Exact and Approximate Algorithms for Finding k-Shortest Paths with Limited Overlap

Theodoros Chondrogiannis[1] Panagiotis Bouros[2] Johann Gamper[1] Ulf Leser[3]

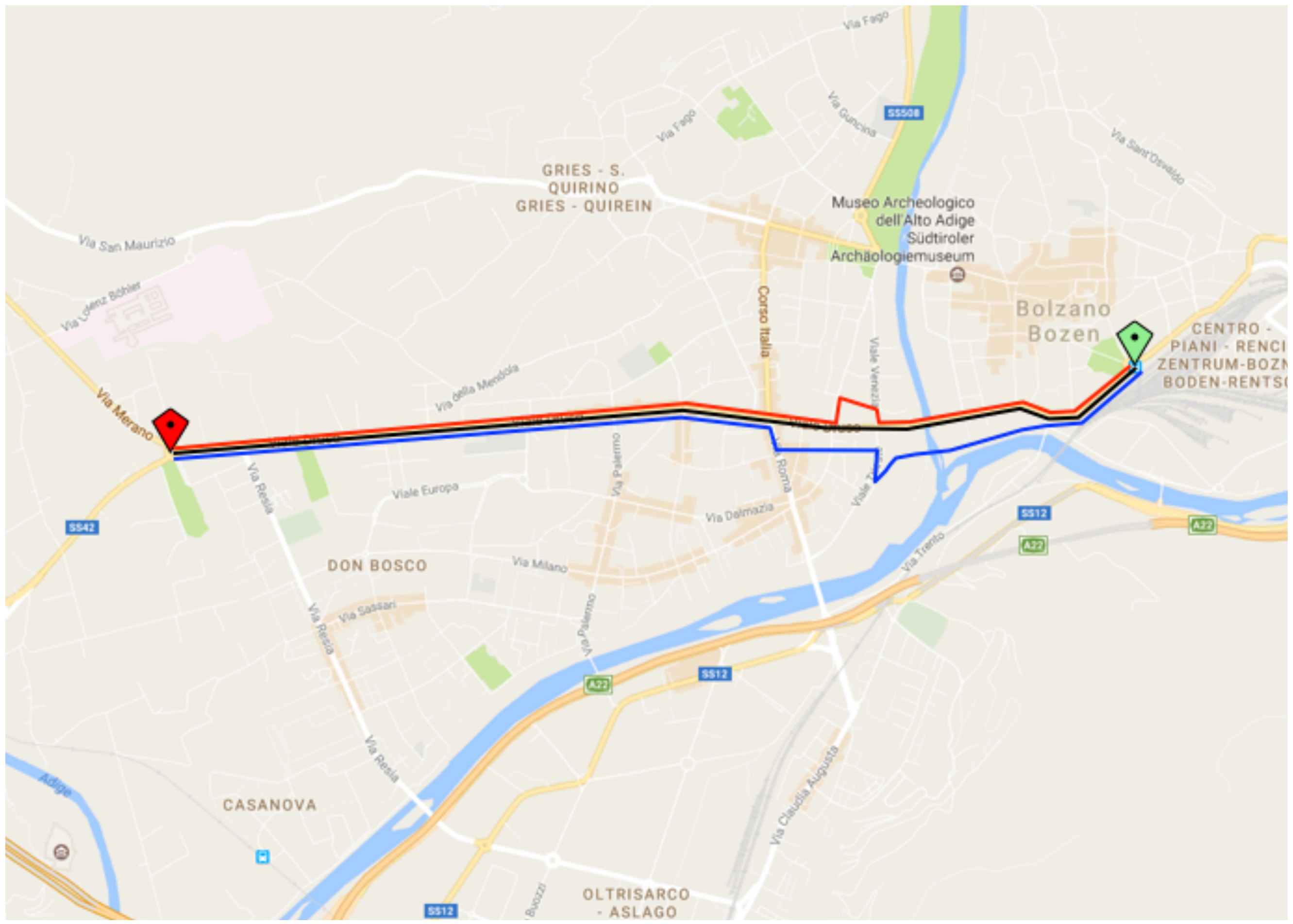
[1] Free University of Bozen-Bolzano

[2] Aarhus University

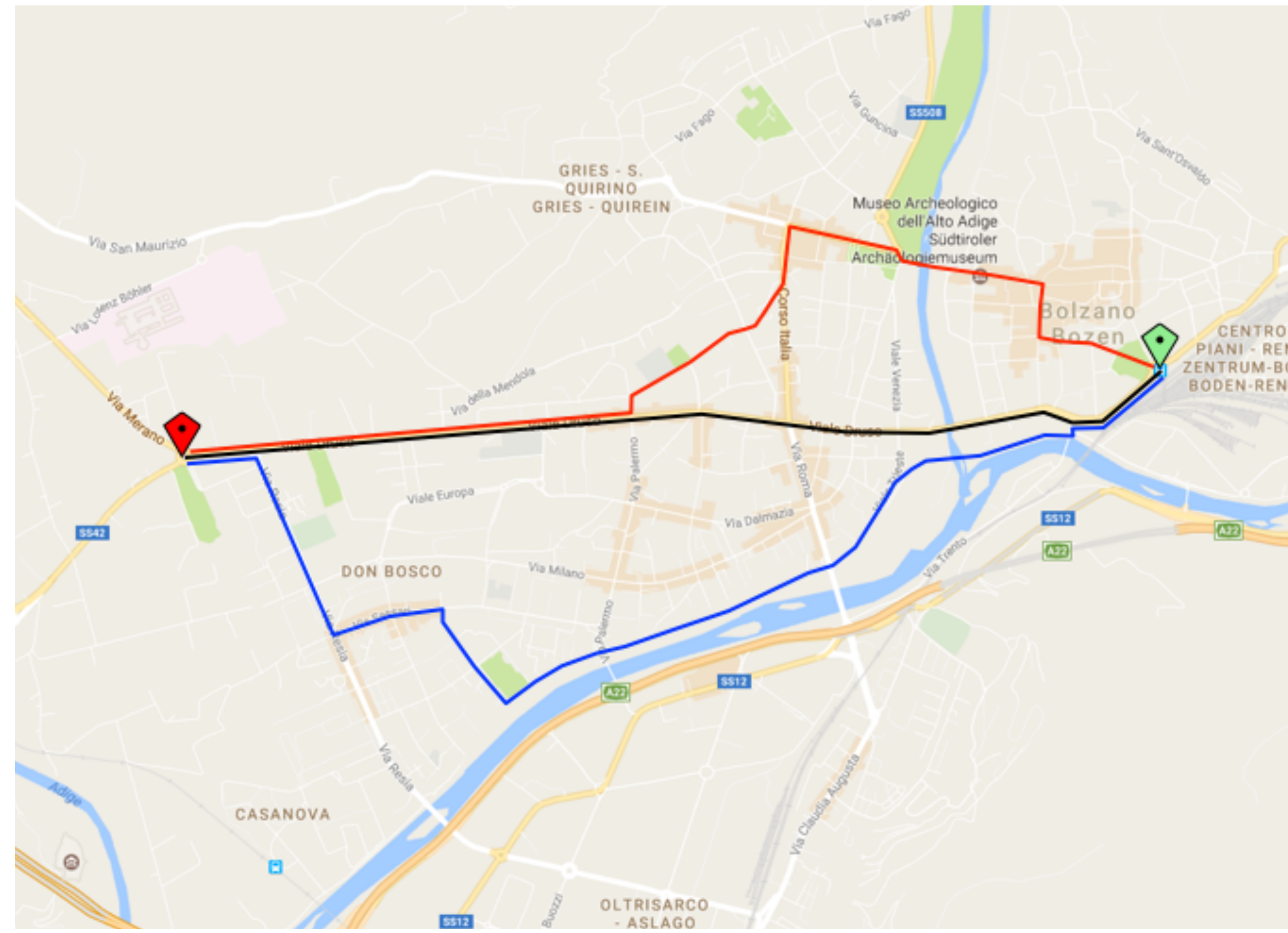
[3] Humboldt-Universität zu Berlin

Venice, 24th March 2017

Alternative Routing on Road Networks



Short but very similar paths



Slightly longer but more dissimilar paths

Overview

- kSPwLO Overview
- MultiPass Algorithm
- Approximate Algorithms
  - OnePass$^+$
  - ESX
- Experimental Evaluation
- Conclusions & Future Work

Overview

- kSPwLO Overview
- MultiPass Algorithm
- Approximate Algorithms
  - OnePass$^+$
