Finding The Most Preferred Path

Dimitris Sacharidis[1] Panagiotis Bouros[2] **Theodoros Chondrogiannis[3]**

[1] Technische Universität Wien

[2] Aarhus University

[3] Free University of Bozen-Bolzano

ACM SIGSPATIAL GIS 2017, November 7-10 2017, Redondo Beach, California, USA

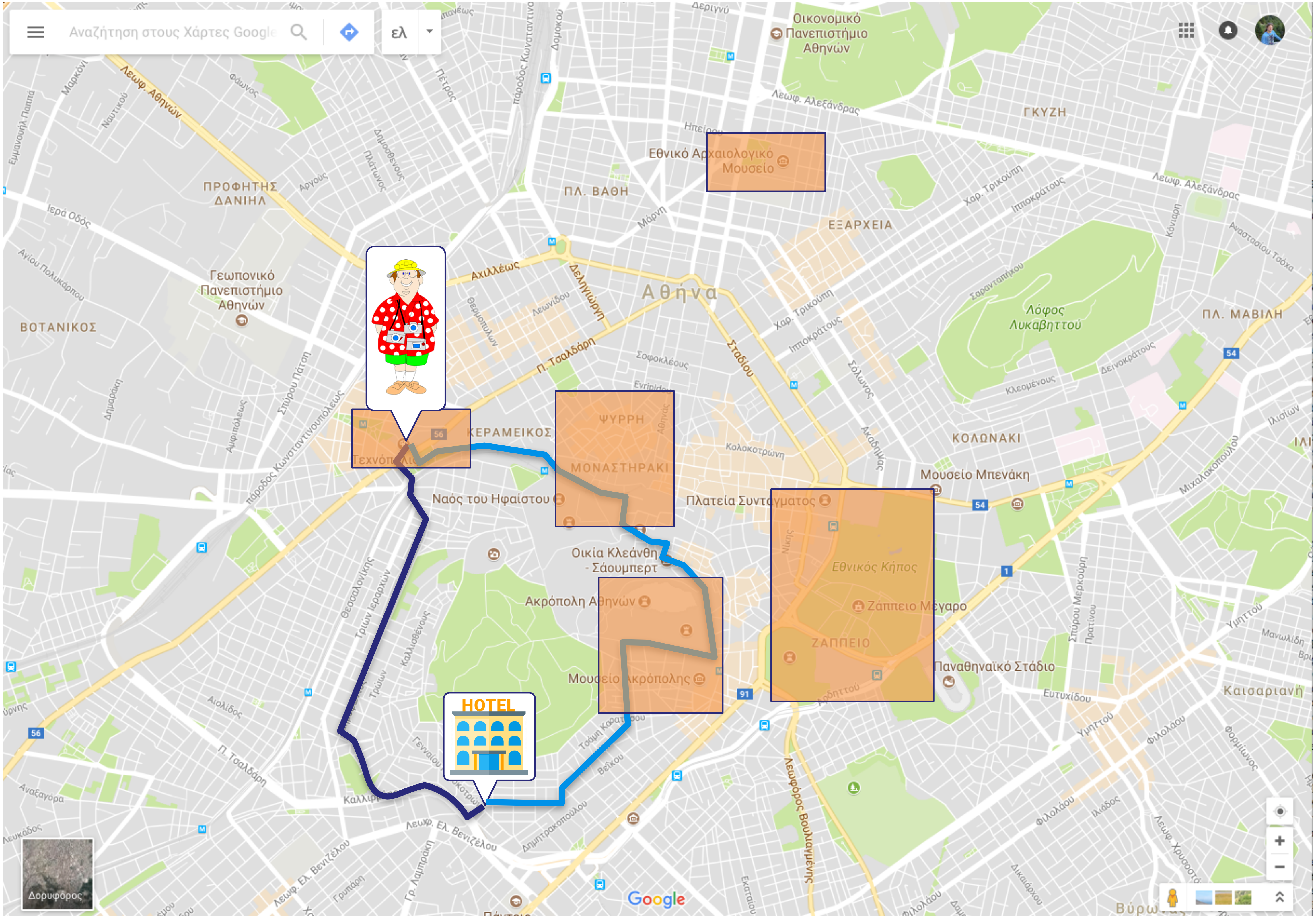
Overview

- Motivating Scenarios
- Preferred Network
- Most Preferred Unrestricted Path
- Most Preferred Near Shortest Path
- Conclusions & Future Work

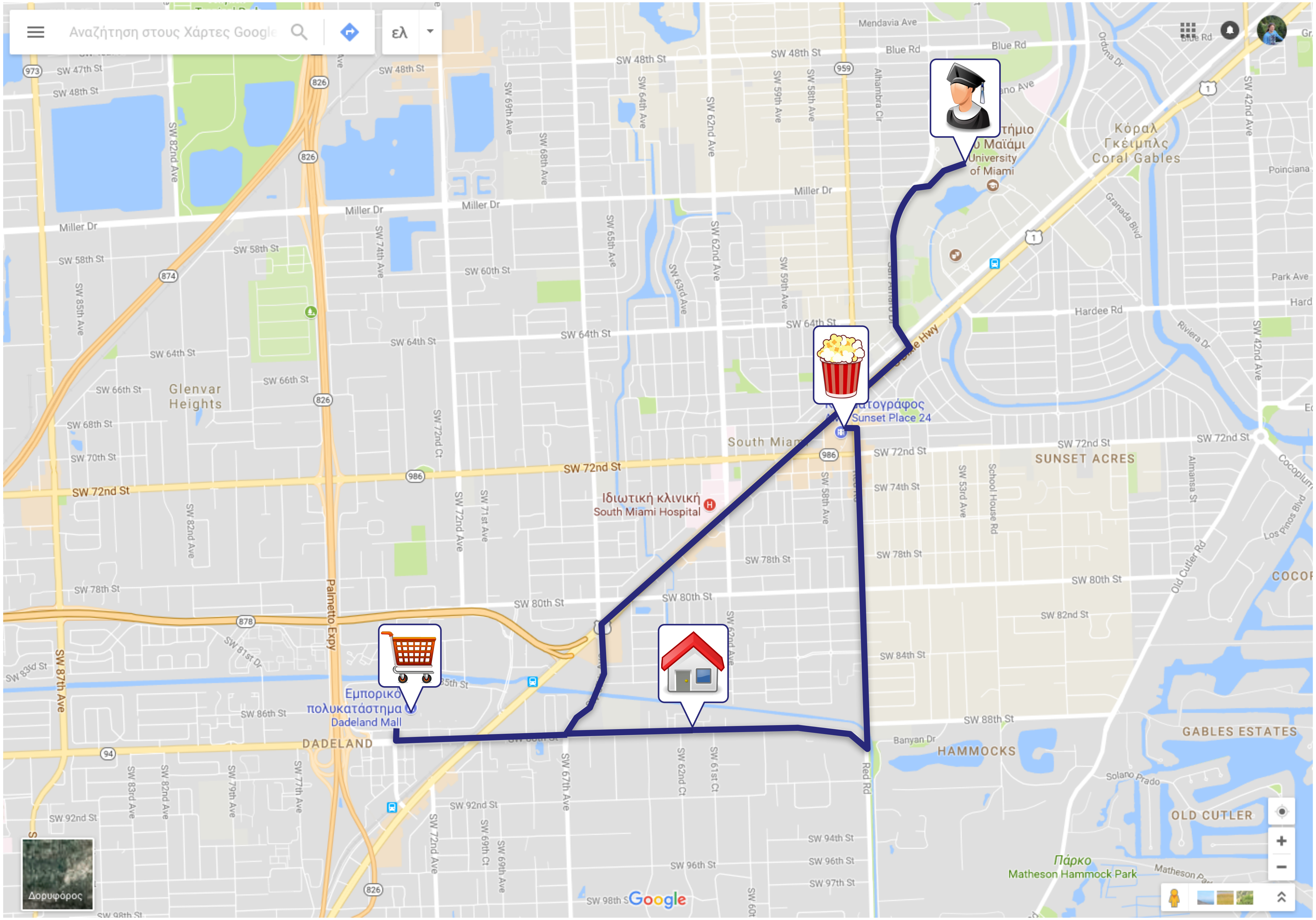
Overview

- Motivating Scenarios
- Preferred Network
- Most Preferred Unrestricted Path
- Most Preferred Near Shortest Path
- Conclusions & Future Work

