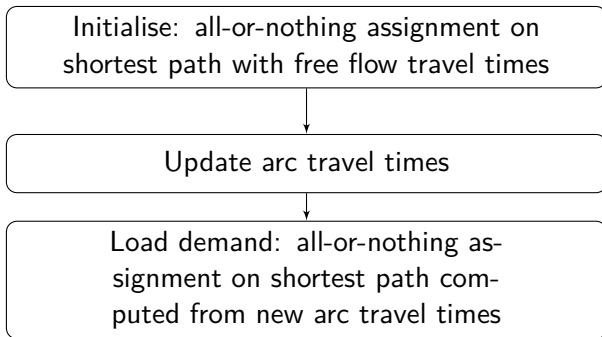
Initialise: all-or-nothing assignment on  
shortest path with free flow travel times



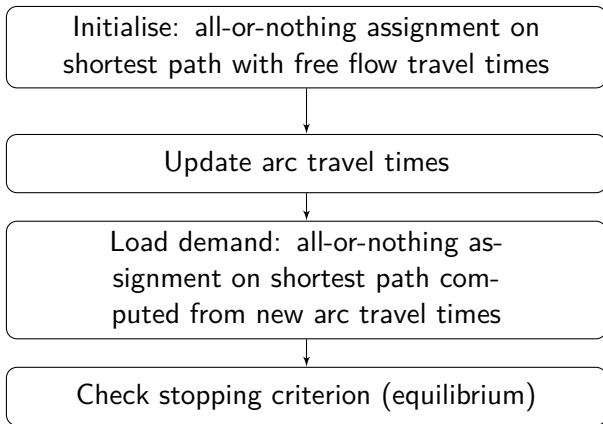
Update arc travel times



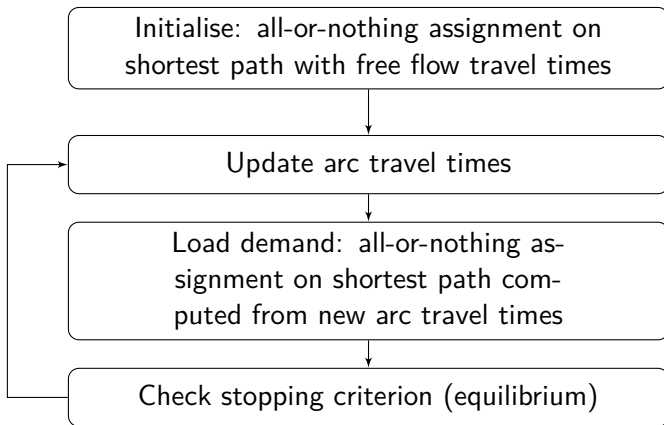
- transportation network with supply and demand nodes
- arcs have non-linear costs for capturing congestion effects
- all-or-nothing assignment
  - send traffic all at once to a node via the shortest path



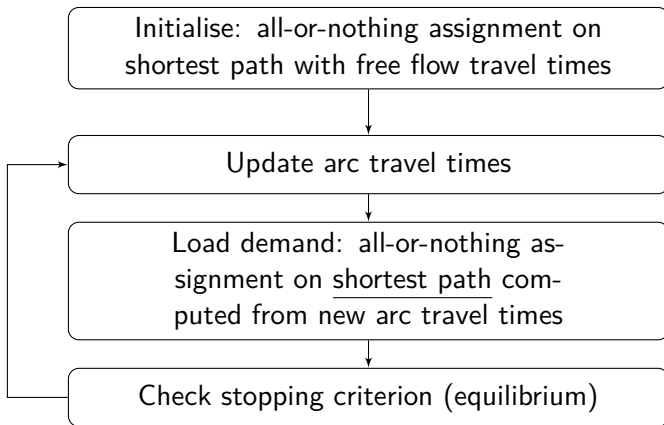
- transportation network with supply and demand nodes
- arcs have non-linear costs for capturing congestion effects
- all-or-nothing assignment
  - send traffic all at once to a node via the shortest path



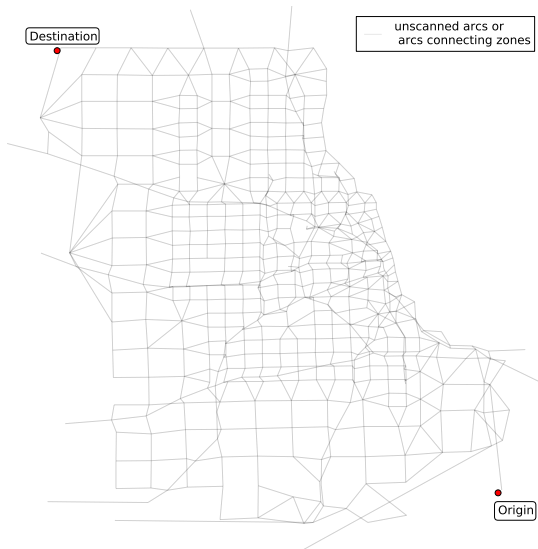
- transportation network with supply and demand nodes
- arcs have non-linear costs for capturing congestion effects
- all-or-nothing assignment
  - send traffic all at once to a node via the shortest path



