Query-by-Sketch: Scaling Shortest Path Graph Queries on Very Large Networks

Ye Wang[1], Qing Wang[1], Henning Koehler[2], Yu Lin[1]

[1]School of Computing, Australian National University
[2]School of Fundamental Sciences, Massey University

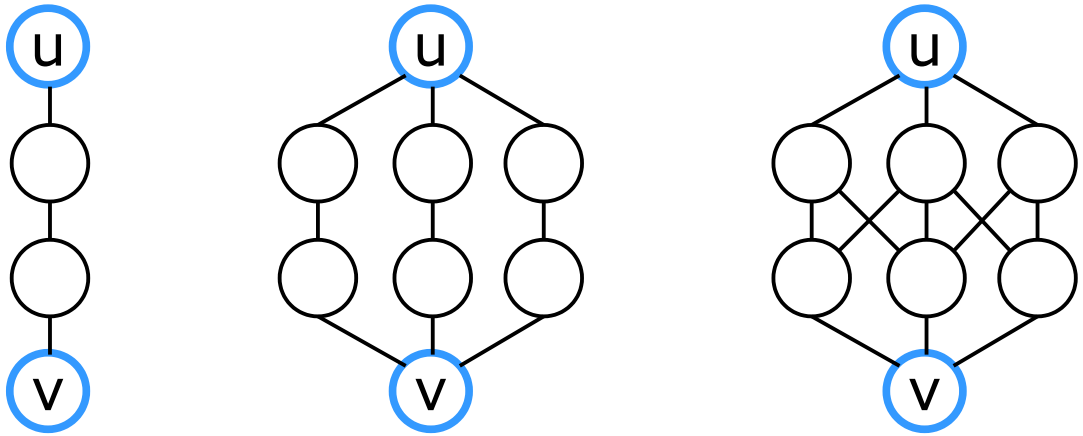
Australian National University

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

{ye.wang; qing.wang}@anu.edu.au; h.koehler@massey.ac.nz; yu.lin@anu.edu.au

Shortest Path Graph Problem

Given a graph G and two vertices u, v,

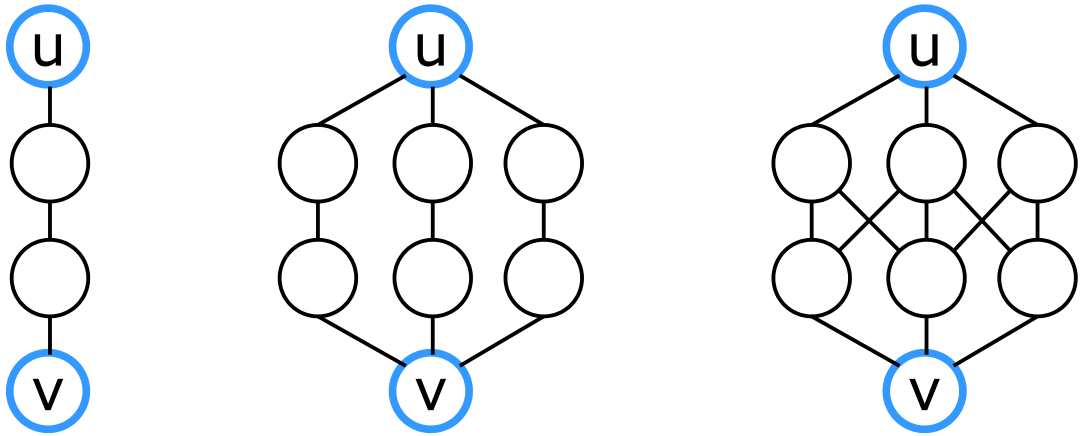


(a) (b) (c)

Shortest Path Graph Problem

Given a graph G and two vertices u, v,

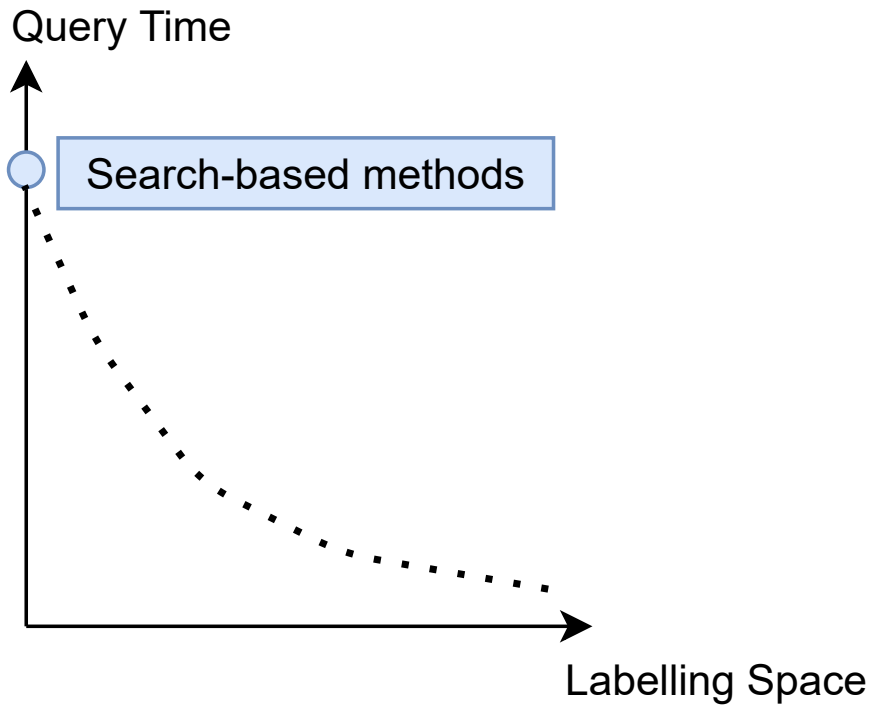
find the shortest path graph G_{uv} that contains exactly all shortest paths between u and v.