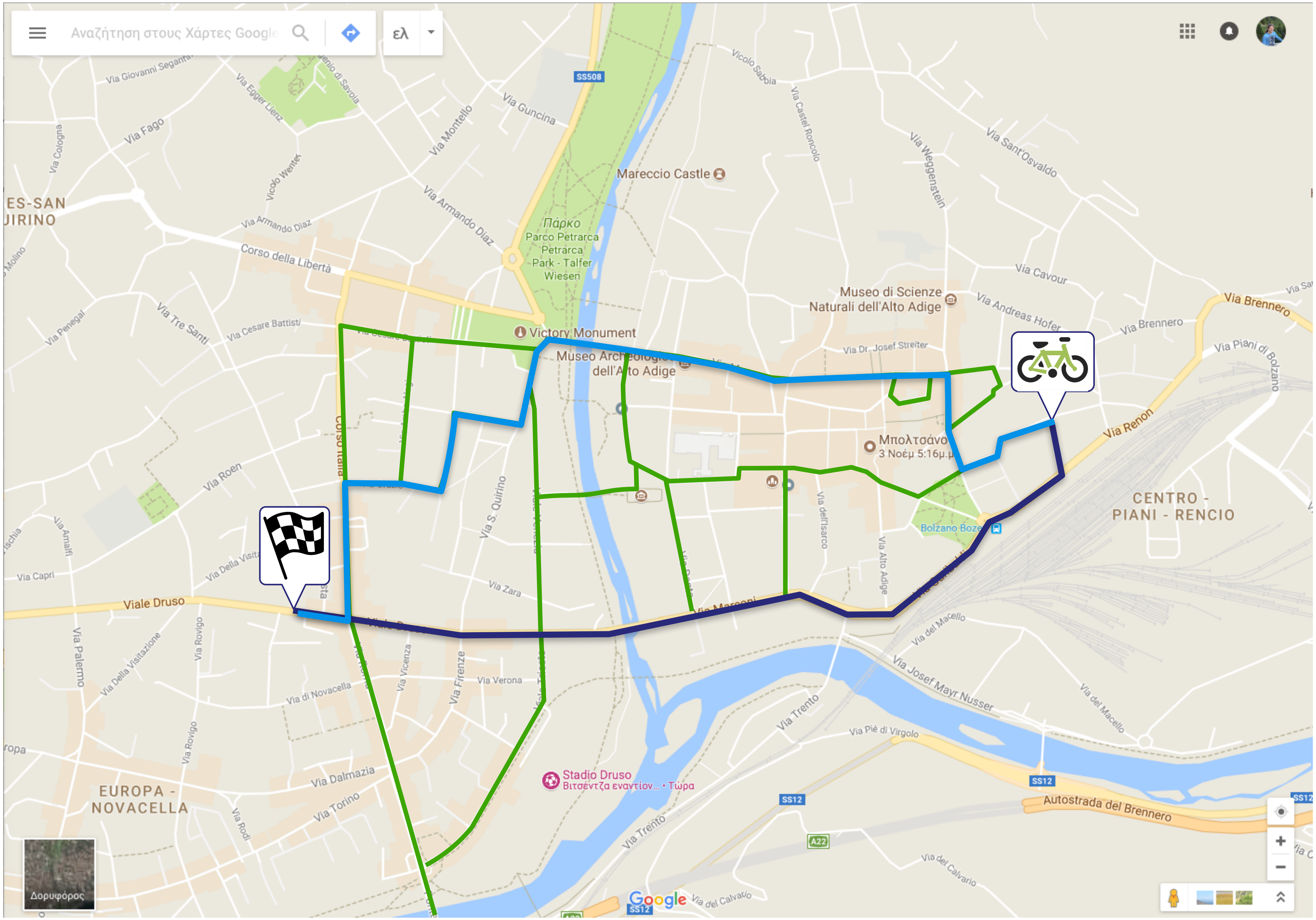
Example 1 - Scenic Route Planning



Example 2 - Familiar Roads



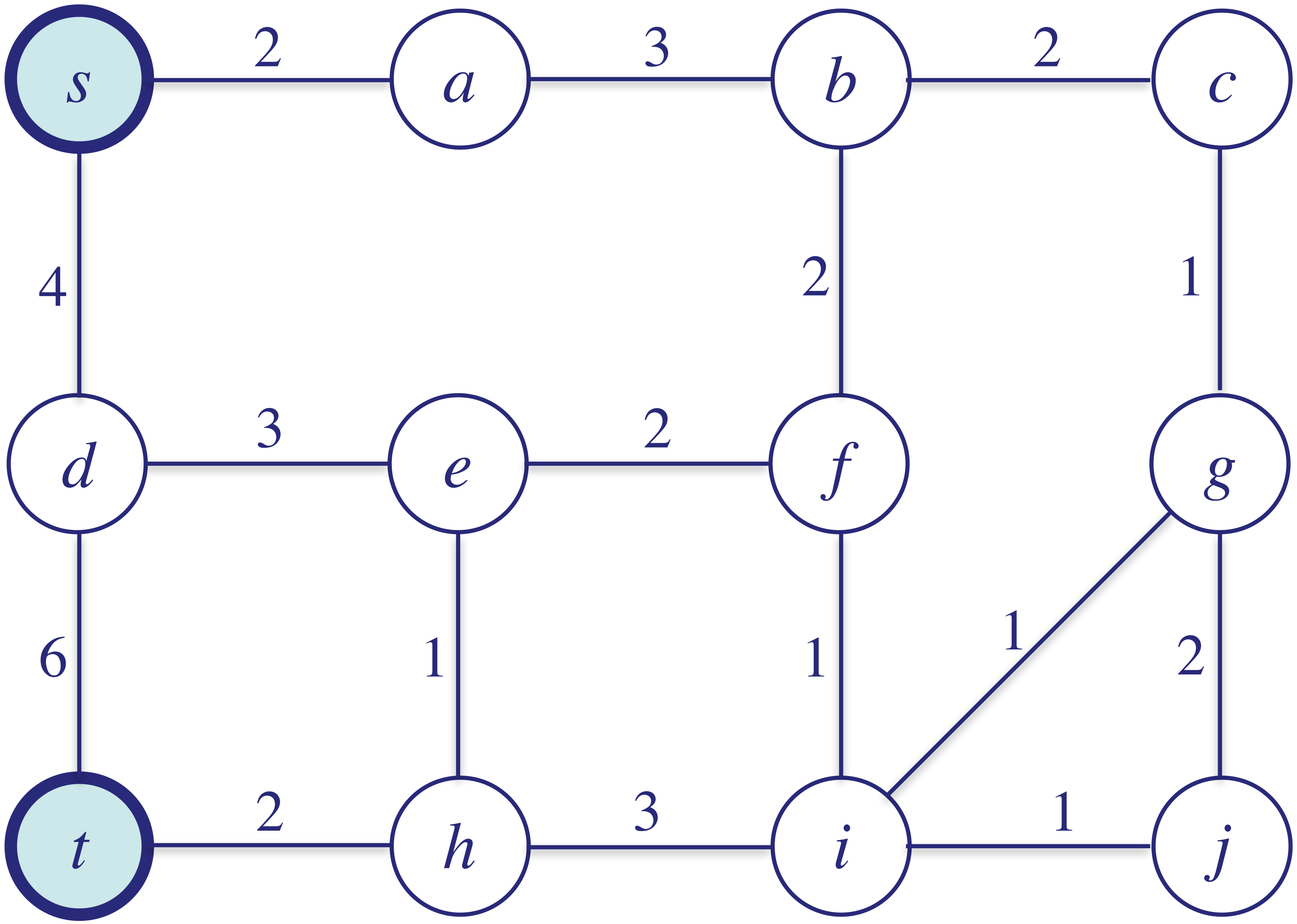
Example 3 - Bicycle Routes



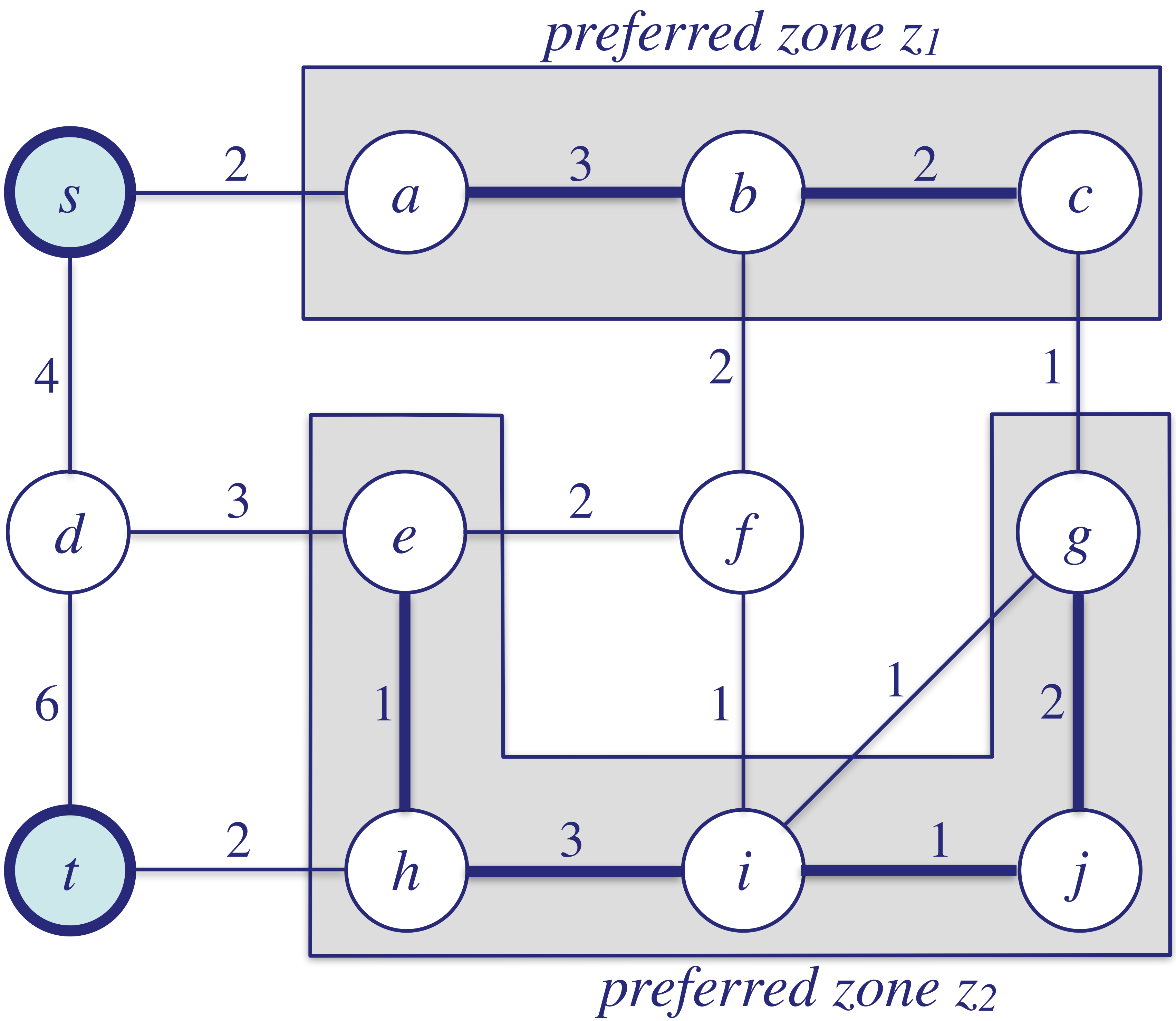
Overview

- Motivating Scenarios
- Preferred Network
- Most Preferred Unrestricted Path
- Most Preferred Near Shortest Path
- Conclusions & Future Work