  - ESX
- Experimental Evaluation
- Conclusions & Future Work

k-SPwLO Definition [GIS'16]

- **k-Shortest Paths with Limited Overlap**
- A *k-SPwLO* query ***q(G,s,t,k,θ)*** returns a set of *k* paths from *s* to *t* in *G*, sorted by length in increasing order, such that:

  (a) the set includes the shortest path $p_0(s \longrightarrow t)$,

  (b) every path is dissimilar to its predecessors (all the shorter paths in the set) w.r.t. a similarity threshold θ,

  (c) each alternative path added to the *k-SPwLO* set is the shortest among all other alternatives

Path Similarity [GIS'16]

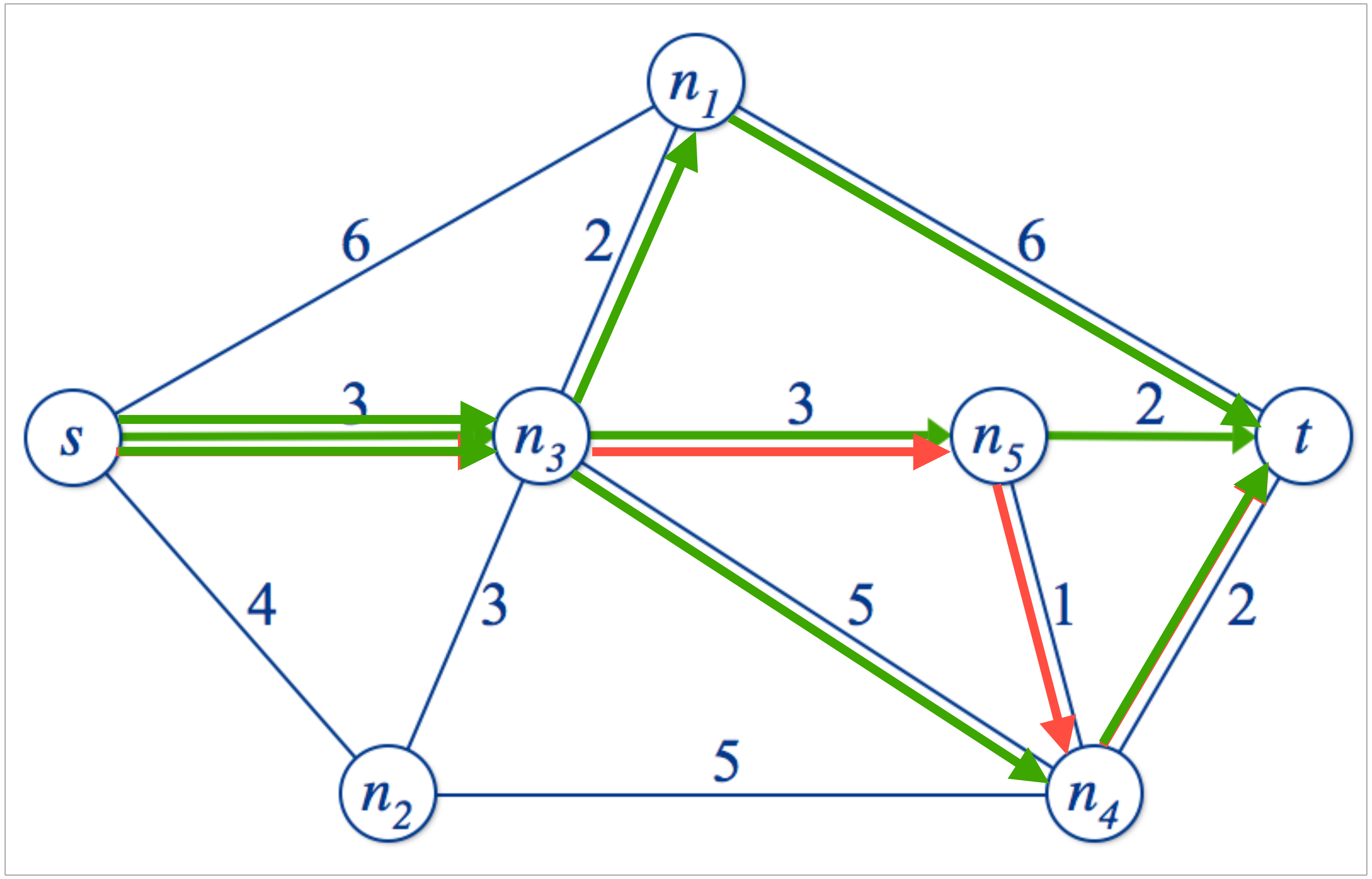
- The similarity of a path p to another path p' is determined by their **overlap ratio**:

$$Sim(p, p') = \frac{\sum_{e \in p \cap p'} w(e)}{\ell(p')}$$

Naive Solution [GIS'16]

- Compute all paths and examine them in increasing order of their length

q(G, s, t, 3, 50%)



path vertices	length
<s,n3,n5,t>	8
<s,n3,n5,n4,t>	9
<s,n3,n4,t>	10
<s,n3,n4,n5,t>	11
<s,n3,n1,t>	11
<s,n2,n4,t>	11
<s,n2,n4,n5,t>	12
<s,n2,n3,n5,t>	12
<s,n1,t>	12
<s,n3,n2,n4,t>	13
(…)	

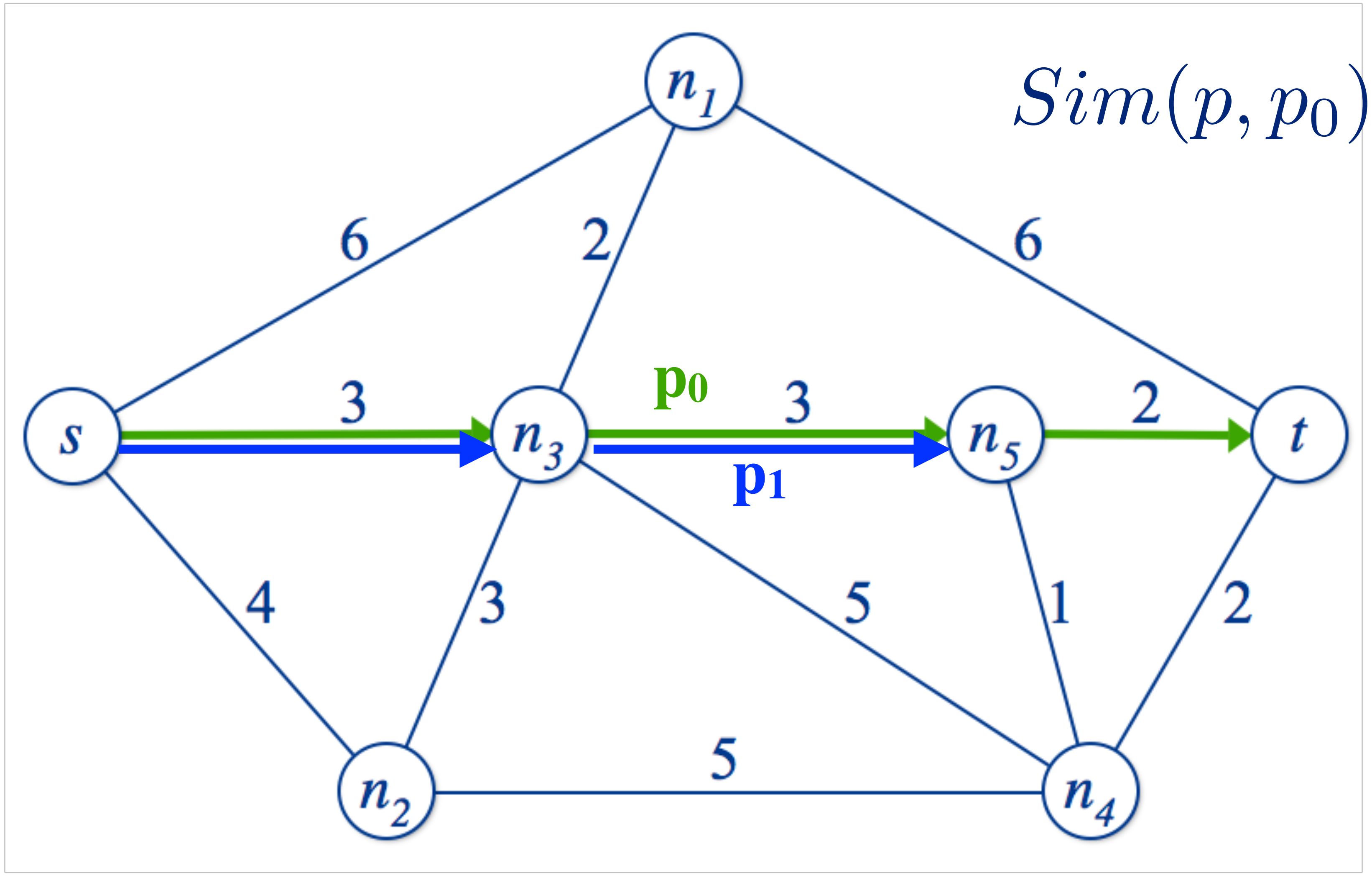
Algorithms [GIS'16]