(a) (b) (c)

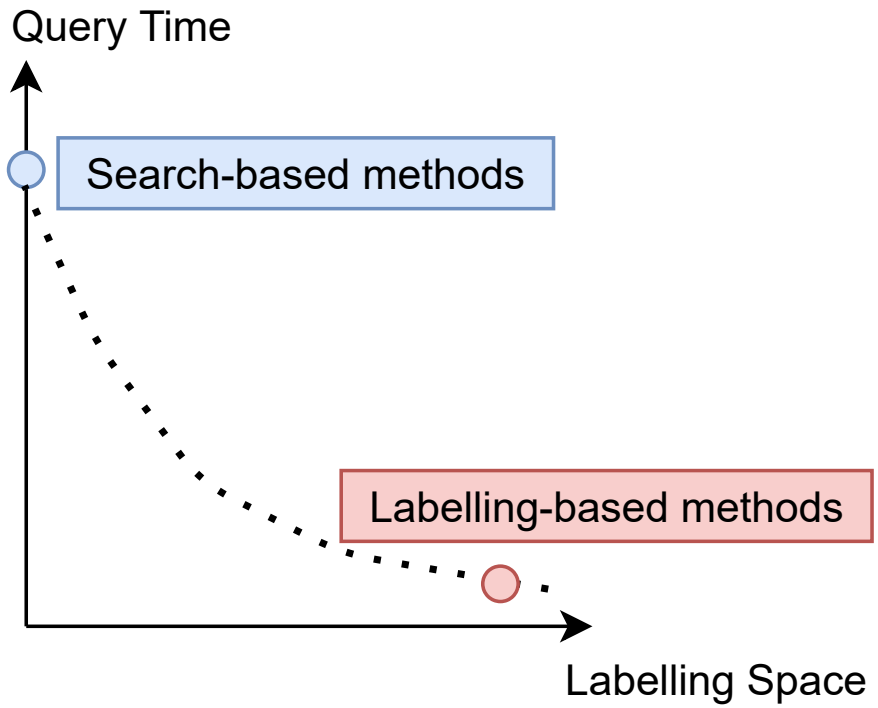
Related Work

- Search-based methods
  - Dijkstra: $O(|E|log|V|)$ query time
  - Breadth-first search: $O(|E|)$ query time



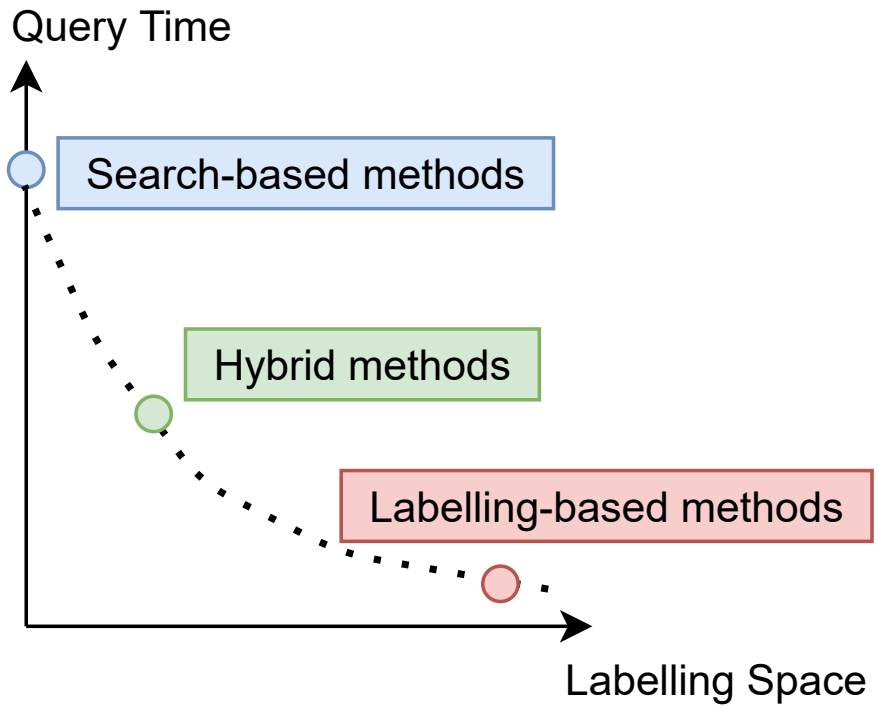
Related Work

- Search-based methods
  - Dijkstra: $O(|E|log|V|)$ query time
  - Breadth-first search: $O(|E|)$ query time
- Labelling-based methods
  - Pruned path labelling: $O(|V|^2)$ labelling space
  - Parent pruned path labelling: $O(|V||E|)$ labelling space



Related Work

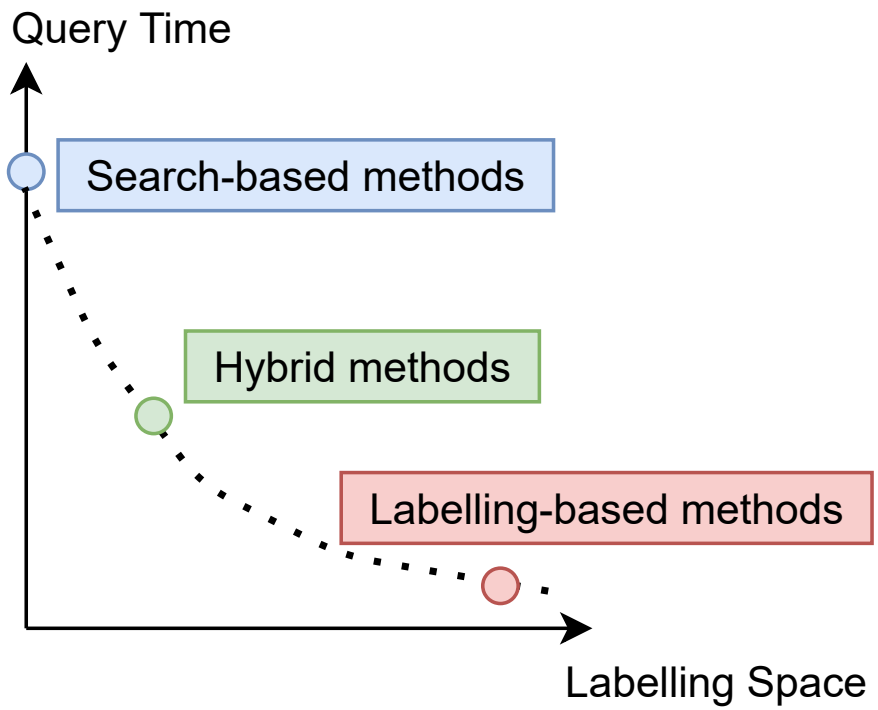
- Search-based methods
  - Dijkstra: $O(|E|log|V|)$ query time
  - Breadth-first search: $O(|E|)$ query time
- Labelling-based methods
  - Pruned path labelling: $O(|V|^2)$ labelling space
  - Parent pruned path labelling: $O(|V||E|)$ labelling space
- Hybrid methods?
  - A trade-off