Road Network



Preferred Network

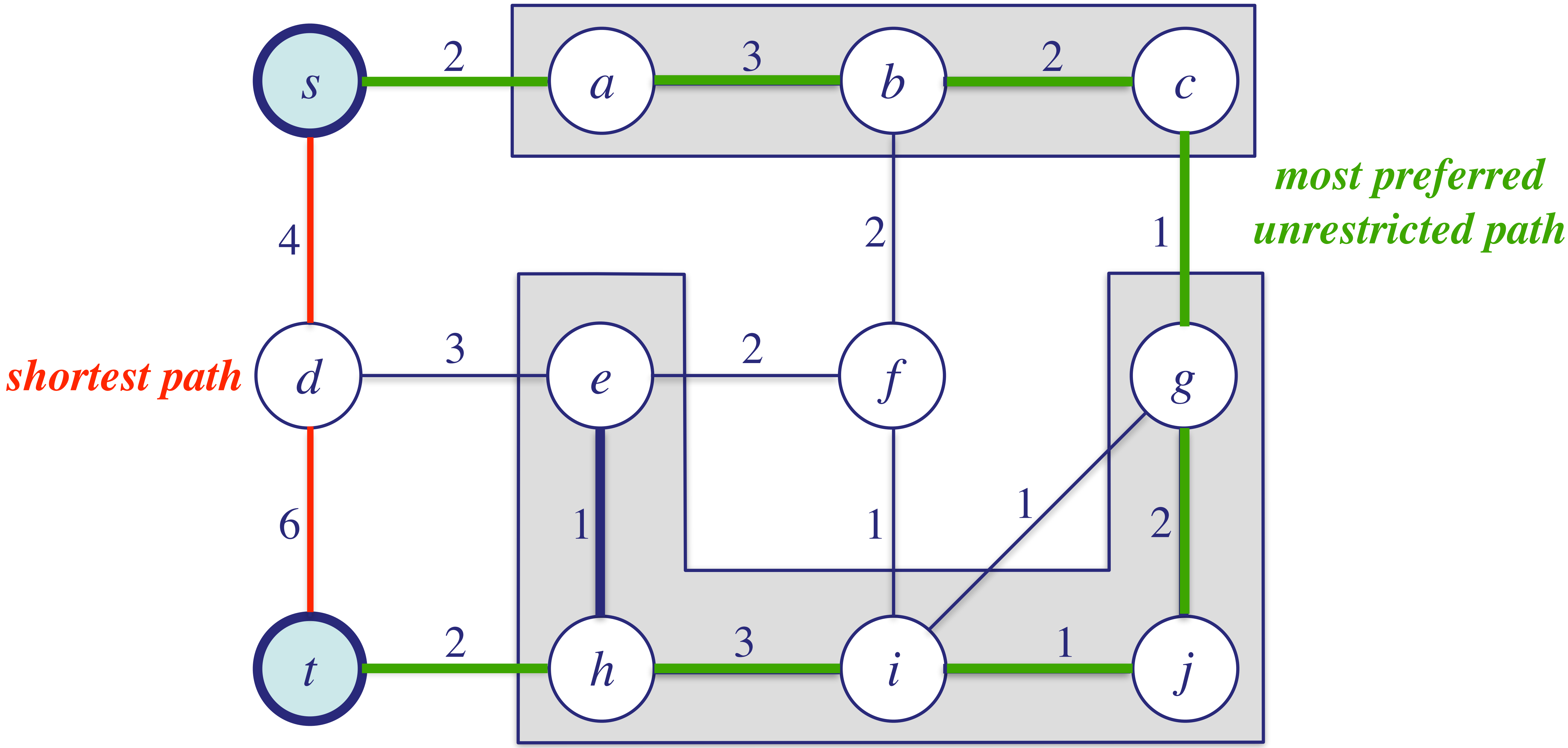


Overview

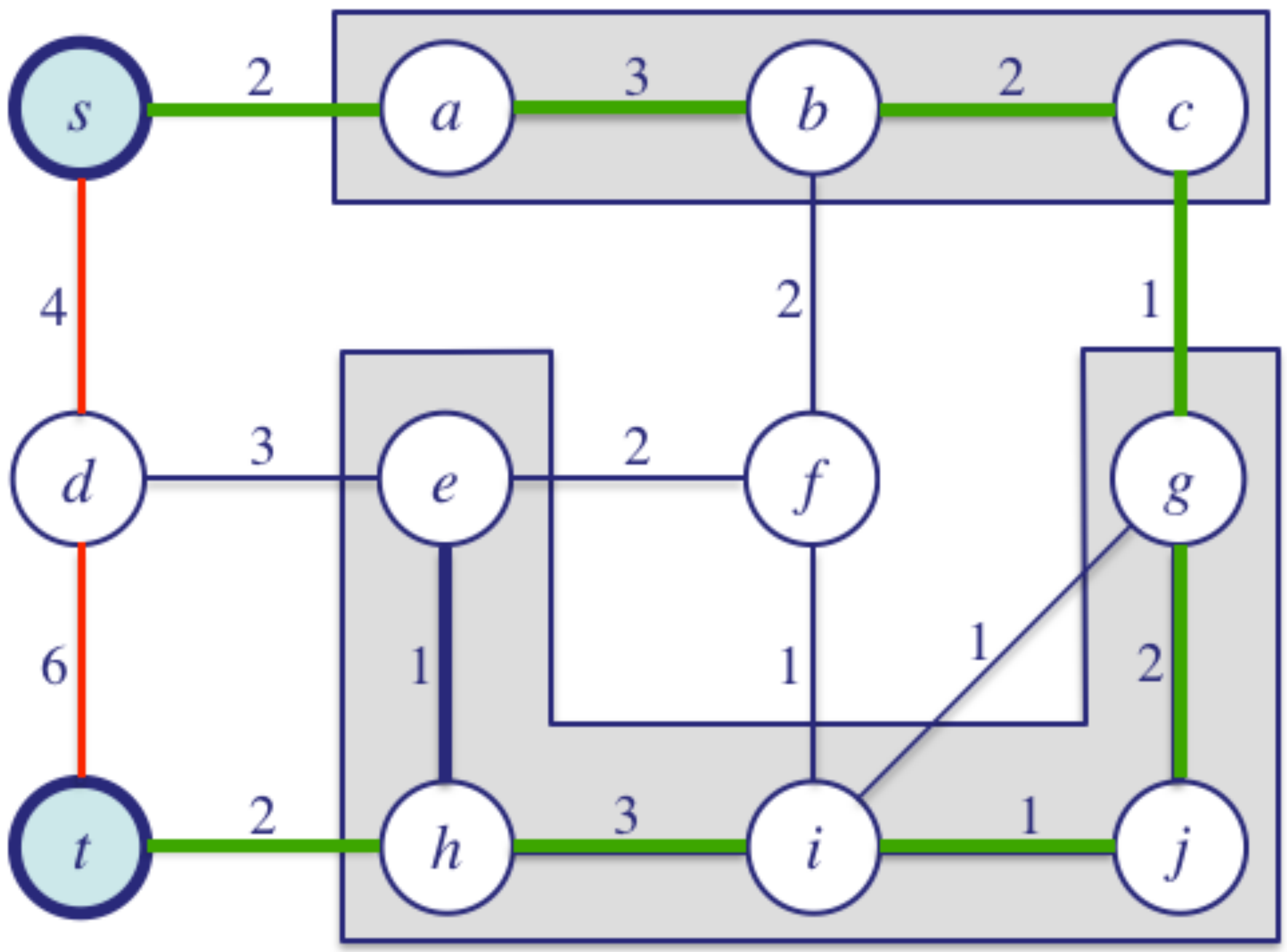
- Motivating Scenarios
- Preferred Network
- Most Preferred Unrestricted Path
- Most Preferred Near Shortest Path
- Conclusions & Future Work

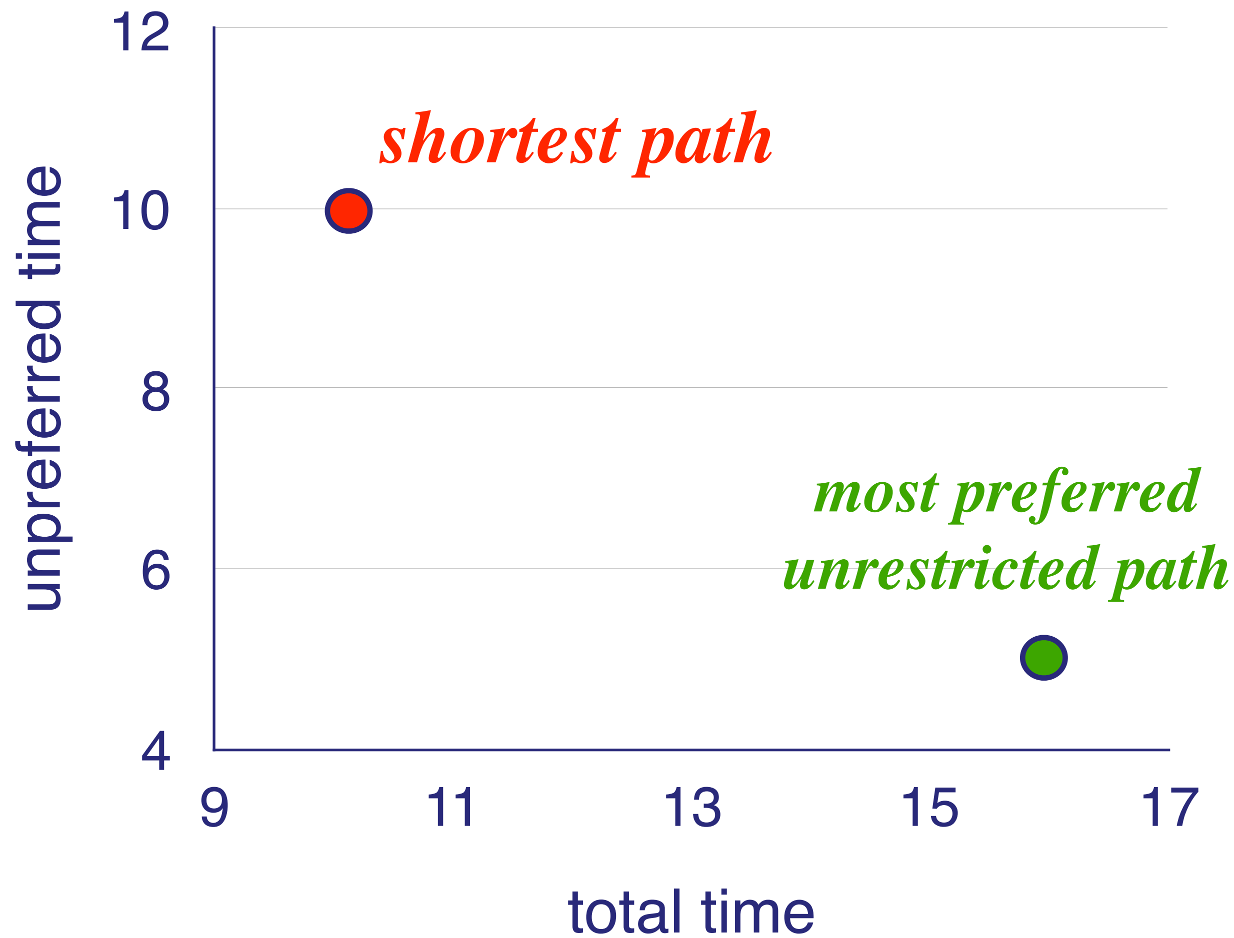
Most Preferred Unrestricted Path

- Find the path which ***minimizes*** the time spent ***outside*** the Preferred Network



Most Preferred Unrestricted Path





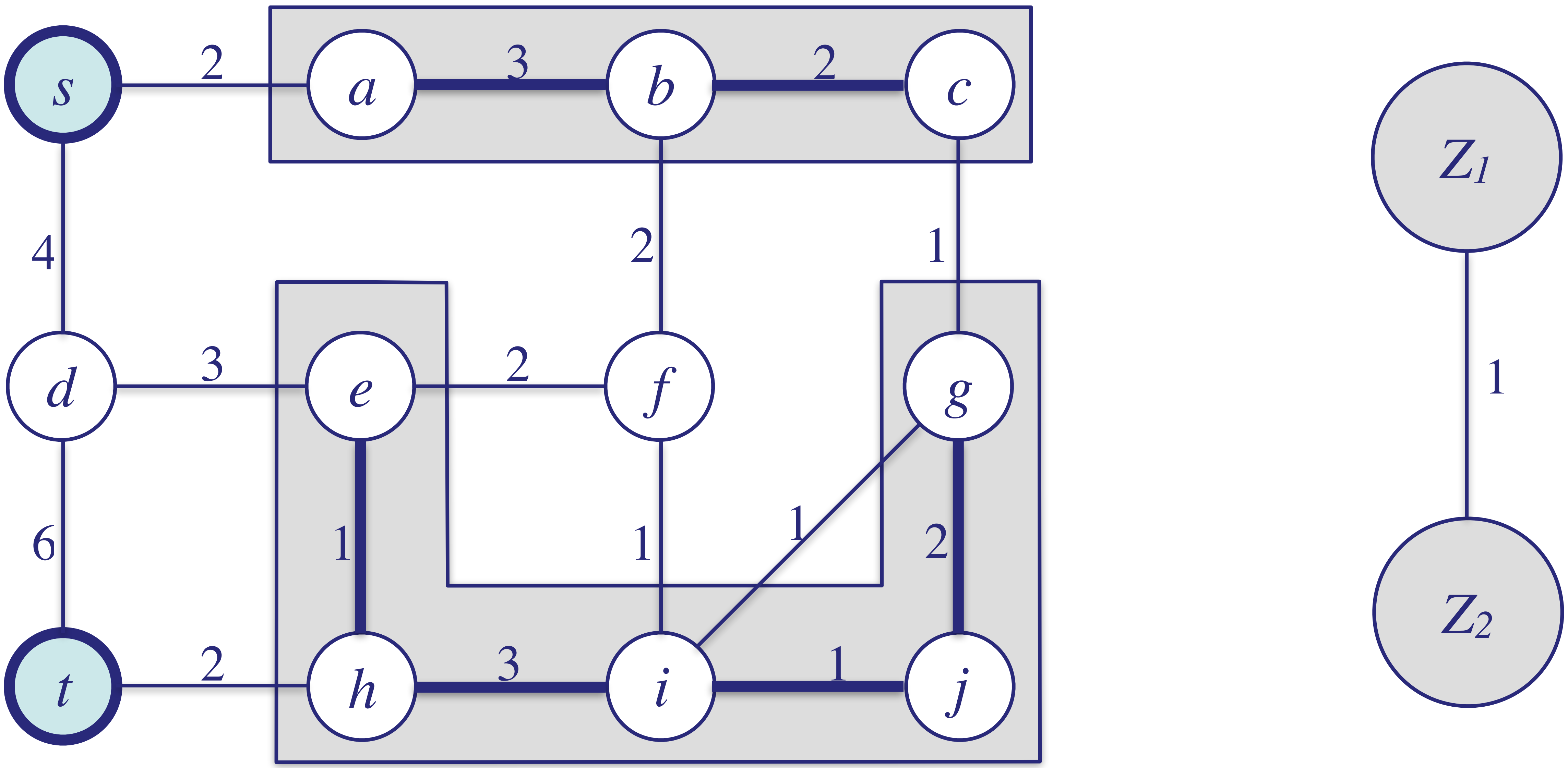
HyperEdges

- Introduced by *Aljubayrin et al* in *ICDE 2015 (Safest Path via Safe Zones)*
- Offline phase: Hypergraph construction
  1. Zones become HyperNodes
  2. HyperEdges are added between HyperNodes/Zones
  3. Weights are determined by shortest paths connecting different zones

