Karlsruhe Institute of Technology

Integrating ULTRA and Trip-Based Routing

ATMOS · September 7, 2020

Jonas Sauer, Dorothea Wagner, and Tobias Zündorf

INSTITUTE OF THEORETICAL INFORMATICS · ALGORITHMICS GROUP

Multi-Modal Journey Planning

Goals:

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

Integrating ULTRA and Trip-Based Routing
Jonas Sauer, Dorothea Wagner, and Tobias Zündorf

Institute of Theoretical Informatics
Algorithmics Group

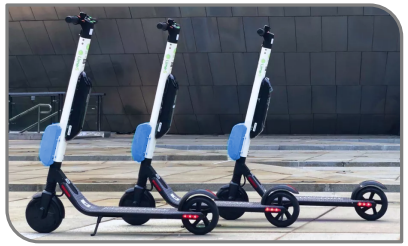
Multi-Modal Journey Planning

Goals:

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

But also:

- Allow secondary transfer mode
- Non-schedule based (walking, bike, e-scooter, ...)



Integrating ULTRA and Trip-Based Routing
Jonas Sauer, Dorothea Wagner, and Tobias Zündorf

Institute of Theoretical Informatics
Algorithmics Group

Problem Statement

Karlsruhe Institute of Technology

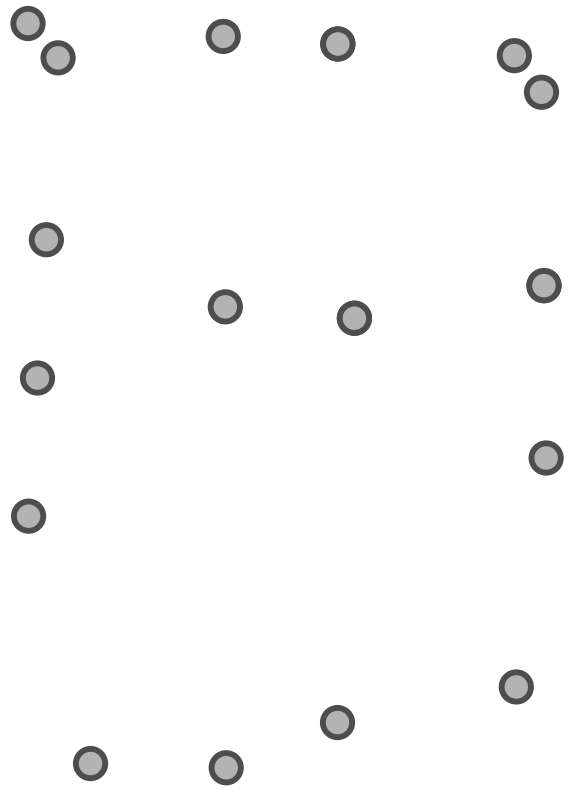
Given:

- Public transit network (timetable)

Problem Statement

Given:

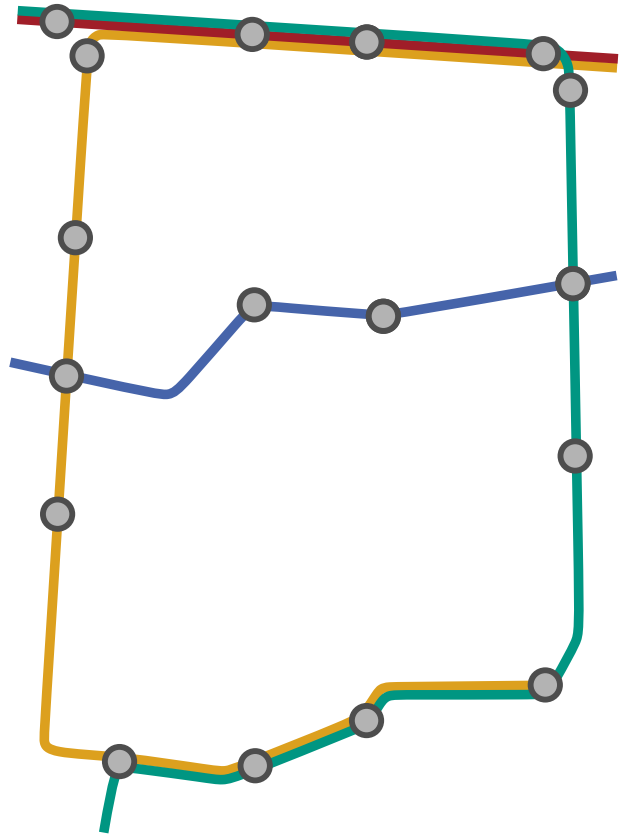
- Public transit network (timetable)
  - Stops (bus stops, stations)



Problem Statement

Given:

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



Problem Statement

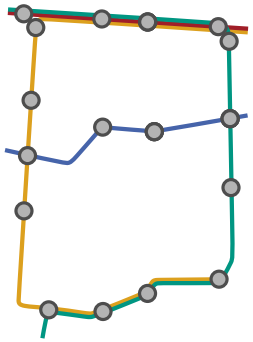
Given:

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

Problem Statement

Given:

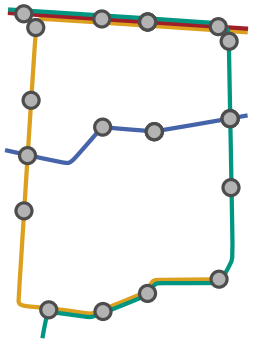
- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)
- Transfer graph (non schedule-based)

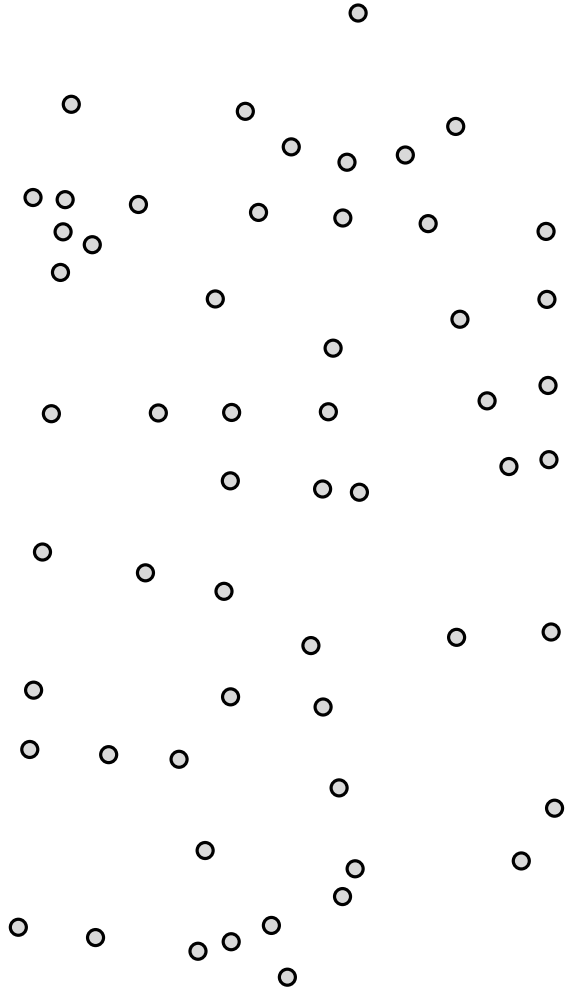


Problem Statement

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)

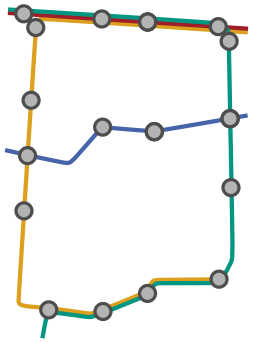


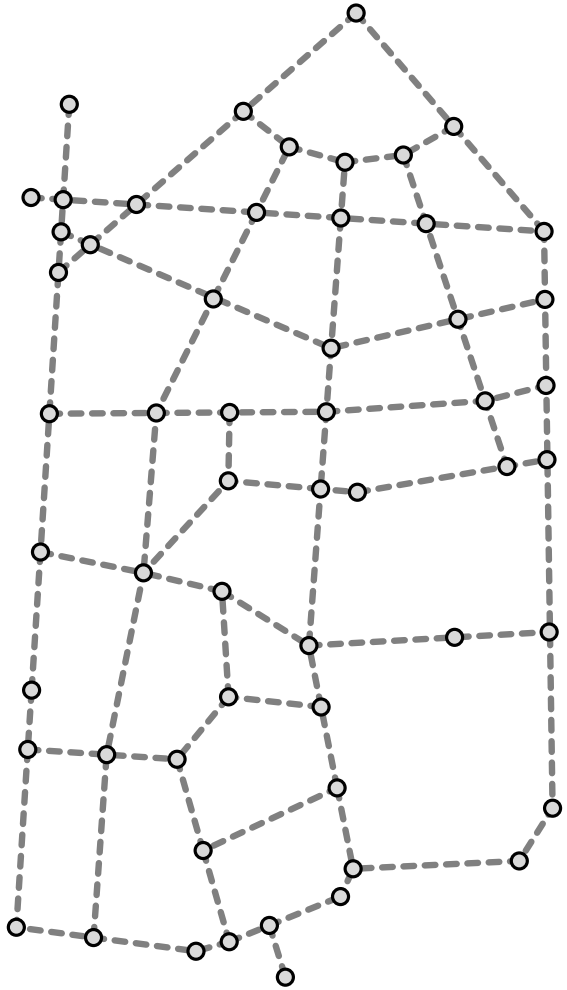


Problem Statement

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)

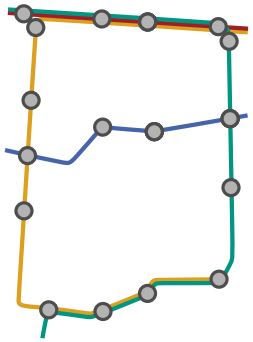


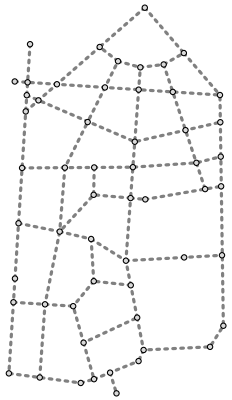


Problem Statement

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)