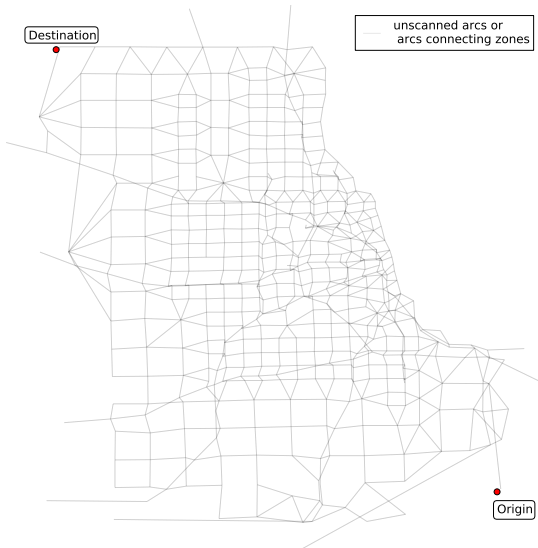
- transportation network with supply and demand nodes
- arcs have non-linear costs for capturing congestion effects
- all-or-nothing assignment
  - send traffic all at once to a node via the shortest path



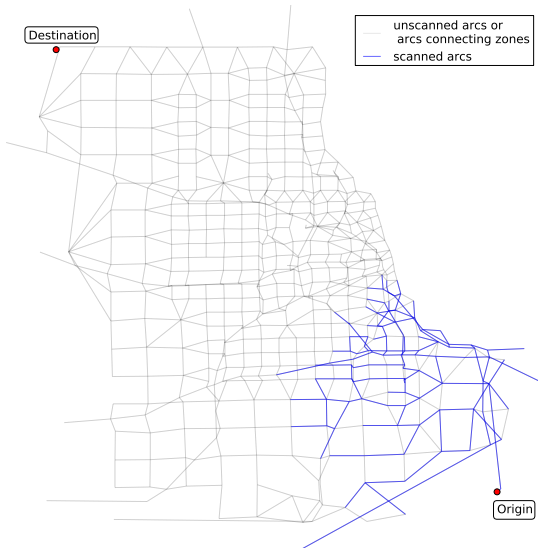
# The Graph - 93,135 Origin-Destinations Pairs



