Karlsruhe Institute of Technology

Faster Multi-Modal Route Planning with Bike Sharing Using ULTRA

SEA · June 17, 2020

Jonas Sauer, Dorothea Wagner, and Tobias Zündorf

INSTITUTE OF THEORETICAL INFORMATICS · ALGORITHMICS GROUP

Multi-Modal Route Planning

Goals:

- Journey planning for public transit
- Find **optimal** journeys
- Consider modes of transportation:
  - All timetable-based modes (trains, trams, buses, ...)

Multi-Modal Route Planning

Karlsruhe Institute of Technology

Goals:

- Journey planning for public transit
- Find **optimal** journeys
- Consider modes of transportation:
  - All timetable-based modes (trains, trams, buses, ...)
  - Walking (from, to, and between stops)

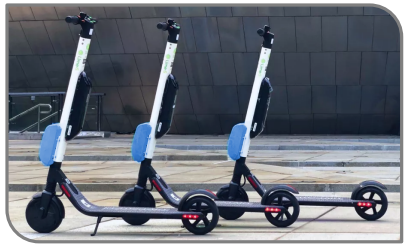
Faster Multi-Modal Route Planning with Bike Sharing Using ULTRA
Jonas Sauer, Dorothea Wagner, and Tobias Zündorf

Institute of Theoretical Informatics
Algorithmics Group

Multi-Modal Route Planning

Goals:

- Journey planning for public transit
- Find **optimal** journeys
- Consider modes of transportation:
  - All timetable-based modes (trains, trams, buses, ...)
  - Walking (from, to, and between stops)
  - Bike sharing (or other rental based services)



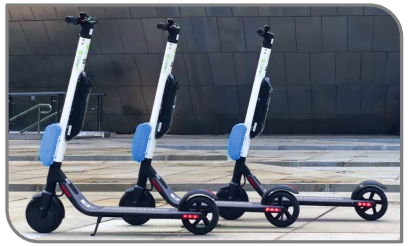
Faster Multi-Modal Route Planning with Bike Sharing Using ULTRA
Jonas Sauer, Dorothea Wagner, and Tobias Zündorf

Karlsruhe Institute of Technology
Institute of Theoretical Informatics
Algorithmics Group

Multi-Modal Route Planning

Goals:

- Journey planning for public transit
- Find **optimal** journeys
- Consider modes of transportation:
  - All timetable-based modes (trains, trams, buses, ...)
  - Walking (from, to, and between stops)
  - Bike sharing (or other rental based services)
  - No limits on any of the transportation modes



Faster Multi-Modal Route Planning with Bike Sharing Using ULTRA
Jonas Sauer, Dorothea Wagner, and Tobias Zündorf

Karlsruhe Institute of Technology
Institute of Theoretical Informatics
Algorithmics Group

Problem Statement

Karlsruhe Institute of Technology

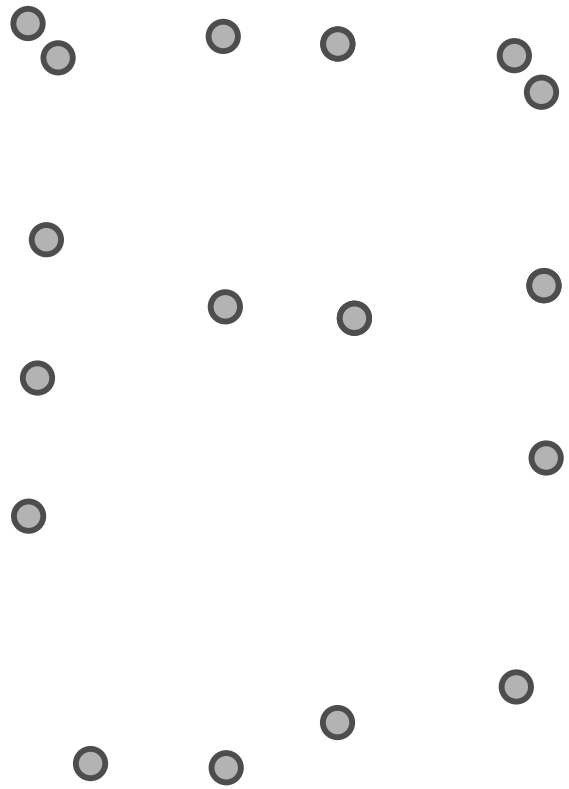
Given:

- Public transit network (timetable)

Problem Statement

Given:

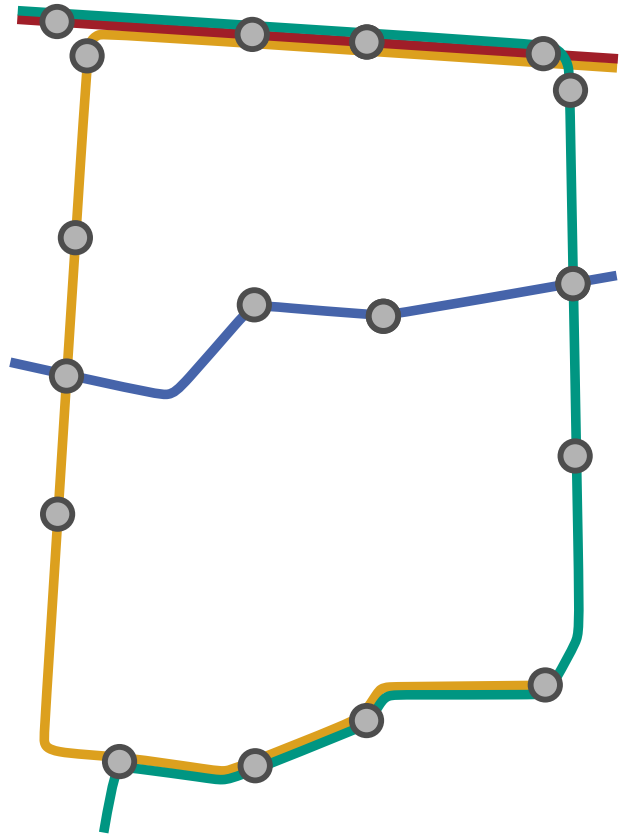
- Public transit network (timetable)
  - Stops (bus stops, stations)



Problem Statement

Given:

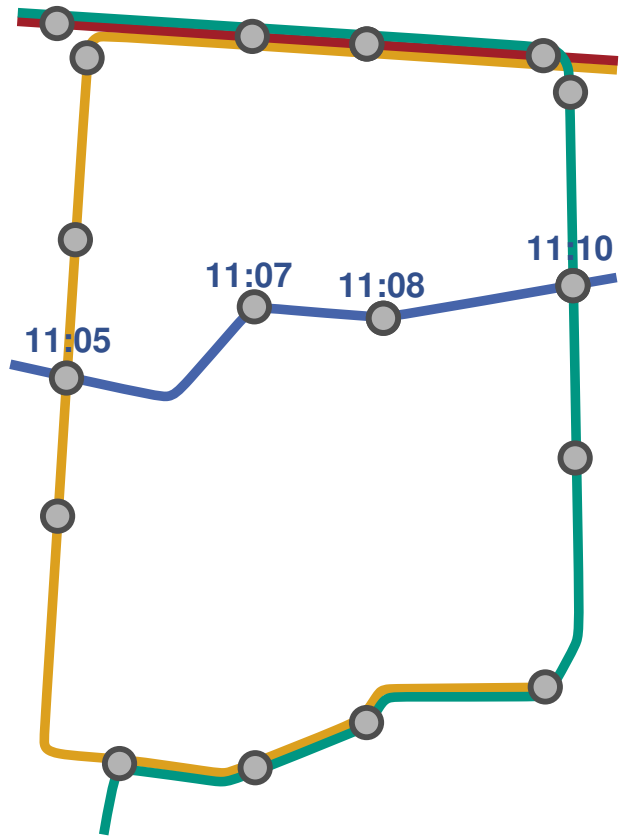
- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)



Problem Statement

Given:

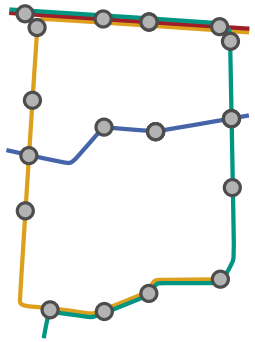
- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)



Problem Statement

Given:

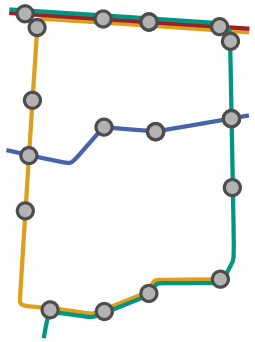
- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)

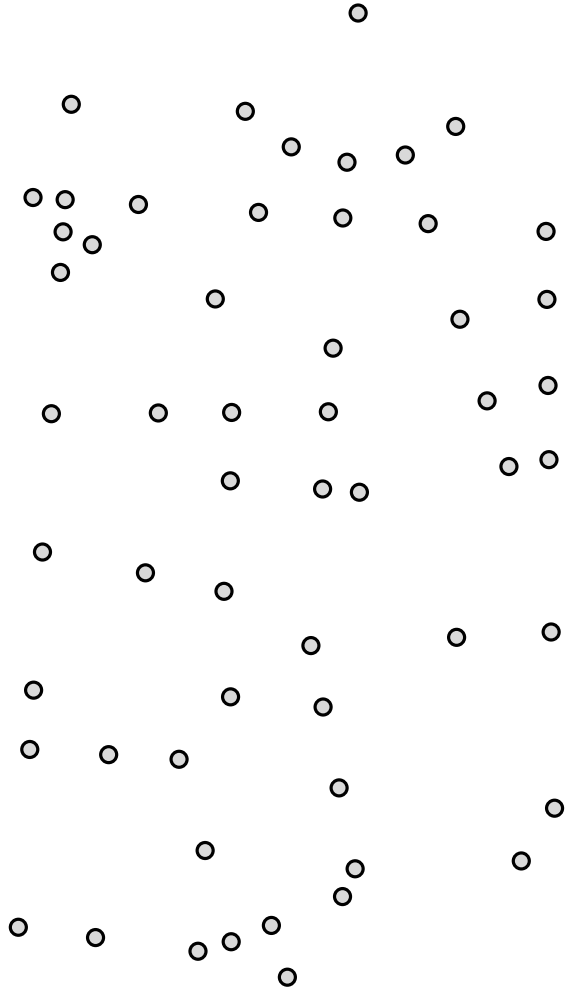


Problem Statement

Given:

- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
  - Vertices (crossings, places)

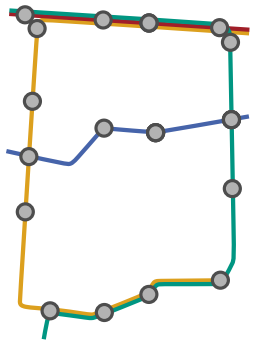


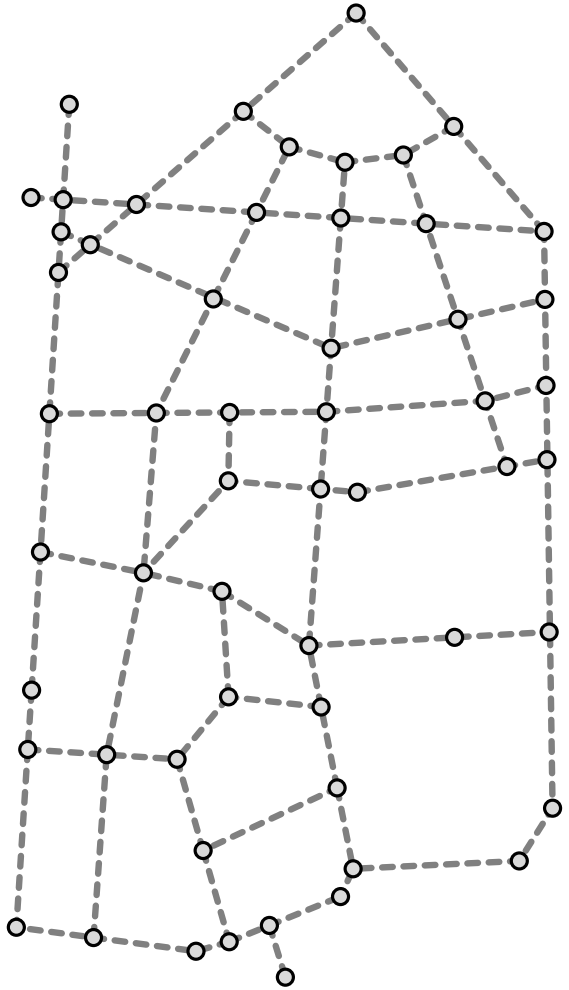


Problem Statement

Given:

- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
  - Vertices (crossings, places)
  - Edges (roads, paths)

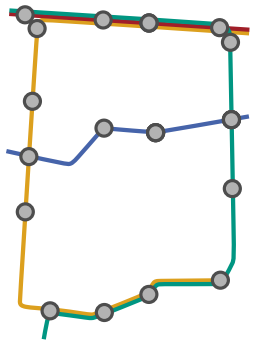


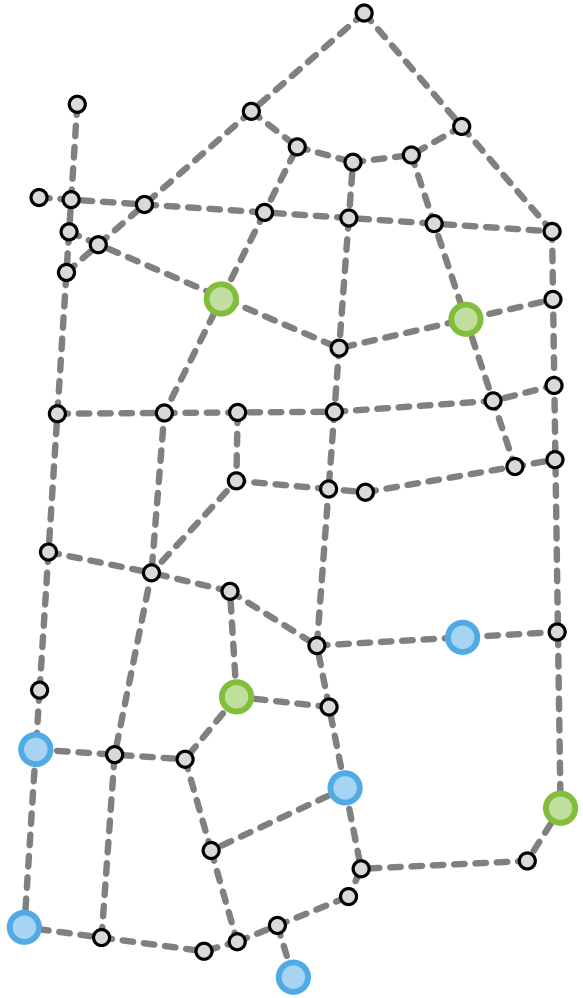


Problem Statement

Given:

- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
  - Vertices (crossings, places)
  - Edges (roads, paths)
  - Bike sharing stations (per operator)

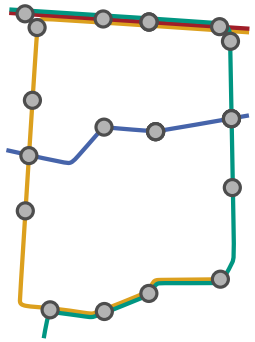


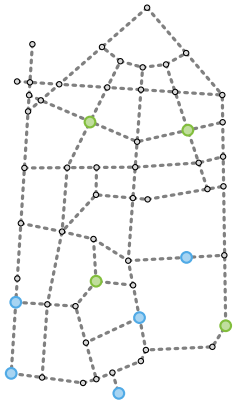


Problem Statement

Given:

- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
  - Vertices (crossings, places)
  - Edges (roads, paths)
  - Bike sharing stations (per operator)