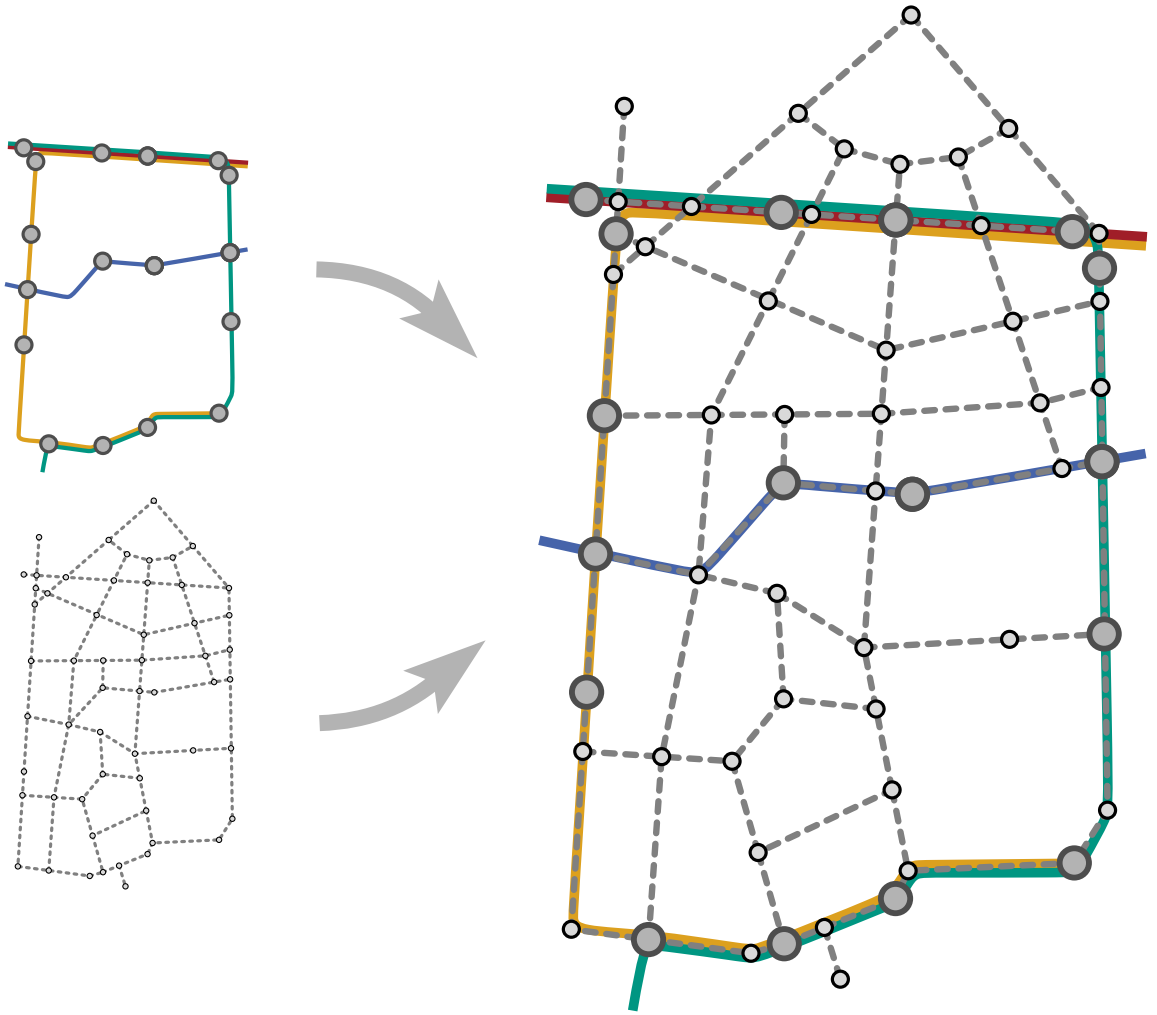




Problem Statement

Given:

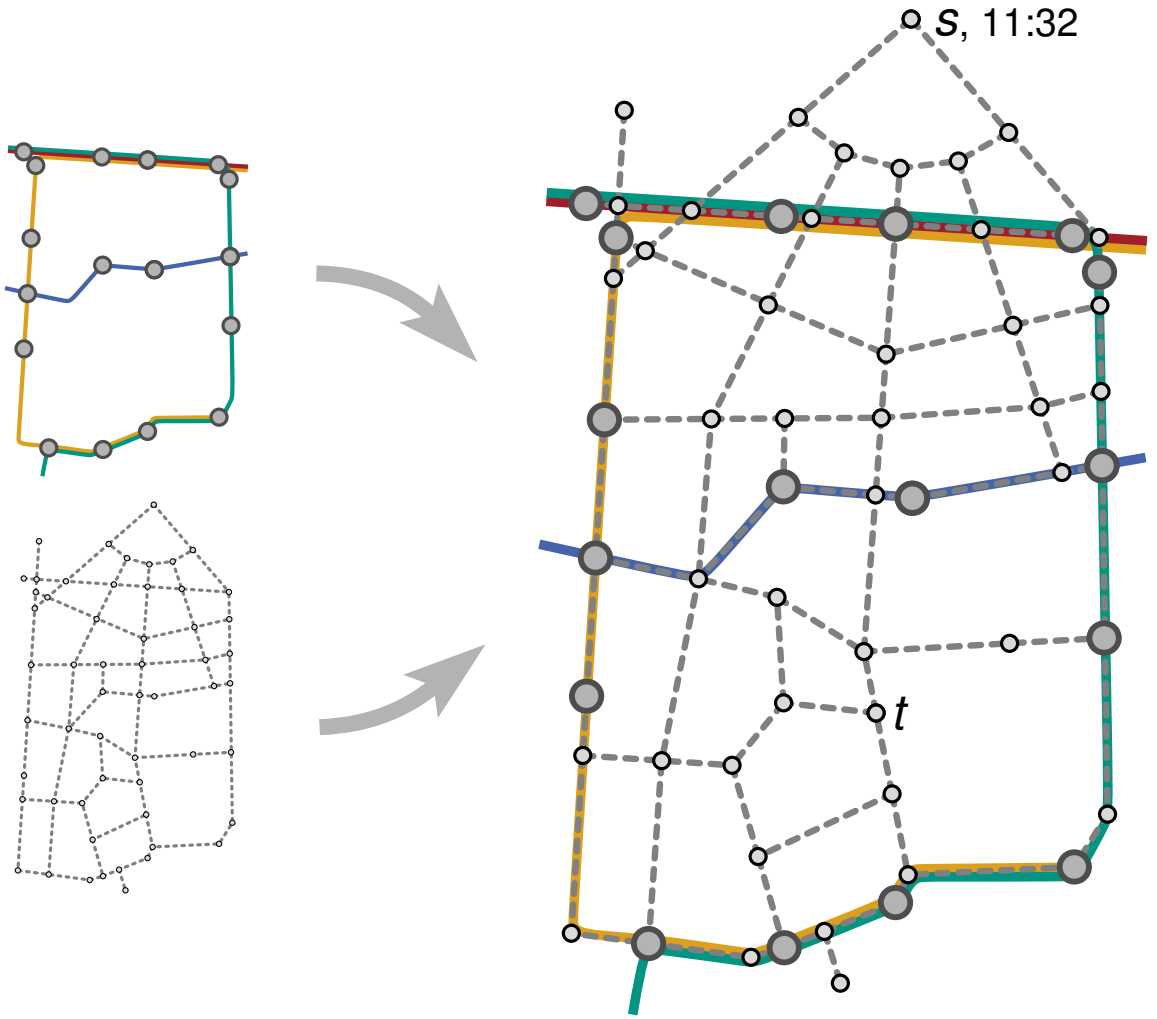
- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)
- Transfer graph (non schedule-based)
  - Vertices (crossings, places)
  - Edges (roads, paths)



Problem Statement

Given:

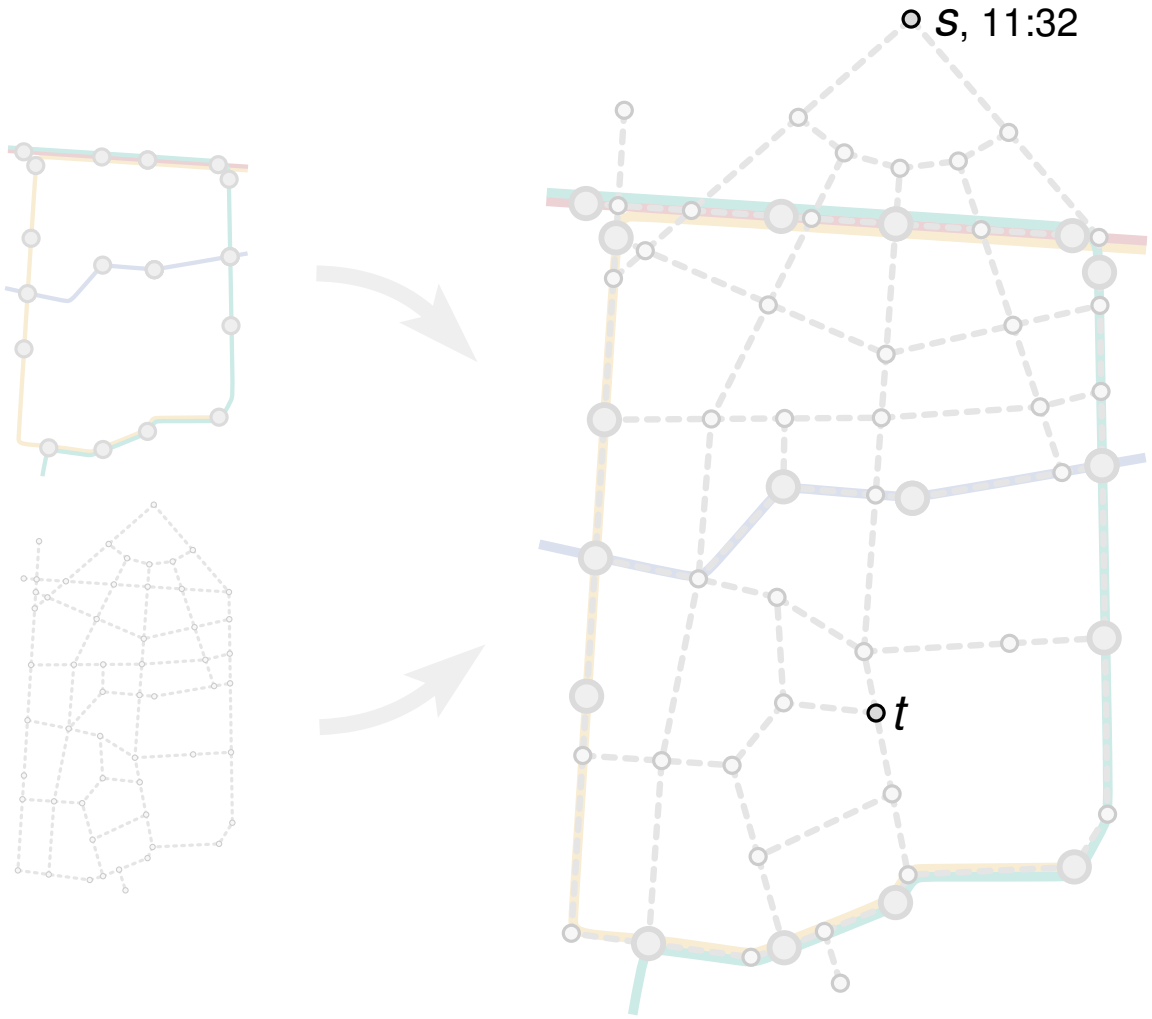
- Public transit network (timetable)
  - Stops (bus stops, stations)
  - Routes (bus lines, train lines)
  - Trips (schedule of a vehicle)
- Transfer graph (non schedule-based)
  - Vertices (crossings, places)
  - Edges (roads, paths)
- Source *s*, target *t*, and a departure time



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)
- Source s, target t, and a departure time