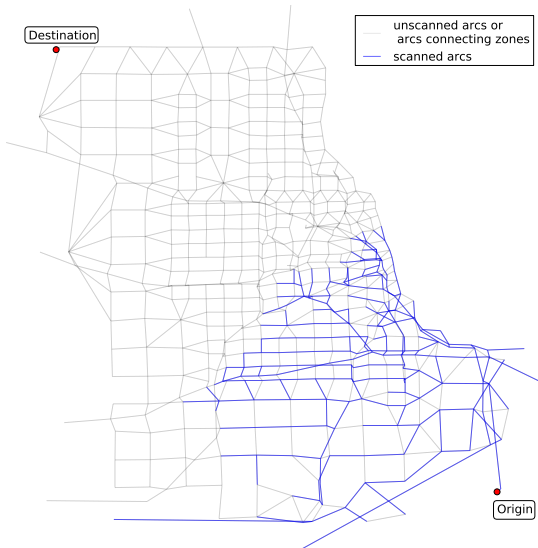
# Dijkstra's Algorithm



# Dijkstra's Algorithm

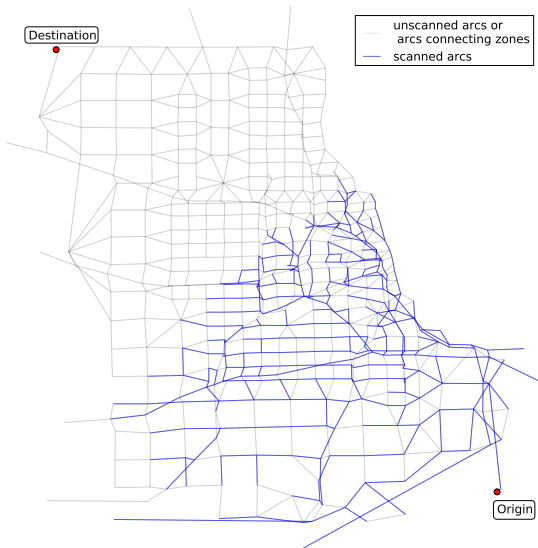


# Dijkstra's Algorithm

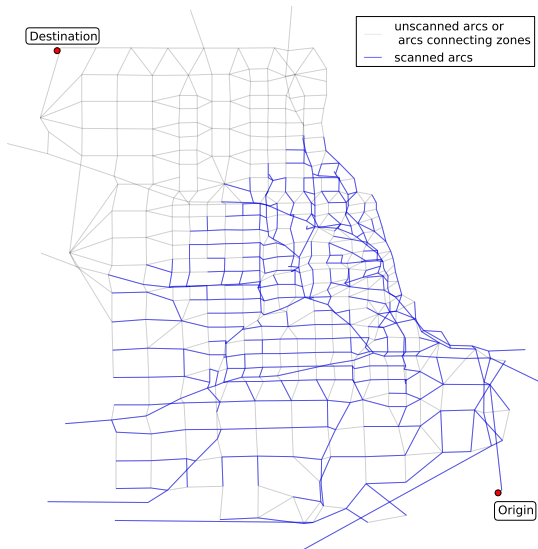




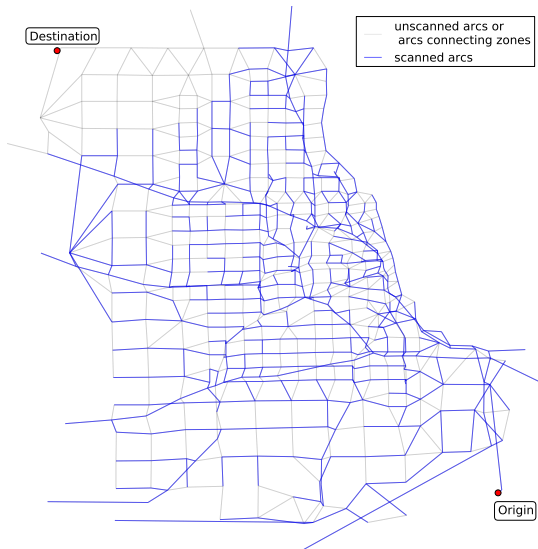
# Dijkstra's Algorithm



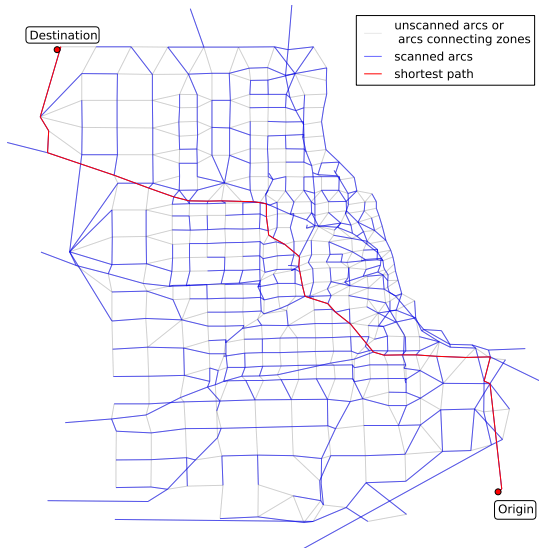
# Dijkstra's Algorithm



# Dijkstra's Algorithm



# Dijkstra's Algorithm



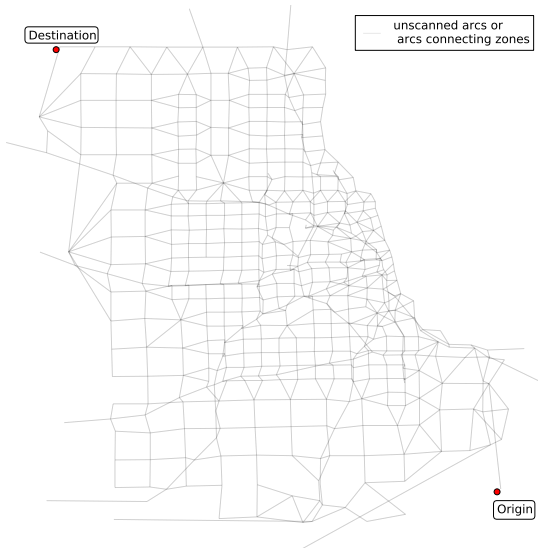
# Dijkstra's Algorithm - Priority Queue

- Pointer based Heap (C++ boost library)
  - Binomial
  - Pairing
  - Binary
  - Ternary
  - Skew
  - Fibonacci
- (red-black) binary search tree (C++ STL `<set>`)
- array based Heap (C++ STL `<priority_queue>`)