- BSL
  - Builds upon the K-Shortest Paths
  - Constructs all paths iteratively
  - Stops when k dissimilar paths are found
- OnePass
  - Traverses the road network once
  - **Prunes (sub-)paths which violate the similarity constraint**
  - Stops when k paths to the target are found

Pruning Criterion 1 [GIS'16]

- Pruning paths that exceed the similarity threshold θ

<u>EXAMPLE (θ = 50%):</u>



$$Sim(p, p_0) = \frac{3+3}{3+3+2} = \frac{6}{8} > 0.5$$

Path *p* is pruned

Overview

- kSPwLO Overview
- MultiPass Algorithm
- Approximate Algorithms
  - OnePass^{+}
  - ESX
- Experimental Evaluation
- Conclusions & Future Work

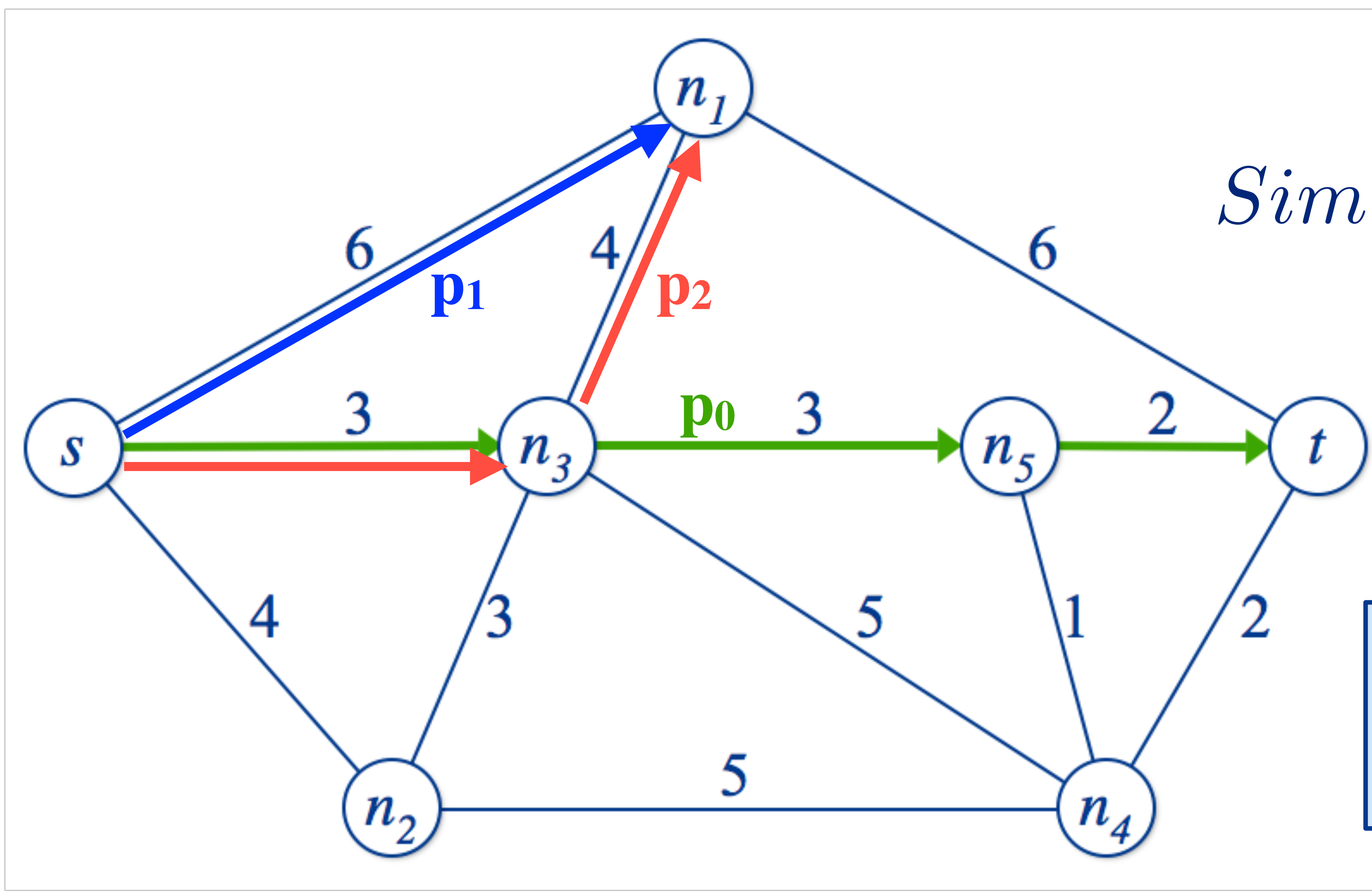
MultiPass Algorithm Overview

- Employs two pruning criteria
  - Prunes sub-paths which violate the similarity constraint (PC1)
  - Prunes non promising paths (PC2)