Research Goal

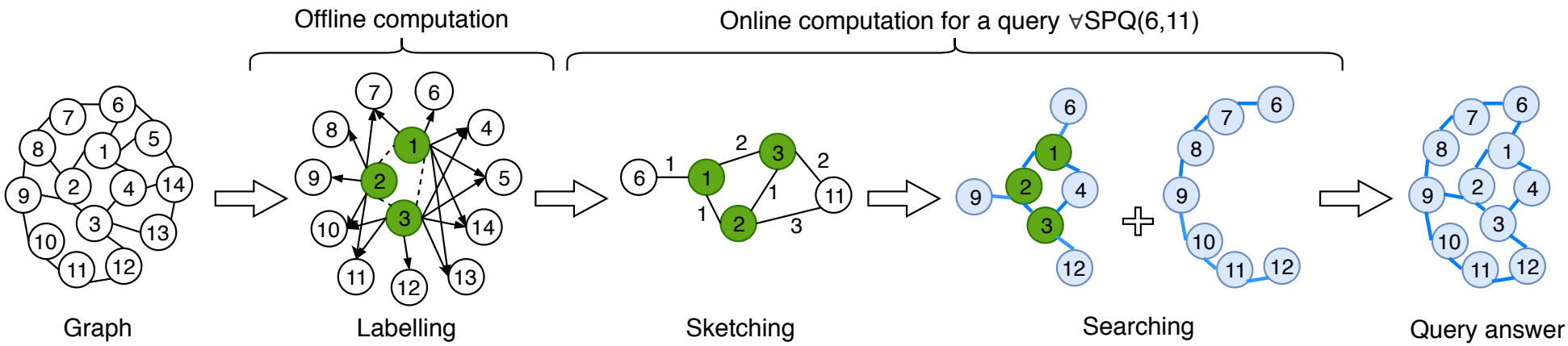
Design a hybrid method that scales over networks with millions or billions of vertices and edges:

- **Query time efficiency:** less than 1 second
- **Labelling space efficiency:** comparable size with the original graph



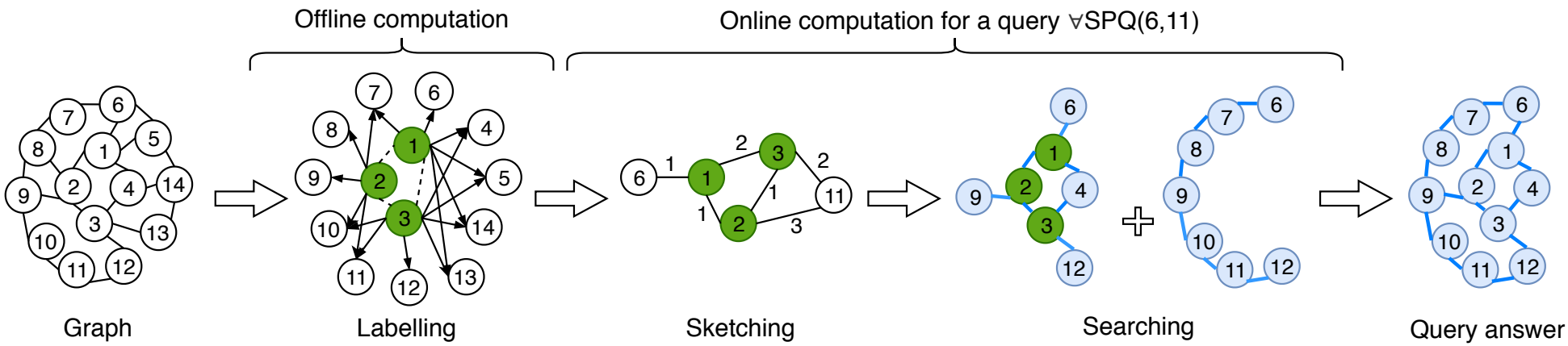
Query-by-Sketch

- Propose a novel method, called *Query-by-Sketch*



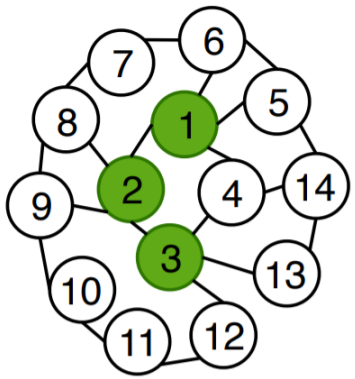
Query-by-Sketch

- Propose a novel method, called *Query-by-Sketch*
- Three key ideas:
  - (1) Labelling scheme;
  - (2) Fast sketching;
  - (3) Guided searching.

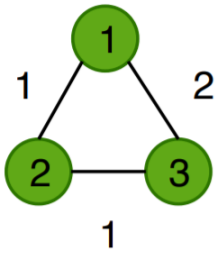


Query-by-Sketch - Labelling Scheme

- Let $G = (V, E)$ and R be a set of landmarks, and $|R| \ll |V|$. A labelling scheme $\mathcal{L} = (M, L)$ consists of
  - M: a meta-graph over R
  - L: a path labelling over G



Graph



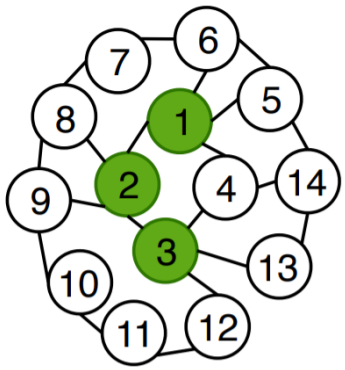
Meta-graph

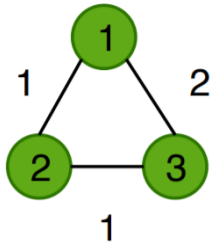
Query-by-Sketch - Labelling Scheme

- Let $G = (V, E)$ and R be a set of landmarks, and $|R| \ll |V|$. A labelling scheme $\mathcal{L} = (M, L)$ consists of
  - M: a meta-graph over R
  - L: a path labelling over G

(1,1): d(4,1)=1
(3,1): d(4,3)=1

Label	Labelling Entries
L(4)	(1,1) (3,1)
L(6)	(1,1)
L(7)	(1,2) (2,2)
L(8)	(2,1)
L(9)	(2,1)
L(10)	(2,2) (3,3)
L(11)	(2,3) (3,2)
L(12)	(3,1)
L(13)	(1,3) (3,1)
L(14)	(1,2) (3,2)

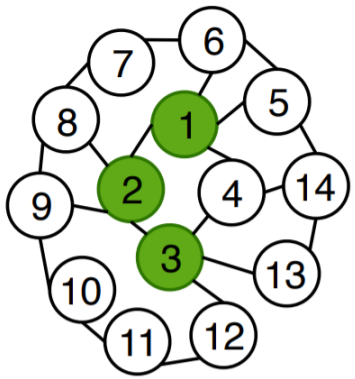




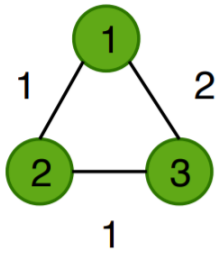
Graph Meta-Graph Path Labelling

Query-by-Sketch - Labelling Scheme

- Let $G = (V, E)$ and R be a set of landmarks, and $|R| \ll |V|$. A labelling scheme $\mathcal{L} = (M, L)$ consists of
  - M: a meta-graph over R
  - L: a path labelling over G