HyperEdges

Original Network

Hypergraph

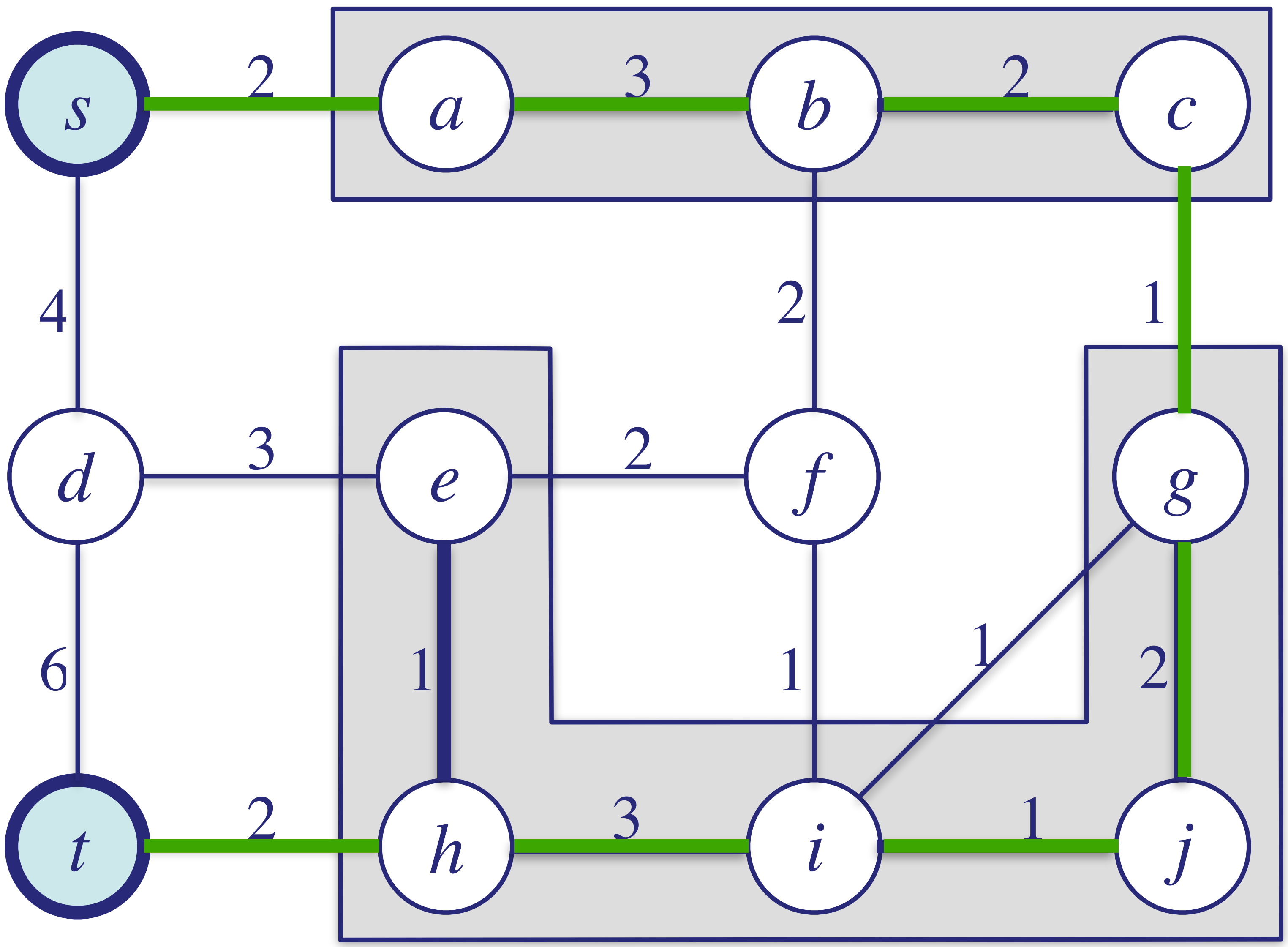


HyperEdges

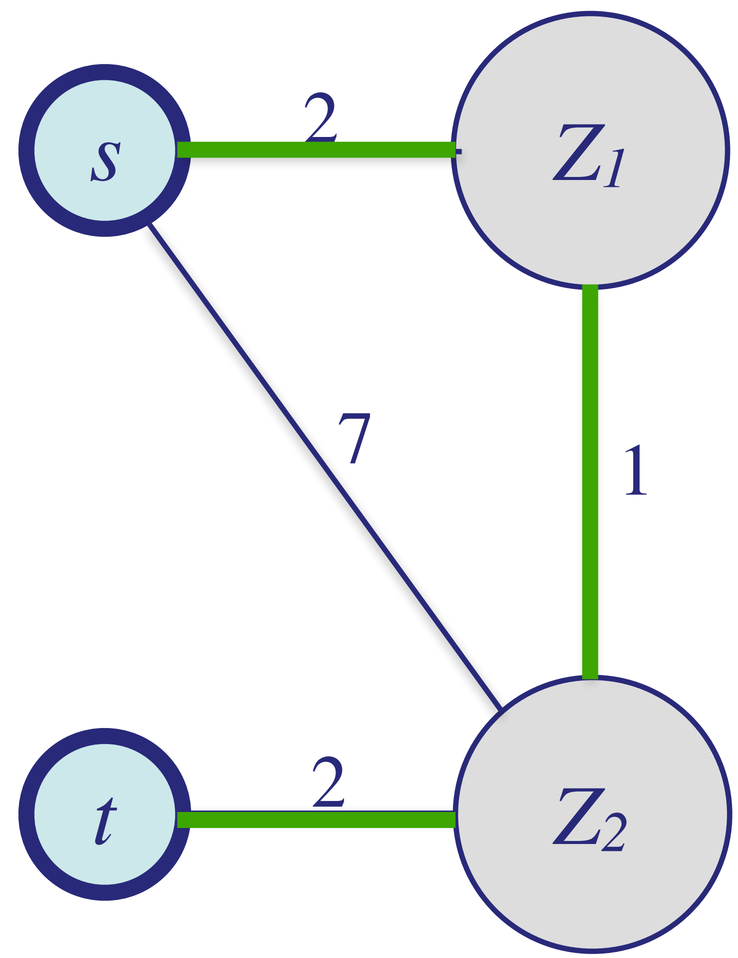
- Introduced by *Aljubayrin et al* [ICDE'15] *(Safest Path via Safe Zones)*
- Offline phase: Hypergraph construction
  1. Zones become HyperNodes
  2. HyperEdges are added between HyperNodes/Zones
  3. Weights are determined by shortest paths connecting different zones
- Online phase: Query Processing
  1. Add source and target nodes to Hypergraph
  2. Run shortest path query over the Hypergraph

HyperEdges

Original Network



Hypergraph

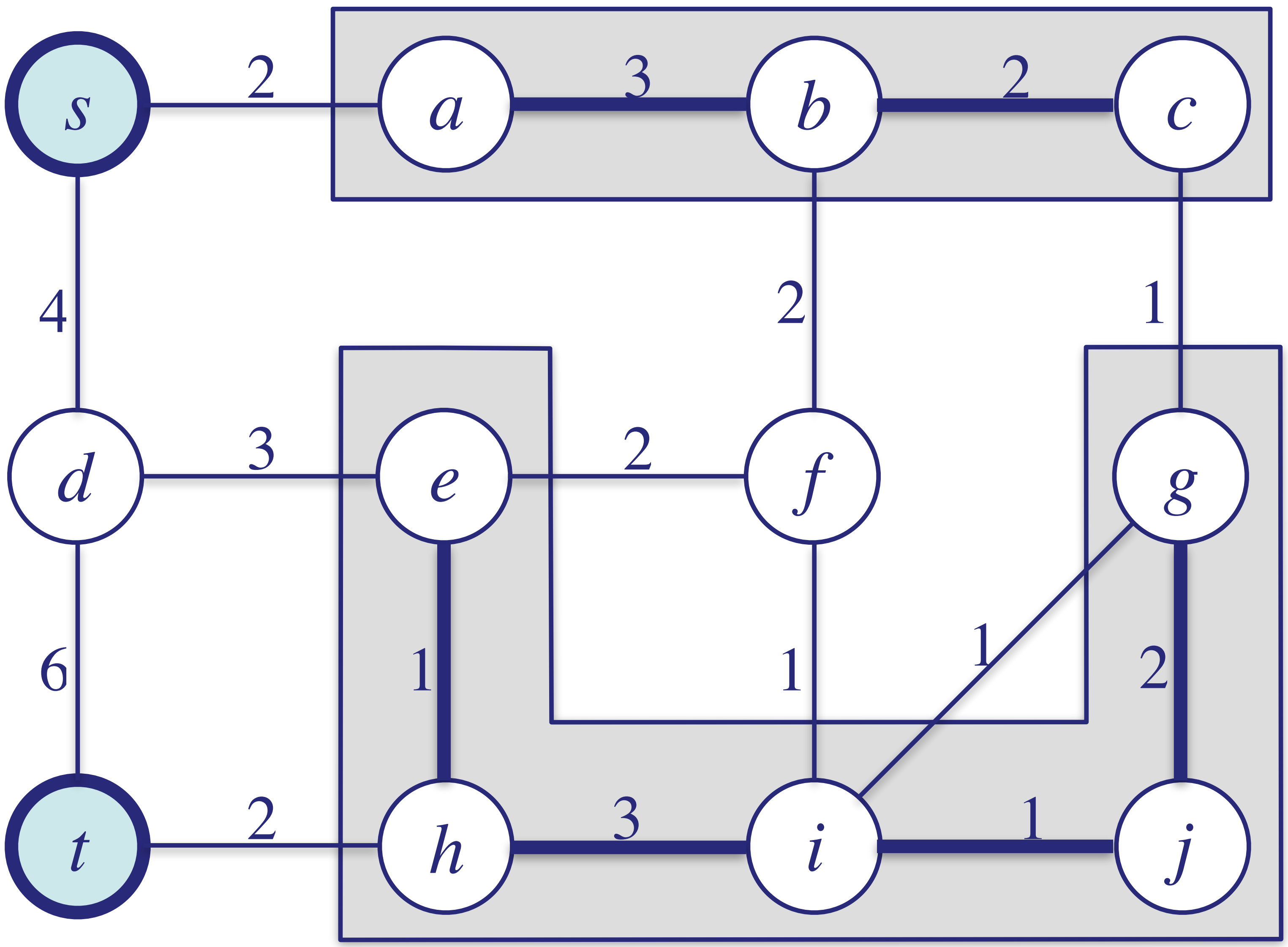


Compressed Network

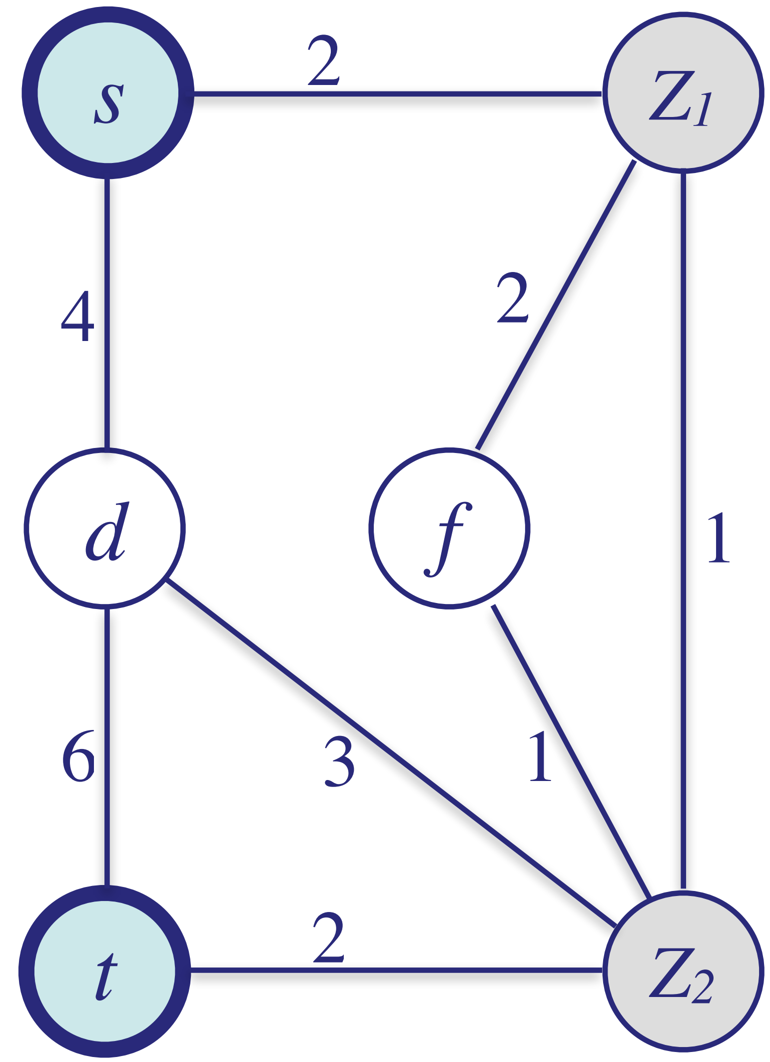
- Offline phase: Build Compressed Network
  1. Replace every zone with a node
  2. Add edges between each new node and each node previously connected with the associated zone