Pruning Criterion 2

- Pruning paths that cannot lead to a solution

EXAMPLE (θ = 50%):



$$\ell(p_1) < \ell(p_2)$$
$$and$$
$$Sim(p_1, p_0) < Sim(p_2, p_0)$$

Path p_2 is pruned

MultiPass Algorithm Overview

- Employs two pruning criteria
  - Prunes sub-paths which violate the similarity constraint (PC1)
  - Prunes non promising paths (PC2)
- Traverses the road network k-1 times
  - After each round, MultiPass **restarts** the expansion
- Terminates when:
  - k dissimilar paths have been found
  - the last round failed to find an alternative path

Overview

- kSPwLO Overview
- MultiPass Algorithm
- Approximate Algorithms
  - OnePass$^+$
  - ESX
- Experimental Evaluation
- Conclusions & Future Work

OnePass+ Algorithm

- OnePass
  - Employs PC1
  - Traverses the road network once
- MultiPass
  - Employs PC1 and PC2
  - Traverses the road network k-1 times
- **OnePass+**
  - **Employs PC1 and PC2**
  - **Traverses the road network once**

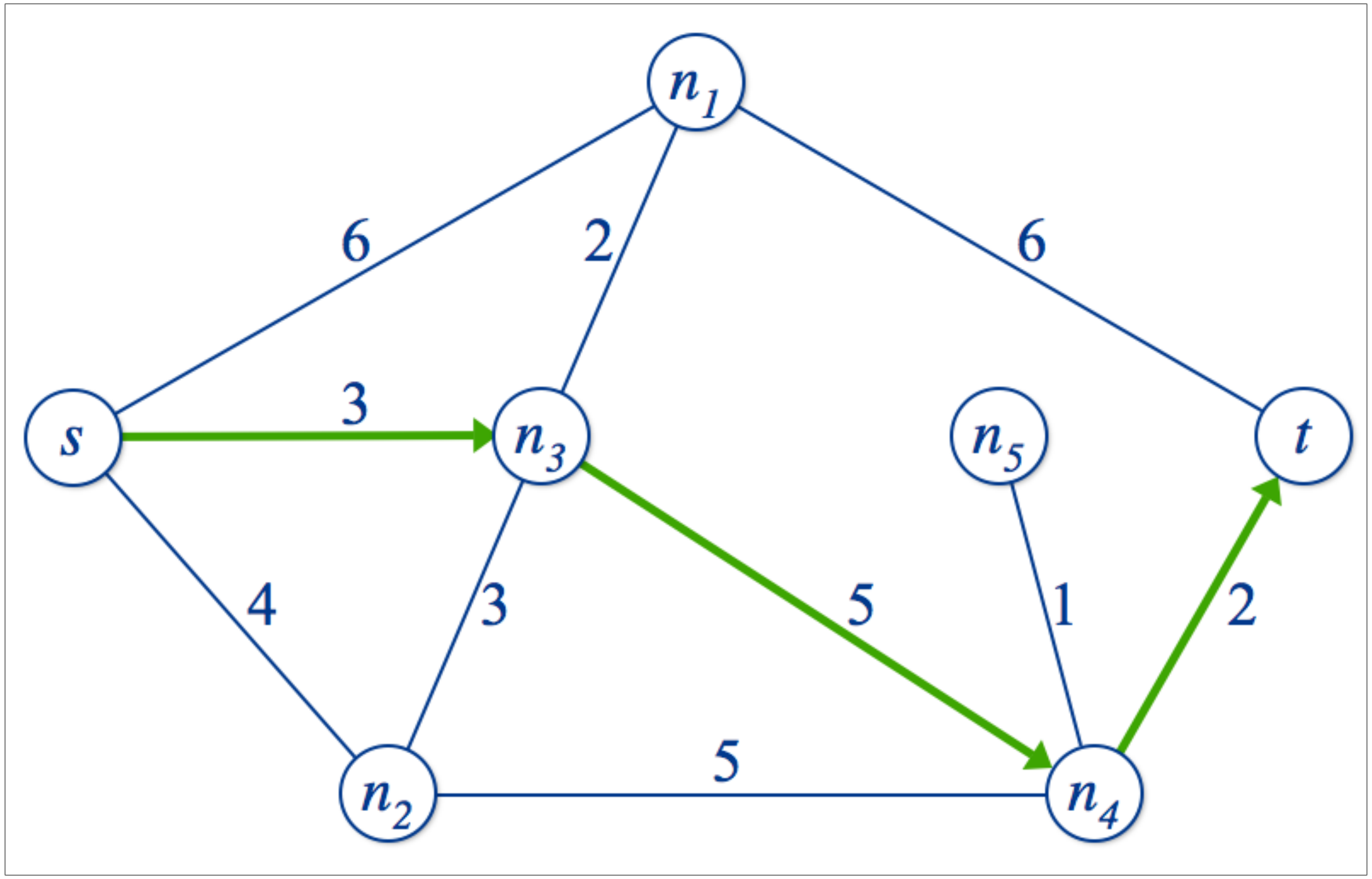
ESX Algorithm Overview

- Main idea:
  - Remove an edge from the road network that lies on some already computed alternative path
  - Compute the shortest path on the updated graph
  - Continue until a sufficiently dissimilar path is found