Graph



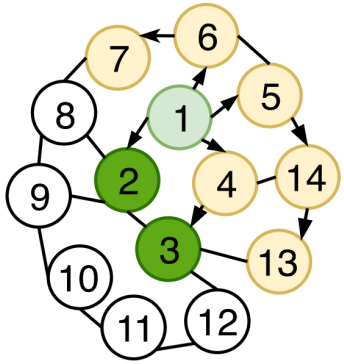
Meta-graph

Label	Labelling Entries
L(4)	(1,1) (3,1)
L(5)	(1,1) (3,3)
L(6)	(1,1)
L(7)	(1,2) (2,2)
L(8)	(2,1)
L(9)	(2,1)
L(10)	(2,2) (3,3)
L(11)	(2,3) (3,2)
L(12)	(3,1)
L(13)	(1,3) (3,1)
L(14)	(1,2) (3,2)

Path-Labelling

Query-by-Sketch - Labelling Scheme

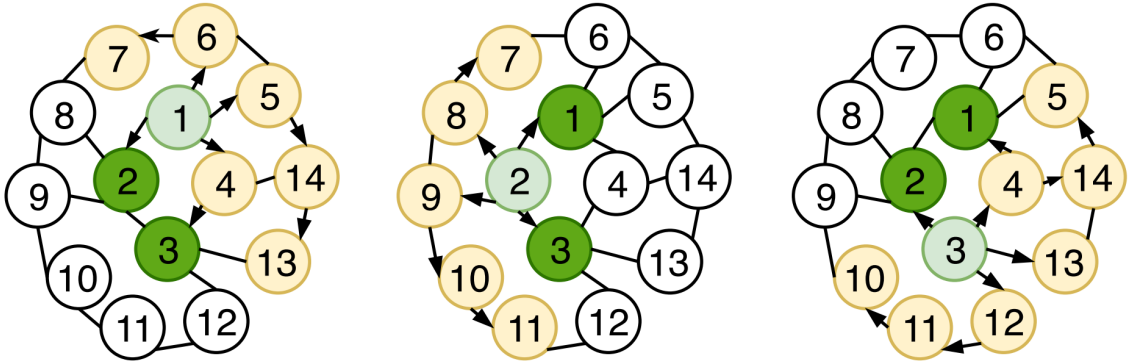
- Let $G = (V, E)$ and R be a set of landmarks, and $|R| \ll |V|$. A labelling scheme $\mathcal{L} = (M, L)$ consists of
  - M: a meta-graph over R
  - L: a path labelling over G



(a)

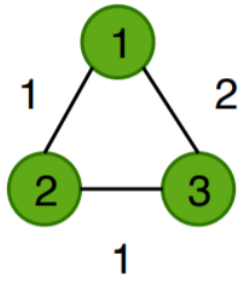
Query-by-Sketch - Labelling Scheme

- Let $G = (V, E)$ and R be a set of landmarks, and $|R| \ll |V|$. A labelling scheme $\mathcal{L} = (M, L)$ consists of
  - M: a meta-graph over R
  - L: a path labelling over G

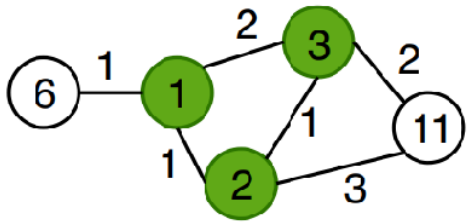


Query-by-Sketch - Fast Sketching

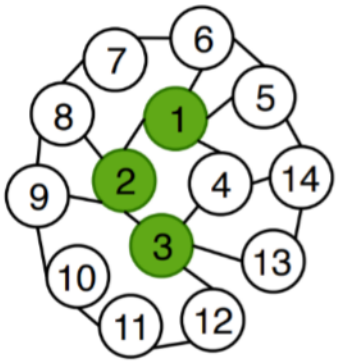
- A sketch S_{uv} for u and v estimates how u and v are connected.



Label	Labelling Entries
L(4)	(1,1) (3,1)
L(5)	(1,1) (3,3)
L(6)	(1,1)
L(7)	(1,2) (2,2)
L(8)	(2,1)
L(9)	(2,1)
L(10)	(2,2) (3,3)
L(11)	(2,3) (3,2)
L(12)	(3,1)
L(13)	(1,3) (3,1)
L(14)	(1,2) (3,2)