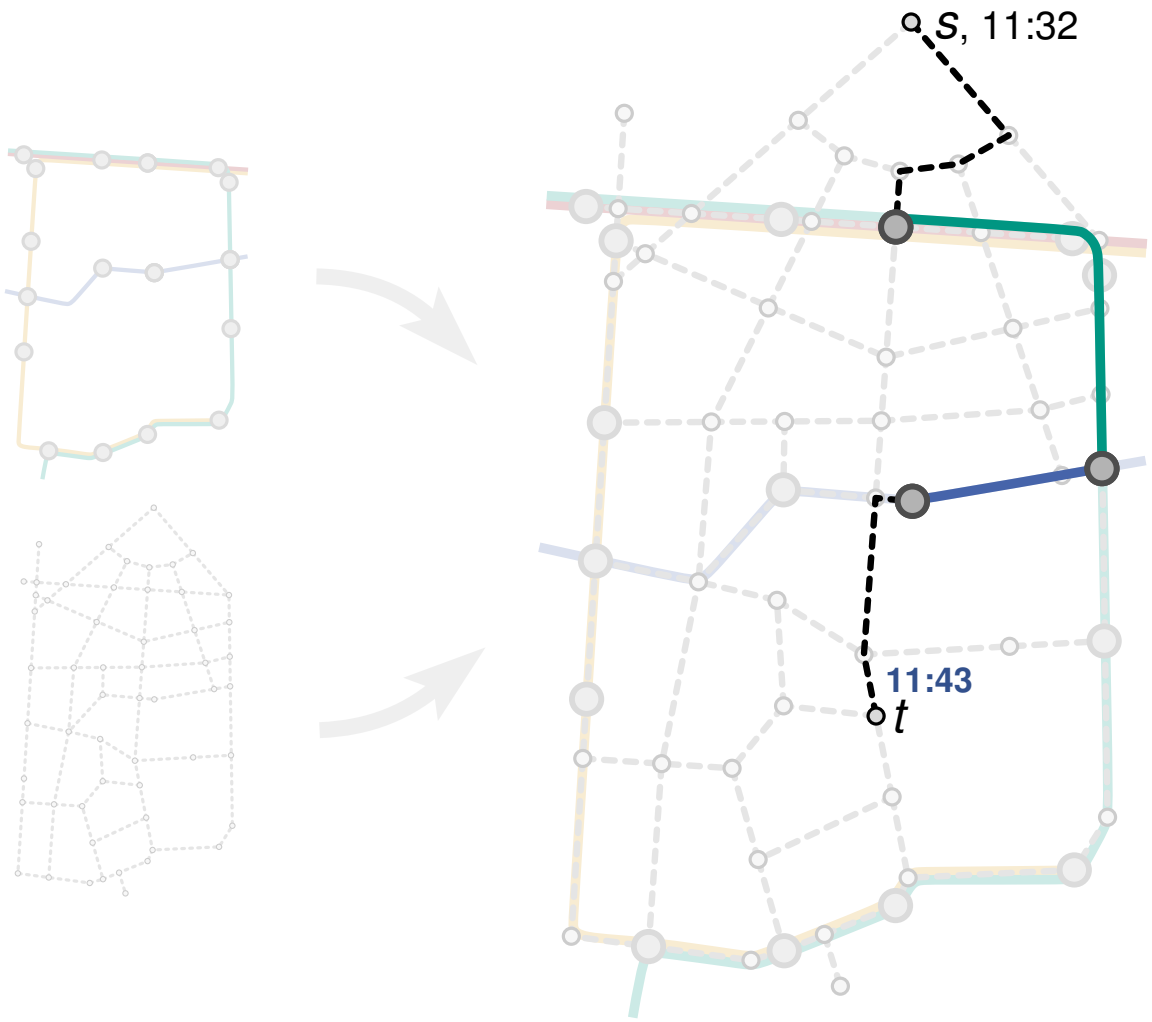
Objective:

- 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)
- Source *s*, target *t*, and a departure time



Objective:

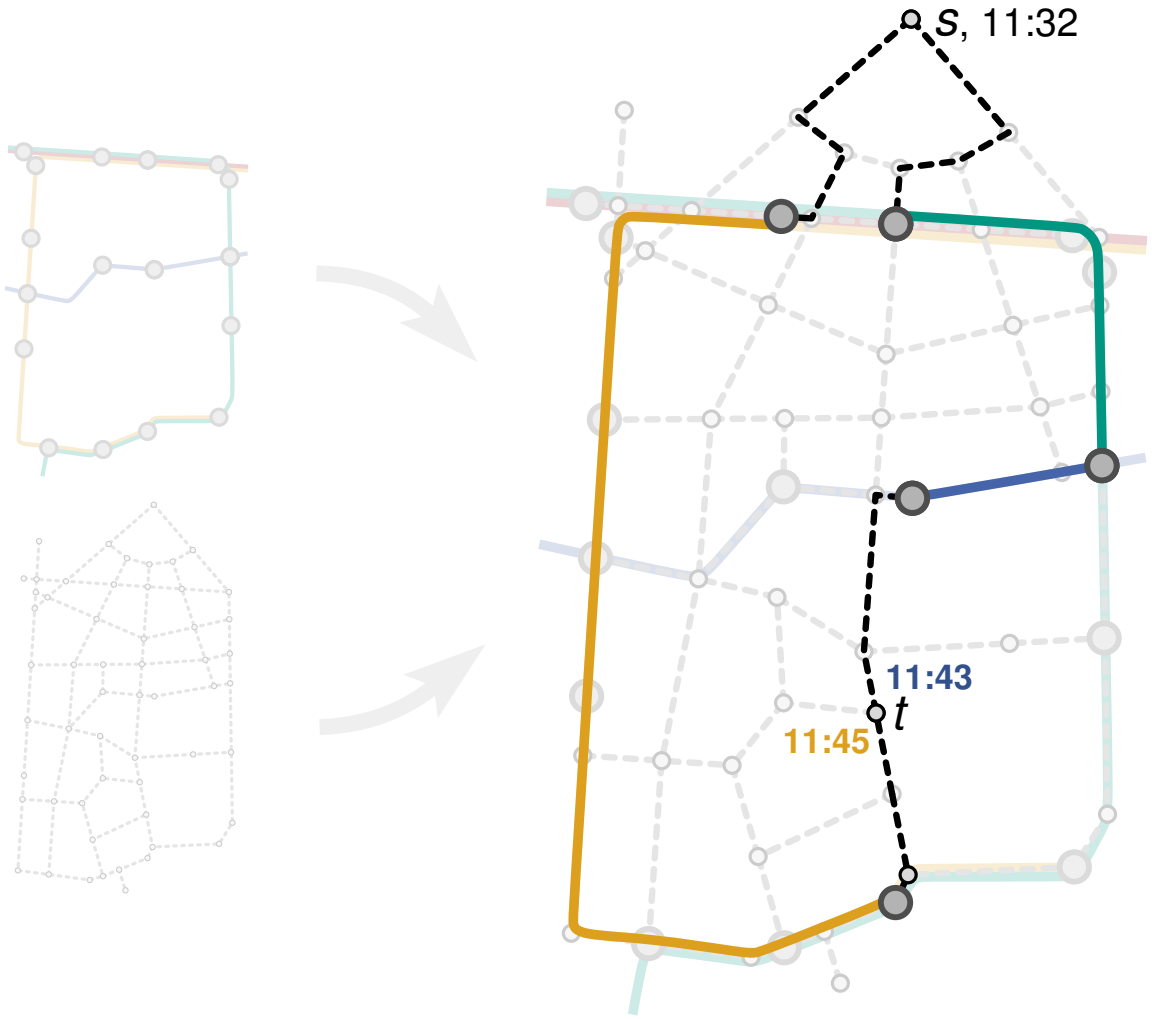
- Find all Pareto-optimal journeys w.r.t. **arrival time** and **number of trips**

Institute of Theoretical Informatics
Algorithmics Group

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)
- Source *s*, target *t*, and a departure time

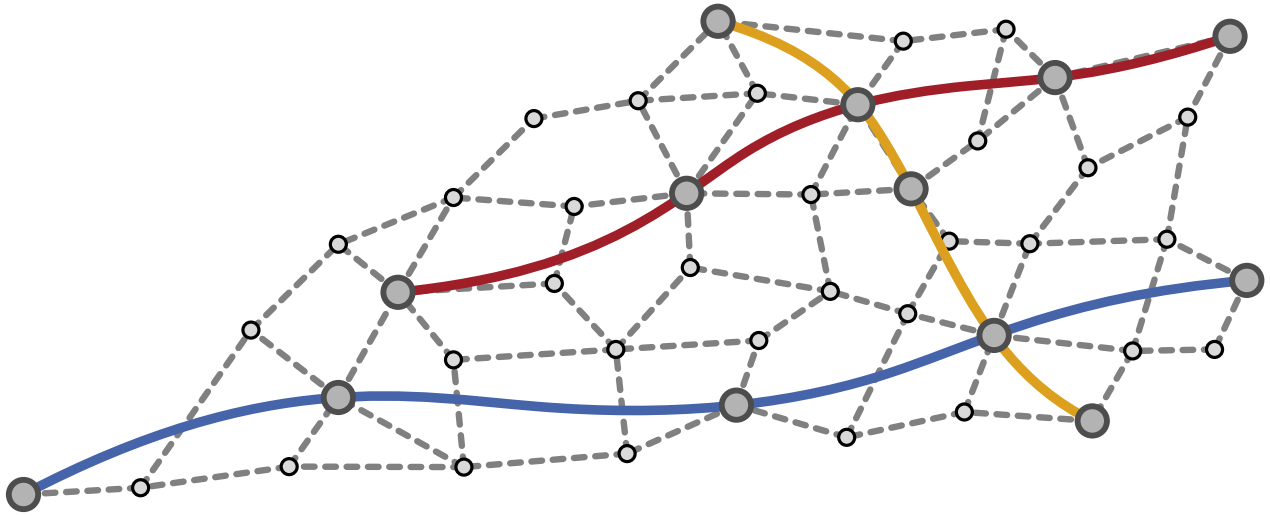


Objective:

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

Related Work

ULTRA: (multimodal journey planning)