ESX Example

- Gradually removing edges until a sufficiently dissimilar path is found

<u>EXAMPLE (θ = 50%):</u>



ESX Algorithm Overview

- Main idea:
  - Remove an edge from the road network that lies on some already computed alternative path
  - Compute the shortest path on the updated graph
  - Continue until a sufficiently dissimilar path is found
- *prio(e)* = # of shortest paths that contain *e*
  - NP-hard - ESX performs a local check
- ESX ensures that the road networks remains connected at all times

Overview

- kSPwLO Overview
- MultiPass Algorithm
- Approximate Algorithms
  - OnePass+
  - ESX
- Experimental Evaluation
- Conclusions & Future Work

Experimental Evaluation

- Performance measurements
  - varying $k \in \{2,3,4,5\}$ (θ = *50%*)
  - varying $\theta \in$ {10%, 30%, 50%, 70%, 90%} (k = *3*)
- Scalability test varying $k \in \{4,8,12,16\}$
- Result quality
  - Average length of recommended path to the length of the shortest path

Experimental Evaluation - Datasets

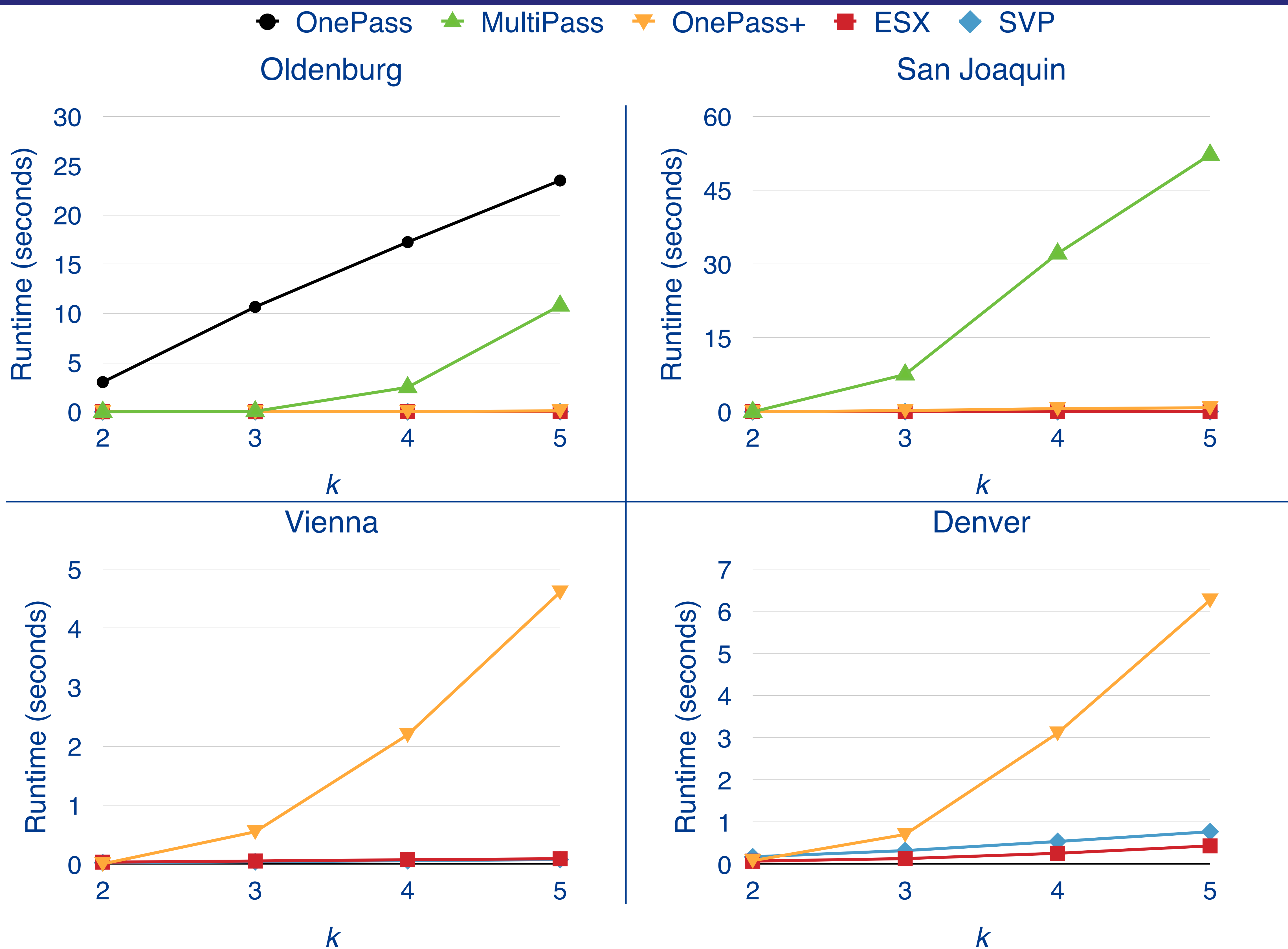
- Road Networks

Road Network	# Nodes	# Edges
Oldenburg	6,105	14,058
San Joaquin	18,263	47,594
Vienna	19,826	54,918
Denver	73,166	196,630
San Francisco	174,956	443,604
Colorado	435,666	1,057,066