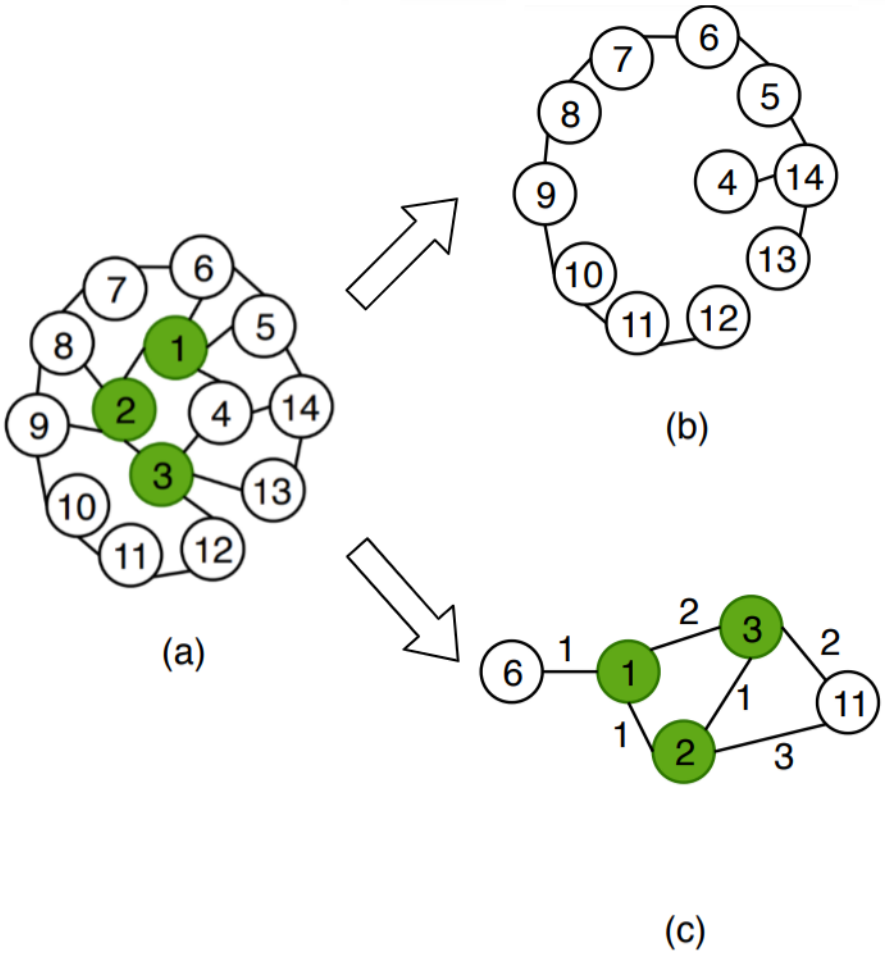


Query-by-Sketch - Guided Searching



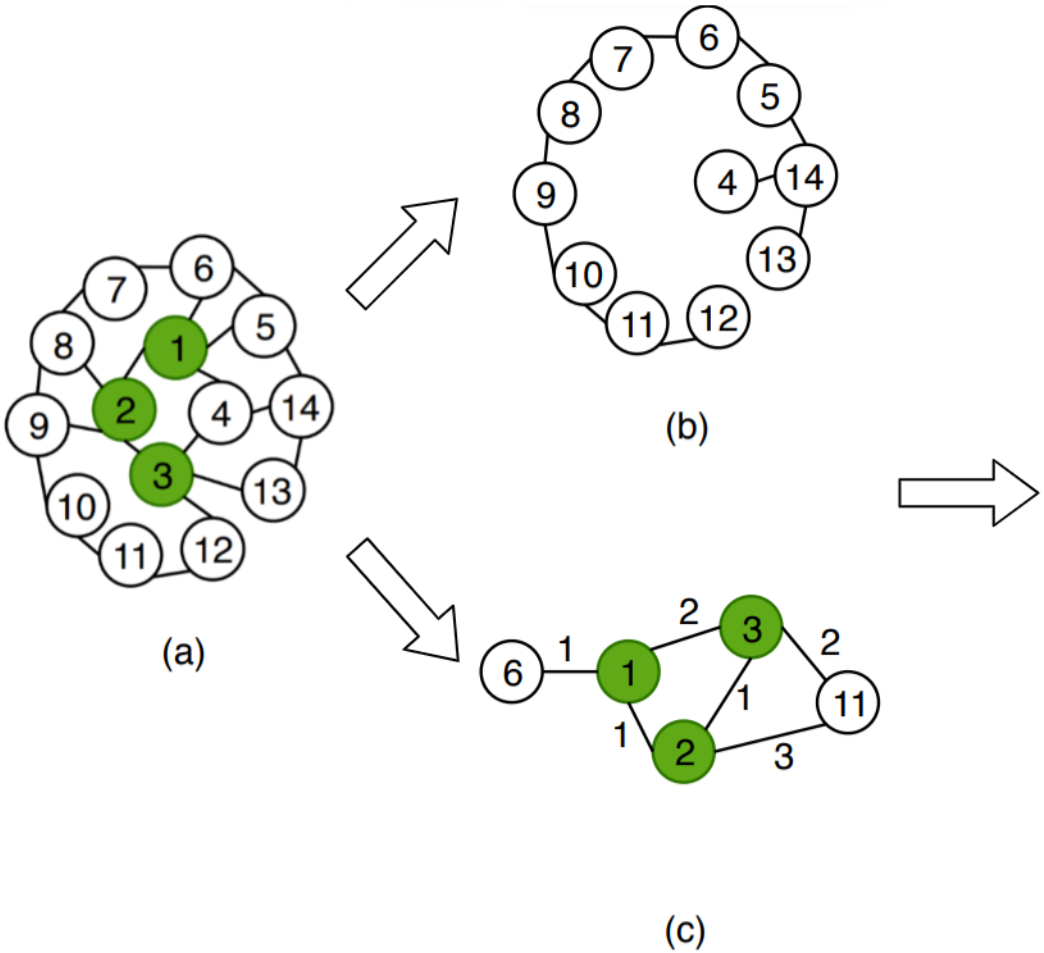
(a)