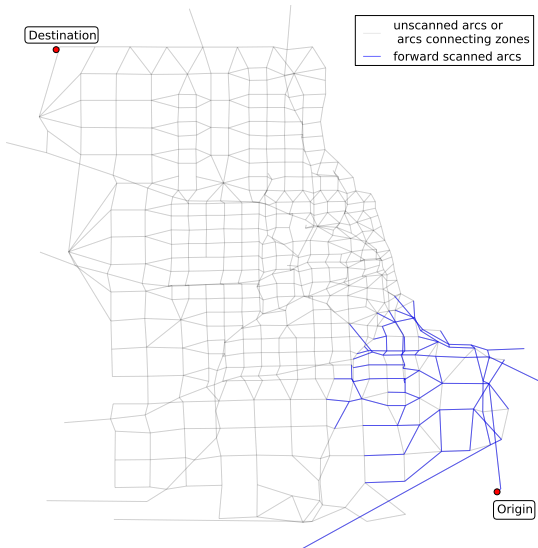
# Dijkstra's Algorithm - Priority Queue

- Pointer based Heap (C++ boost library)
  - Binomial
  - Pairing
  - Binary
  - Ternary
  - Skew
  - Fibonacci
- (red-black) binary search tree (C++ STL `<set>`)
- array based Heap (C++ STL `<priority_queue>`) ← winner