Compressed Network

Original Network



Compressed Network

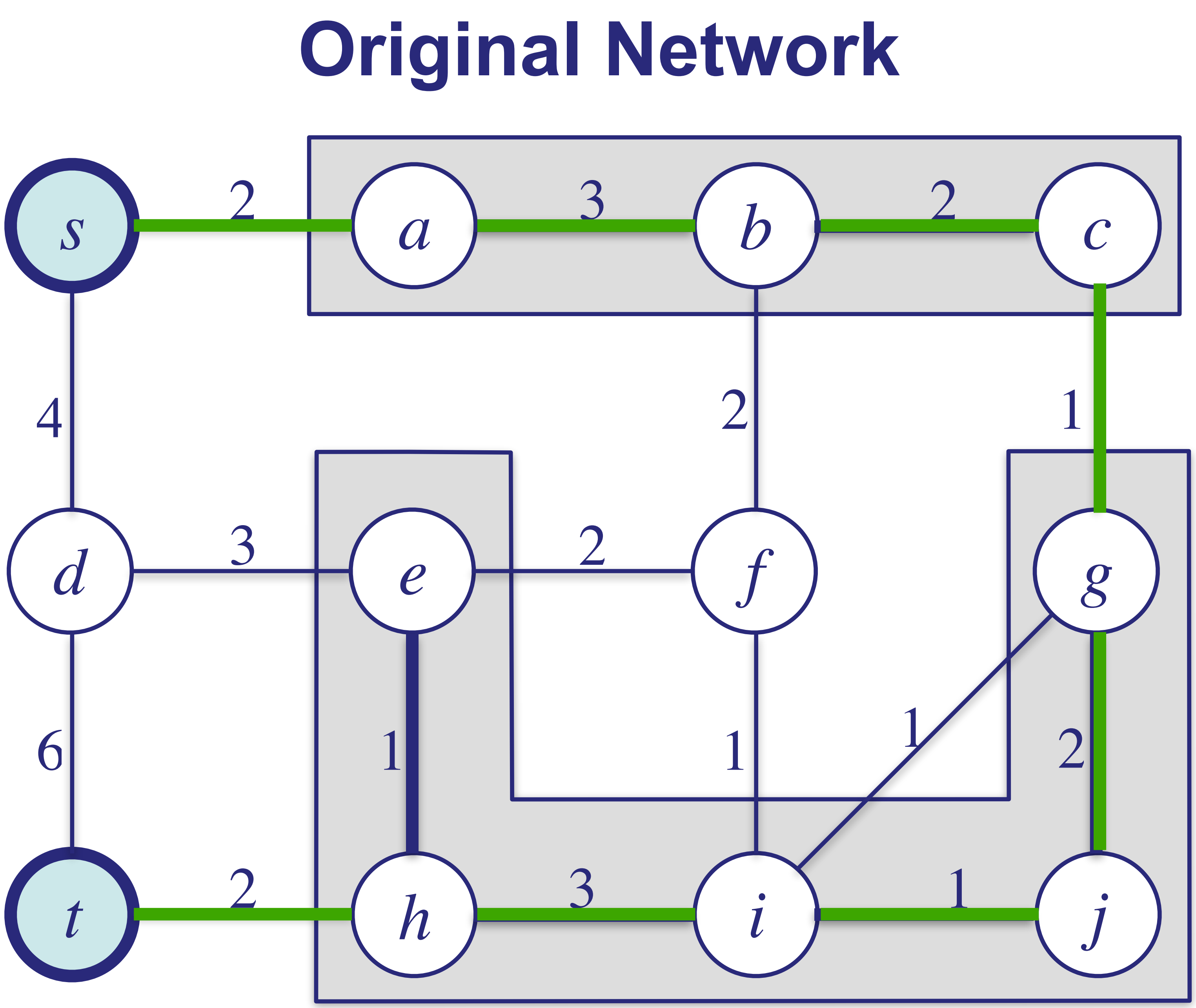


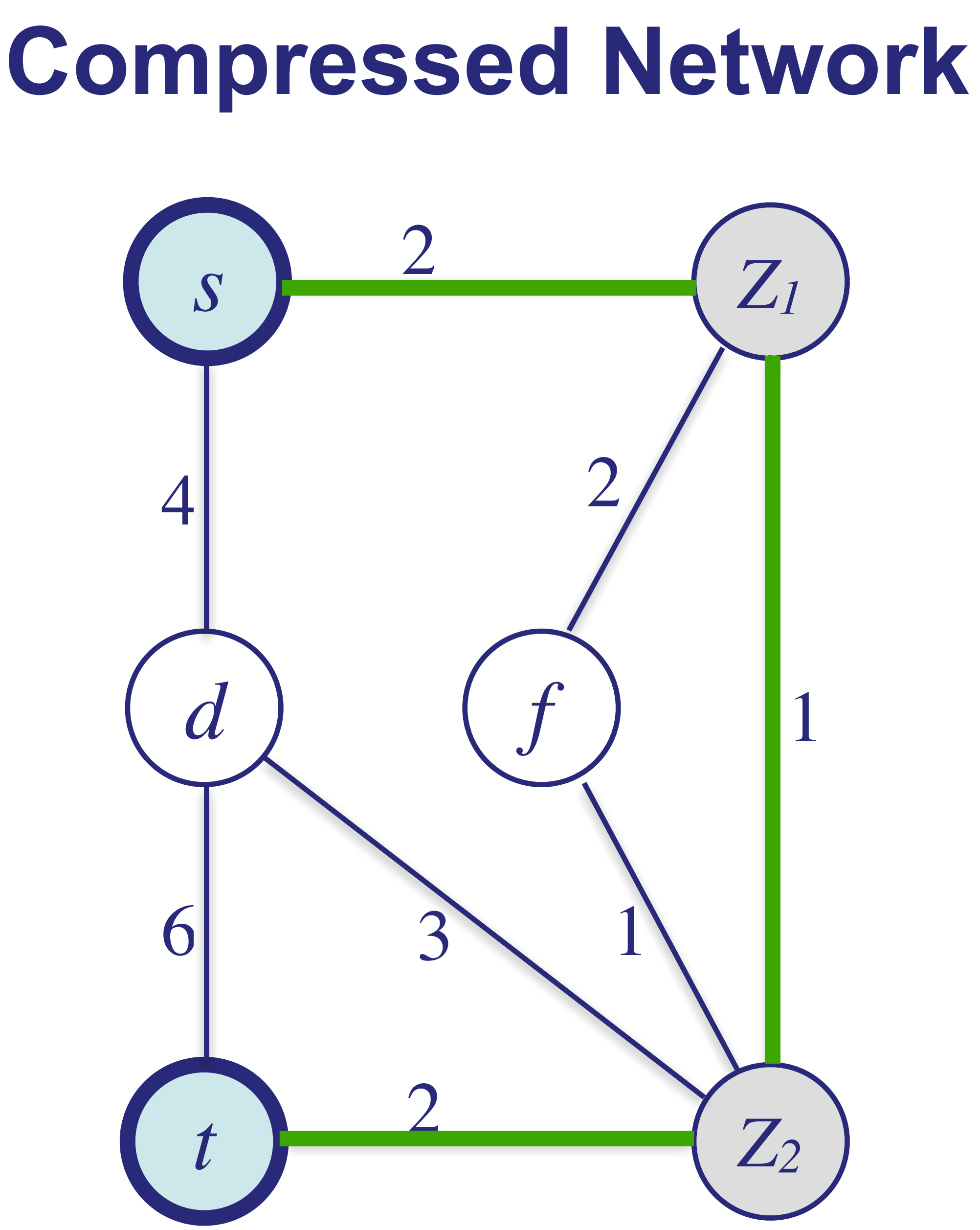
Compressed Network

- Offline phase: Build Compressed Network
  1. Replace every zone with a node
  2. Add edges between each new node and each node previously connected with the associated zone
- Online phase: Query Processing