Query-by-Sketch - Guided Searching



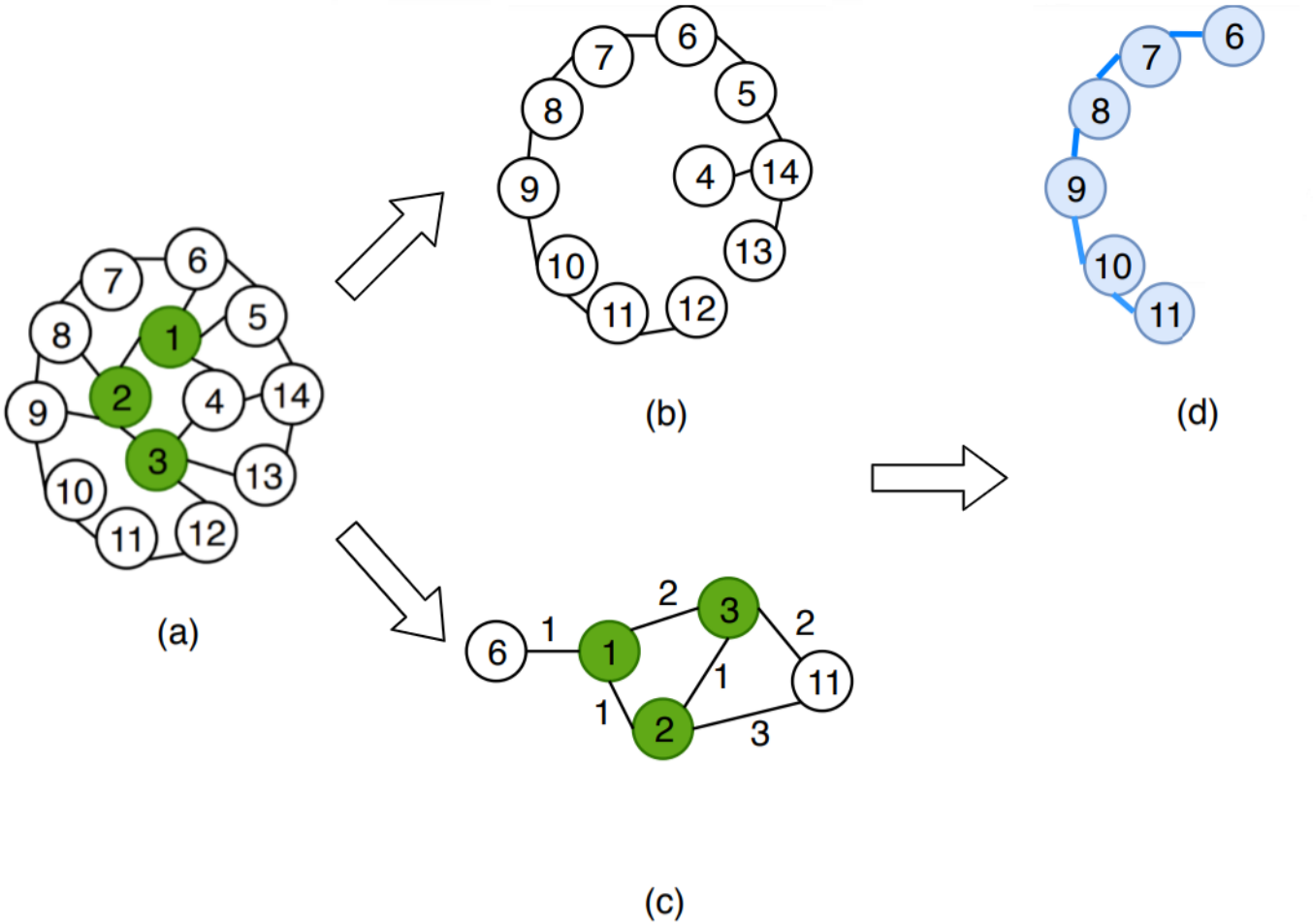
Query-by-Sketch - Guided Searching

- Searching shortest paths on the sparsified graph $G^- = [G \backslash R]$



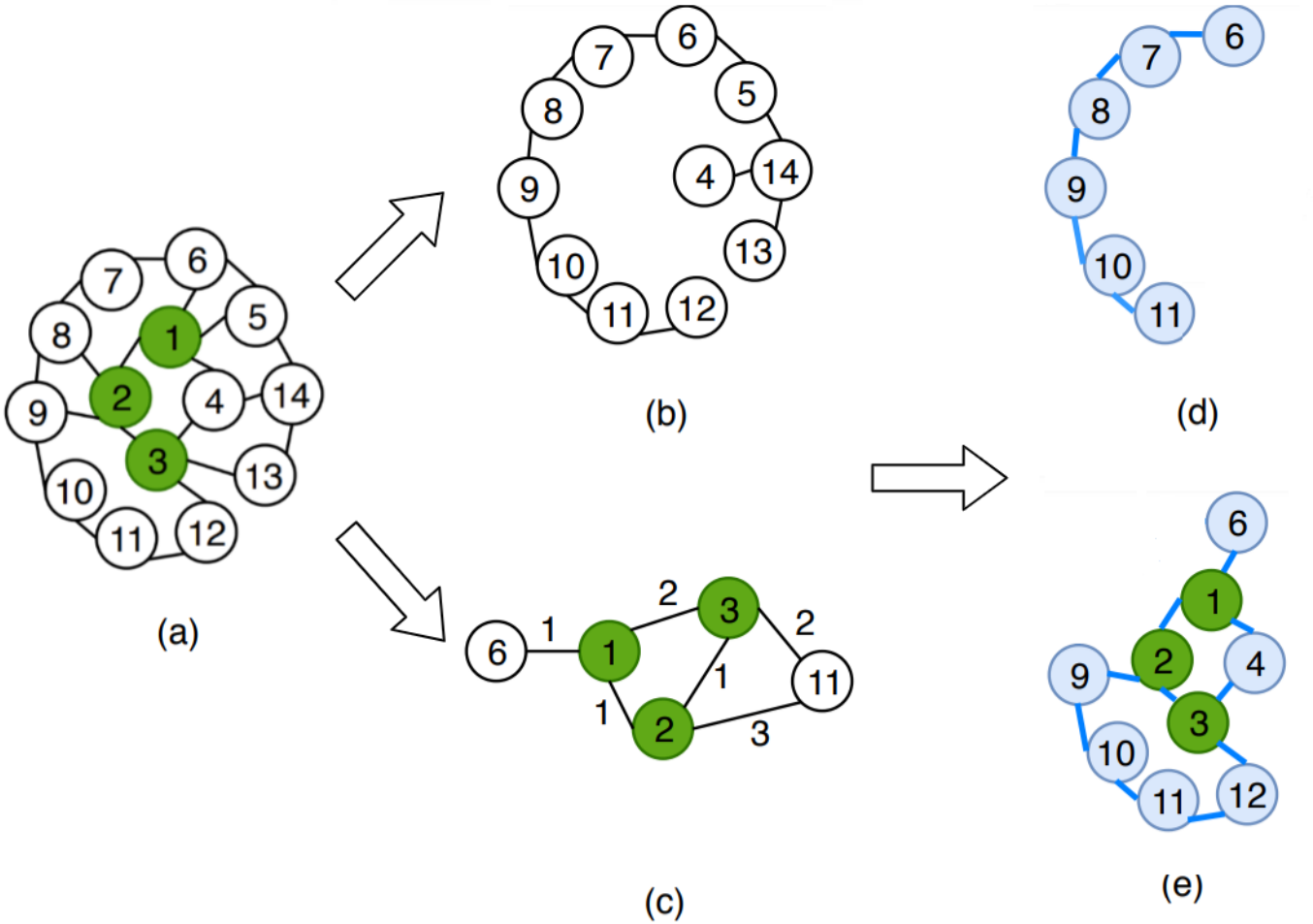
Query-by-Sketch - Guided Searching

- Searching shortest paths on the sparsified graph $G^- = [G \backslash R]$



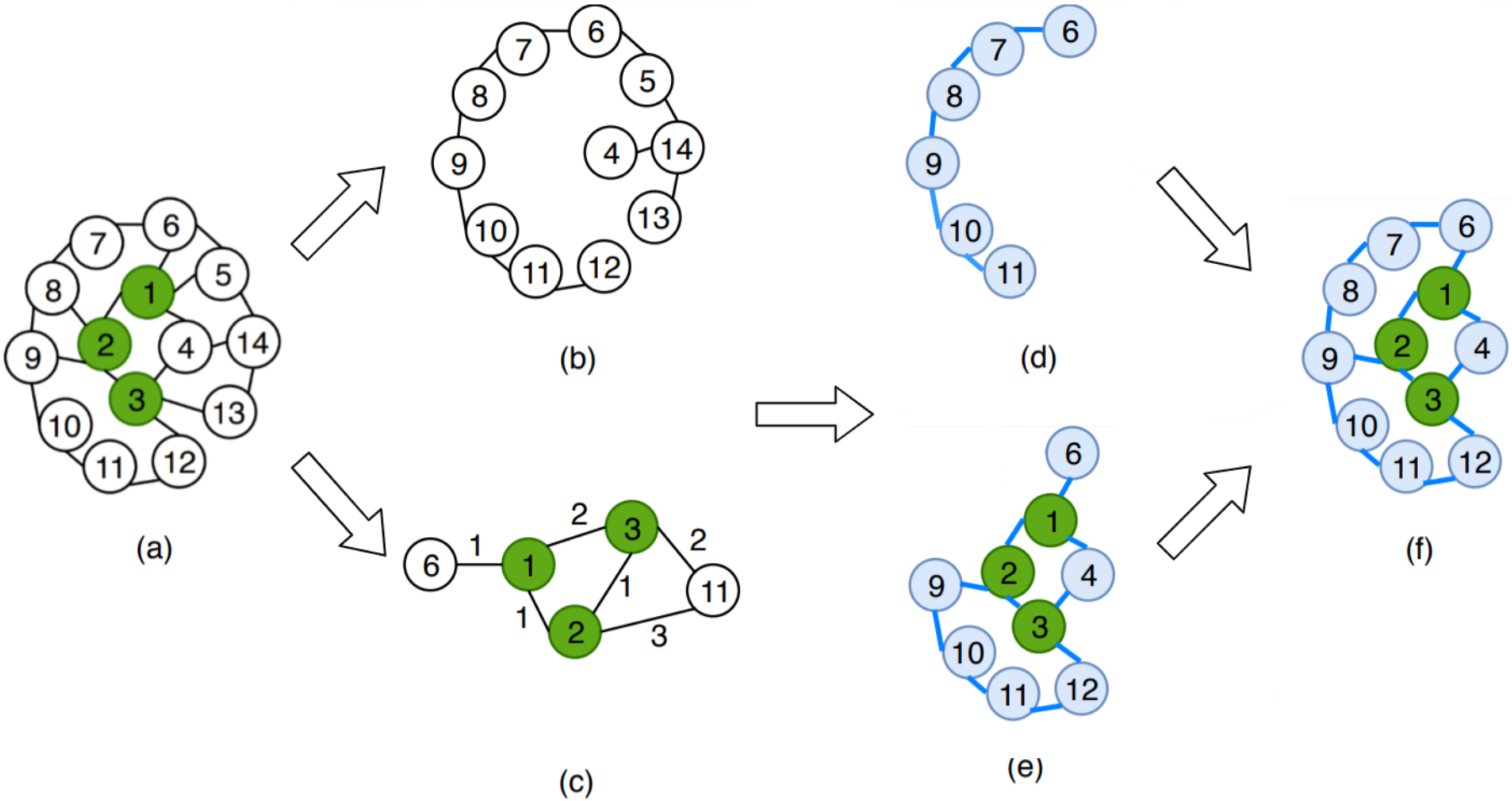
Query-by-Sketch - Guided Searching

- Searching shortest paths on the sparsified graph $G^- = [G \backslash R]$
- Searching shortest paths captured by the sketch



Query-by-Sketch - Guided Searching

- Searching shortest paths on the sparsified graph $G^- = [G \backslash R]$
- Searching shortest paths captured by the sketch



Experiments

We evaluate Query-by-sketch (QbS) against the baselines:

- Search-based methods:
  - Bi-directional BFS (Bi-BFS)
- Labelling-based methods:
  - Pruned path labelling (PPL)
  - Parent pruned path labelling (ParentPPL)

Experiments

- **Datasets:** 12 real-world complex networks

Dataset	$\|V\|$	$\|E^{un}\|$	max. deg	avg. deg	avg. dist
Douban (DO)	0.2M	0.3M	287	4.2	5.2
DBLP (DB)	0.3M	1.1M	343	6.6	6.8
Youtube (YT)	1.1M	3.0M	28,754	5.27	5.3