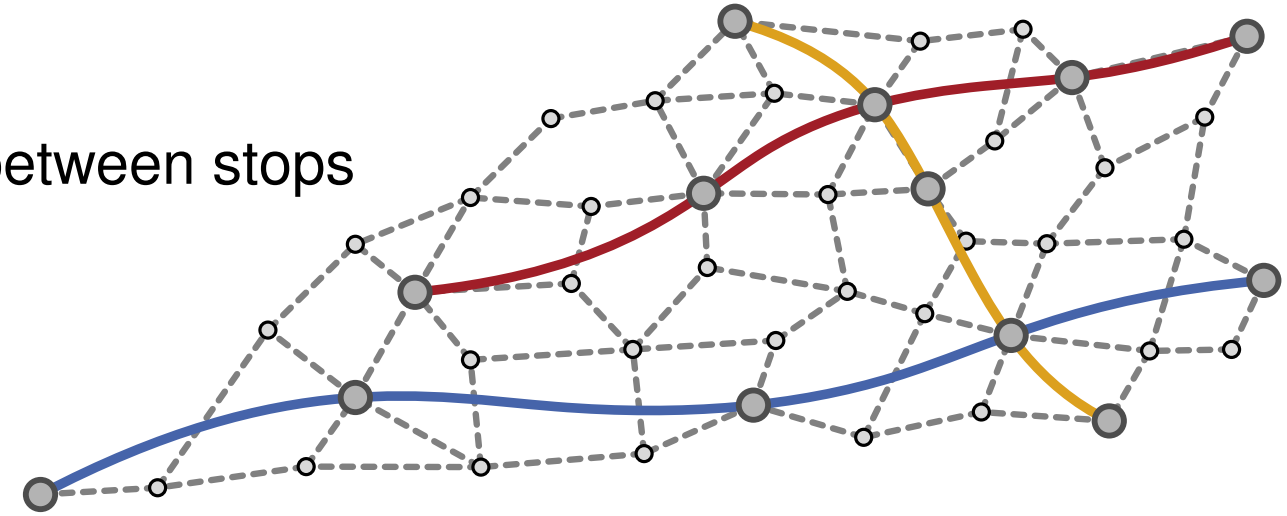


Institute of Theoretical Informatics
Algorithmics Group

Related Work

ULTRA: (multimodal journey planning)

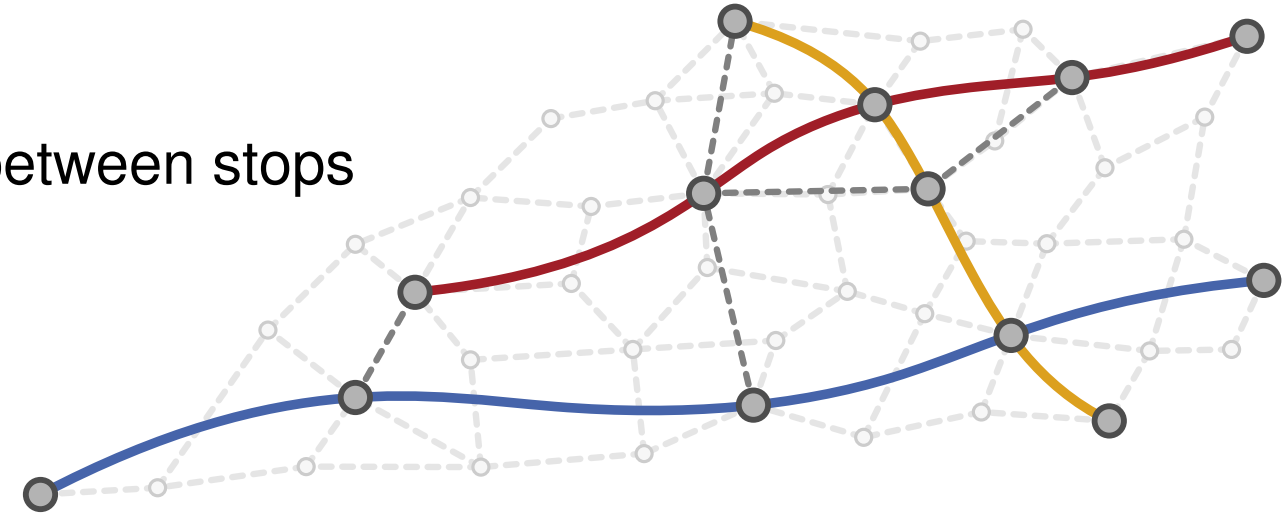
- Complex transfer graph $\rightsquigarrow$ Shortcuts between stops



Related Work

ULTRA: (multimodal journey planning)

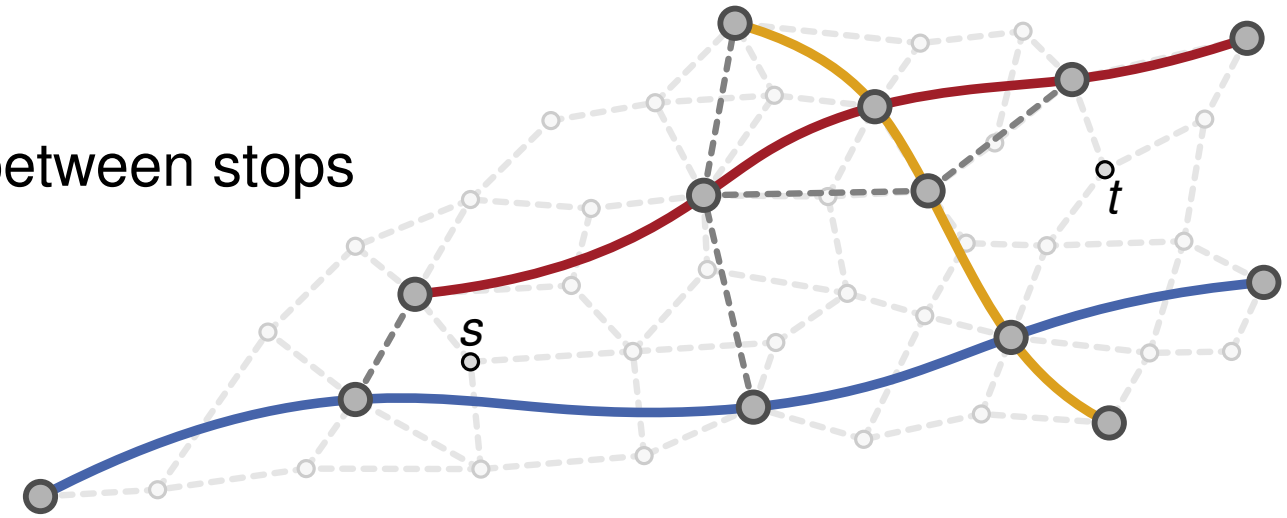
- Complex transfer graph $\rightsquigarrow$ Shortcuts between stops



Related Work

ULTRA: (multimodal journey planning)

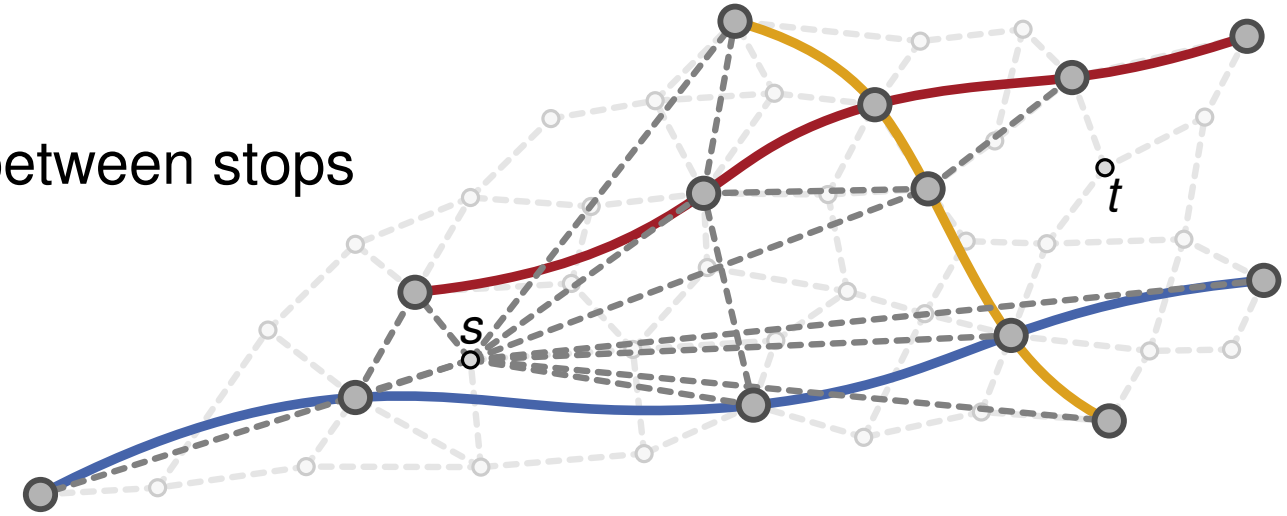
- Complex transfer graph $\rightsquigarrow$ Shortcuts between stops
- Add transfers from source / to target



Related Work

ULTRA: (multimodal journey planning)

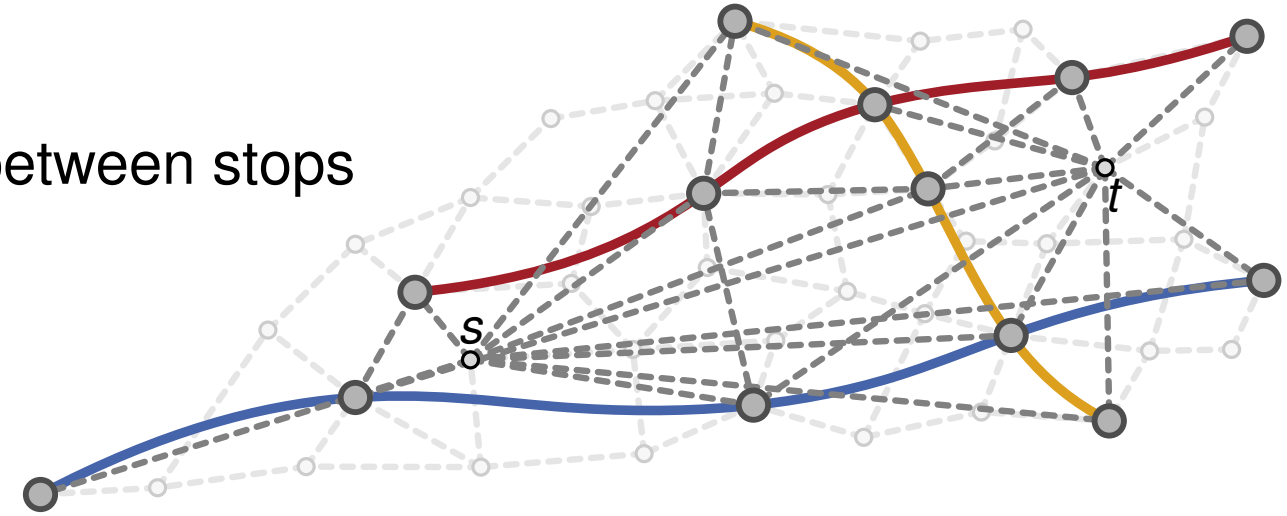
- Complex transfer graph $\rightsquigarrow$ Shortcuts between stops
- Add transfers from source / to target



Related Work

ULTRA: (multimodal journey planning)