# Bidirectional Dijkstra's Algorithm

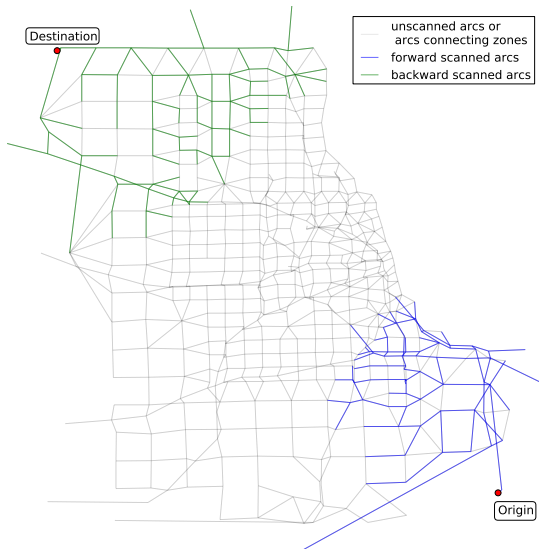


# Bidirectional Dijkstra's Algorithm

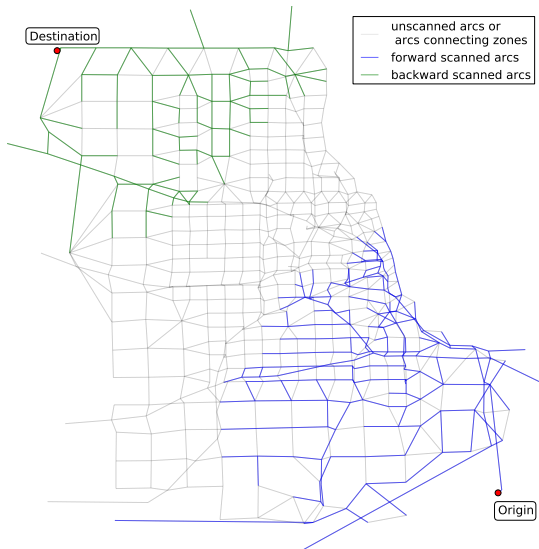




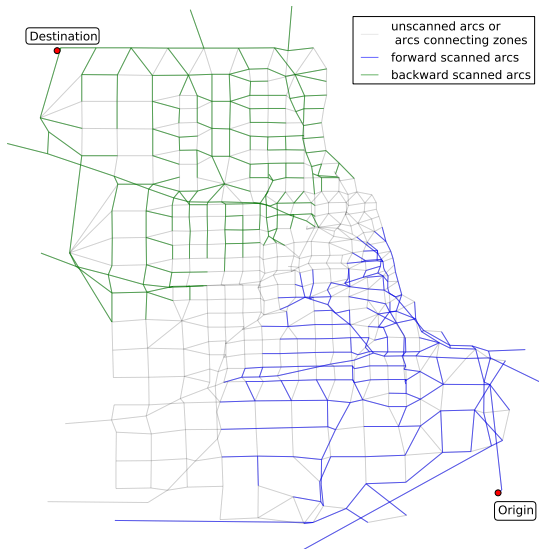
# Bidirectional Dijkstra's Algorithm



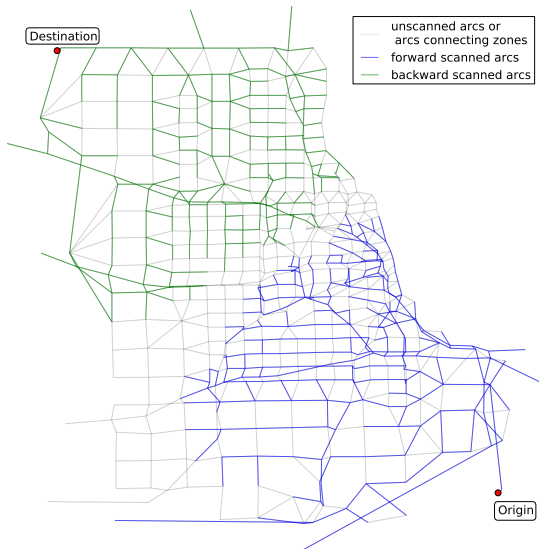
# Bidirectional Dijkstra's Algorithm



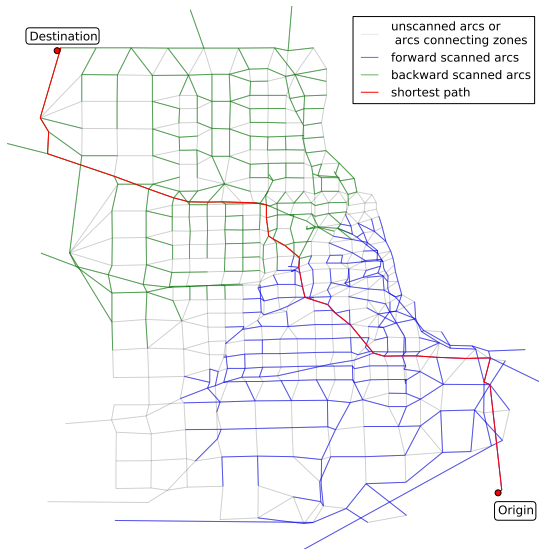
# Bidirectional Dijkstra's Algorithm



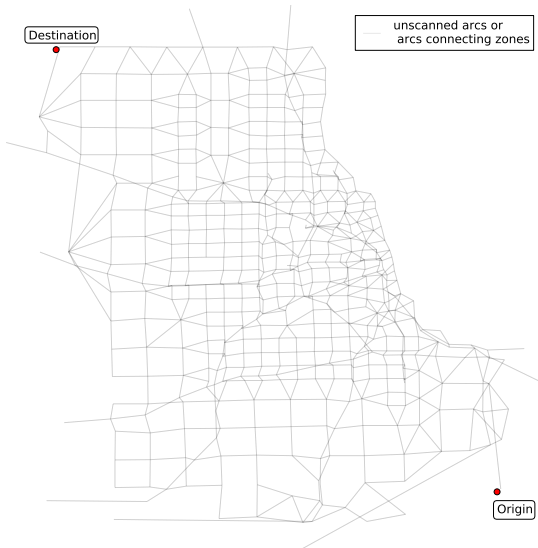