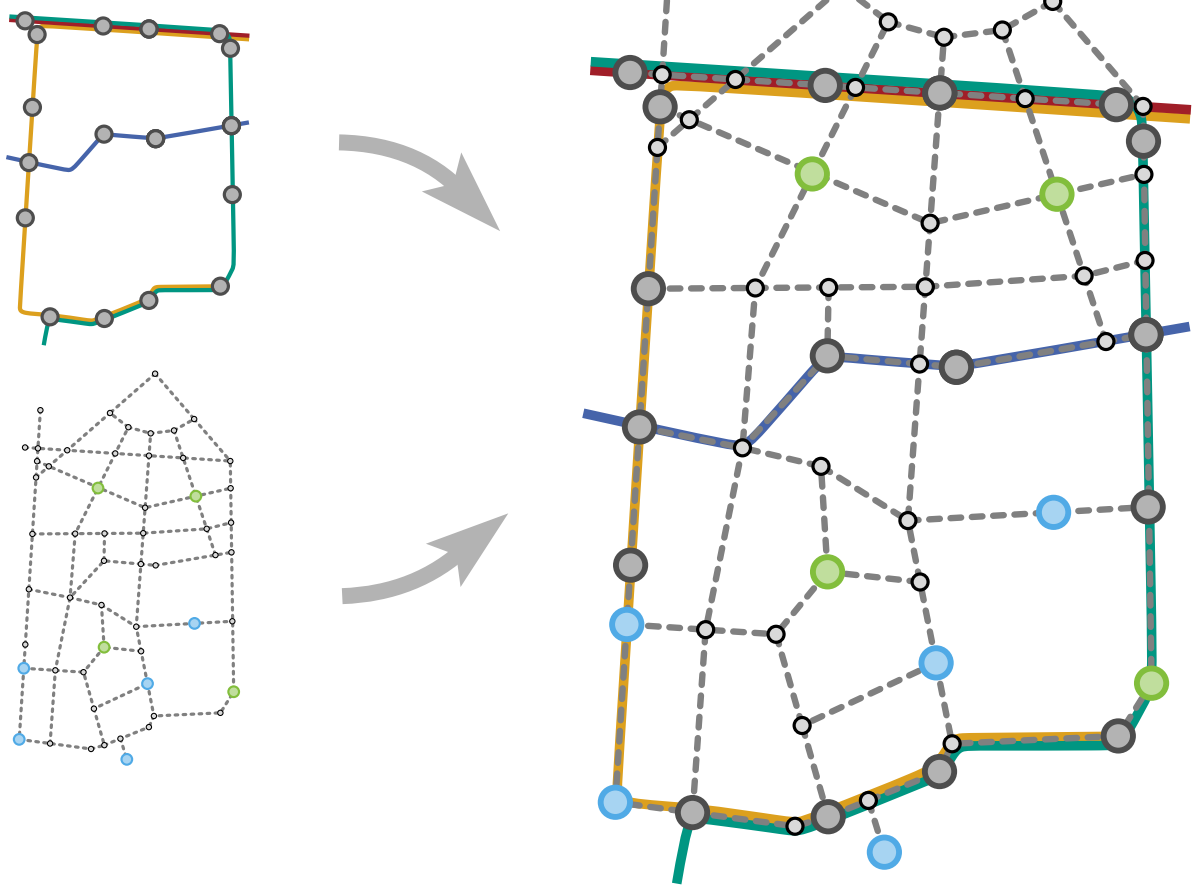




Problem Statement

Given:

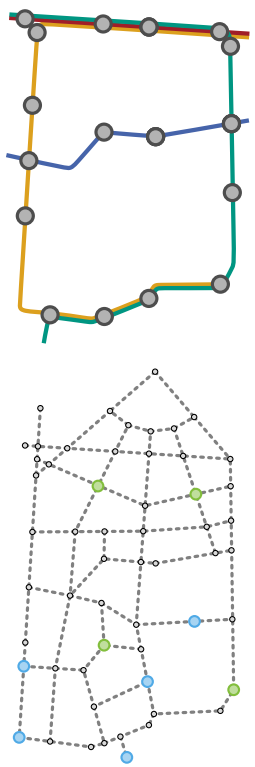
- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
  - Vertices (crossings, places)
  - Edges (roads, paths)
  - Bike sharing stations (per operator)

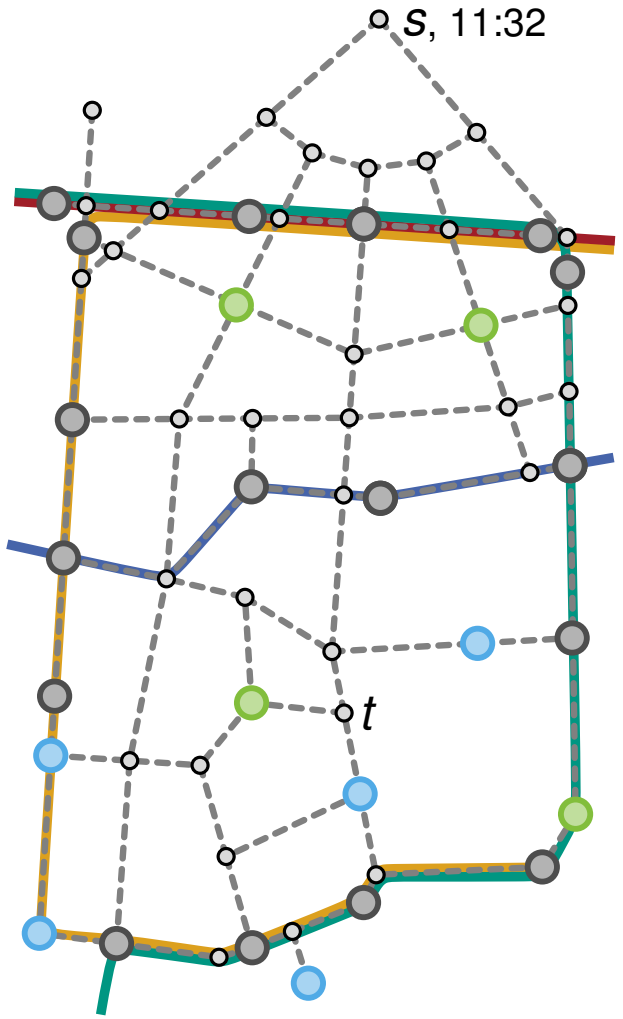


Problem Statement

Given:

- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
  - Vertices (crossings, places)
  - Edges (roads, paths)
  - Bike sharing stations (per operator)
- Source *s*, target *t*, and a departure time

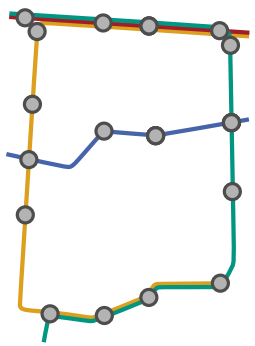


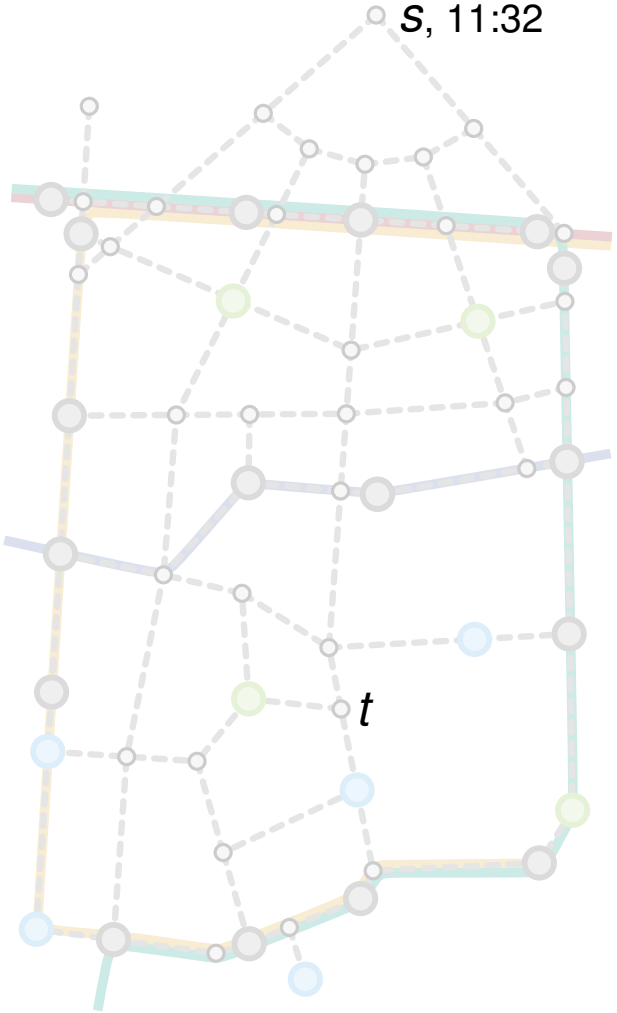


Problem Statement

Given:

- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
  - Vertices (crossings, places)
  - Edges (roads, paths)
  - Bike sharing stations (per operator)
- Source *s*, target *t*, and a departure time



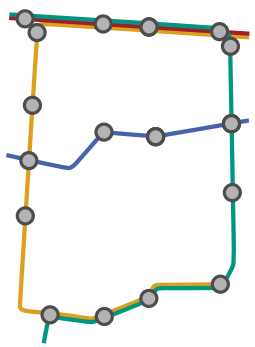


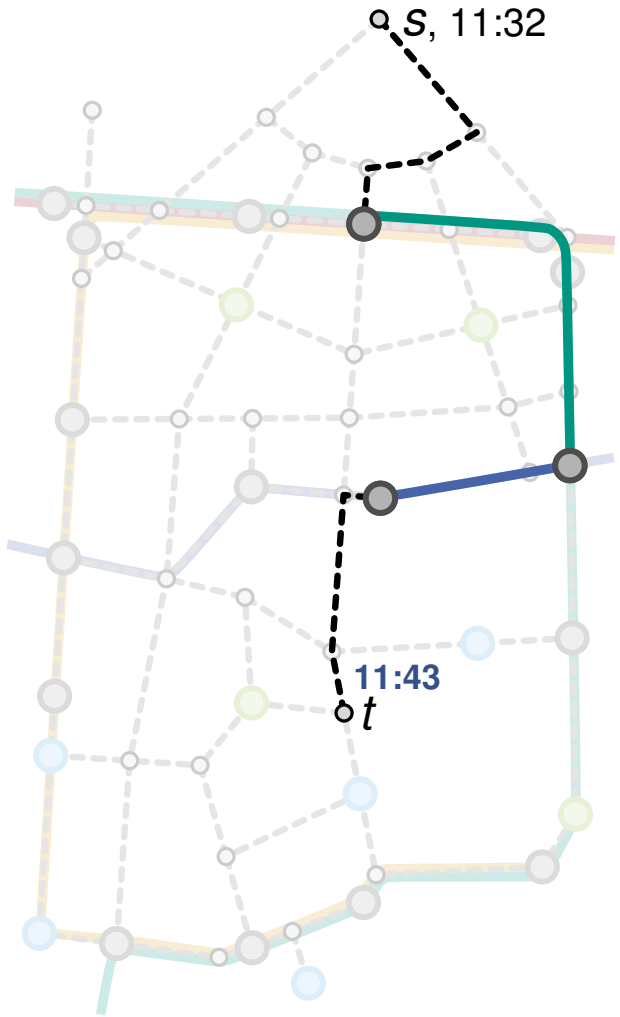
Objective: Find all Pareto-optimal journeys w.r.t. **arrival time** and **number of trips**

Problem Statement

Given:

- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
  - Vertices (crossings, places)
  - Edges (roads, paths)
  - Bike sharing stations (per operator)
- Source *s*, target *t*, and a departure time



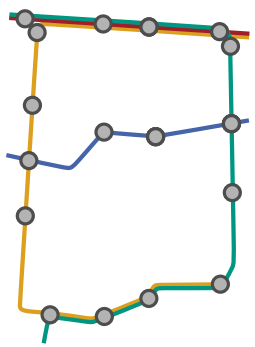


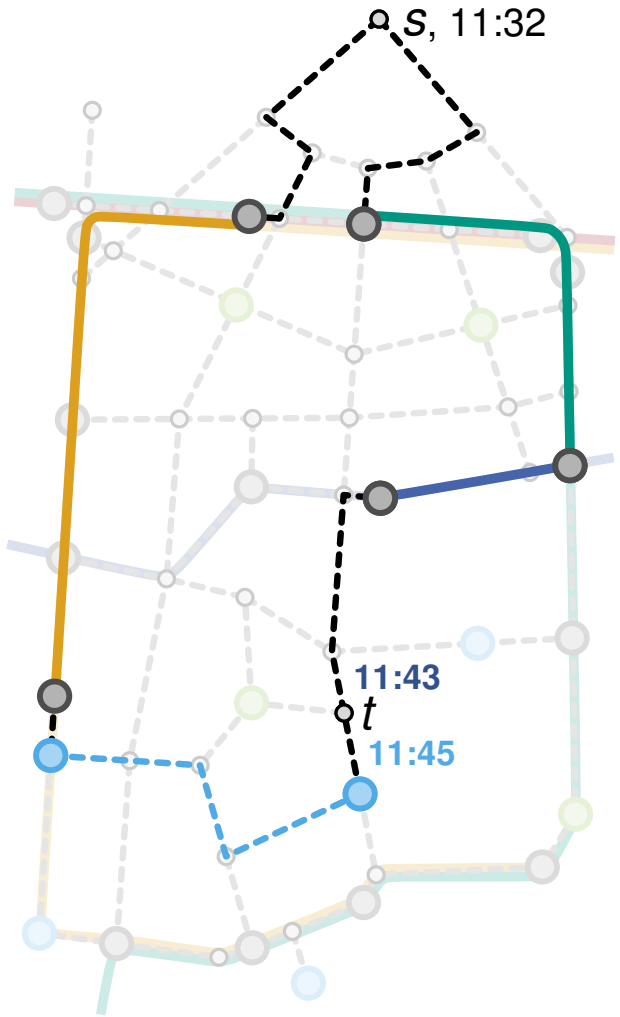
Objective: Find all Pareto-optimal journeys w.r.t. **arrival time** and **number of trips**

Problem Statement

Given:

- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
  - Vertices (crossings, places)
  - Edges (roads, paths)
  - Bike sharing stations (per operator)
- Source *s*, target *t*, and a departure time



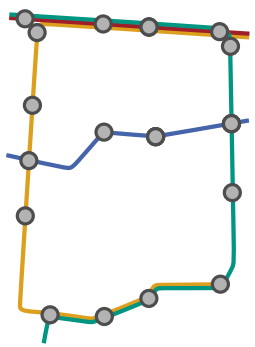


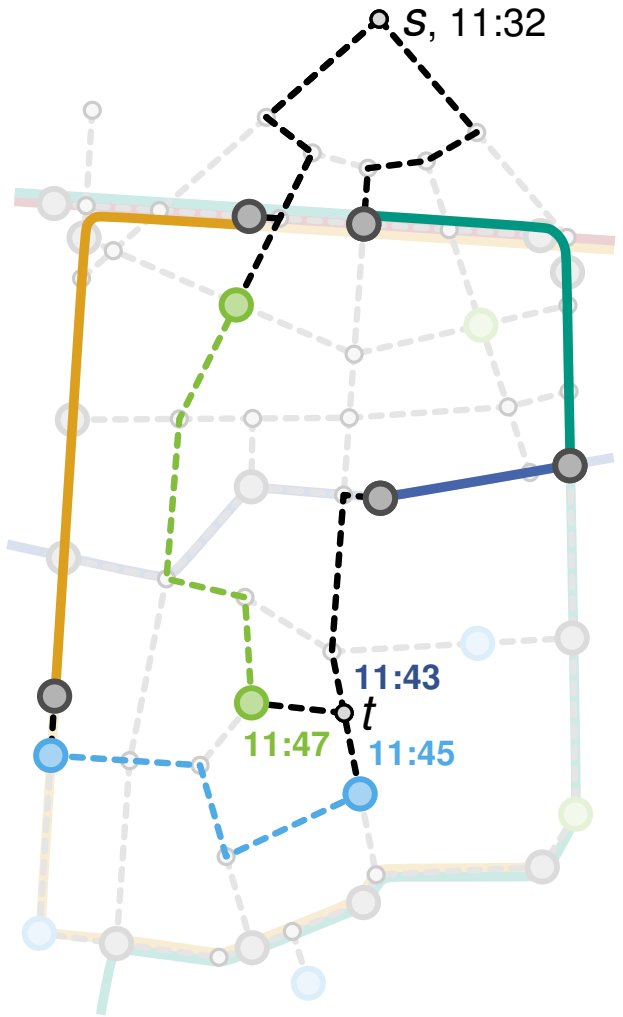
Objective: Find all Pareto-optimal journeys w.r.t. **arrival time** and **number of trips**

Problem Statement

Given:

- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)
- Transfer graph (non-schedule based)
  - Vertices (crossings, places)
  - Edges (roads, paths)
  - Bike sharing stations (per operator)
- Source *s*, target *t*, and a departure time