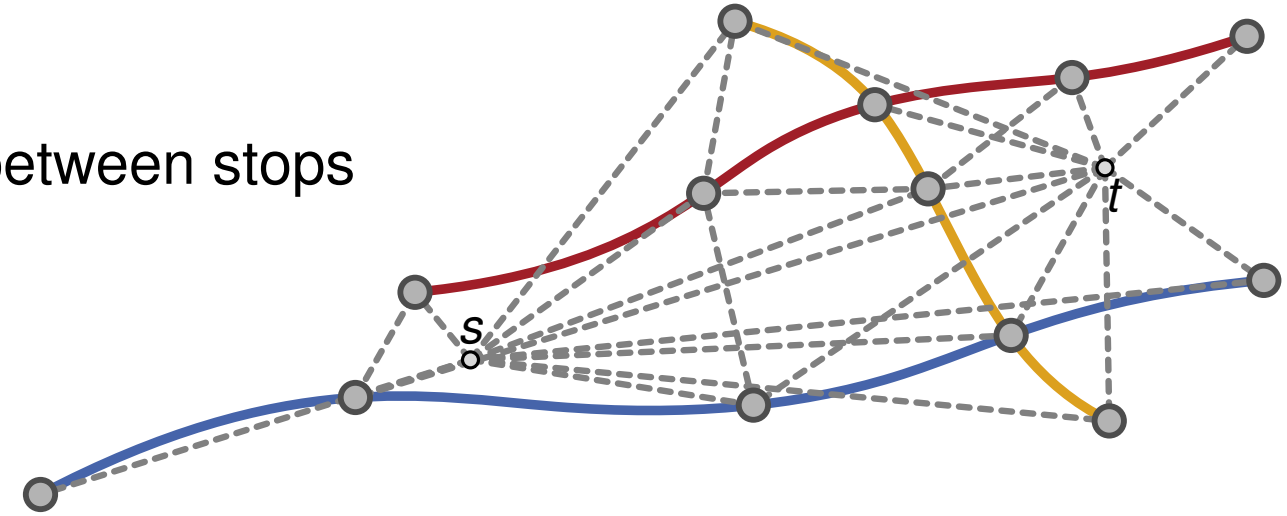
- Complex transfer graph $\rightsquigarrow$ Shortcuts between stops
- Add transfers from source / to target



Related Work

ULTRA: (multimodal journey planning)

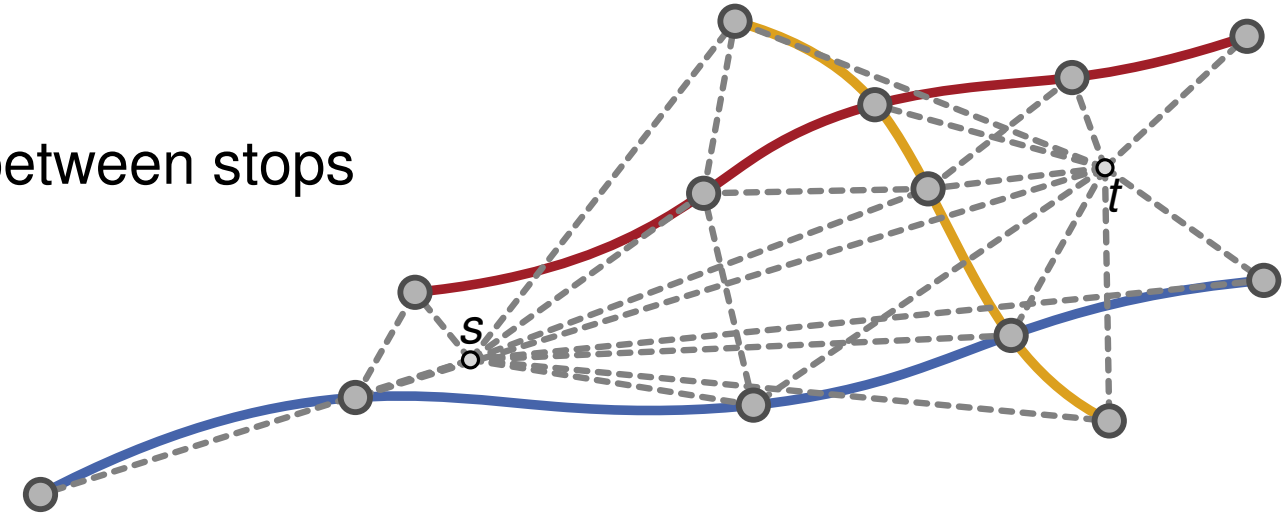
- Complex transfer graph $\rightsquigarrow$ Shortcuts between stops
- Add transfers from source / to target
- Use public transit algorithm



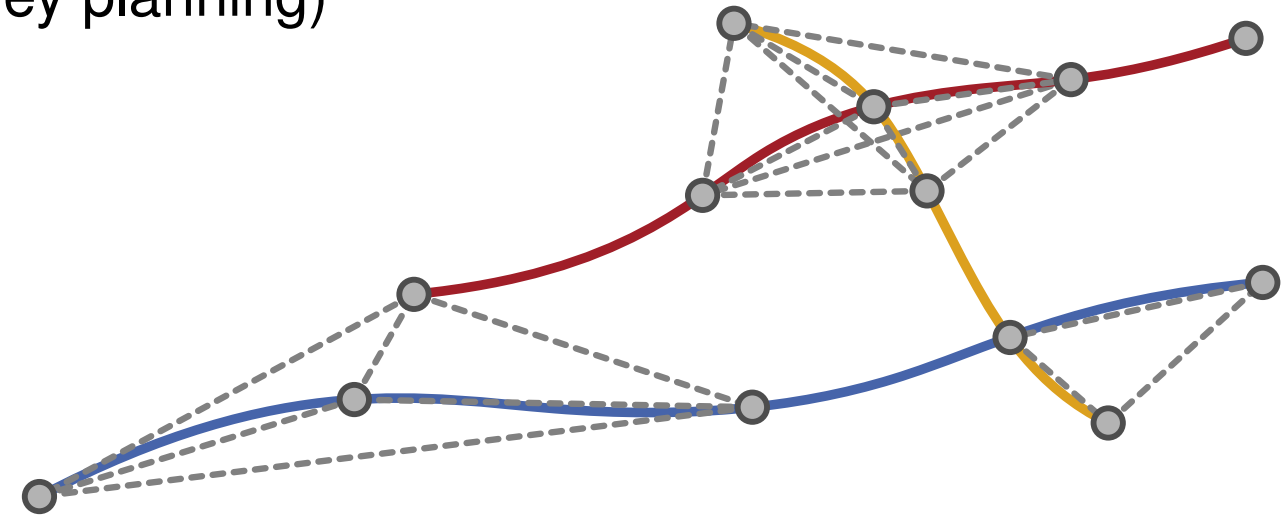
Related Work

ULTRA: (multimodal journey planning)

- Complex transfer graph $\rightsquigarrow$ Shortcuts between stops
- Add transfers from source / to target
- Use public transit algorithm



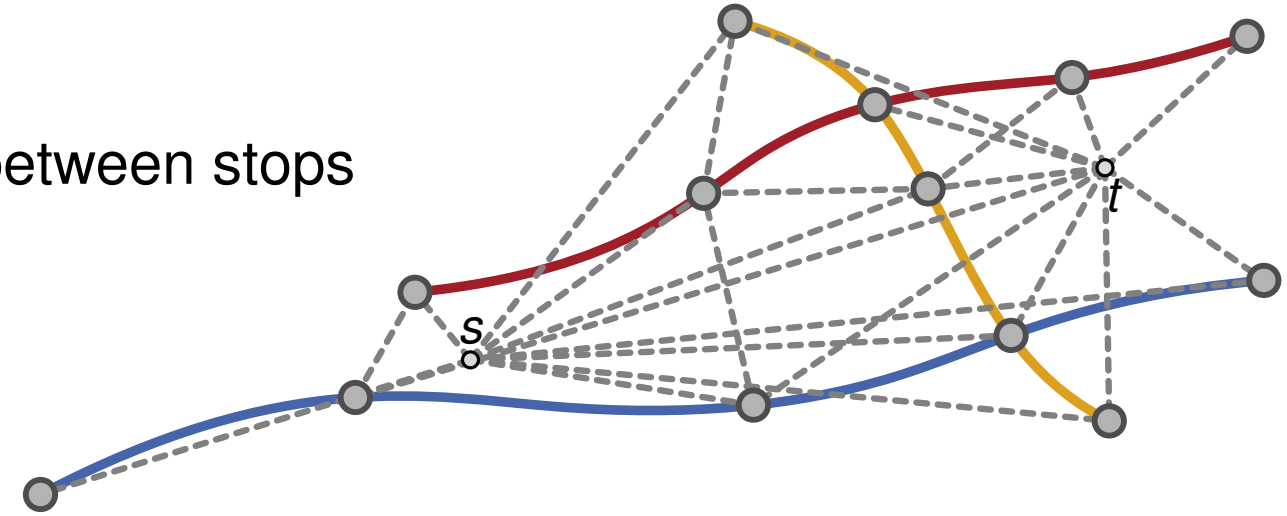
Trip-Based Routing: (public transit journey planning)



Related Work

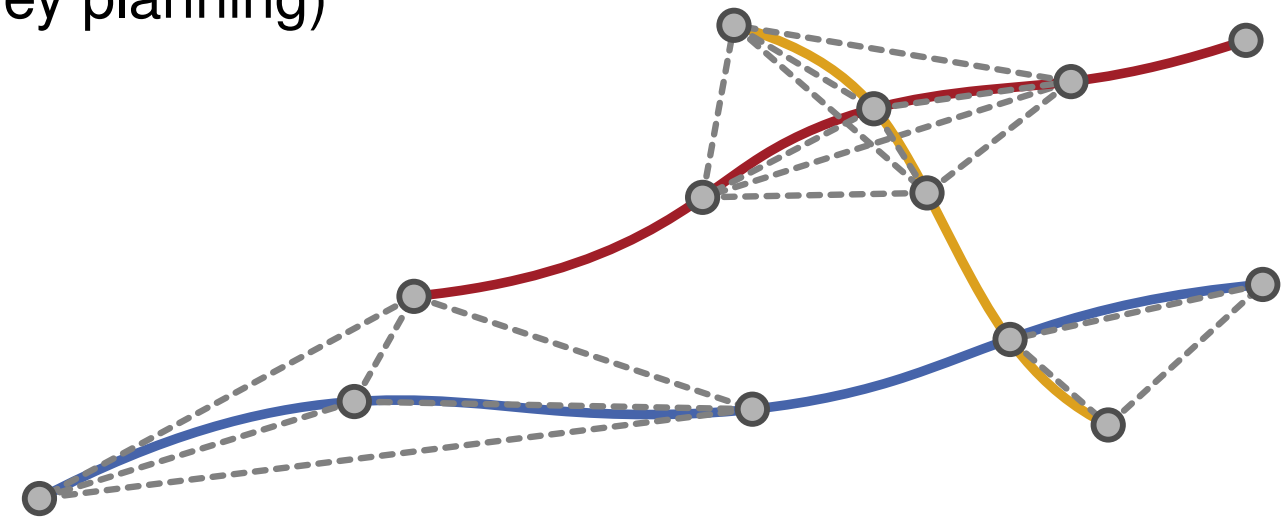
ULTRA: (multimodal journey planning)

- Complex transfer graph $\rightsquigarrow$ Shortcuts between stops
- Add transfers from source / to target
- Use public transit algorithm



Trip-Based Routing: (public transit journey planning)