WikiTalk (WK)	2.4M	4.7M	100,029	3.89	3.9
Skitter (SK)	1.7M	11.1M	35,455	13.08	5.1
Baidu (BA)	2.1M	17.0M	97,848	15.89	4.1
LiveJournal (LJ)	4.8M	43.1M	20,334	17.79	5.5
Orkut (OR)	3.1M	117M	33,313	76.28	4.2
Twitter (TW)	41.7M	1.2B	2,997,487	57.74	3.6
Friendster (FR)	65.6M	1.8B	5,214	55.06	4.8
uk2007 (UK)	106M	3.3B	979,738	62.77	5.6
ClueWeb09 (CW)	1.7B	7.8B	6,444,720	9.27	7.5

Experiments

Q1: *How efficiently can QbS construct labelling?*

Dataset	Construction Time (sec.)		
	QbS	PPL	ParentPPL
Douban	0.3	154	2,736
DBLP	1.1	2,610	11,049
Youtube	4.4	22,601	DNF
WikiTalk	4.9	8,662	DNF
Skitter	12.7	86,326	DNF
Baidu	18.9	DNF	ROM
LiveJournal	52.2	DNF	ROM
Orkut	73.2	DNF	ROM
Twitter	1,345	DNF	ROM
Friendster	2,354	DNF	ROM
uk2007	1,485	ROM	ROM
ClueWeb09	17,060	ROM	ROM

Experiments

Q2: *How does QbS perform in terms of labelling size?*

Dataset	QbS	PPL	ParentPPL	$\|G\|$
Douban	2.98MB	0.4GB	0.8GB	2.5MB
DBLP	6.08MB	1.2GB	2.4GB	8.0MB
Youtube	22.2MB	1.7GB	–	23MB
WikiTalk	46.4MB	2.1GB	–	36MB
Skitter	52.7MB	9.2GB	–	85MB
Baidu	45.6MB	–	–	130MB
LiveJournal	93.6MB	–	–	329MB
Orkut	62.1MB	–	–	894MB
Twitter	1.54GB	–	–	9.0GB
Friendster	1.23GB	–	–	13.0GB
uk2007	2.06GB	–	–	24.8GB
ClueWeb09	31.9GB	–	–	58.2GB