# Bidirectional Dijkstra's Algorithm



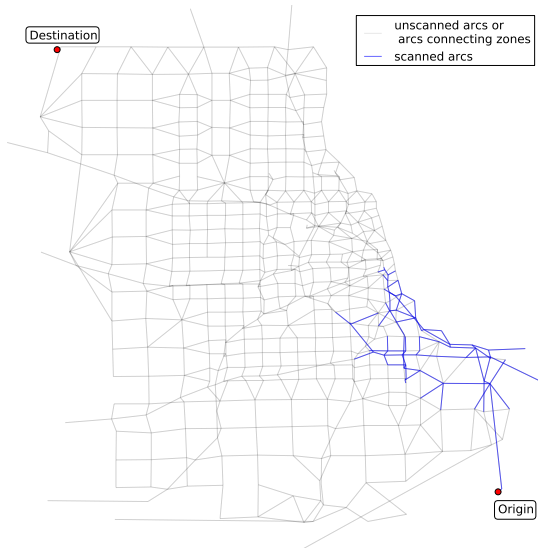
# Bidirectional Dijkstra's Algorithm



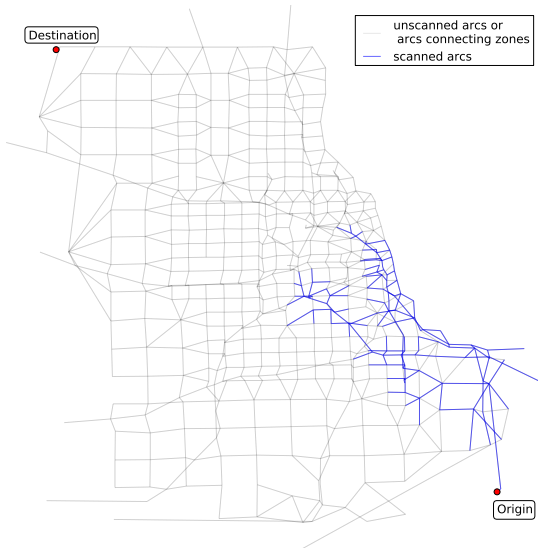
# A\* Search



# A\* Search

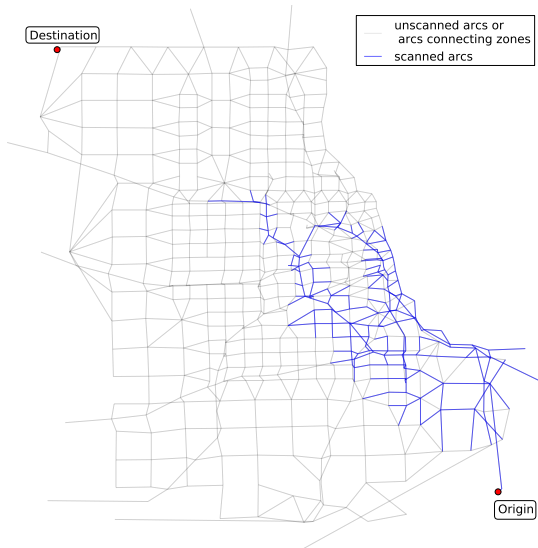


# A\* Search

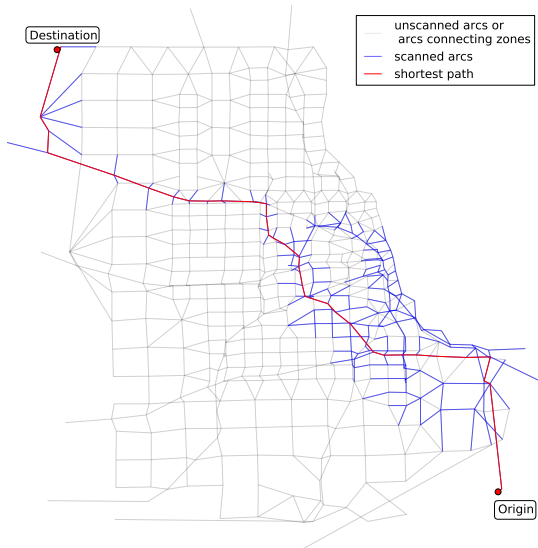




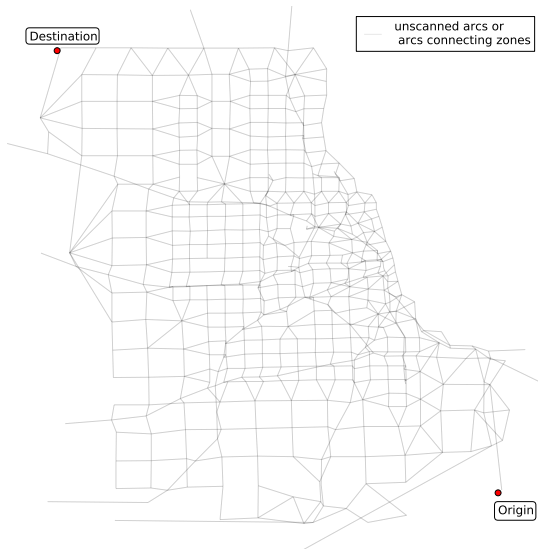
# A\* Search



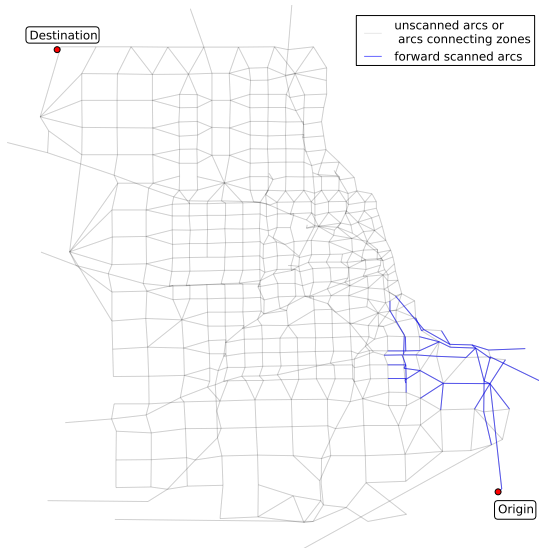
# A\* Search



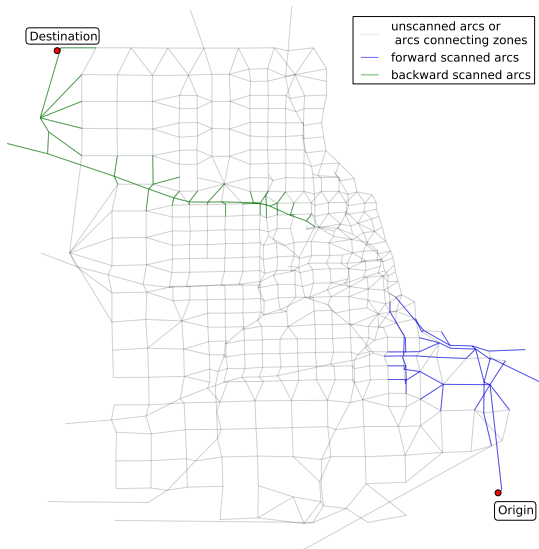