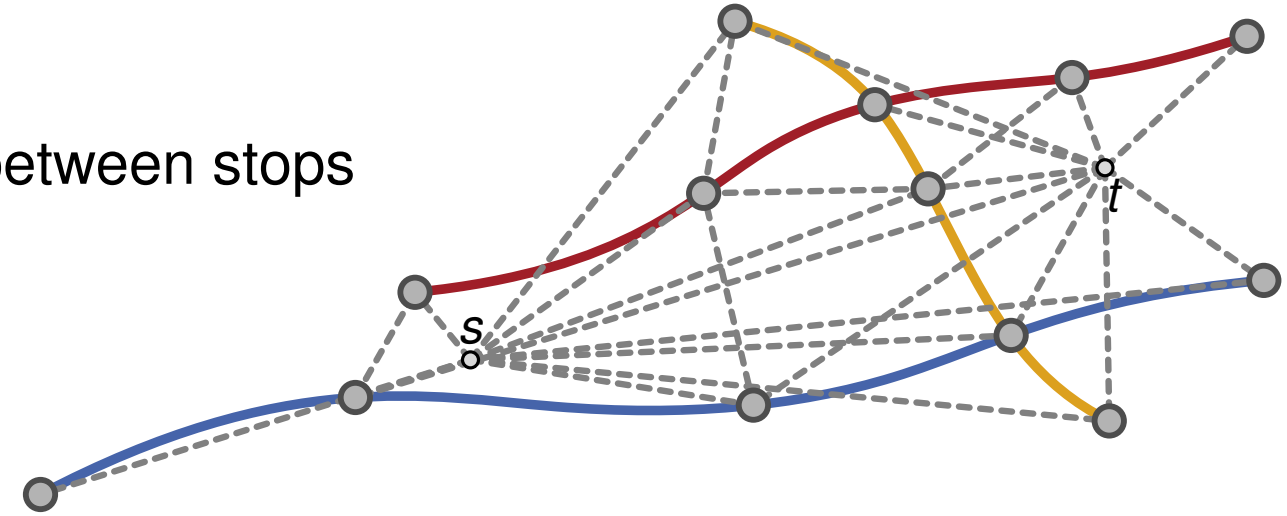
- Operates on the trips of the network



Related Work

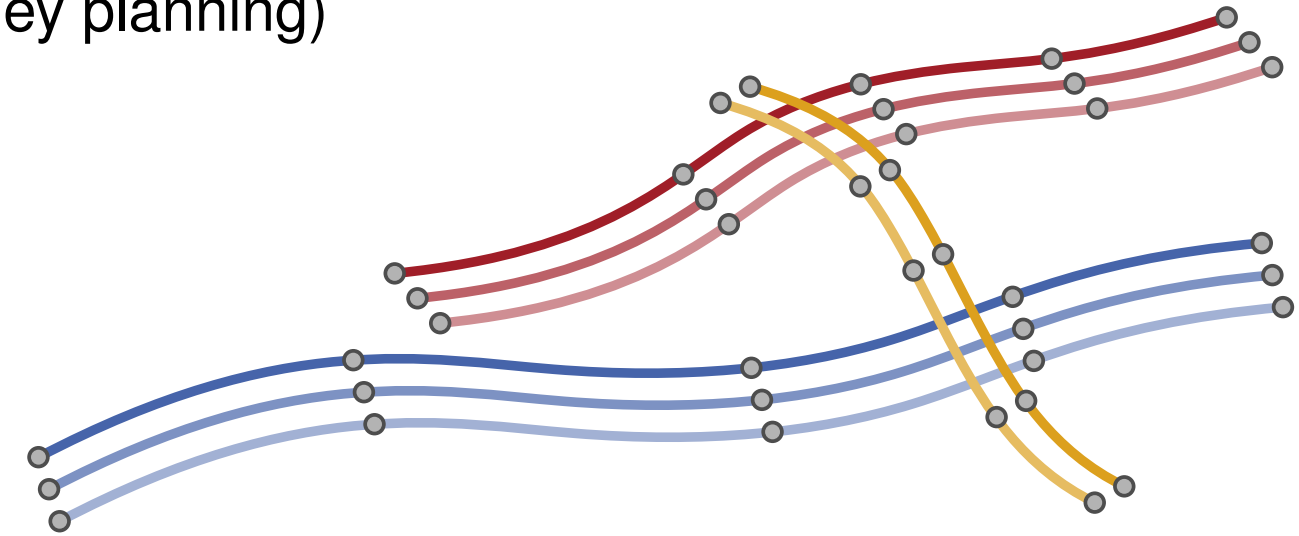
ULTRA: (multimodal journey planning)

- Complex transfer graph $\rightsquigarrow$ Shortcuts between stops
- Add transfers from source / to target
- Use public transit algorithm



Trip-Based Routing: (public transit journey planning)

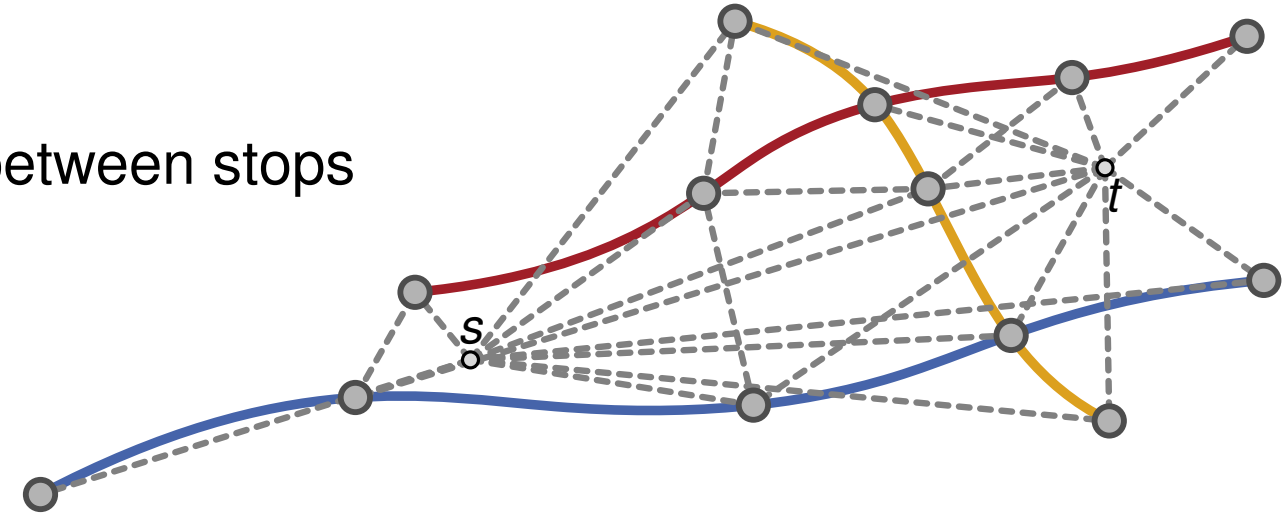
- Operates on the trips of the network



Related Work

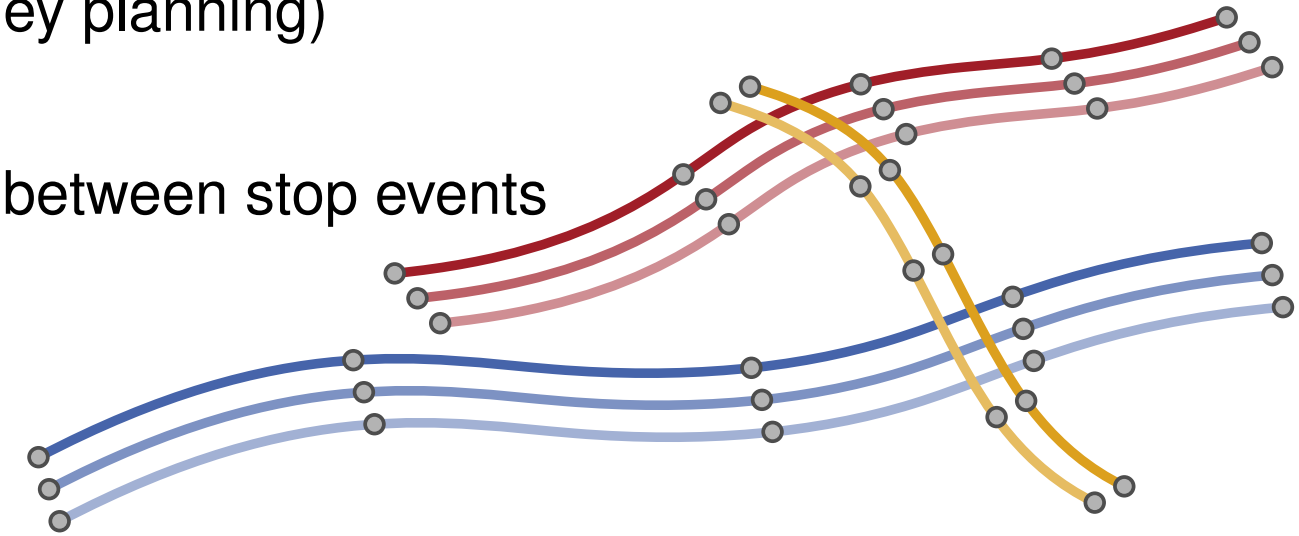
ULTRA: (multimodal journey planning)

- Complex transfer graph ⇝ Shortcuts between stops
- Add transfers from source / to target
- Use public transit algorithm



Trip-Based Routing: (public transit journey planning)

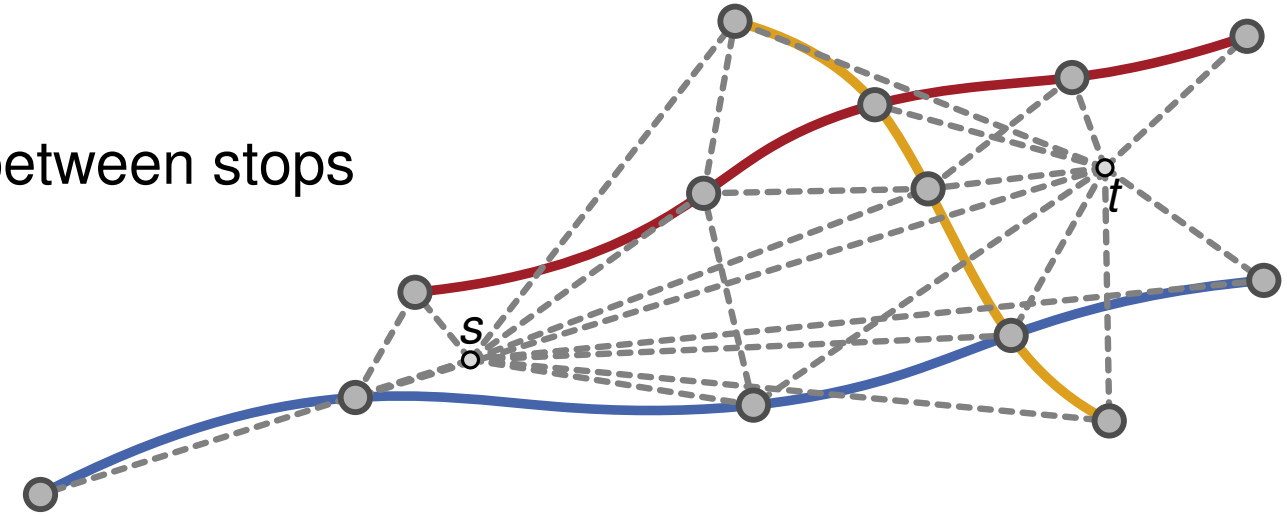
- Operates on the trips of the network
- Transitive transfer graph ⇝ Shortcuts between stop events