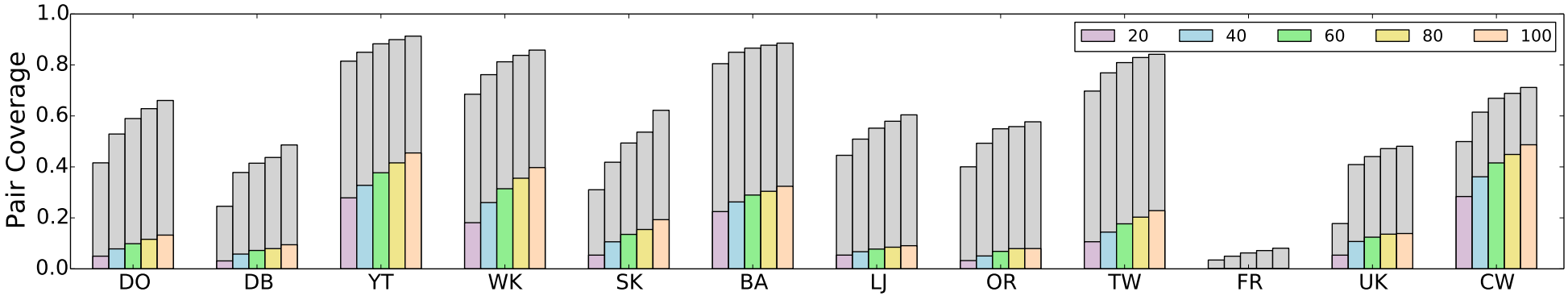
Experiments

Q3: *How does QbS perform in terms of query time?*

Dataset	Average Query Time (ms.)			
	QbS	PPL	ParentPPL	Bi-BFS
Douban	0.037	1.414	0.038	0.585
DBLP	0.097	1.782	0.052	2.995
Youtube	0.218	5.314	-	23.809
WikiTalk	0.693	3.536	-	6.984
Skitter	0.951	16.978	-	44.685
Baidu	0.845	-	-	174.412
LiveJournal	1.095	-	-	84.967
Orkut	4.237	-	-	207.541
Twitter	164.333	-	-	4,817.774
Friendster	11.972	-	-	3,600.362
uk2007	77.830	-	-	5,264.101
ClueWeb09	480.443	-	-	DNF

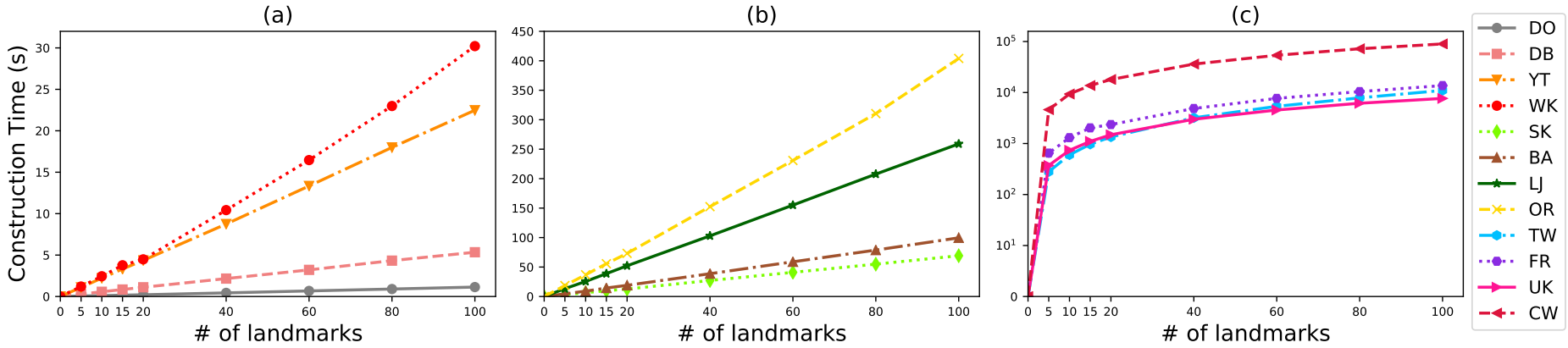
Experiments

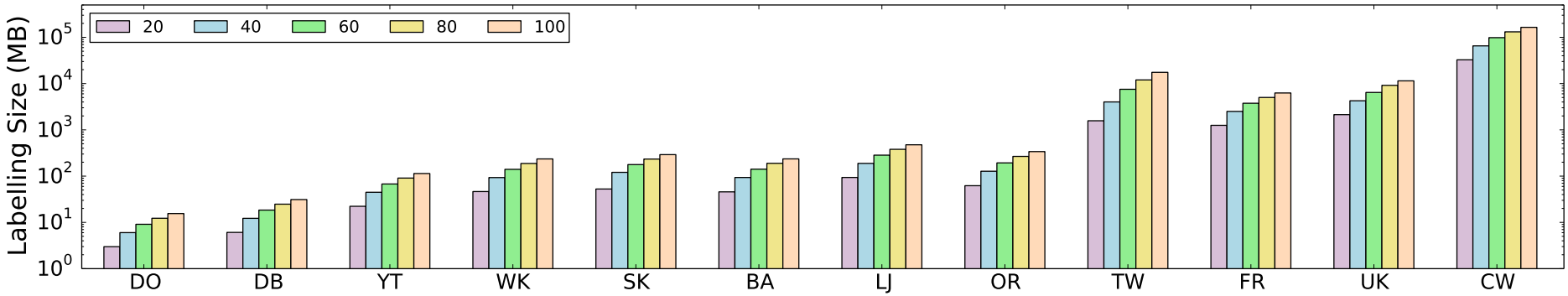
Q4: *How well can "sketching" help improve the performance?*



Experiments

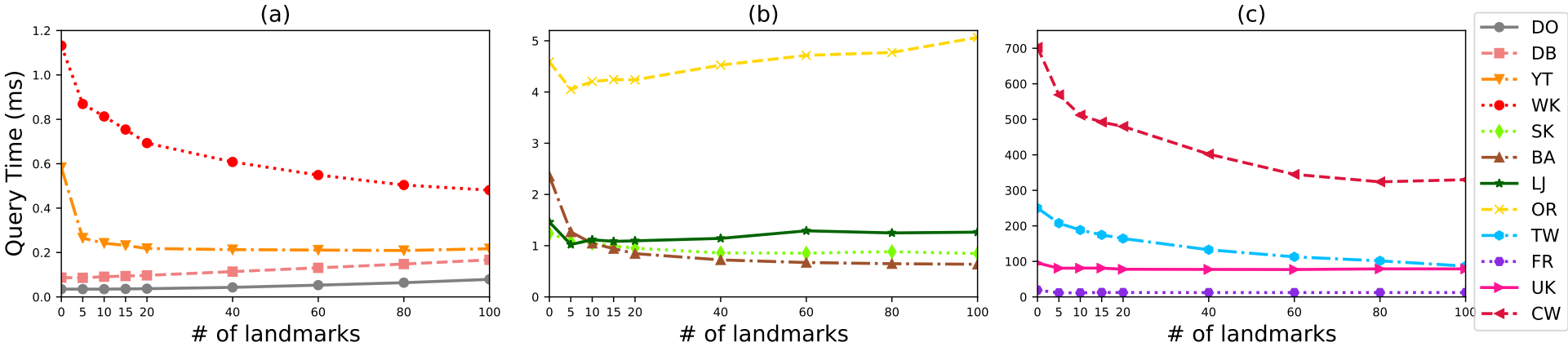
Q5: *How does the number of landmarks affect the performance?*





Experiments

Q5: *How does the number of landmarks affect the performance?*



Summary

- Proposed Query-by-Sketch (QbS) to answer shortest-path graph queries.
- Conducted experiments on real-world datasets.

Summary

- Proposed Query-by-Sketch (QbS) to answer shortest-path graph queries.
- Proved the correctness and analysed the complexity of QbS.
- Conducted experiments on real-world datasets.