Objective: Find all Pareto-optimal journeys w.r.t. **arrival time** and **number of trips**

Approaches

Greatest Challenge:

- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared

Approaches

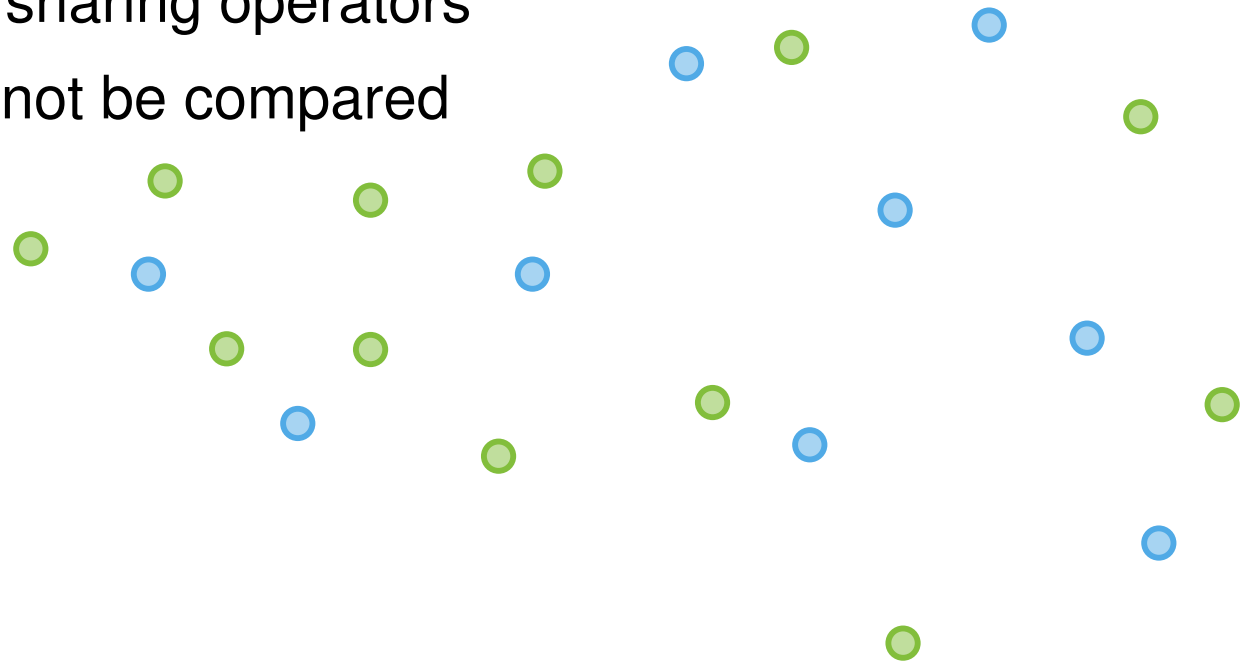
Greatest Challenge:

- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared

Approaches

Greatest Challenge:

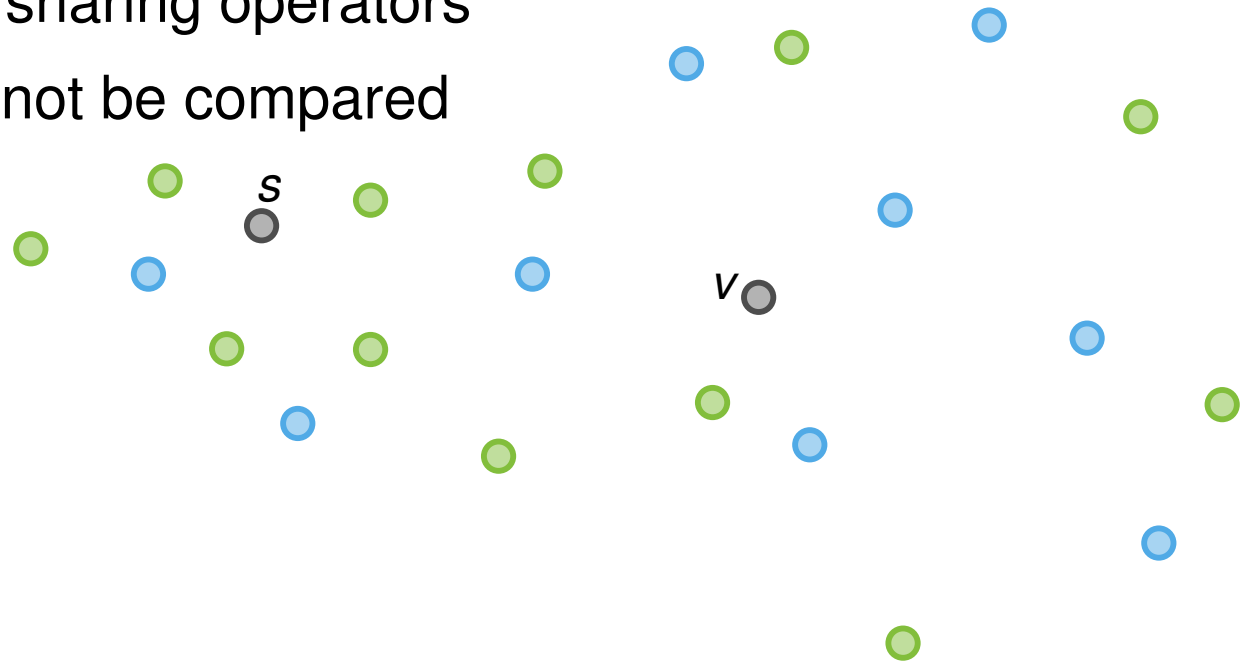
- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared



Approaches

Greatest Challenge:

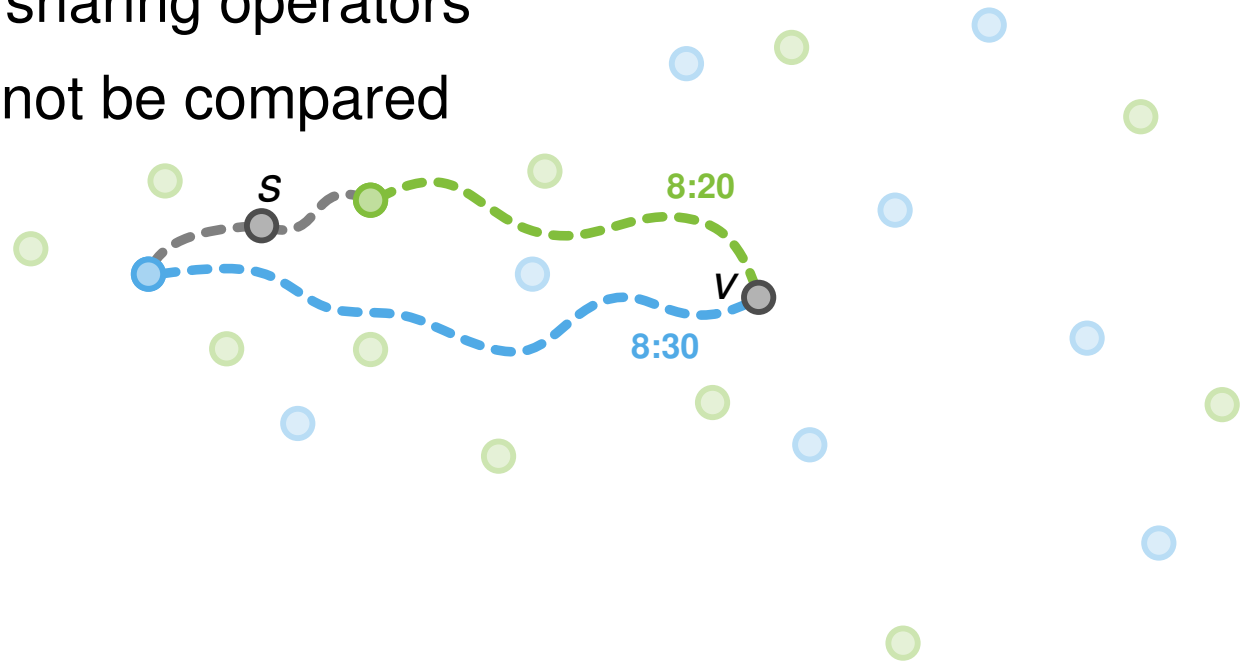
- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared



Approaches

Greatest Challenge:

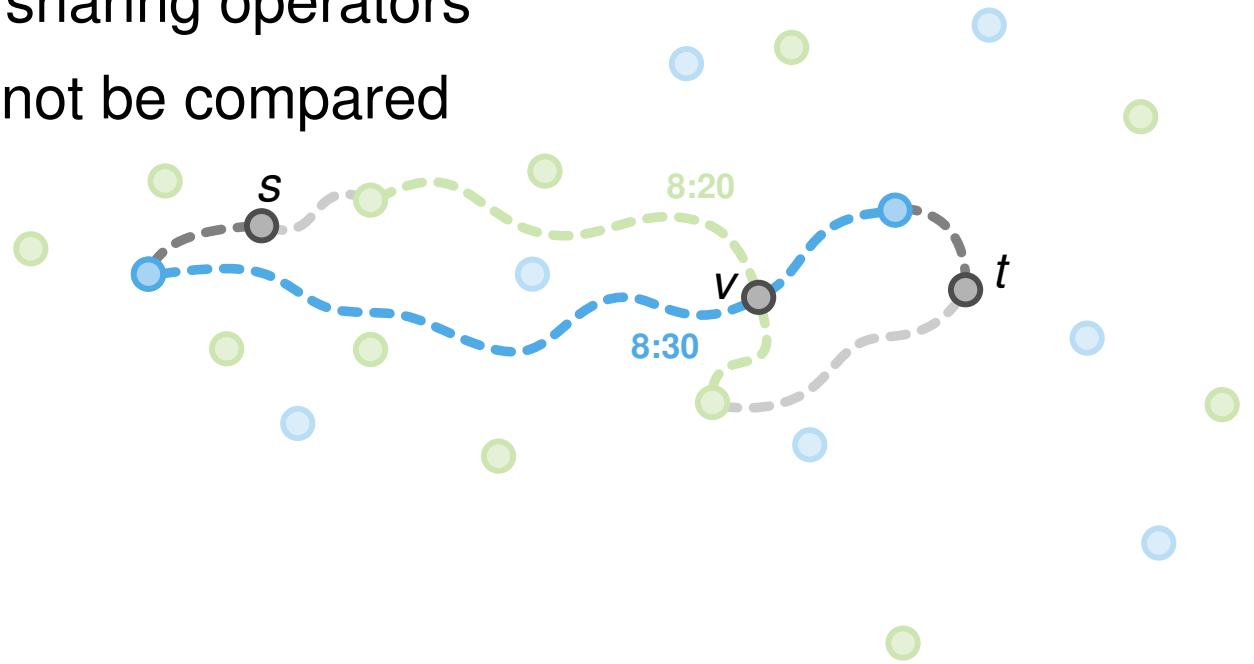
- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared



Approaches

Greatest Challenge:

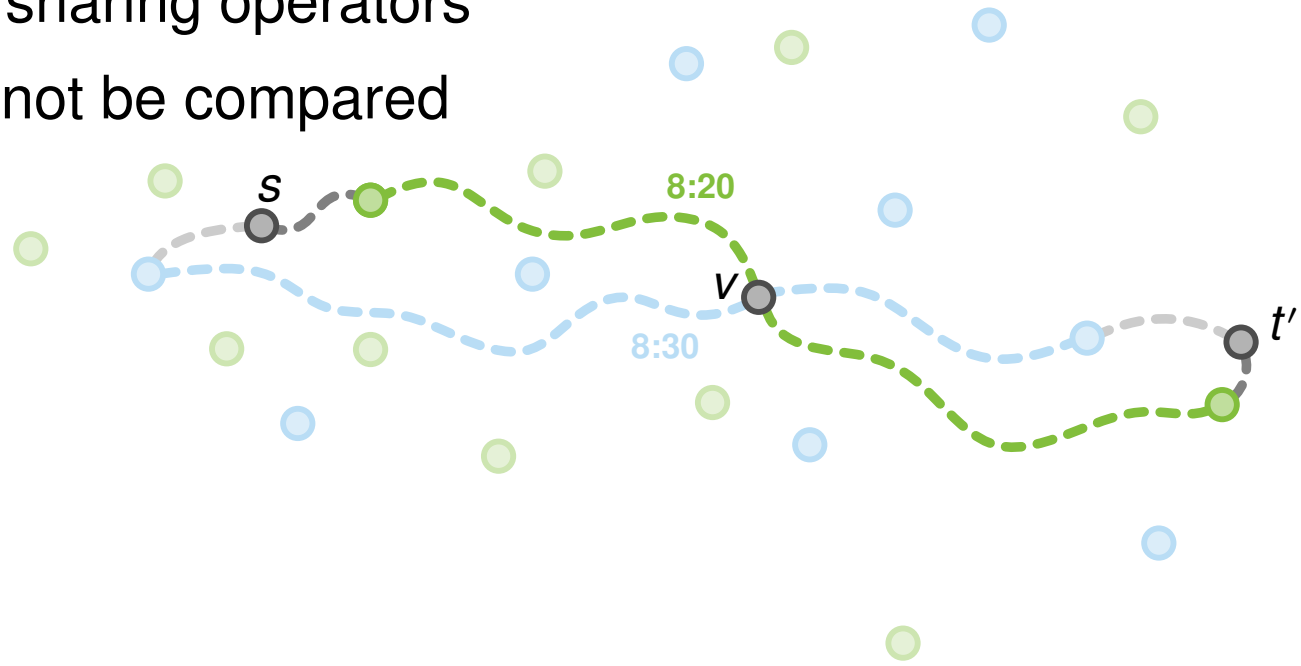
- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared



Approaches

Greatest Challenge:

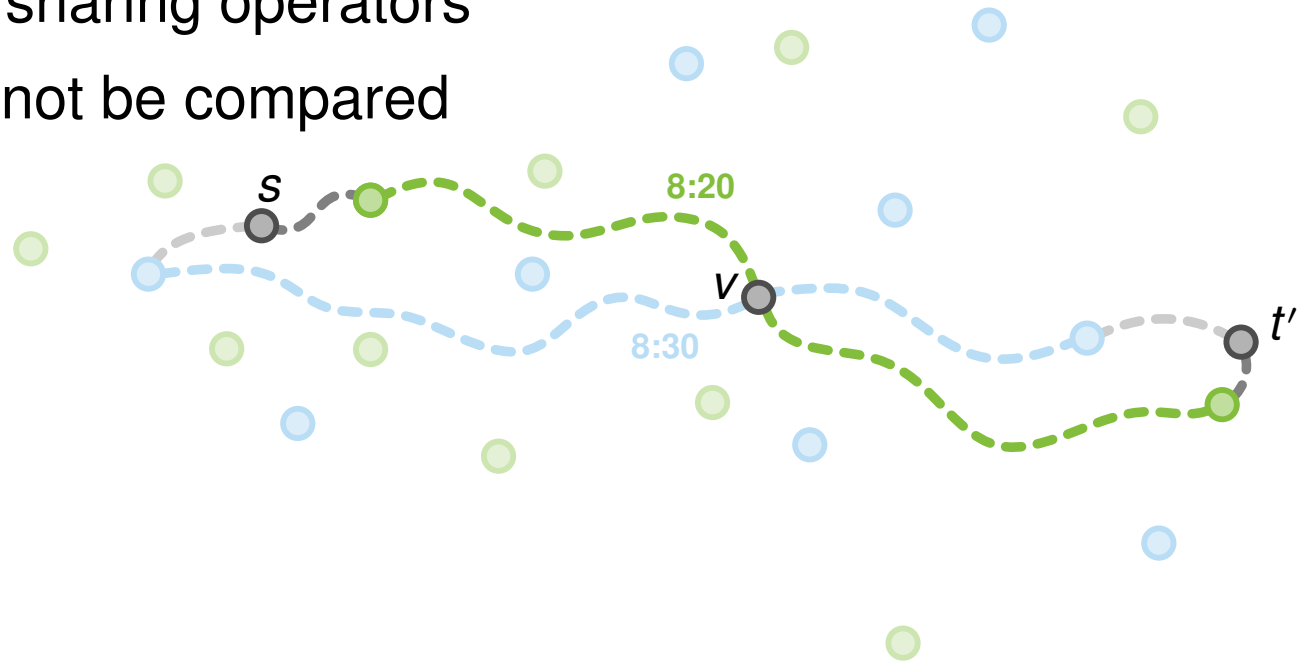
- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared



Approaches

Greatest Challenge:

- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared

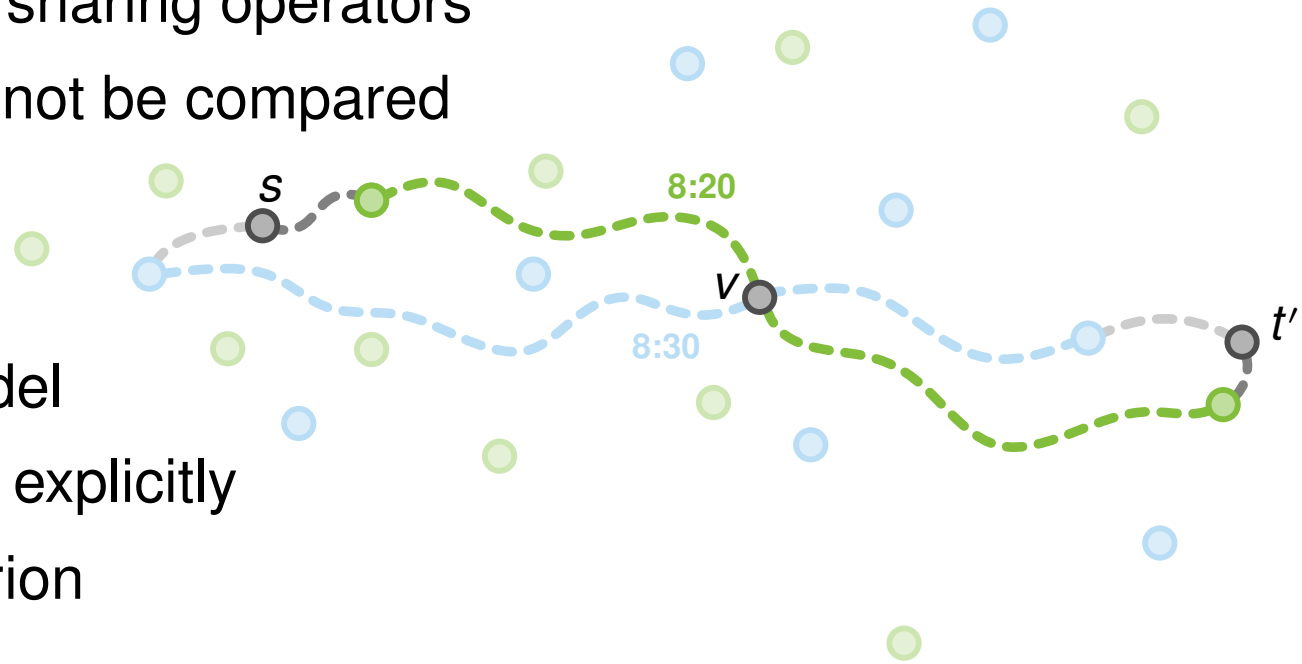


Two Possible Solutions:

Approaches

Greatest Challenge:

- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared



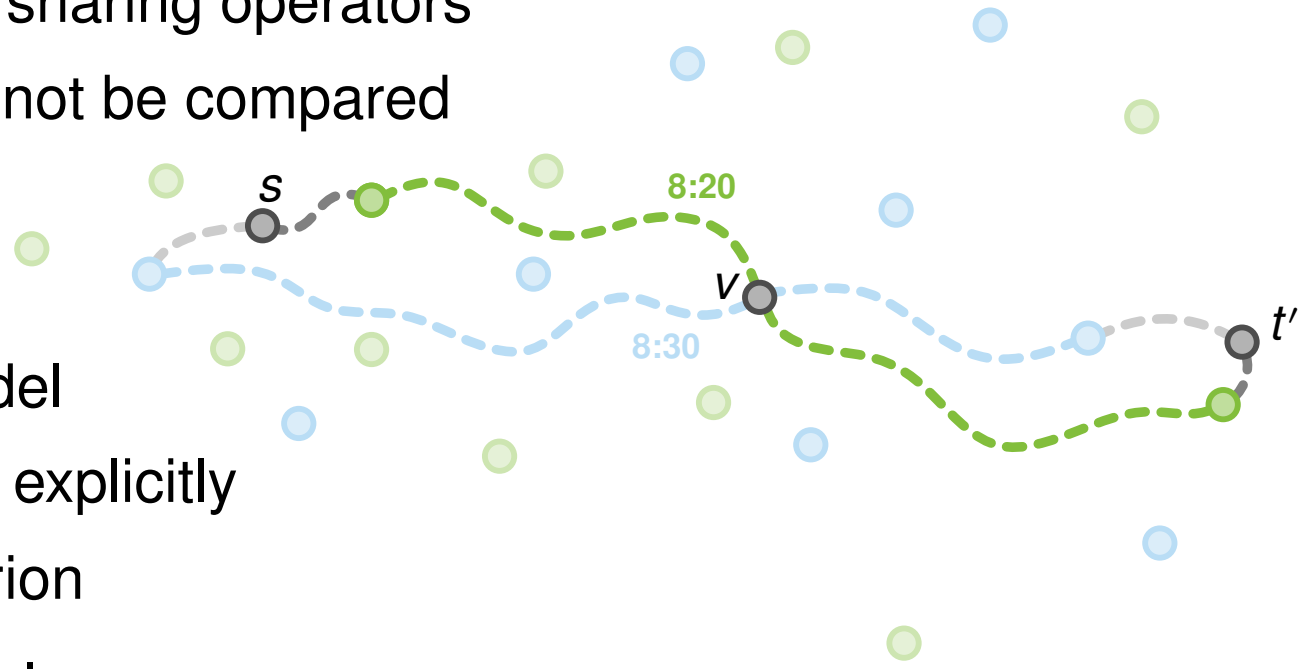
Two Possible Solutions:

- The **Operator-Dependent** (OD) model
  - Handle operators in the algorithm explicitly
  - Similar to a third dominance criterion

Approaches

Greatest Challenge:

- Distinguish and handle multiple bike sharing operators
- Labels with different rental bikes cannot be compared



Two Possible Solutions:

- The **Operator-Dependent** (OD) model
  - Handle operators in the algorithm explicitly
  - Similar to a third dominance criterion
- The **Operator-Expanded** (OE) model
  - Encode operators within a “normal” network
  - Use an existing algorithm with the modified network

The Operator-Dependent (OD) Model

Karlsruhe Institute of Technology

Basic Idea:

- Treat bike sharing as an additional optimization criterion
- Handle renting and returning of bicycles with the algorithm

The Operator-Dependent (OD) Model

Karlsruhe Institute of Technology

Basic Idea:

- Treat bike sharing as an additional optimization criterion
- Handle renting and returning of bicycles with the algorithm

Integration into multi-Modal multi-Criteria RAPTOR (MCR): [Delling et al. 2013]

- Naive:
  - Use label-bags of MCR for bike sharing operators

The Operator-Dependent (OD) Model

Basic Idea:

- Treat bike sharing as an additional optimization criterion
- Handle renting and returning of bicycles with the algorithm

Integration into multi-Modal multi-Criteria RAPTOR (MCR): [Delling et al. 2013]

- Naive:
  - Use label-bags of MCR for bike sharing operators

Bag per **stop**, **#trips**

The Operator-Dependent (OD) Model

Basic Idea:

- Treat bike sharing as an additional optimization criterion
- Handle renting and returning of bicycles with the algorithm

Integration into multi-Modal multi-Criteria RAPTOR (MCR): [Delling et al. 2013]

- Naive:
  - Use label-bags of MCR for bike sharing operators
- Observation:
  - Bike sharing operators are few and discrete
  - Scan routes separately for each operator

Bag per **stop**, **#trips**

The Operator-Dependent (OD) Model

Basic Idea:

- Treat bike sharing as an additional optimization criterion
- Handle renting and returning of bicycles with the algorithm

Integration into multi-Modal multi-Criteria RAPTOR (MCR): [Delling et al. 2013]

- Naive:
  - Use label-bags of MCR for bike sharing operators
- Observation:
  - Bike sharing operators are few and discrete
  - Scan routes separately for each operator

Bag per **stop**, **#trips**

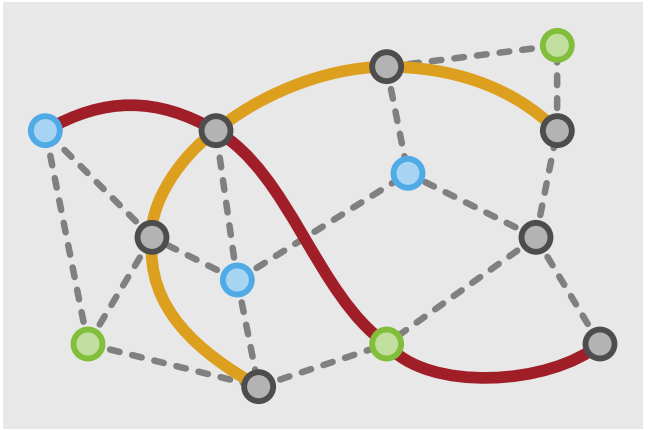
⇓

Entry per **stop**, **#trips**, **operator**

The Operator-Expanded (OE) Model

Basic Idea:

- Encode bike sharing within a “normal” network

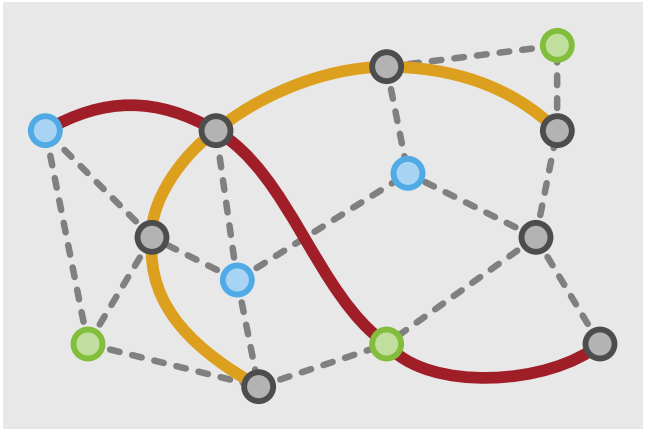


The Operator-Expanded (OE) Model

Basic Idea:

- Encode bike sharing within a “normal” network

Our Approach:

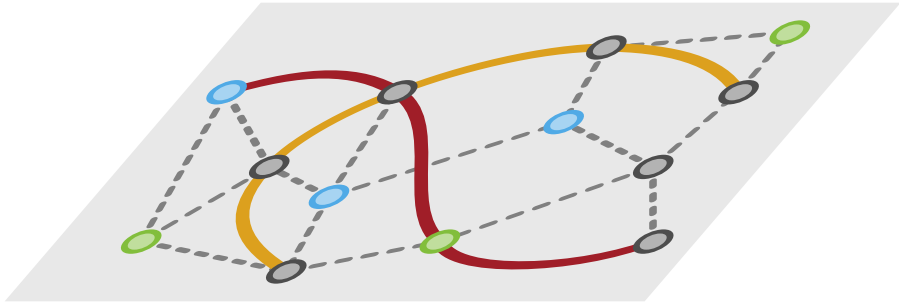


The Operator-Expanded (OE) Model

Basic Idea:

- Encode bike sharing within a “normal” network

Our Approach:



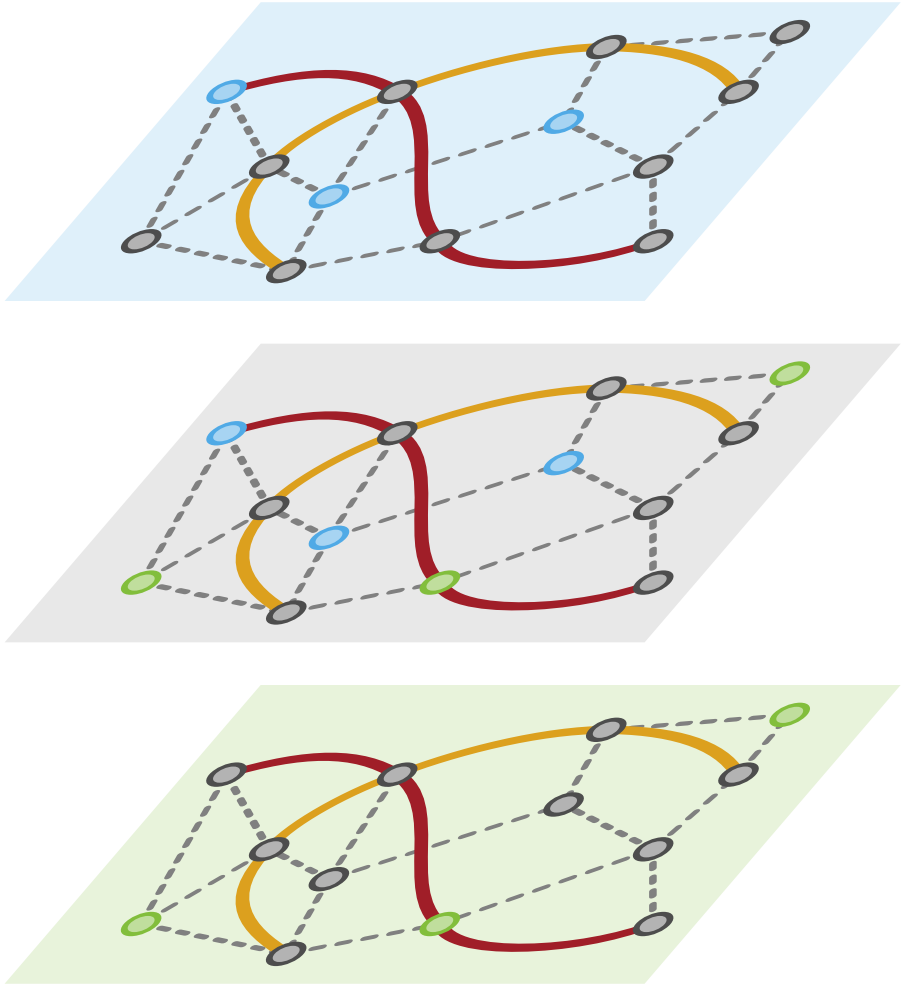
The Operator-Expanded (OE) Model

Basic Idea:

- Encode bike sharing within a “normal” network

Our Approach:

- Copy network once per operator



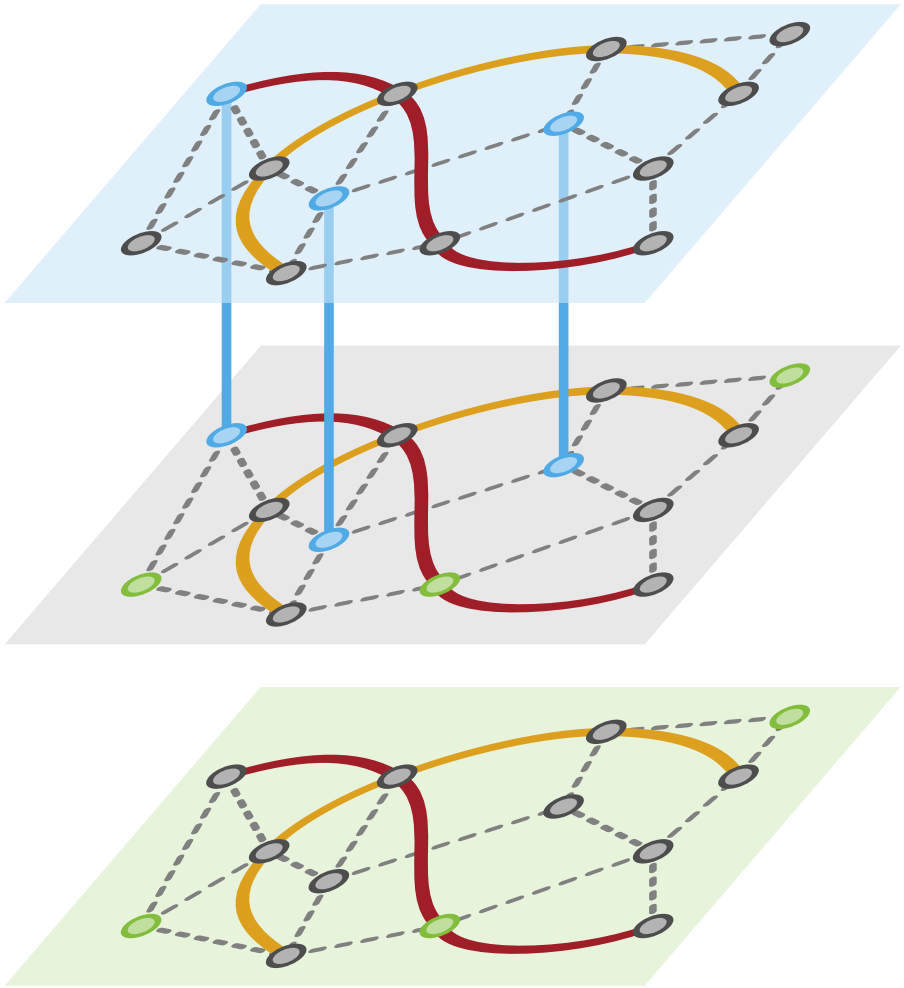
The Operator-Expanded (OE) Model

Basic Idea:

- Encode bike sharing within a “normal” network

Our Approach:

- Copy network once per operator
- Connect networks at bike sharing stations



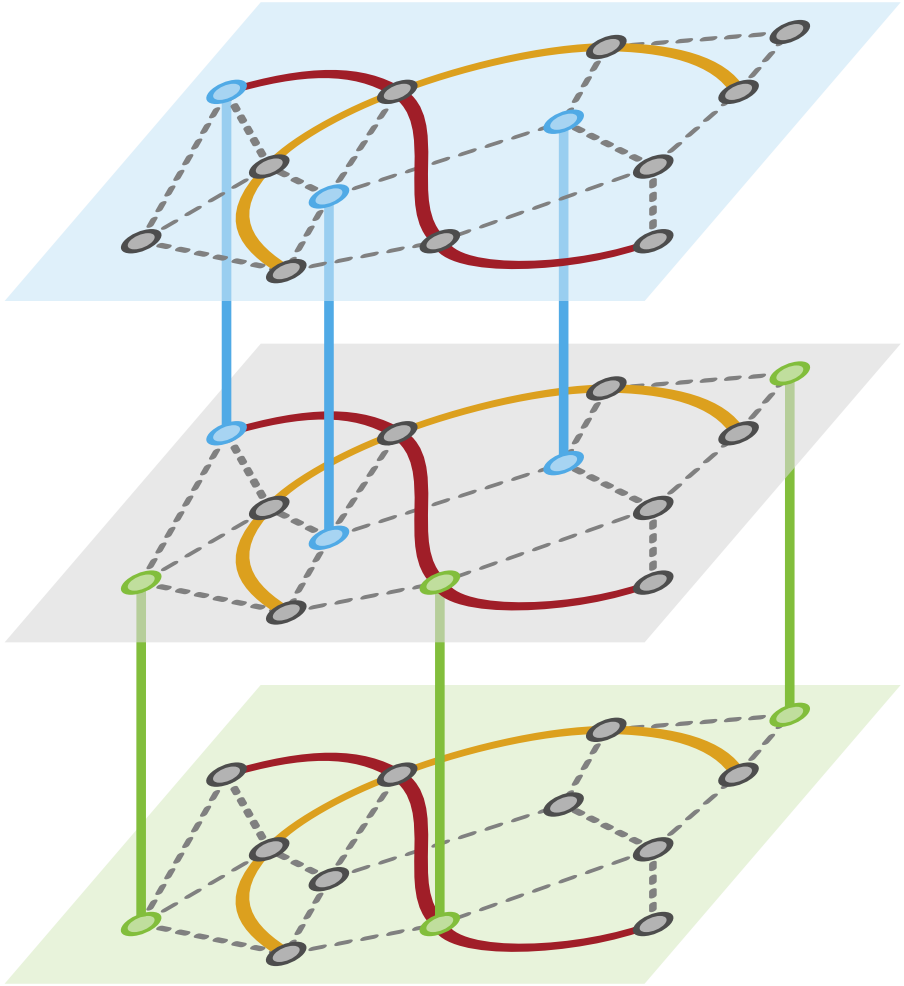
The Operator-Expanded (OE) Model

Basic Idea:

- Encode bike sharing within a “normal” network

Our Approach:

- Copy network once per operator
- Connect networks at bike sharing stations



The Operator-Expanded (OE) Model

Basic Idea:

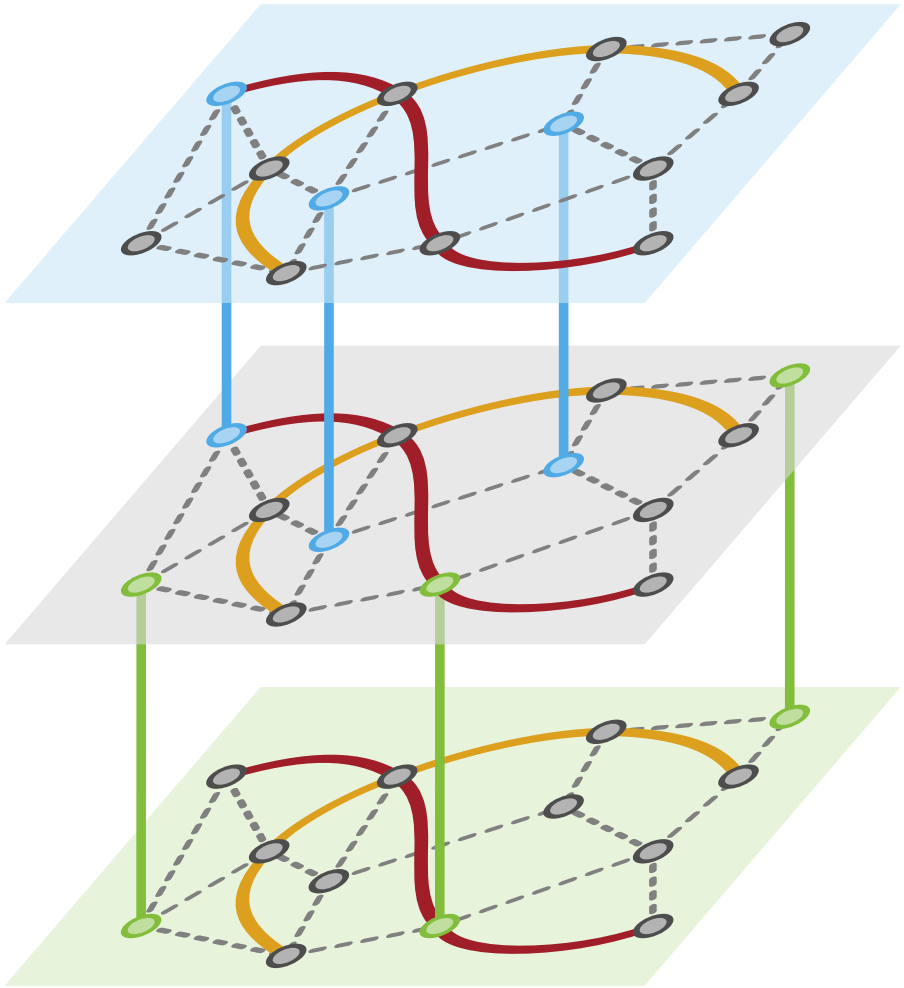
- Encode bike sharing within a “normal” network

Our Approach:

- Copy network once per operator
- Connect networks at bike sharing stations

Properties:

- Any existing algorithm can run on this network
- Using the green network ⇔ Renting a green bike
  (Using the blue network ⇔ Renting a blue bike)



The Operator-Expanded (OE) Model

Basic Idea:

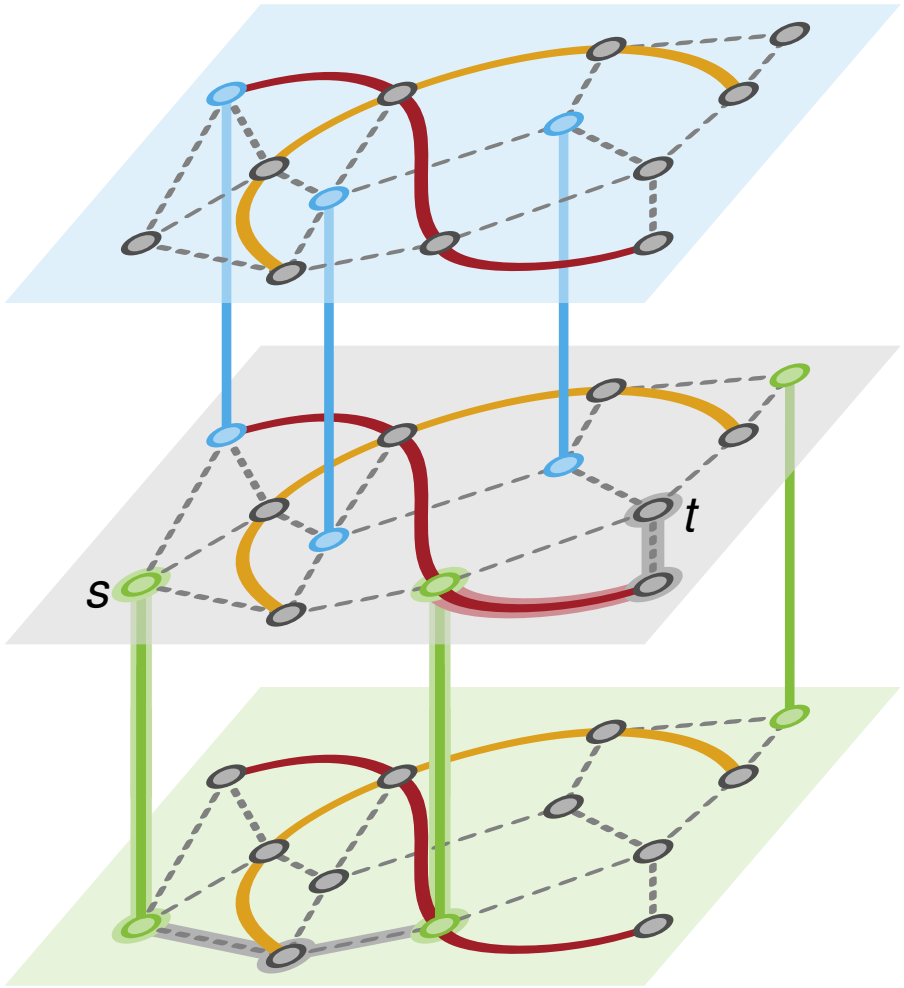
- Encode bike sharing within a “normal” network

Our Approach:

- Copy network once per operator
- Connect networks at bike sharing stations

Properties:

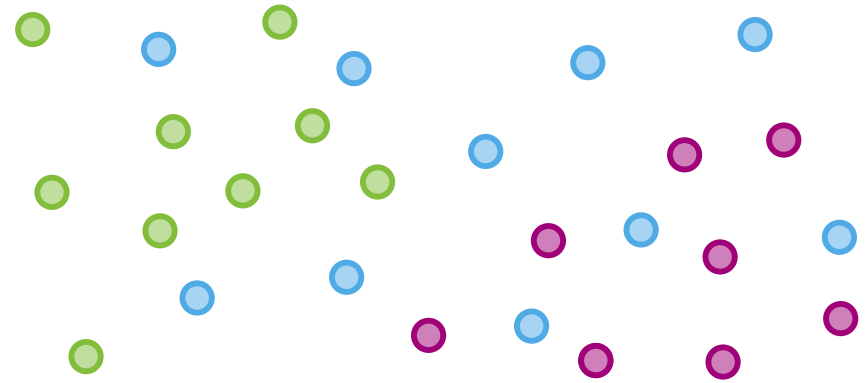
- Any existing algorithm can run on this network
- Using the green network ⇔ Renting a green bike
  (Using the blue network ⇔ Renting a blue bike)



Speed-up Technique: Operator Pruning (OP)

Observation:

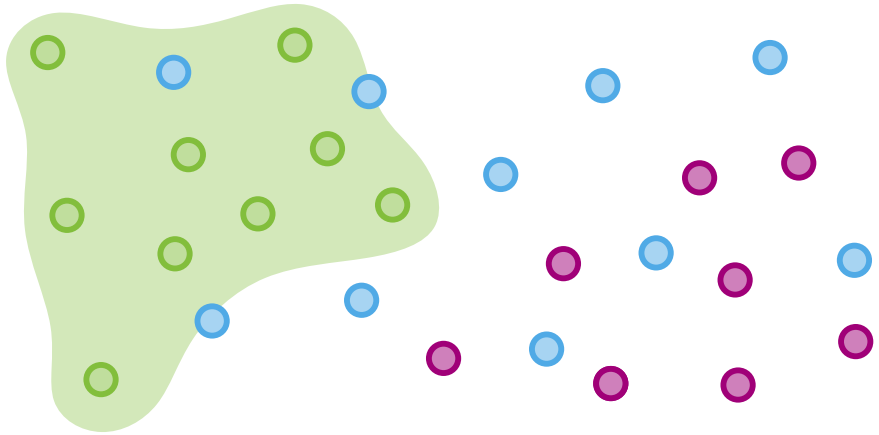
- Not every rental bike is useful throughout the whole network



Speed-up Technique: Operator Pruning (OP)

Observation:

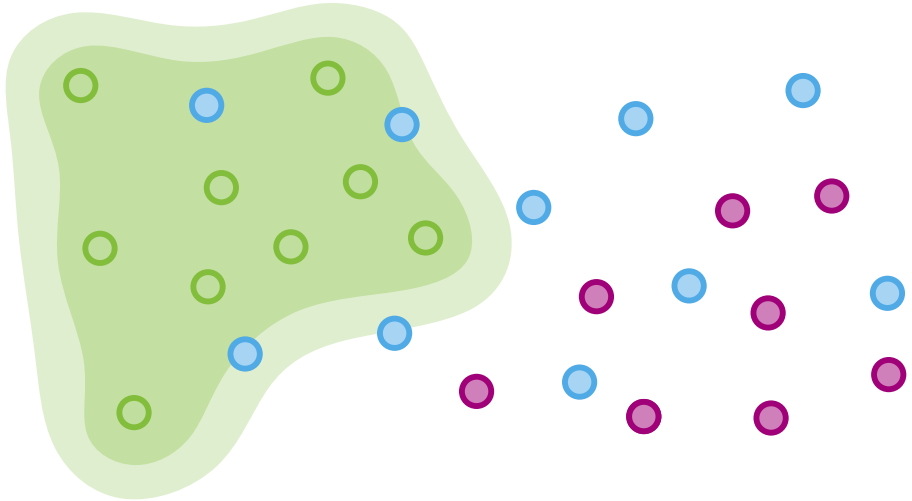
- Not every rental bike is useful throughout the whole network



Speed-up Technique: Operator Pruning (OP)

Observation:

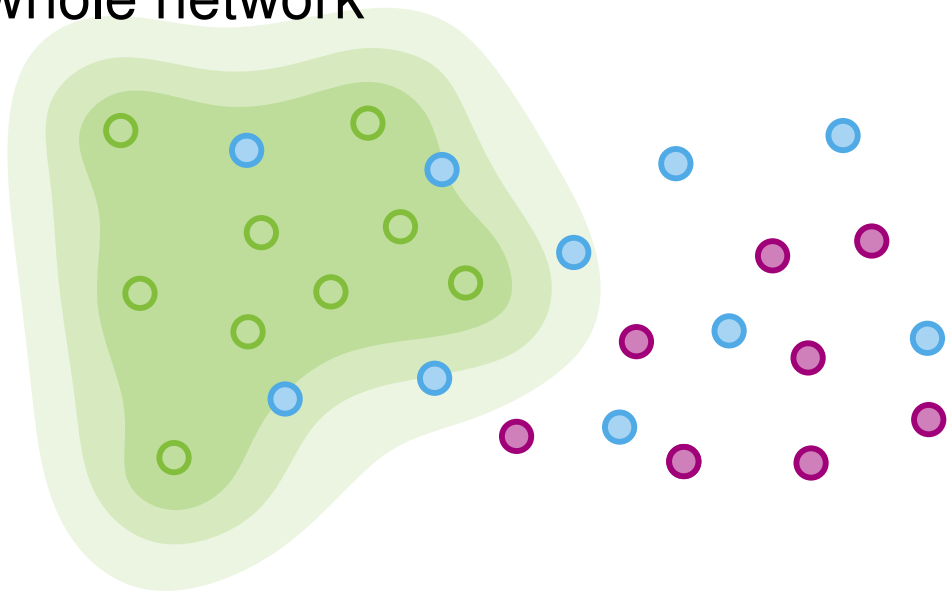
- Not every rental bike is useful throughout the whole network



Speed-up Technique: Operator Pruning (OP)

Observation:

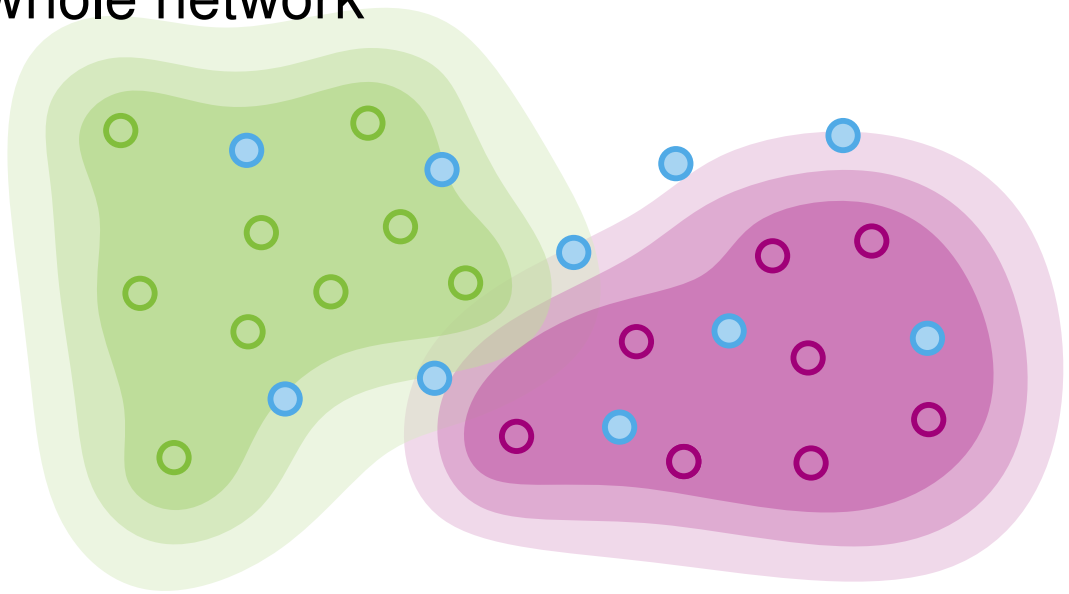
- Not every rental bike is useful throughout the whole network



Speed-up Technique: Operator Pruning (OP)

Observation:

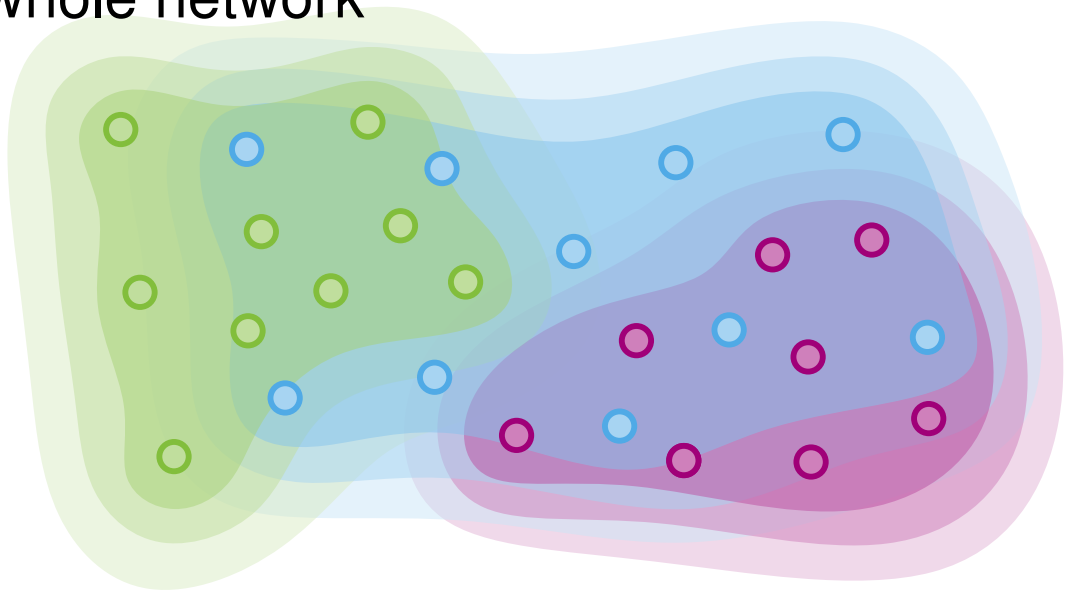
- Not every rental bike is useful throughout the whole network



Speed-up Technique: Operator Pruning (OP)

Observation:

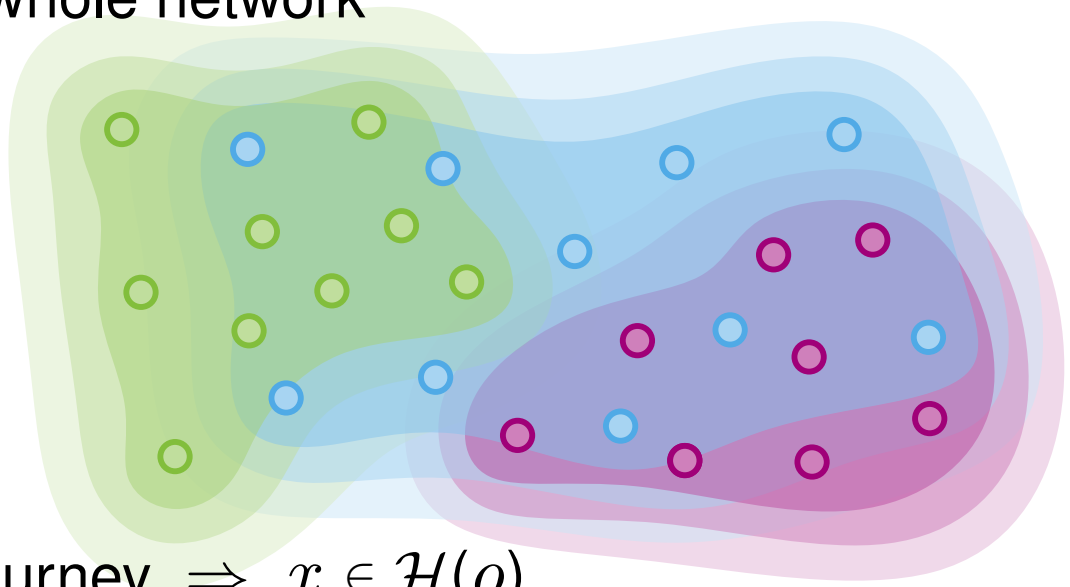
- Not every rental bike is useful throughout the whole network



Speed-up Technique: Operator Pruning (OP)

Observation:

- Not every rental bike is useful throughout the whole network



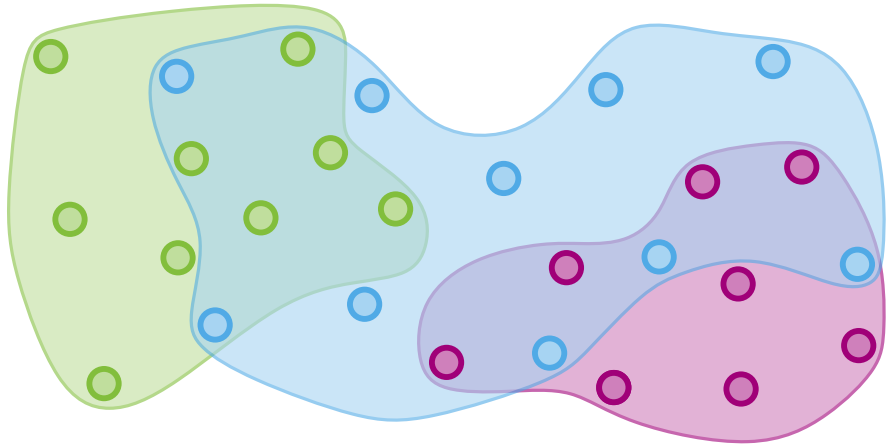
Operator Hull $\mathcal{H}$:

- Subset of the network
- For every bike sharing operator o
- For every vertex/edge/trip x in the network
- If x is used with a bike of o in some optimal journey $\Rightarrow$ $x \in \mathcal{H}(o)$

Speed-up Technique: Operator Pruning (OP)

Observation:

- Not every rental bike is useful throughout the whole network



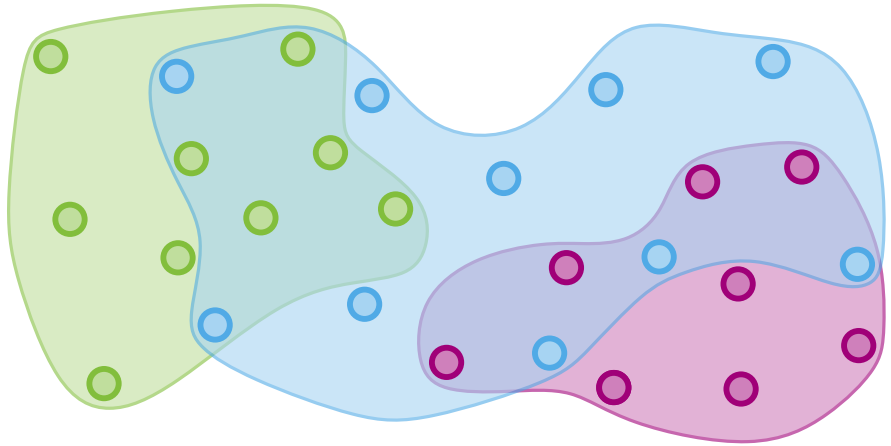
Operator Hull $\mathcal{H}$:

- Subset of the network
- For every bike sharing operator o
- For every vertex/edge/trip x in the network
- If x is used with a bike of o in some optimal journey $\Rightarrow$ $x \in \mathcal{H}(o)$

Speed-up Technique: Operator Pruning (OP)

Observation:

- Not every rental bike is useful throughout the whole network



Operator Hull $\mathcal{H}$:

- Subset of the network
- For every bike sharing operator o
- For every vertex/edge/trip x in the network
- If x is used with a bike of o in some optimal journey $\Rightarrow$ $x \in \mathcal{H}(o)$

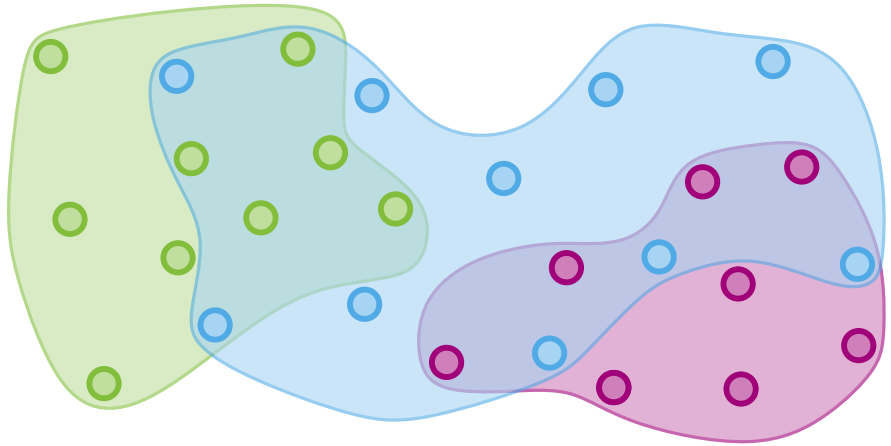
Preprocessing:

- Computing $\mathcal{H}(o)$ can be done with standard MCR

Speed-up Technique: Operator Pruning (OP)

Observation:

- Not every rental bike is useful throughout the whole network



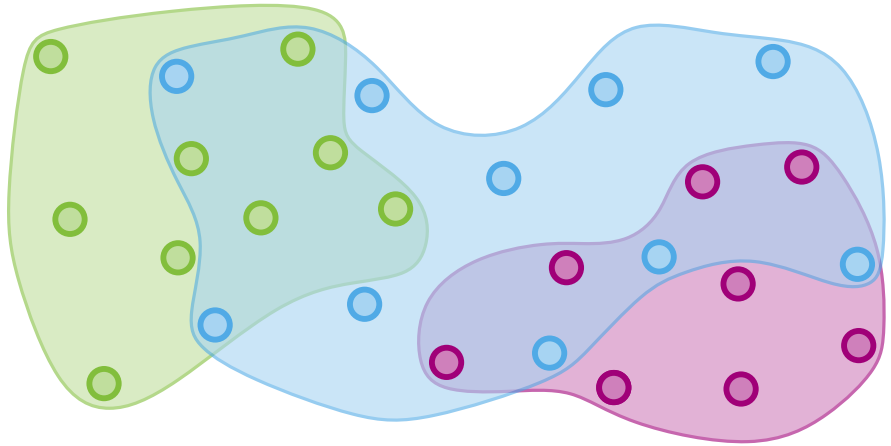
Speed-up Technique: Operator Pruning (OP)

Observation:

- Not every rental bike is useful throughout the whole network

Operator-Dependent Queries:

- Use $\mathcal{H}(o)$ to prune the search space



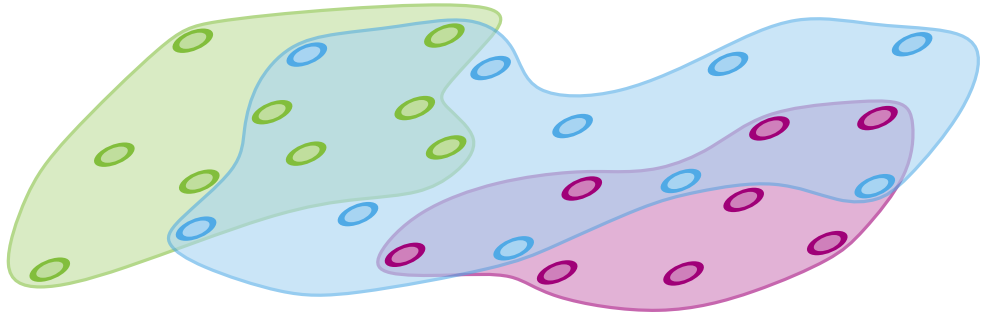
Speed-up Technique: Operator Pruning (OP)

Observation:

- Not every rental bike is useful throughout the whole network

Operator-Dependent Queries:

- Use $\mathcal{H}(o)$ to prune the search space



Operator-Expanded Queries:

- Build a reduced Network
- Do not copy the whole network
- Use $\mathcal{H}(o)$ as copy for operator o

Speed-up Technique: Operator Pruning (OP)

Observation:

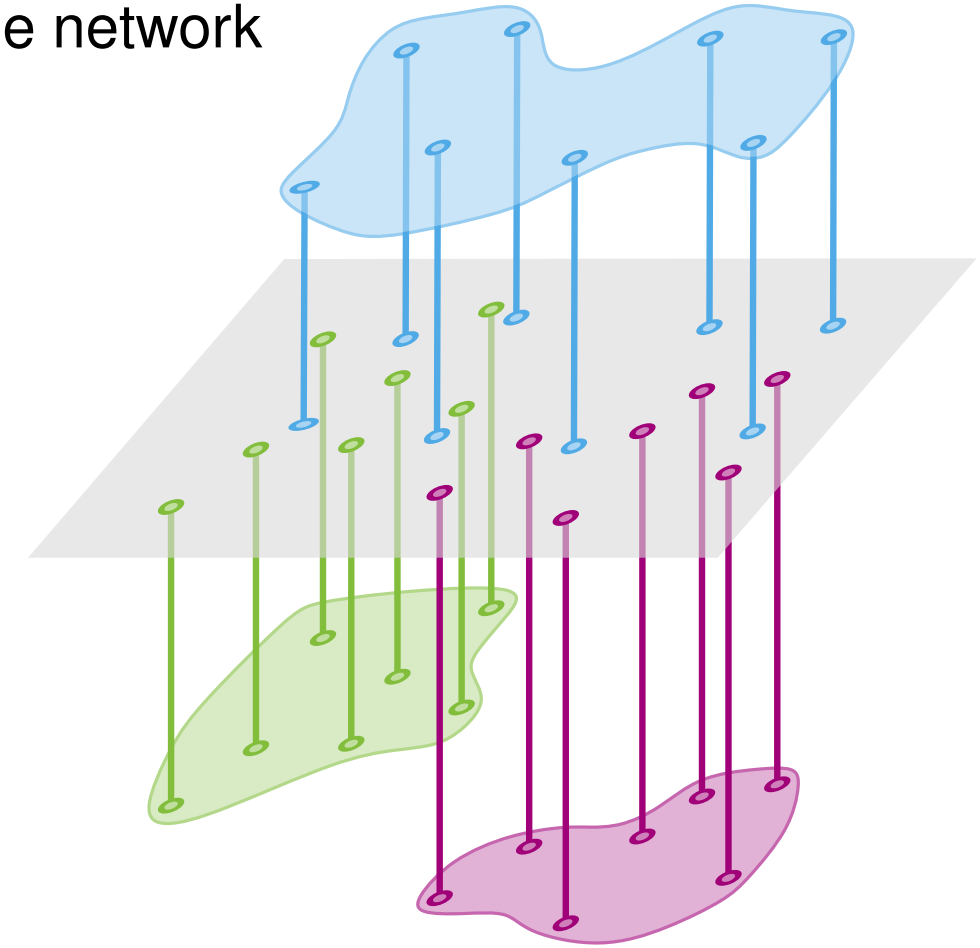
- Not every rental bike is useful throughout the whole network