Related Work

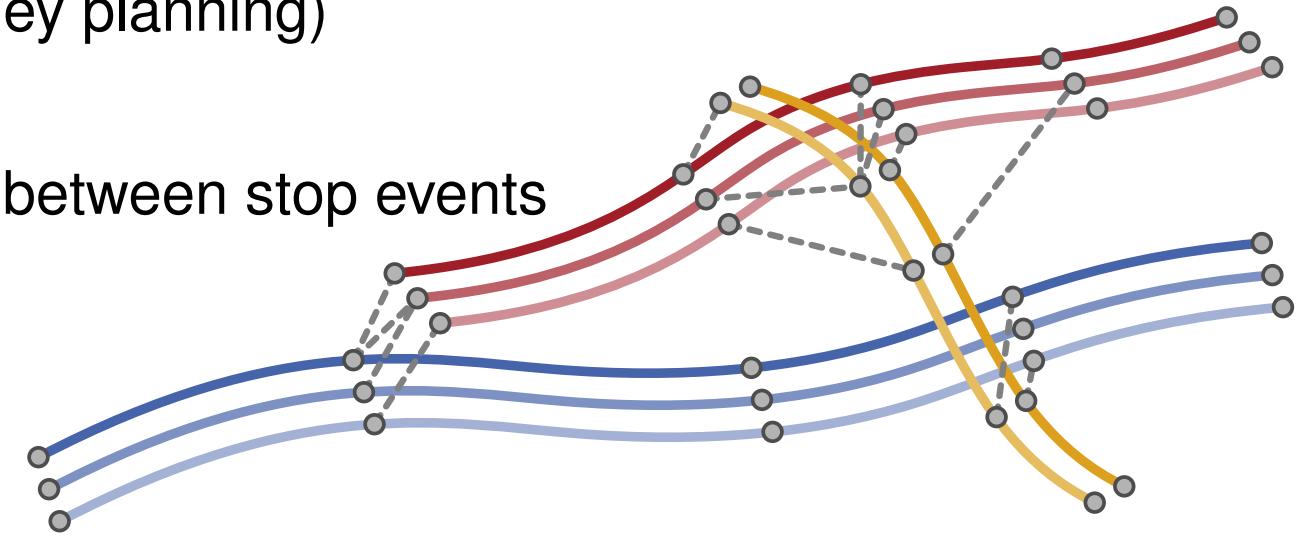
ULTRA: (multimodal journey planning)

- Complex transfer graph $\rightsquigarrow$ Shortcuts between stops
- Add transfers from source / to target
- Use public transit algorithm



Trip-Based Routing: (public transit journey planning)

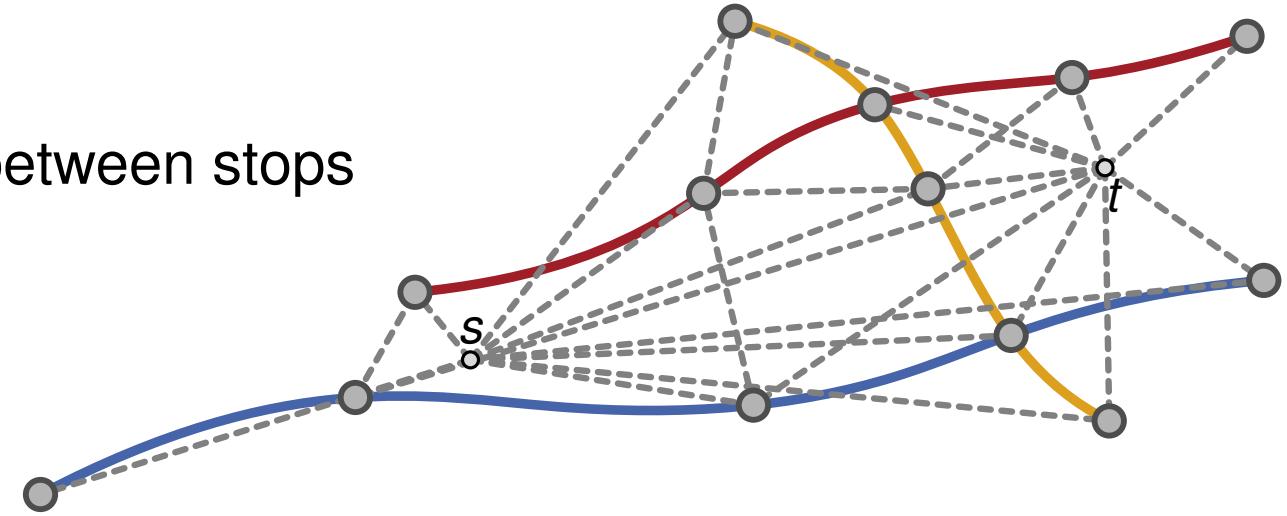
- Operates on the trips of the network
- Transitive transfer graph $\rightsquigarrow$ Shortcuts between stop events



Related Work

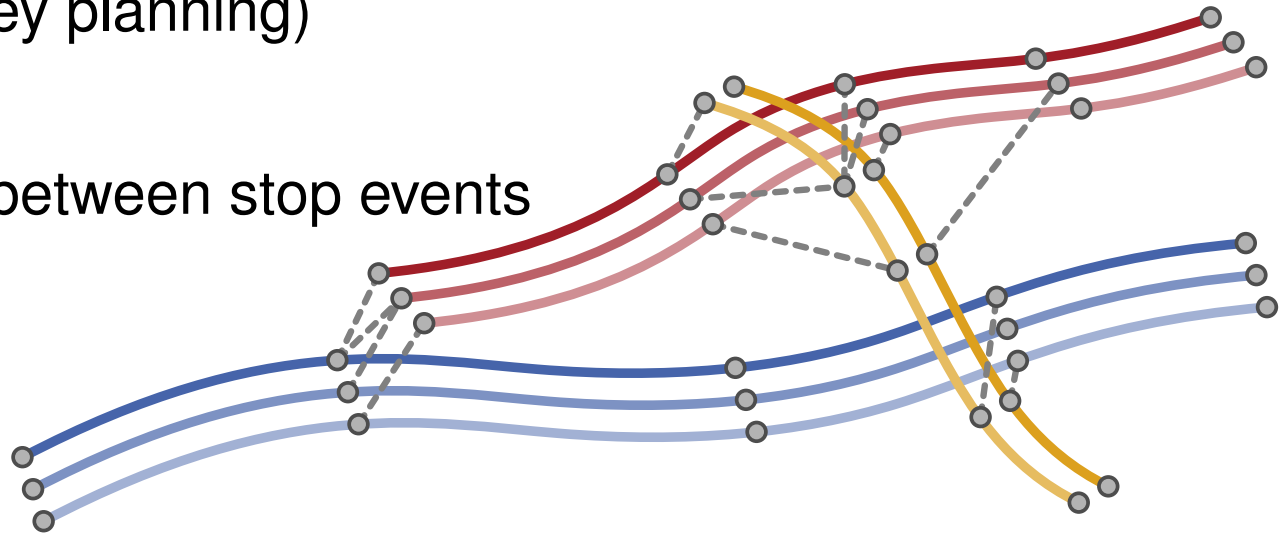
ULTRA: (multimodal journey planning)

- Complex transfer graph $\rightsquigarrow$ Shortcuts between stops
- Add transfers from source / to target
- Use public transit algorithm



Trip-Based Routing: (public transit journey planning)

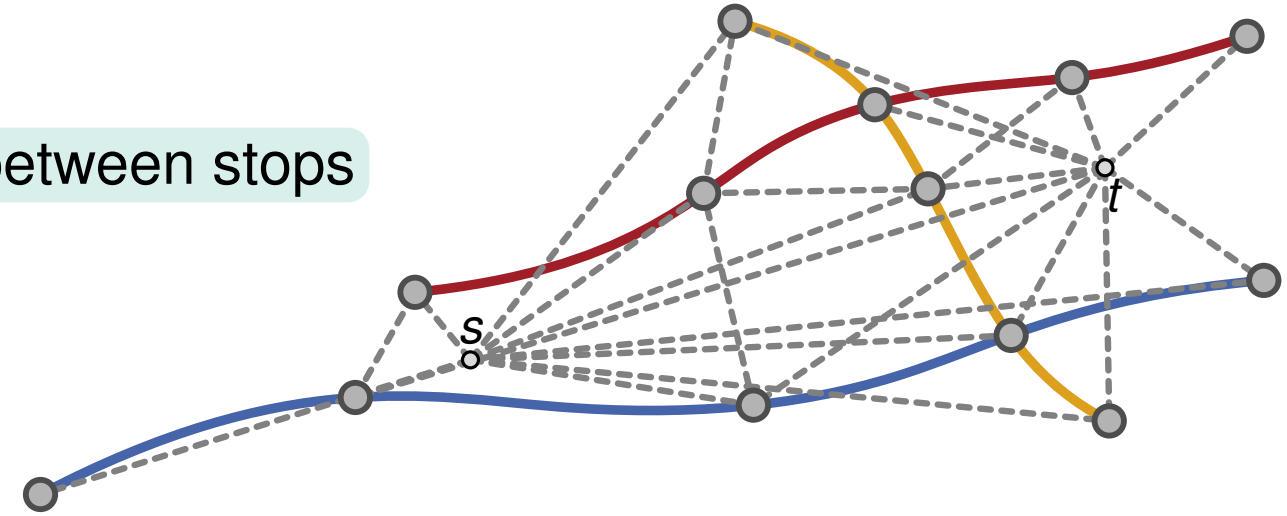
- Operates on the trips of the network
- Transitive transfer graph $\rightsquigarrow$ Shortcuts between stop events
- BFS-like search on the resulting graph