  Run shortest path query over the Compressed Network

Compressed Network

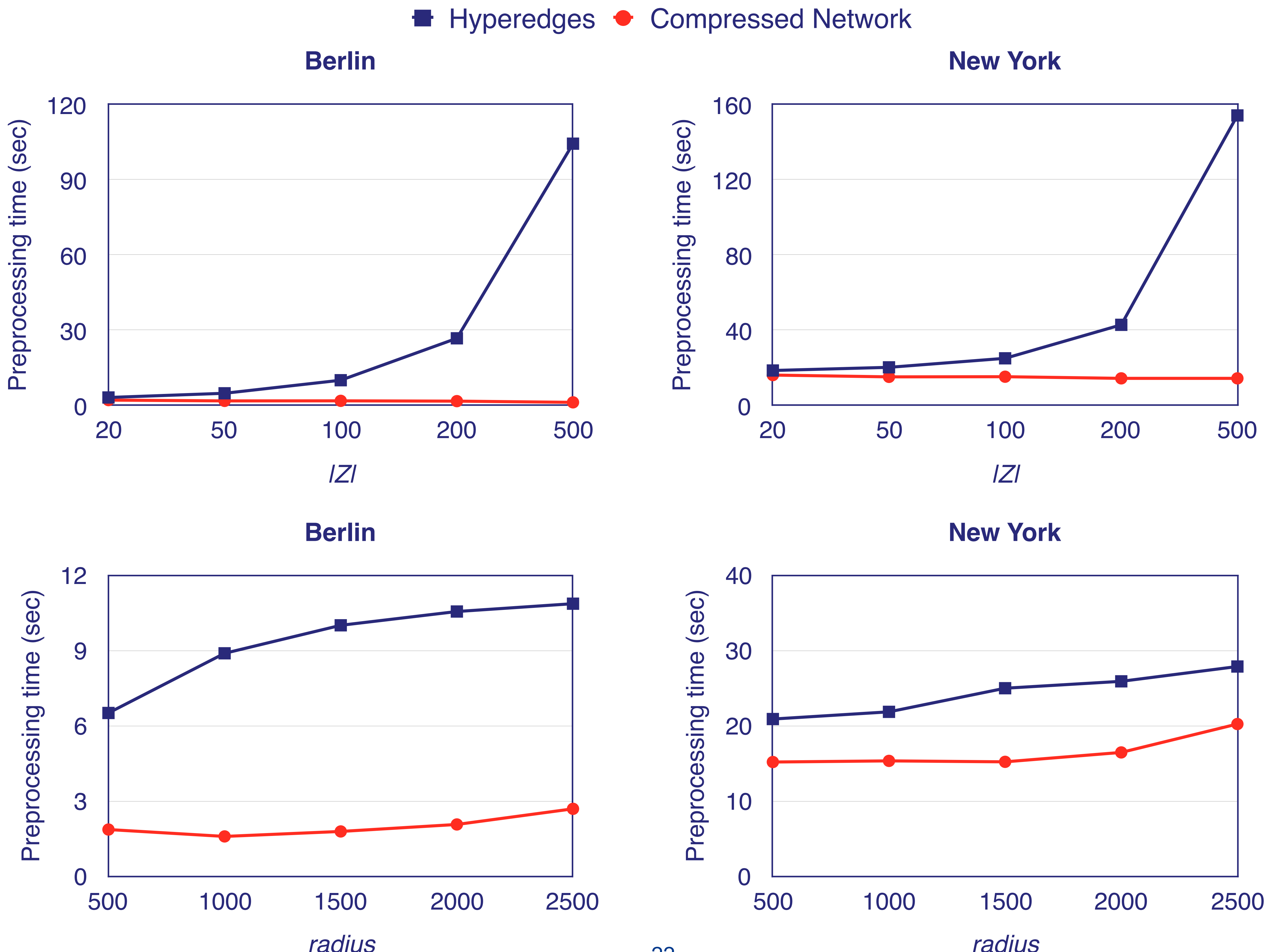




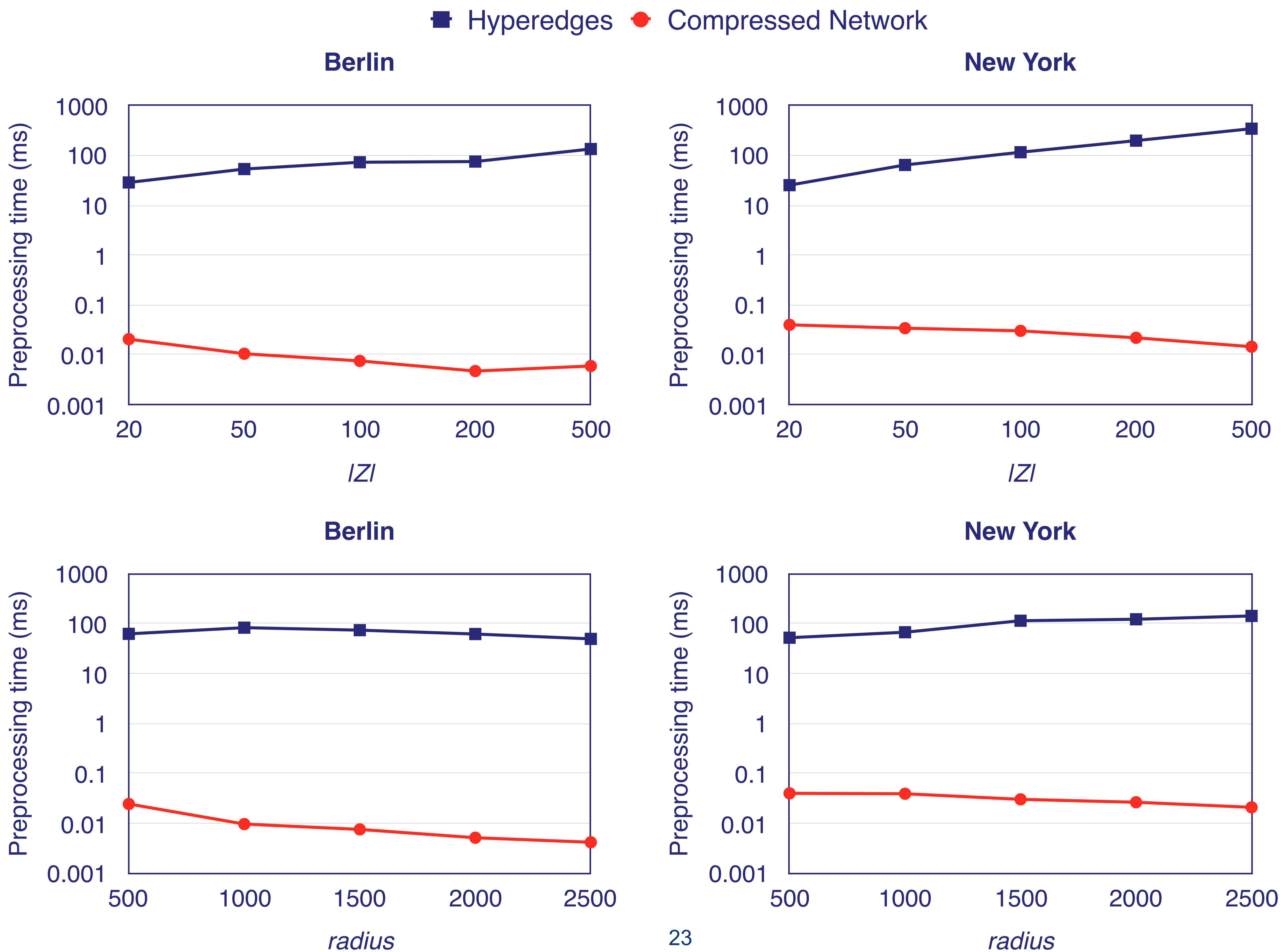
Experimental Evaluation MPUP

- Two datasets:
  - Berlin (37,126 nodes and 102,260 edges)
  - New York (264,346 nodes and 730,100 edges)
- Two experiments:
  - varying number of zones $|Z|$
  - varying radius of zones r
- Default values: $|Z| = 100$, $r = 1500$m
- We use Contraction Hierarchies to optimize both preprocessing and query processing

MPUP - Preprocessing



MPUP - Query Processing



Overview

- Motivating Scenarios
- Preferred Network
- Most Preferred Unrestricted Path
- Most Preferred Near Shortest Path
- Conclusions & Future Work

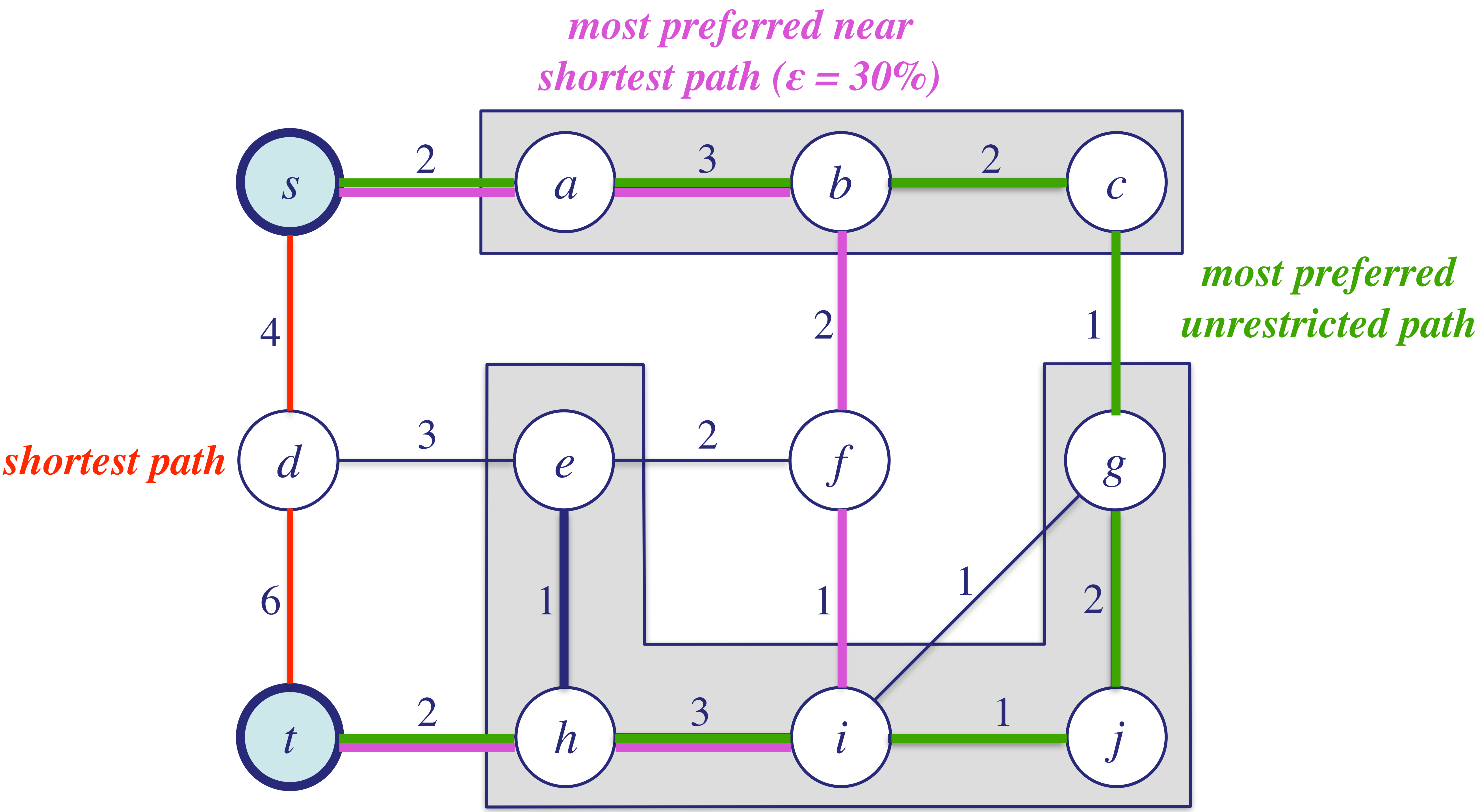
Most Preferred Near Shortest Path

- Find the path which

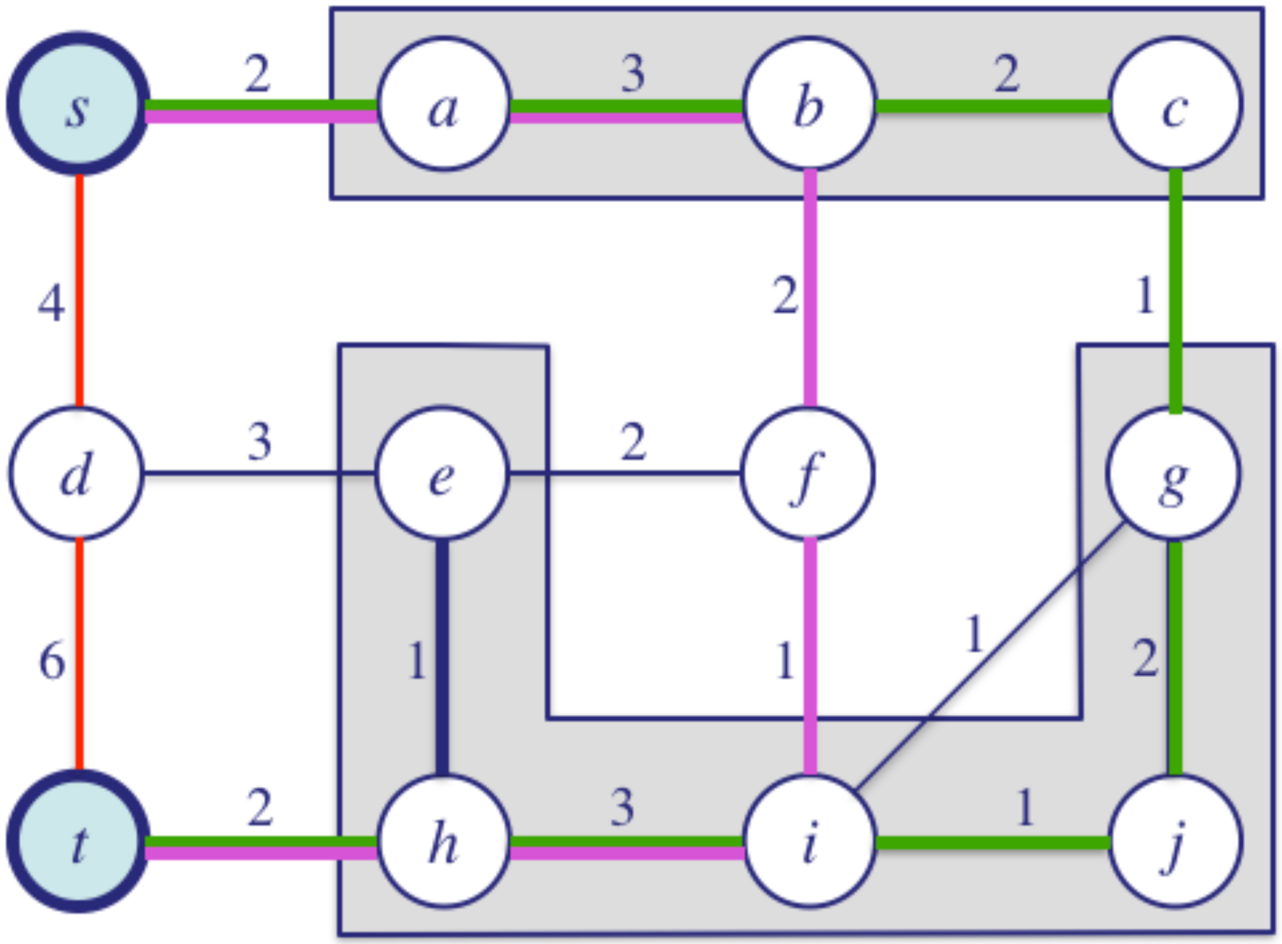
  (1) ***minimizes*** the time spent ***outside*** the Preferred Network