Operator-Dependent Queries:

- Use $\mathcal{H}(o)$ to prune the search space

Operator-Expanded Queries:

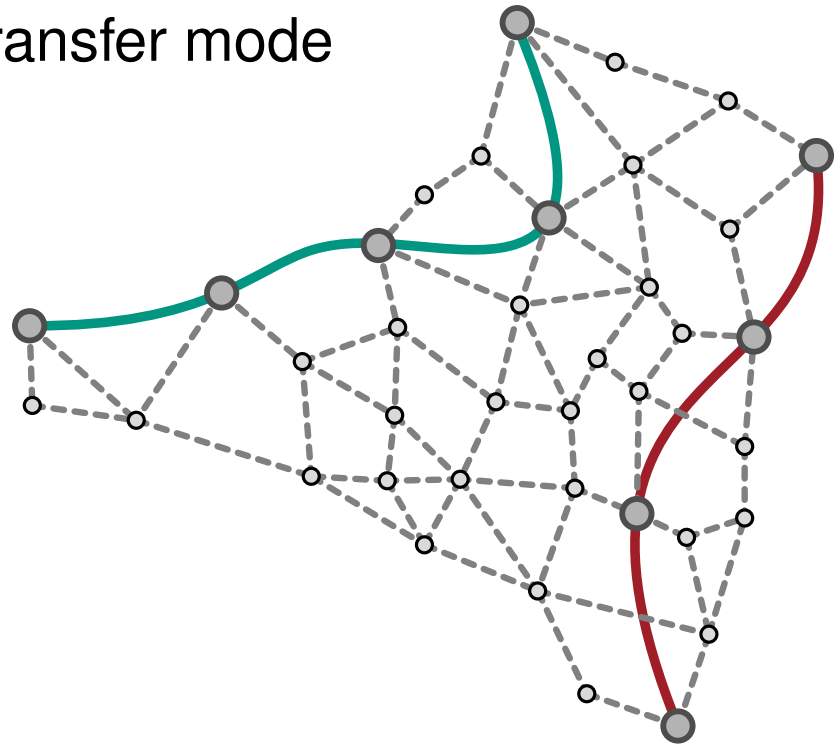
- Build a reduced Network
- Do not copy the whole network
- Use $\mathcal{H}(o)$ as copy for operator o



Speed-up Technique: Integration with ULTRA [Baum et al. 2019]

ULTRA (UnLimited TRAnsfers) overview:

- Speed-up technique for public transit + one additional transfer mode
- Replaces the transfer graph with inter-trip shortcuts

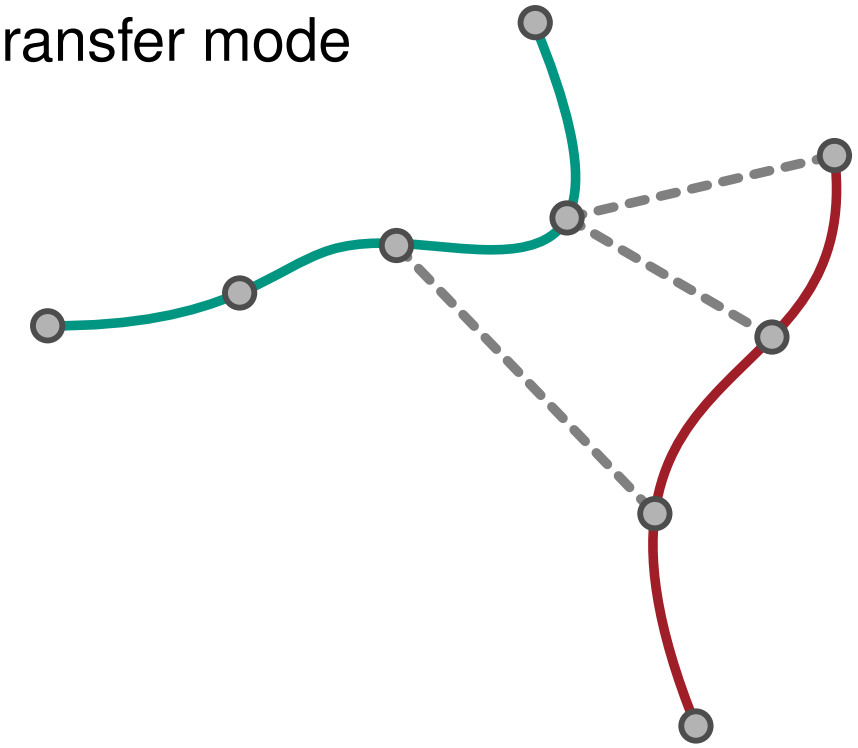


Speed-up Technique: Integration with ULTRA [Baum et al. 2019]

Karlsruhe Institute of Technology

ULTRA (UnLimited TRAnsfers) overview:

- Speed-up technique for public transit + one additional transfer mode
- Replaces the transfer graph with inter-trip shortcuts

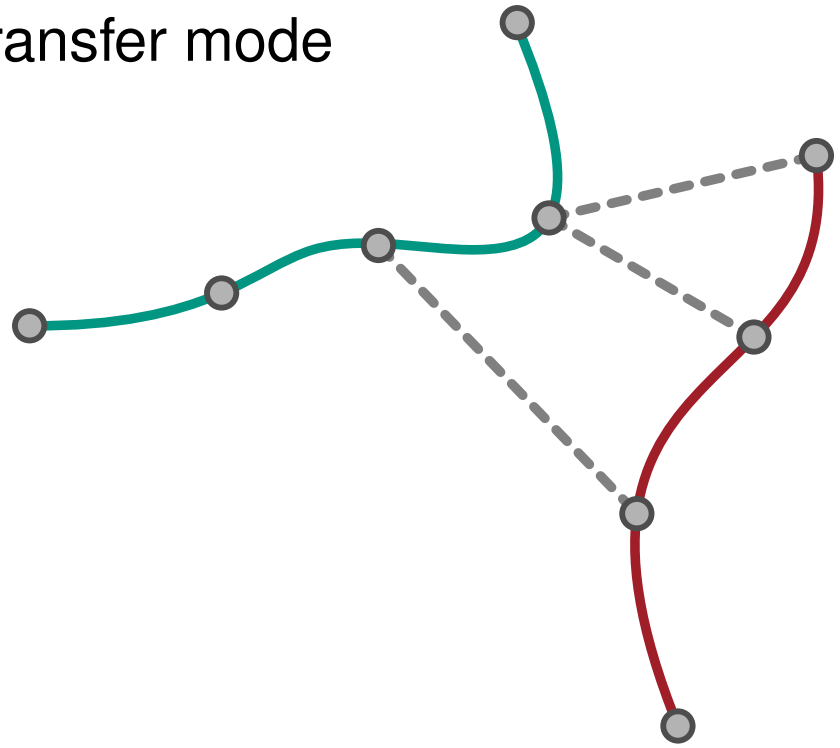


Speed-up Technique: Integration with ULTRA [Baum et al. 2019]

Karlsruhe Institute of Technology

ULTRA (UnLimited TRAnsfers) overview:

- Speed-up technique for public transit + one additional transfer mode
- Replaces the transfer graph with inter-trip shortcuts



Adaptation for Bike Sharing:

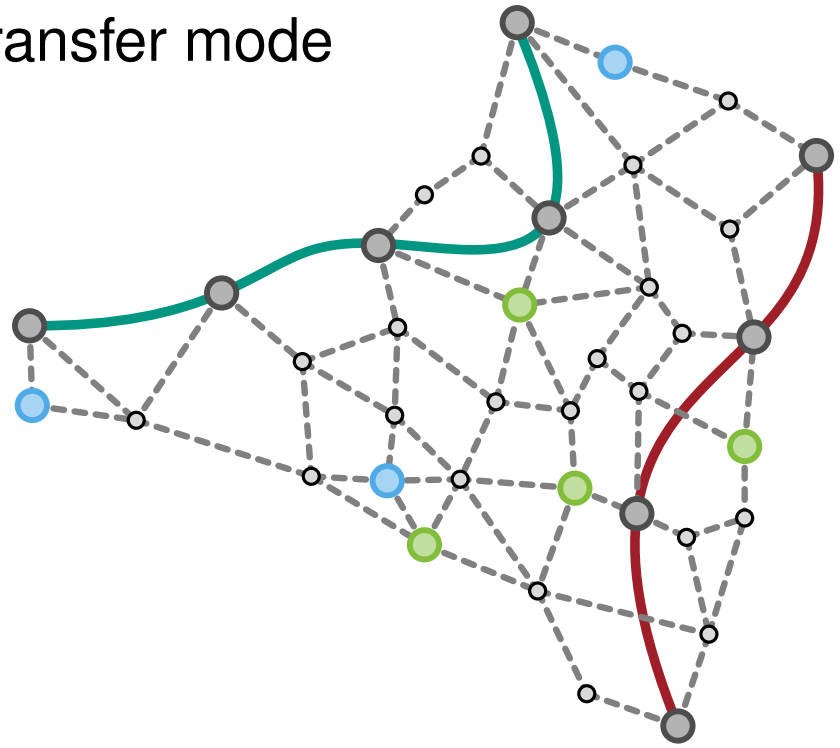
- Check if bike sharing is useful while transferring
- If so, represent the transfer with a single shortcut
- Independent of the number of bikes rented

Speed-up Technique: Integration with ULTRA [Baum et al. 2019]

Karlsruhe Institute of Technology

ULTRA (UnLimited TRAnsfers) overview:

- Speed-up technique for public transit + one additional transfer mode
- Replaces the transfer graph with inter-trip shortcuts



Adaptation for Bike Sharing:

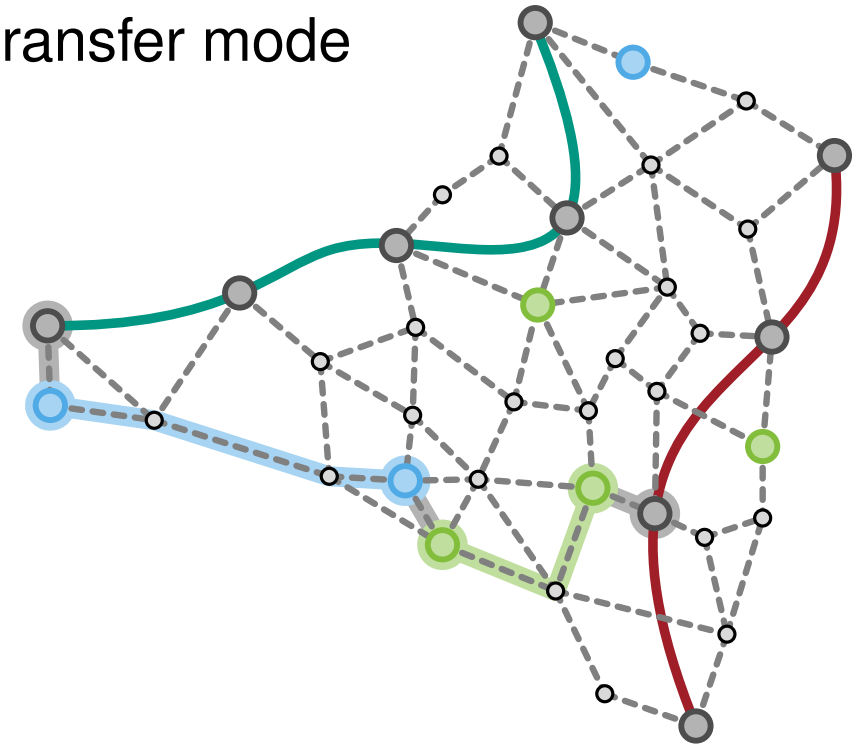
- Check if bike sharing is useful while transferring
- If so, represent the transfer with a single shortcut
- Independent of the number of bikes rented

Speed-up Technique: Integration with ULTRA [Baum et al. 2019]

Karlsruhe Institute of Technology

ULTRA (UnLimited TRAnsfers) overview:

- Speed-up technique for public transit + one additional transfer mode
- Replaces the transfer graph with inter-trip shortcuts



Adaptation for Bike Sharing:

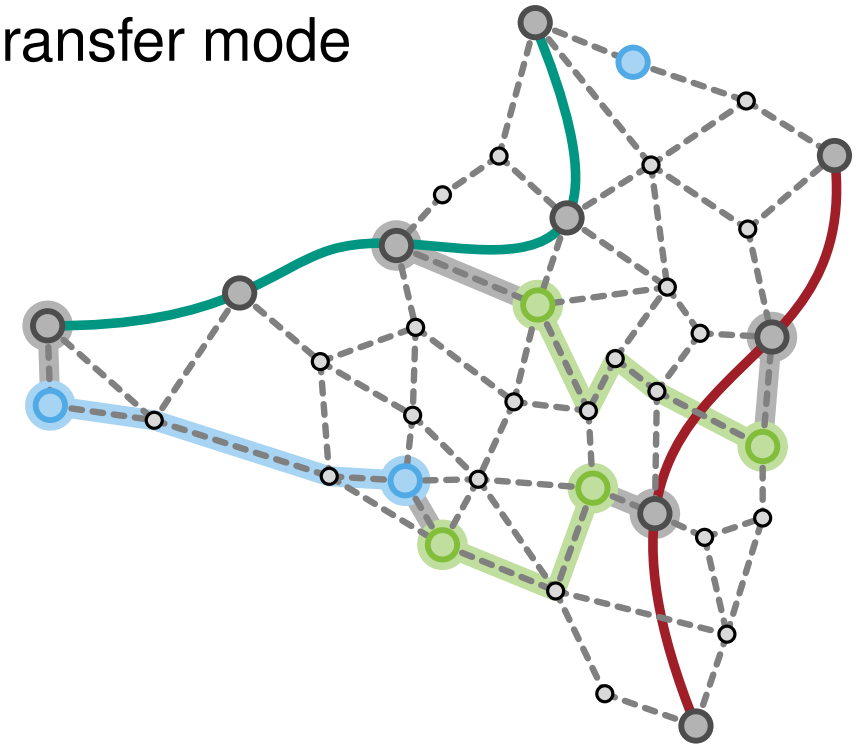
- Check if bike sharing is useful while transferring
- If so, represent the transfer with a single shortcut
- Independent of the number of bikes rented

Speed-up Technique: Integration with ULTRA [Baum et al. 2019]

Karlsruhe Institute of Technology

ULTRA (UnLimited TRAnsfers) overview:

- Speed-up technique for public transit + one additional transfer mode
- Replaces the transfer graph with inter-trip shortcuts



Adaptation for Bike Sharing:

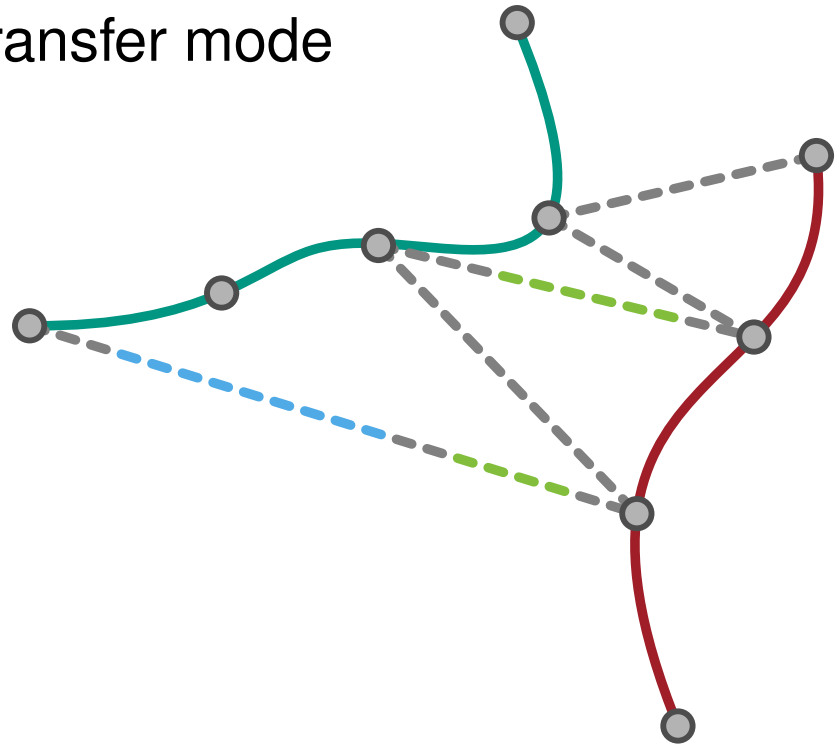
- Check if bike sharing is useful while transferring
- If so, represent the transfer with a single shortcut
- Independent of the number of bikes rented