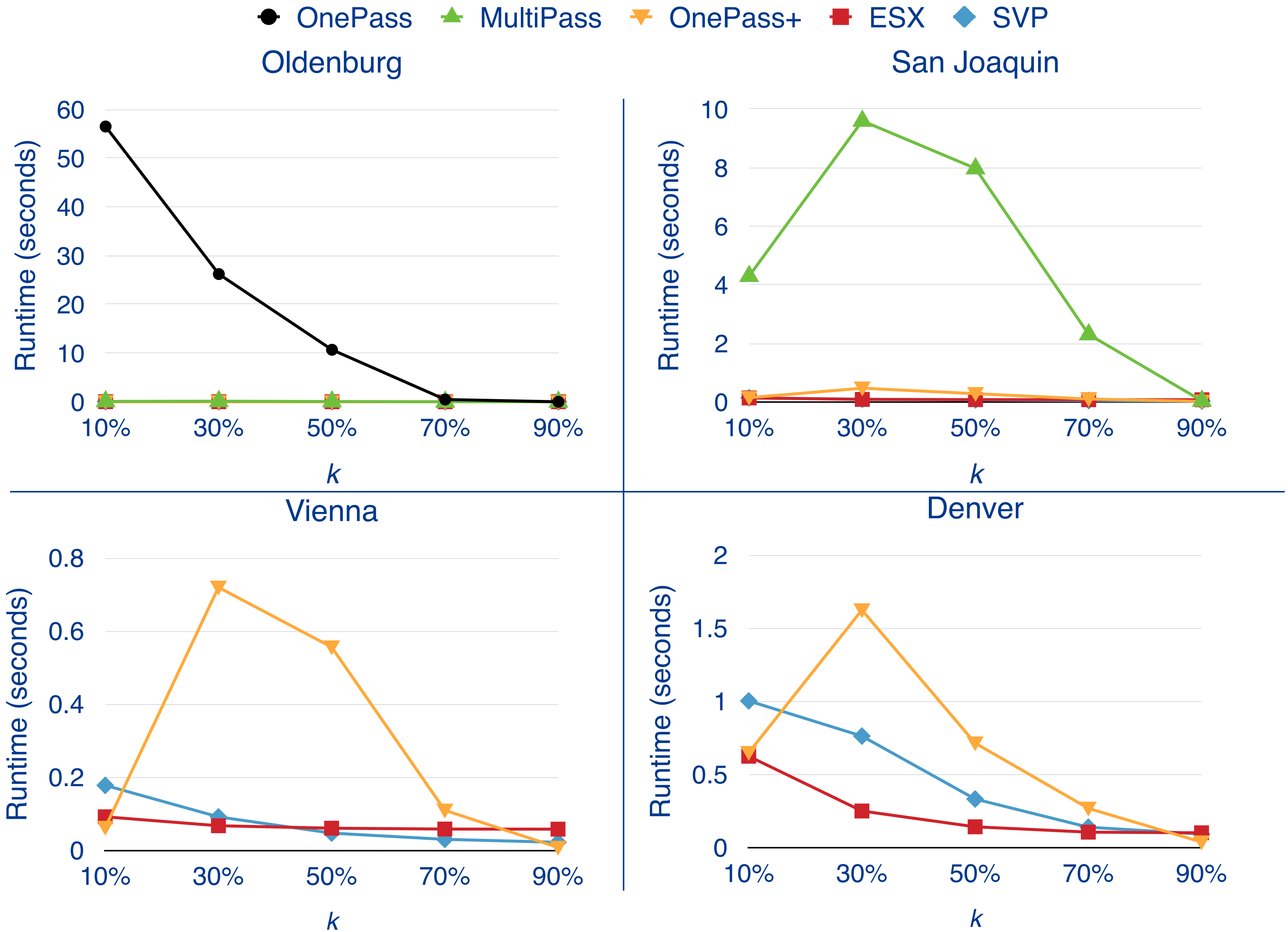
Experimental Evaluation - Algorithms

- Exact algorithms
  - OnePass [GIS'16]
  - MultiPass
- Approximate algorithms
  - $OnePass^+$
  - ESX
  - SVP^+ (adapted from [JEA'13])