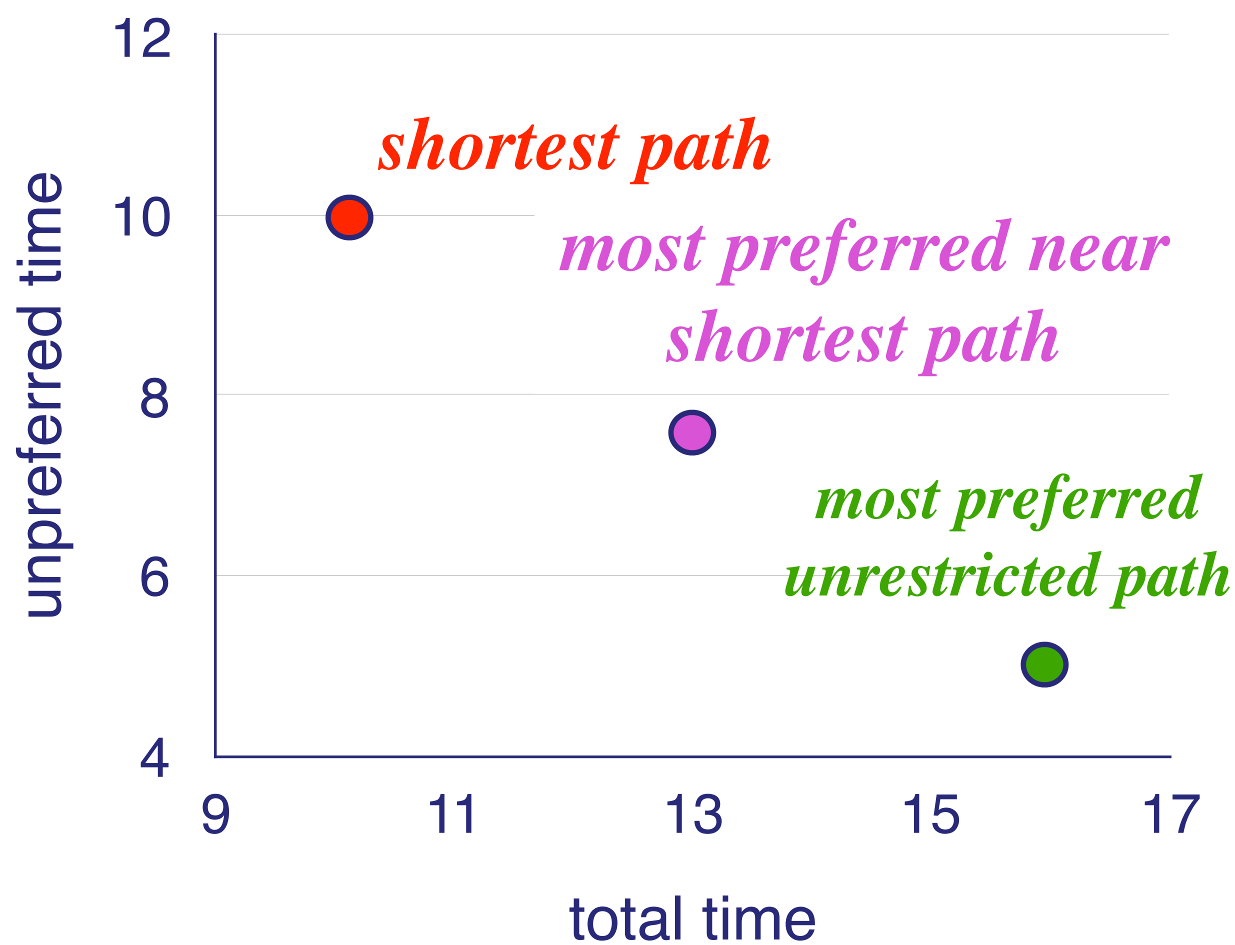
  (2) is at most ***X% longer*** than the ***shortest path***

Most Preferred Near Shortest Path



Most Preferred Near Shortest Path



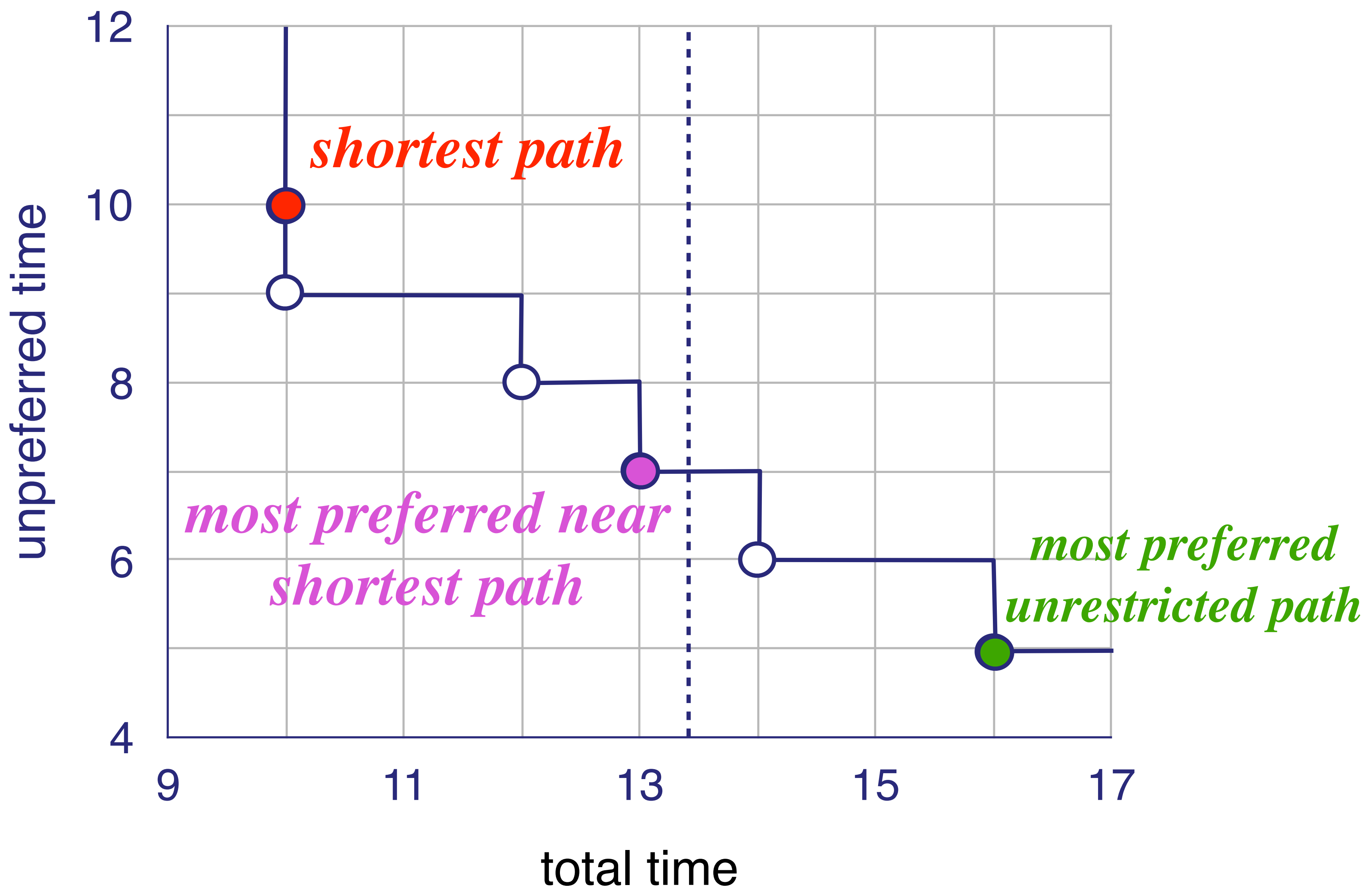


Algorithms

- Advanced Route Skyline Computation (ARSC)

Advanced Route Skyline Computation

- Compute the entire Route Skyline
- Then retrieve the MPNSP

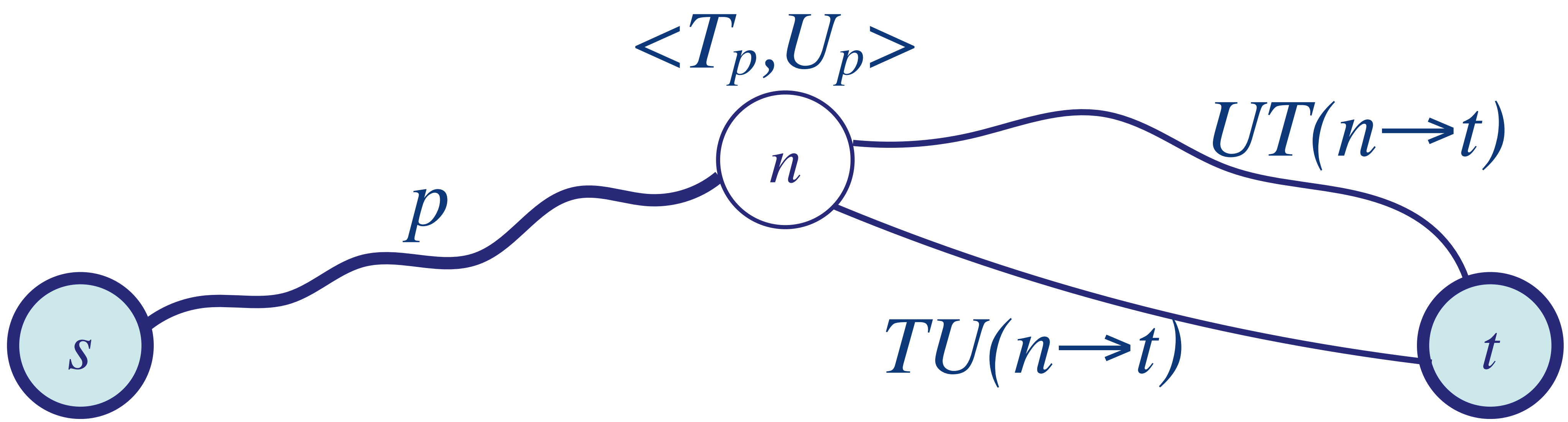


Algorithms

- Advanced Route Skyline Computation (ARSC)
  - prunes dominated paths
- **Algorithm ALGO-U**
  - prunes dominated paths
  - employs upper bounds for the unpreferred time