Speed-up Technique: Integration with ULTRA [Baum et al. 2019]

Karlsruhe Institute of Technology

ULTRA (UnLimited TRAnsfers) overview:

- Speed-up technique for public transit + one additional transfer mode
- Replaces the transfer graph with inter-trip shortcuts



Adaptation for Bike Sharing:

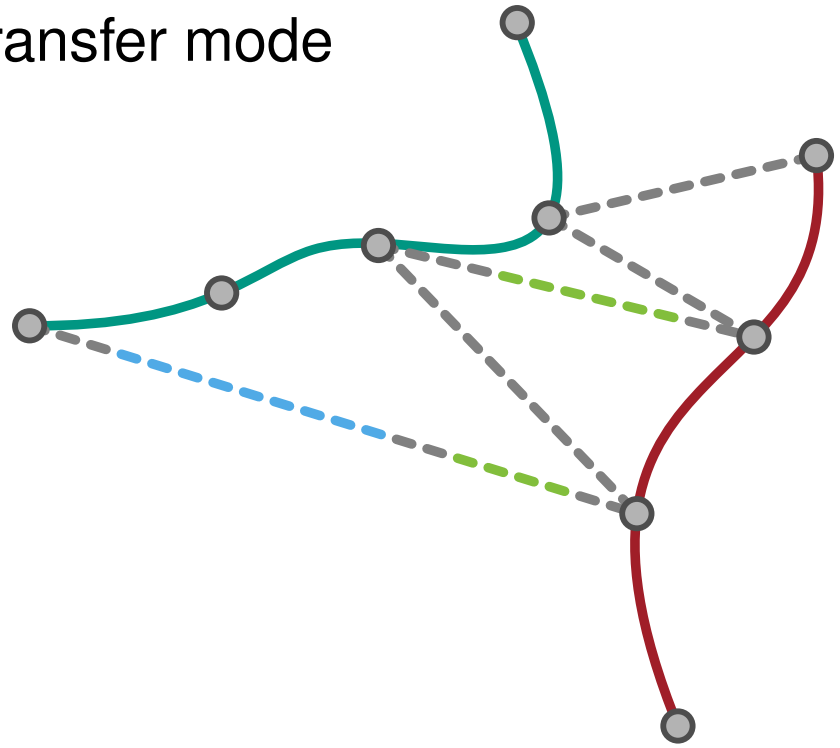
- Check if bike sharing is useful while transferring
- If so, represent the transfer with a single shortcut
- Independent of the number of bikes rented

Speed-up Technique: Integration with ULTRA [Baum et al. 2019]

Karlsruhe Institute of Technology

ULTRA (UnLimited TRAnsfers) overview:

- Speed-up technique for public transit + one additional transfer mode
- Replaces the transfer graph with inter-trip shortcuts



Adaptation for Bike Sharing:

- Check if bike sharing is useful while transferring
- If so, represent the transfer with a single shortcut
- Independent of the number of bikes rented

Solution:

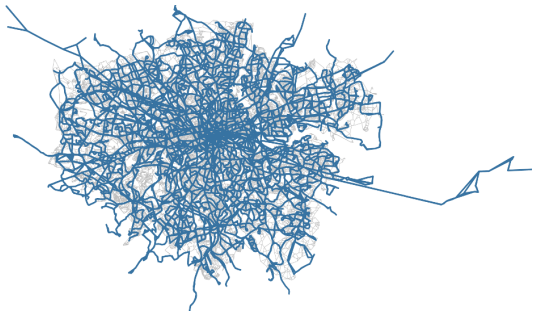
- Perform the ULTRA preprocessing on the operator-expanded network

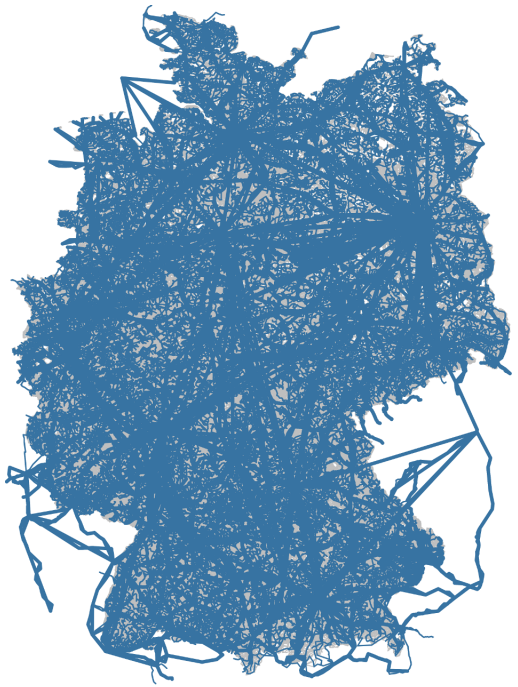
Experimental Evaluation

Instances:

- London, Switzerland, and Germany
- Timetables comprising two days from TfL, GTFS-CH, and DB
- Transfer graphs and bike sharing stations from OpenStreetMap

Network	Stops	Routes	Trips	Vertices	Edges	Stations	Operators
London	20 595	2 107	125 k	183 k	579 k	823	4
Switzerland	25 426	13 934	369 k	604 k	1 847 k	534	11
Germany	244 055	231 089	2 387 k	6 872 k	21 372 k	2 682	22





Experimental Evaluation – Preprocessing

Impact of Operator-Pruning:

- Computation of operator hulls is quite fast
- Leads to significantly smaller operator-expanded networks
- Makes ULTRA on the operator-expanded network feasible

	London		Switzerland		Germany	
	OE	OE-OP	OE	OE-OP	OE	OE-OP
Expanded stops	102 975	31 216	301 500	36 892	5 613 265	411 980
ULTRA shortcuts	1 831 779	521 882	3 389 309	435 514	?	7 873 379
Operator hulls (sequential)	–	3:01:21	–	50:20	–	83:38:15
Operator hulls (parallel 16)	–	15:34	–	4:15	–	8:45:22
Total (CH + OP + ULTRA)	14:15:19	59:33	10:01:54	28:03	≈21 weeks	40:13:48

Experimental Evaluation – Preprocessing

Impact of Operator-Pruning:

- Computation of operator hulls is quite fast
- Leads to significantly smaller operator-expanded networks
- Makes ULTRA on the operator-expanded network feasible

	London		Switzerland		Germany	
	OE	OE-OP	OE	OE-OP	OE	OE-OP
Expanded stops	102 975	31 216	301 500	36 892	5 613 265	411 980
ULTRA shortcuts	1 831 779	521 882	3 389 309	435 514	?	7 873 379
Operator hulls (sequential)	–	3:01:21	–	50:20	–	83:38:15
Operator hulls (parallel 16)	–	15:34	–	4:15	–	8:45:22
Total (CH + OP + ULTRA)	14:15:19	59:33	10:01:54	28:03	≈21 weeks	40:13:48

Experimental Evaluation – Preprocessing

Impact of Operator-Pruning:

- Computation of operator hulls is quite fast
- Leads to significantly smaller operator-expanded networks
- Makes ULTRA on the operator-expanded network feasible

	London		Switzerland		Germany	
	OE	OE-OP	OE	OE-OP	OE	OE-OP
Expanded stops	102 975	31 216	301 500	36 892	5 613 265	411 980
ULTRA shortcuts	1 831 779	521 882	3 389 309	435 514	?	7 873 379
Operator hulls (sequential)	–	3:01:21	–	50:20	–	83:38:15
Operator hulls (parallel 16)	–	15:34	–	4:15	–	8:45:22
Total (CH + OP + ULTRA)	14:15:19	59:33	10:01:54	28:03	≈21 weeks	40:13:48

Experimental Evaluation – Preprocessing

Impact of Operator-Pruning:

- Computation of operator hulls is quite fast
- Leads to significantly smaller operator-expanded networks
- Makes ULTRA on the operator-expanded network feasible

	London		Switzerland		Germany	
	OE	OE-OP	OE	OE-OP	OE	OE-OP
Expanded stops	102 975	31 216	301 500	36 892	5 613 265	411 980
ULTRA shortcuts	1 831 779	521 882	3 389 309	435 514	?	7 873 379
Operator hulls (sequential)	–	3:01:21	–	50:20	–	83:38:15
Operator hulls (parallel 16)	–	15:34	–	4:15	–	8:45:22
Total (CH + OP + ULTRA)	14:15:19	59:33	10:01:54	28:03	≈21 weeks	40:13:48

Experimental Evaluation – Preprocessing

Impact of Operator-Pruning:

- Computation of operator hulls is quite fast
- Leads to significantly smaller operator-expanded networks
- Makes ULTRA on the operator-expanded network feasible

	London		Switzerland		Germany	
	OE	OE-OP	OE	OE-OP	OE	OE-OP
Expanded stops	102 975	31 216	301 500	36 892	5 613 265	411 980
ULTRA shortcuts	1 831 779	521 882	3 389 309	435 514	?	7 873 379
Operator hulls (sequential)	–	3:01:21	–	50:20	–	83:38:15
Operator hulls (parallel 16)	–	15:34	–	4:15	–	8:45:22
Total (CH + OP + ULTRA)	14:15:19	59:33	10:01:54	28:03	≈21 weeks	40:13:48

Experimental Evaluation – Query

Average Running Times:

- Combining ULTRA, OE, and OP yields the fastest algorithm

Network	Algorithm	Preprocessing	Query			
		Time [h:m:s]	Rounds	Vertices	Routes	Time [ms]
Switzerland	MCR-OD	0:56	9.55	840 k	171 k	286.8
	MCR-OE	1:02	9.55	782 k	171 k	345.0
	MCR-OE-OP	5:40	8.35	144 k	43 k	52.8
	ULTRA-OE-OP	28:03	8.48	29 k	44 k	21.0
Germany	MCR-OD	13:19	11.99	17 421 k	2 888 k	9 830.1
	MCR-OE	15:21	11.99	16 120 k	2 889 k	10 599.3
	MCR-OE-OP	9:05:48	10.24	2 091 k	679 k	1 322.7
	ULTRA-OE-OP	40:13:48	10.38	301 k	688 k	649.3

Experimental Evaluation – Query

Average Running Times:

- Combining ULTRA, OE, and OP yields the fastest algorithm

Network	Algorithm	Preprocessing	Query			
		Time [h:m:s]	Rounds	Vertices	Routes	Time [ms]
Switzerland	MCR-OD	0:56	9.55	840 k	171 k	286.8
	MCR-OE	1:02	9.55	782 k	171 k	345.0
	MCR-OE-OP	5:40	8.35	144 k	43 k	52.8
	ULTRA-OE-OP	28:03	8.48	29 k	44 k	21.0
Germany	MCR-OD	13:19	11.99	17 421 k	2 888 k	9 830.1
	MCR-OE	15:21	11.99	16 120 k	2 889 k	10 599.3
	MCR-OE-OP	9:05:48	10.24	2 091 k	679 k	1 322.7
	ULTRA-OE-OP	40:13:48	10.38	301 k	688 k	649.3

Experimental Evaluation – Query

Average Running Times:

- Combining ULTRA, OE, and OP yields the fastest algorithm

Network	Algorithm	Preprocessing	Query			
		Time [h:m:s]	Rounds	Vertices	Routes	Time [ms]
Switzerland	MCR-OD	0:56	9.55	840 k	171 k	286.8
	MCR-OE	1:02	9.55	782 k	171 k	345.0
	MCR-OE-OP	5:40	8.35	144 k	43 k	52.8
	ULTRA-OE-OP	28:03	8.48	29 k	44 k	21.0
Germany	MCR-OD	13:19	11.99	17 421 k	2 888 k	9 830.1
	MCR-OE	15:21	11.99	16 120 k	2 889 k	10 599.3
	MCR-OE-OP	9:05:48	10.24	2 091 k	679 k	1 322.7
	ULTRA-OE-OP	40:13:48	10.38	301 k	688 k	649.3

Experimental Evaluation – Query

Average Running Times:

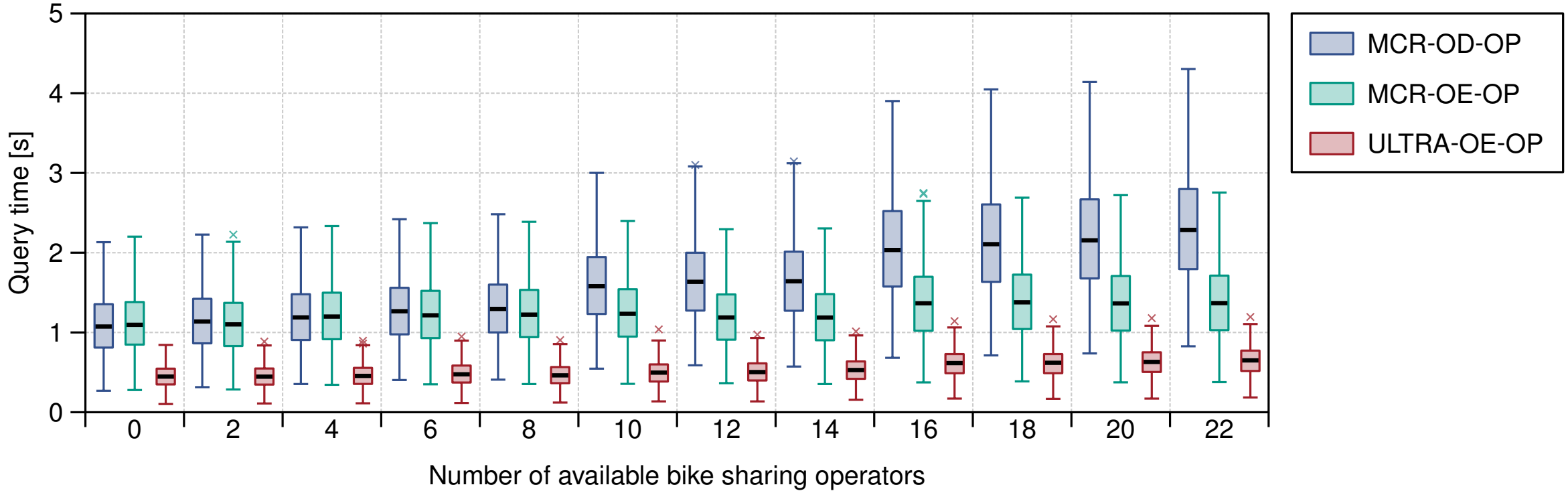
- Combining ULTRA, OE, and OP yields the fastest algorithm

Network	Algorithm	Preprocessing	Query			
		Time [h:m:s]	Rounds	Vertices	Routes	Time [ms]
Switzerland	MCR-OD	0:56	9.55	840 k	171 k	286.8
	MCR-OE	1:02	9.55	782 k	171 k	345.0
	MCR-OE-OP	5:40	8.35	144 k	43 k	52.8
	ULTRA-OE-OP	28:03	8.48	29 k	44 k	21.0
Germany	MCR-OD	13:19	11.99	17 421 k	2 888 k	9 830.1
	MCR-OE	15:21	11.99	16 120 k	2 889 k	10 599.3
	MCR-OE-OP	9:05:48	10.24	2 091 k	679 k	1 322.7
	ULTRA-OE-OP	40:13:48	10.38	301 k	688 k	649.3

Experimental Evaluation – Query

Running Times Depending on Number of Operators:

- Operator-expanded model benefits more from operator-pruning
- ULTRA reduces query time significantly



Conclusion

Our Contribution:

- We introduced two new approaches for modeling bike sharing:
  - Operator-Dependent
  - Operator-Expanded
- We presented a novel speed-up technique: Operator-Pruning
- Overall, we are more than 10 times faster than the base-line