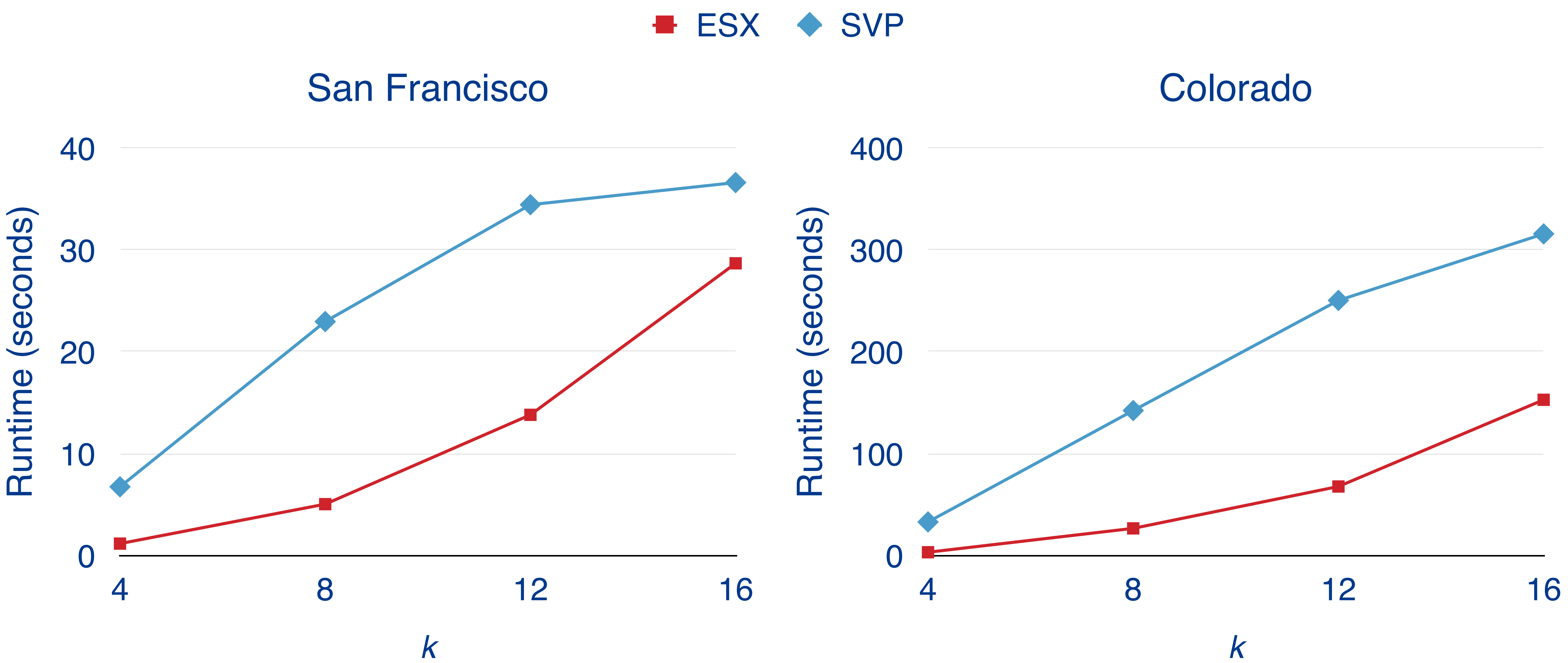
Performance - Varying *k*



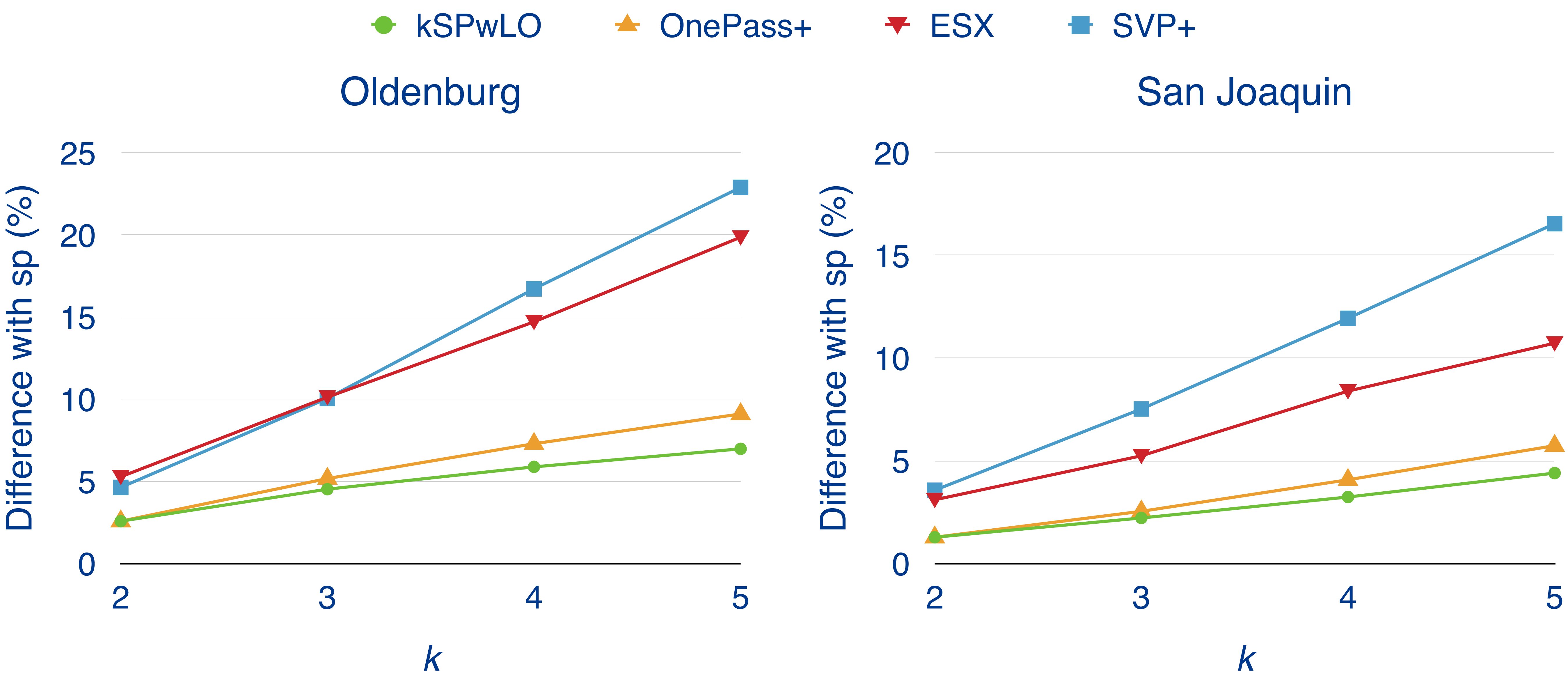
Performance - Varying θ



Scalability - Varying *k*



Result Quality - Average Length



Overview

- kSPwLO Overview
- MultiPass Algorithm
- Approximate Algorithms
  - OnePass^{+}
  - ESX
- Experimental Evaluation
- Conclusions & Future Work

Conclusions and Future Work

- We presented 3 novel algorithms
  - ✓ **MultiPass:** computes the optimal result but is practical only for small road networks
  - ✓ **OnePass^{+}:** good approximation and practical for larger road networks than MultiPass
  - ✓ **ESX:** less accurate but practical even for large road networks and large values of *k*
- Future Work
  - ✓ Support arbitrary similarity metrics
  - ✓ Investigate the computation of dissimilar paths on different types of networks (i.e. social networks)

Thank you!