# Bidirectional A\* search



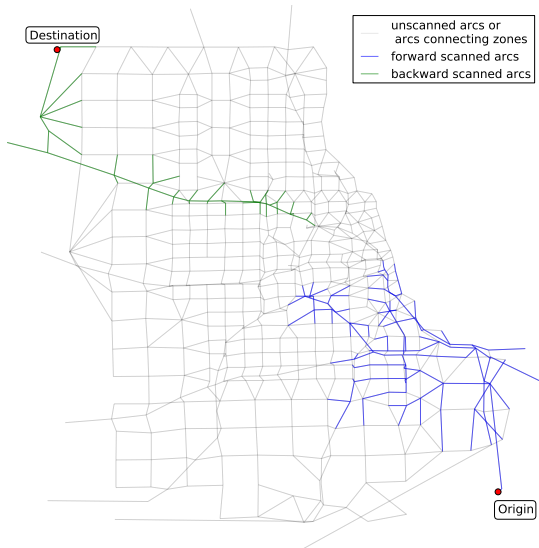
# Bidirectional A\* search



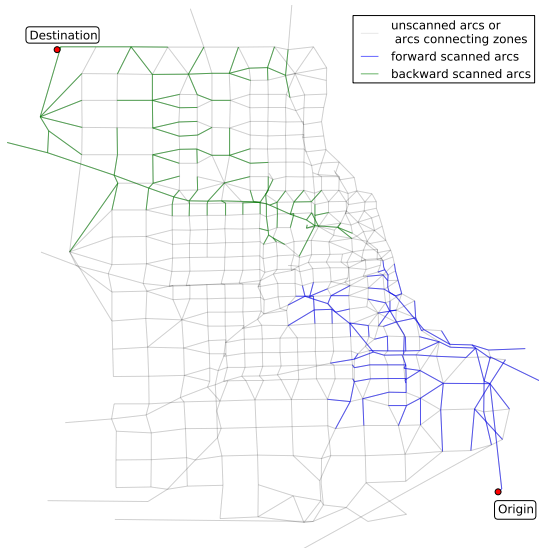
# Bidirectional A\* search



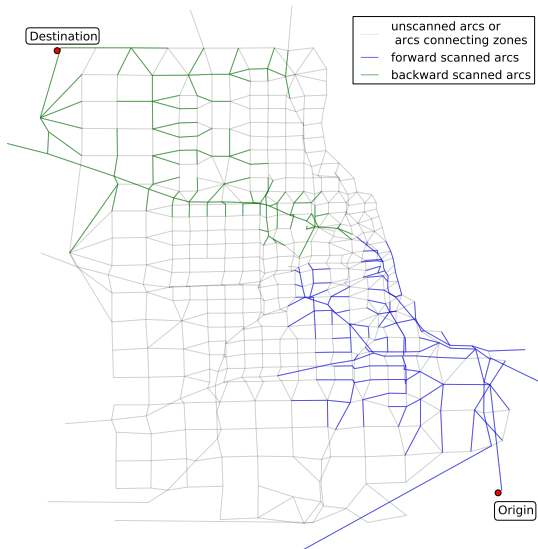
# Bidirectional A\* search



# Bidirectional A\* search

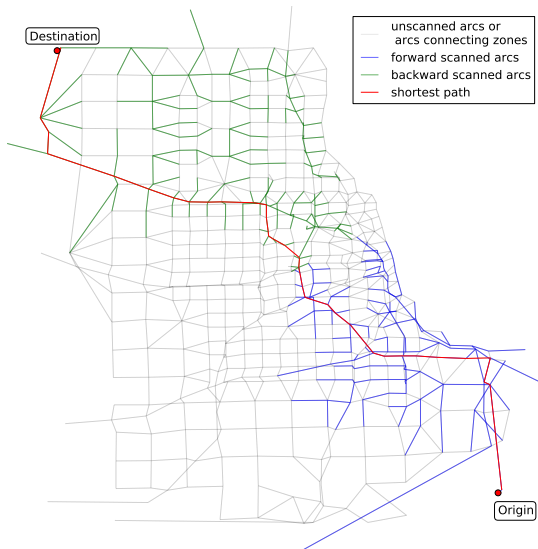


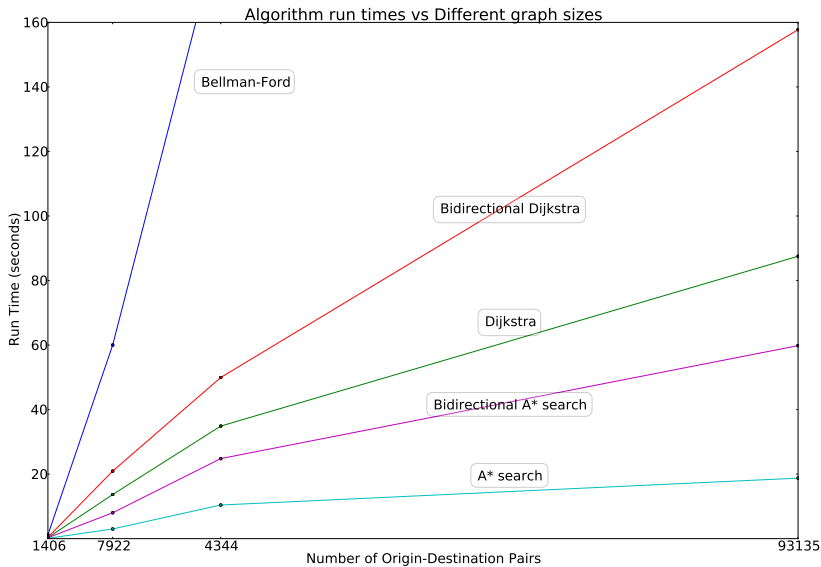
# Bidirectional A\* search

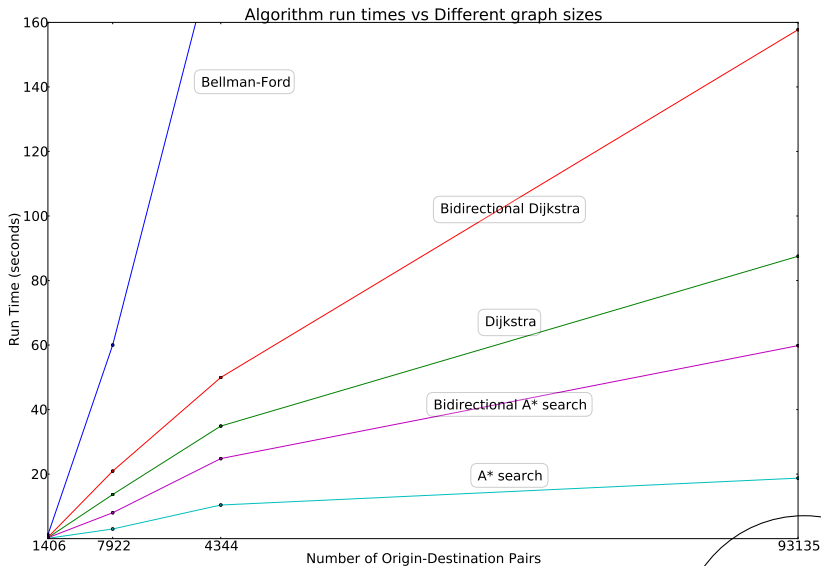




# Bidirectional A\* search







10 iters  
°

27 iters  
○

126 iters  
○

25 iters

# Future Works

- preprocessing

- preprocessing
  - A\* search combined with landmark distances

- preprocessing
  - A\* search combined with landmark distances
- use information from previous iteration

- preprocessing
  - A\* search combined with landmark distances
- use information from previous iteration
  - Incremental heuristic search - Lifelong Planning A\*