ALGO-U directs the search towards the desired result of the skyline

Algorithm ALGO-U



Upper Bound U*

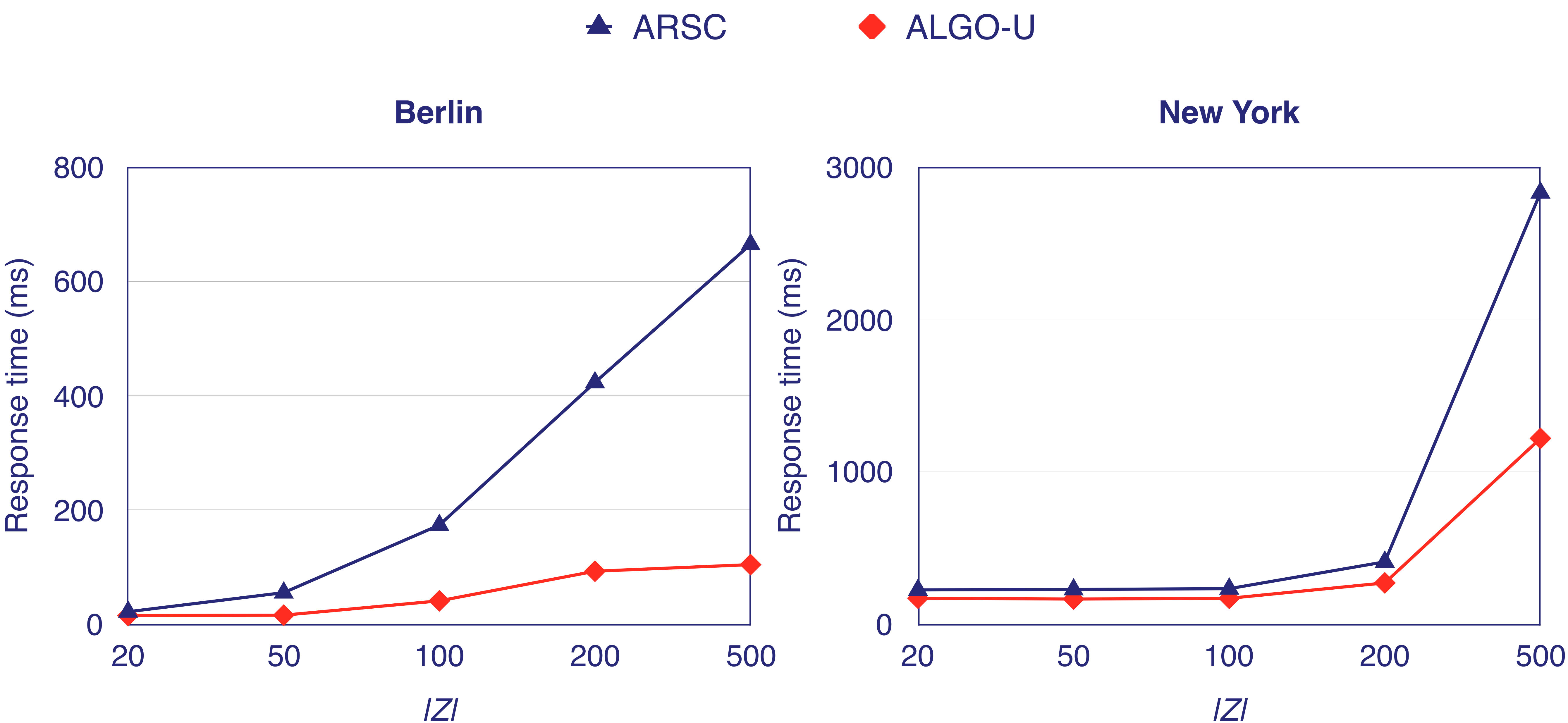
$$\min\begin{cases} U_p + TU(n \rightarrow t).U, & \textbf{\textit{if}}\ \boxed{T_p + TU(n \rightarrow t).T \leq (1+\varepsilon)\cdot d(s,t)} \\ U_p + UT(n \rightarrow t).U, & \textbf{\textit{if}}\ \boxed{T_p + UT(n \rightarrow t).T \leq (1+\varepsilon)\cdot d(s,t)} \end{cases}$$

if the extension does not violate ε

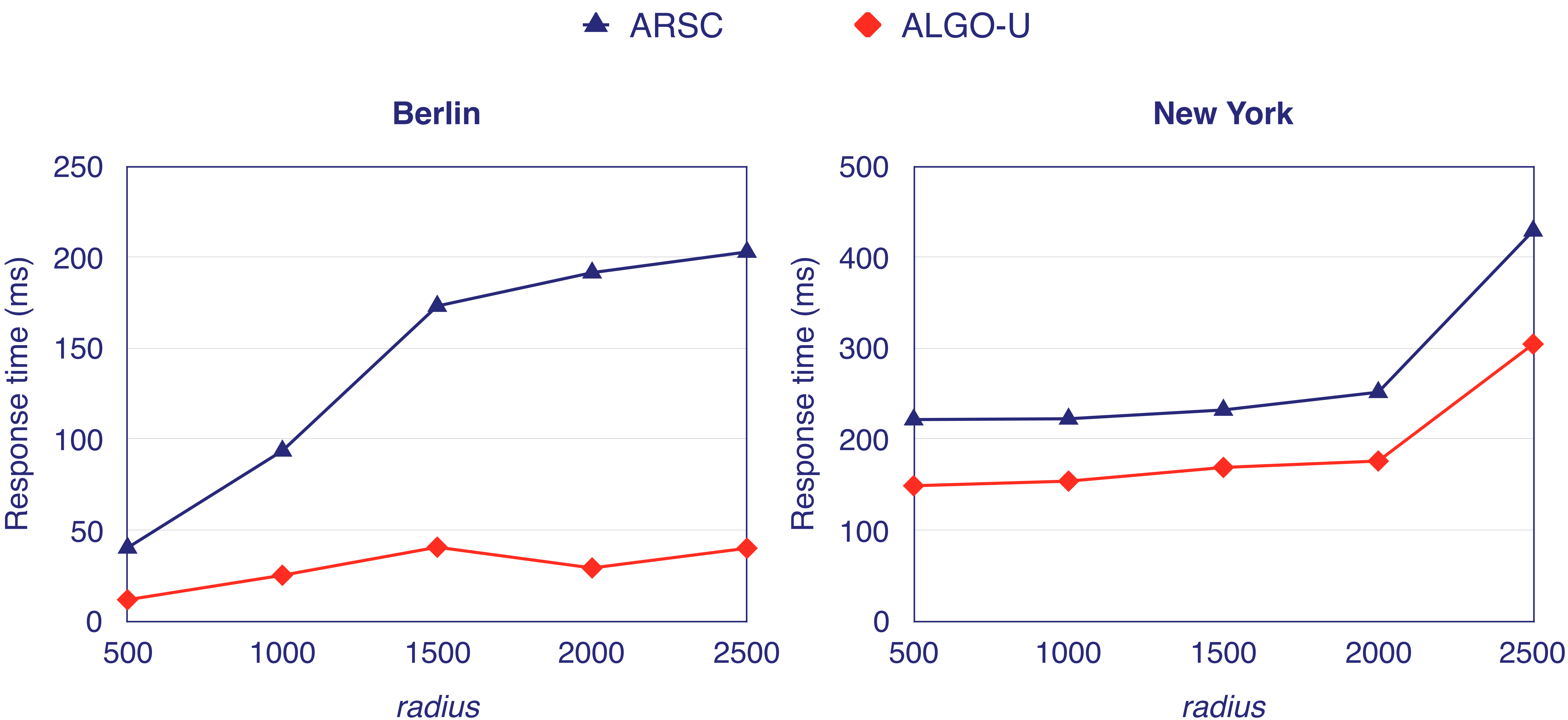
Experimental Evaluation MPNSP

- Two datasets:
  - Berlin (37,126 nodes and 102,260 edges)
  - New York (264,346 nodes and 730,100 edges)
- Two experiments:
  - varying number of zones $|Z|$
  - varying radius of zones r
  - varying threshold ε
- Default values: $|Z| = 100$, $r = 1500$m, $\varepsilon = 30\%$

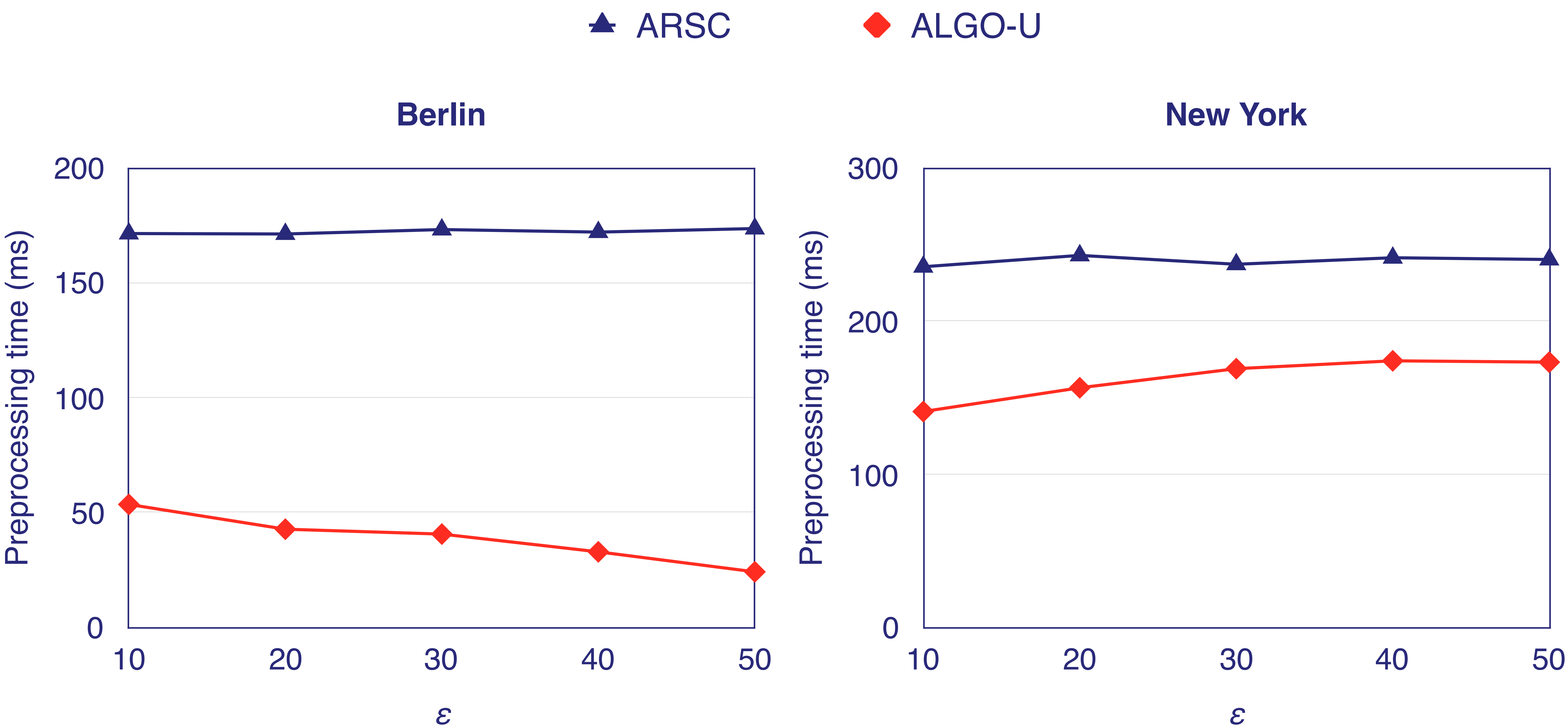
Experimental Evaluation MPNSP



Experimental Evaluation MPNSP



Experimental Evaluation MPNSP



Overview

- Motivating Scenarios
- Preferred Network
- Most Preferred Unrestricted Path
- Most Preferred Near Shortest Path
- Conclusions & Future Work

Conclusions and Future Work

- Conclusions
  - ✓ We studied the problem of Finding the Most Preferred Path on road networks
  - ✓ The Compressed Network approach improves the state-of-the-art for MPUP
  - ✓ We introduced MPNSP along with ALGO-U which also improves the state-of-the-art
- Future Work
  - ✓ Investigate pre-processing methods for MPNSP
  - ✓ Study methods to extract Preferred Zones

Thank you!