Related Work

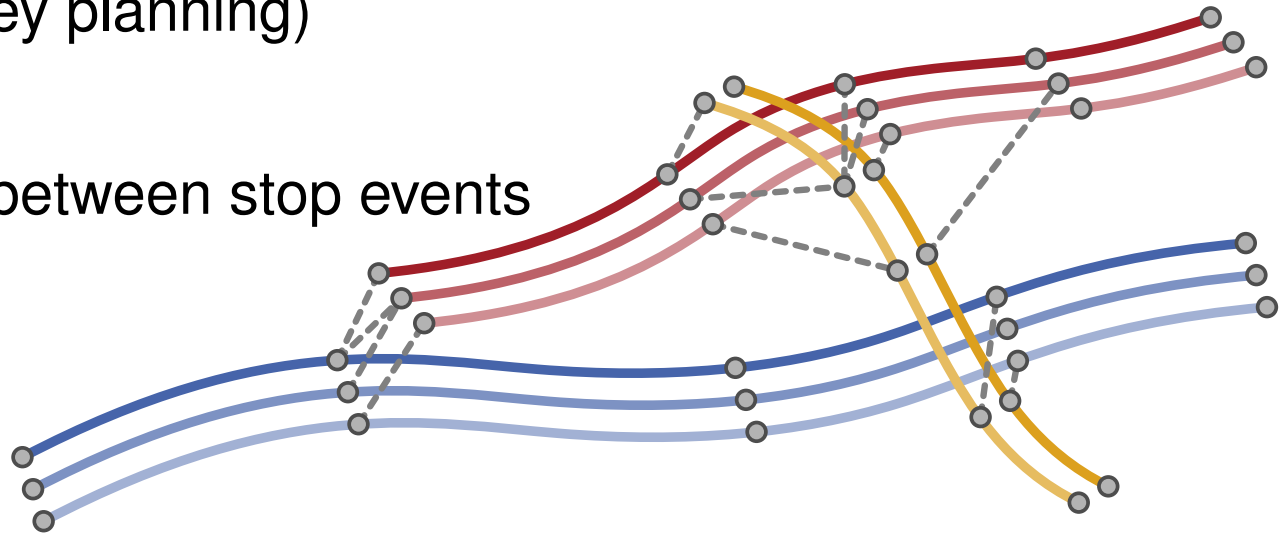
ULTRA: (multimodal journey planning)

- Complex transfer graph $\rightsquigarrow$ Shortcuts between stops
- Add transfers from source / to target
- Use public transit algorithm



Trip-Based Routing: (public transit journey planning)

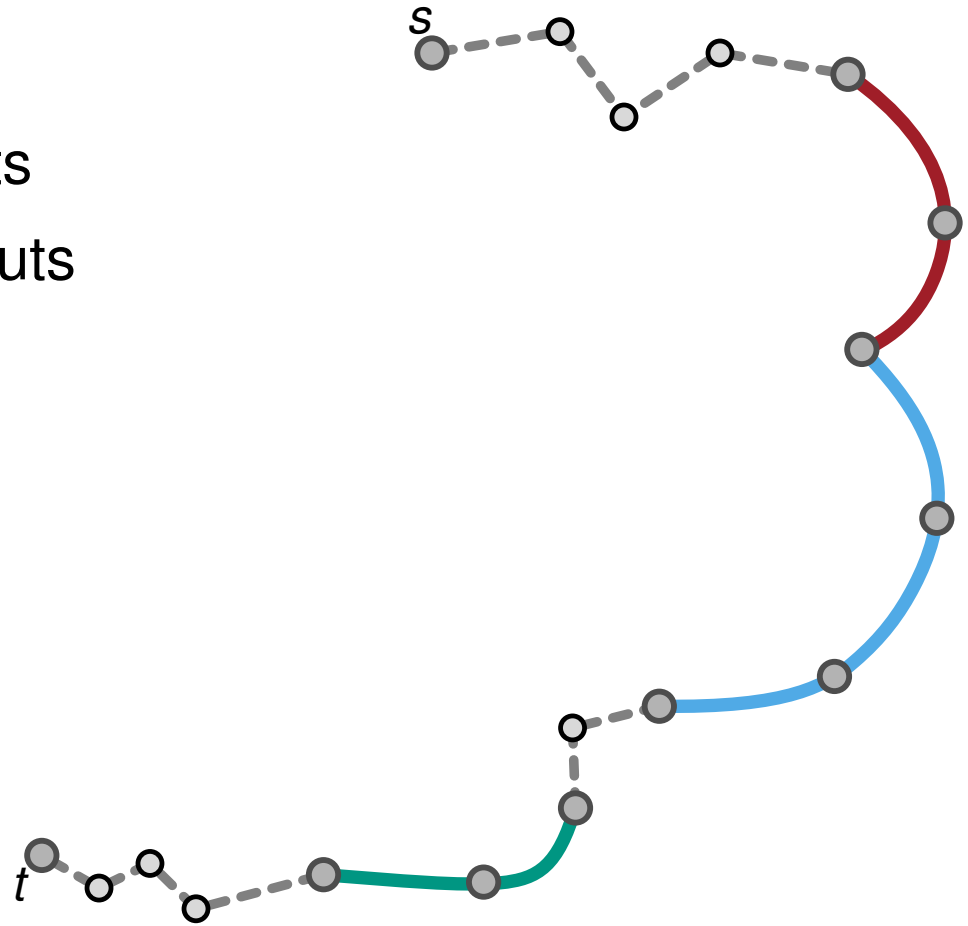
- Operates on the trips of the network
- Transitive transfer graph $\rightsquigarrow$ Shortcuts between stop events
- BFS-like search on the resulting graph



Integrating ULTRA and Trip-Based Preprocessing

Similarities and Differences:

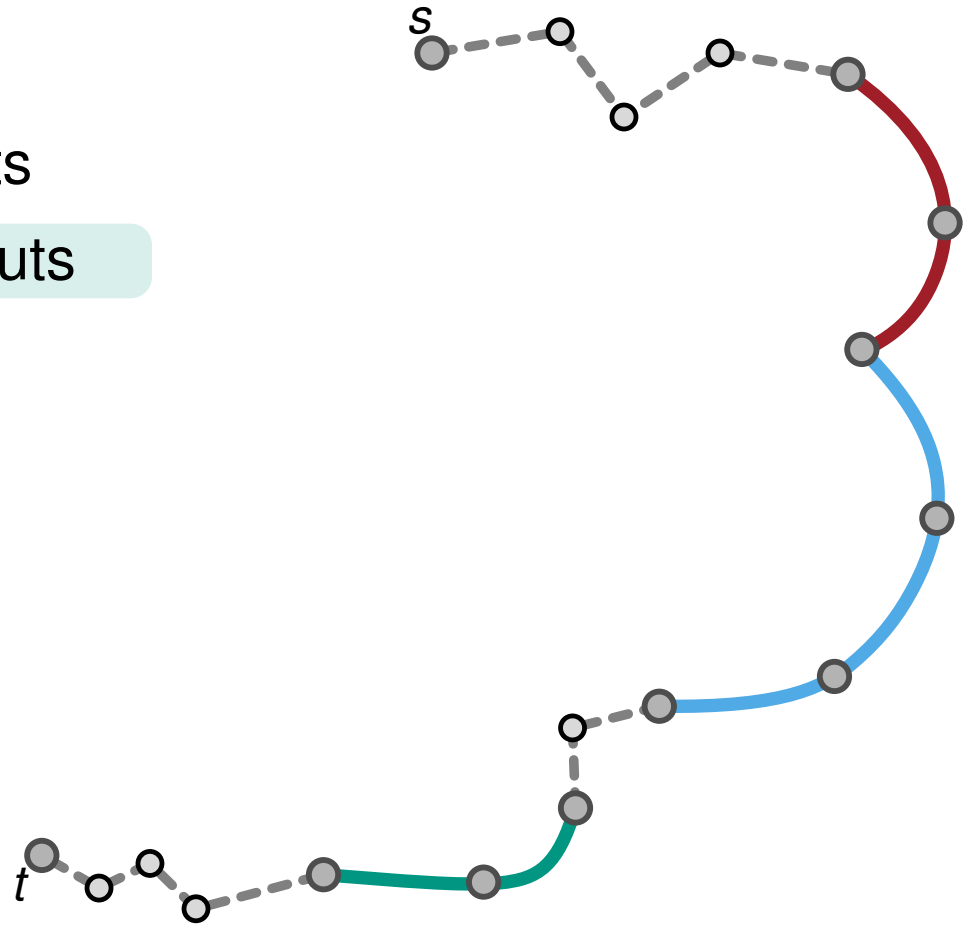
- Precomputed shortcuts represent transfers
  - ULTRA uses **time-independent** shortcuts
  - Trip-Based uses **time-dependent** shortcuts



Integrating ULTRA and Trip-Based Preprocessing

Similarities and Differences:

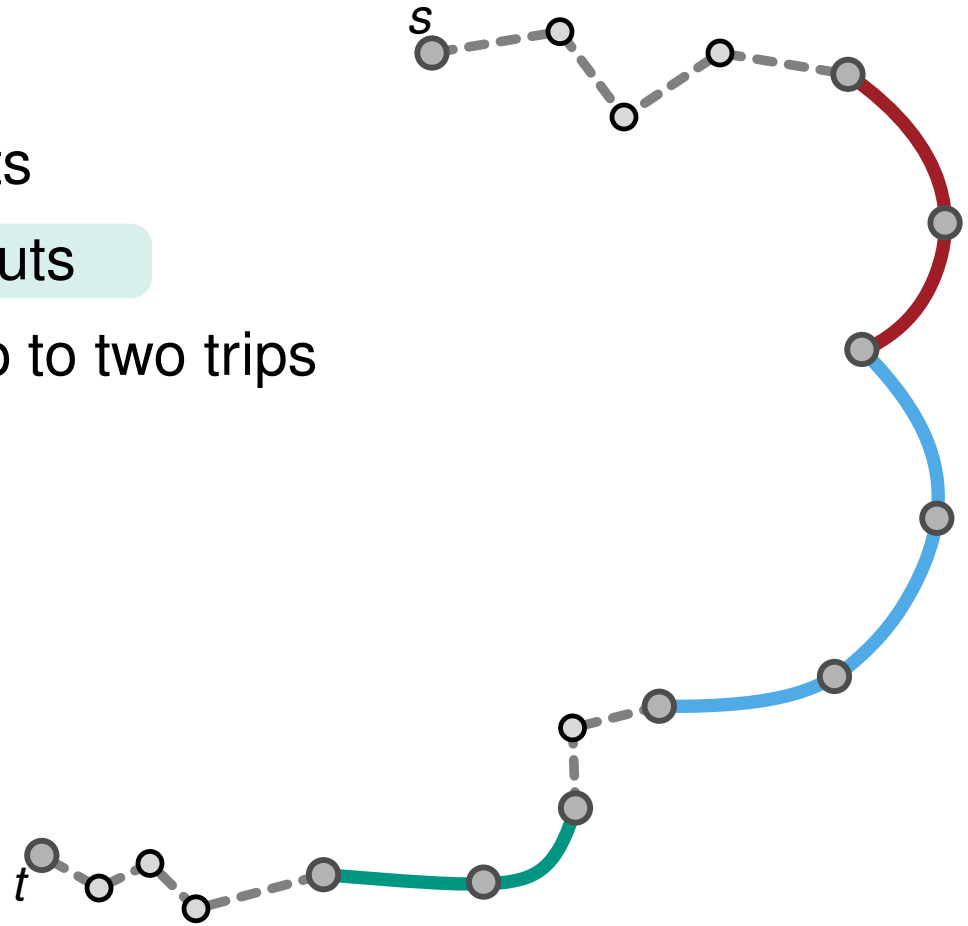
- Precomputed shortcuts represent transfers
  - ULTRA uses **time-independent** shortcuts
  - Trip-Based uses **time-dependent** shortcuts



Integrating ULTRA and Trip-Based Preprocessing

Similarities and Differences:

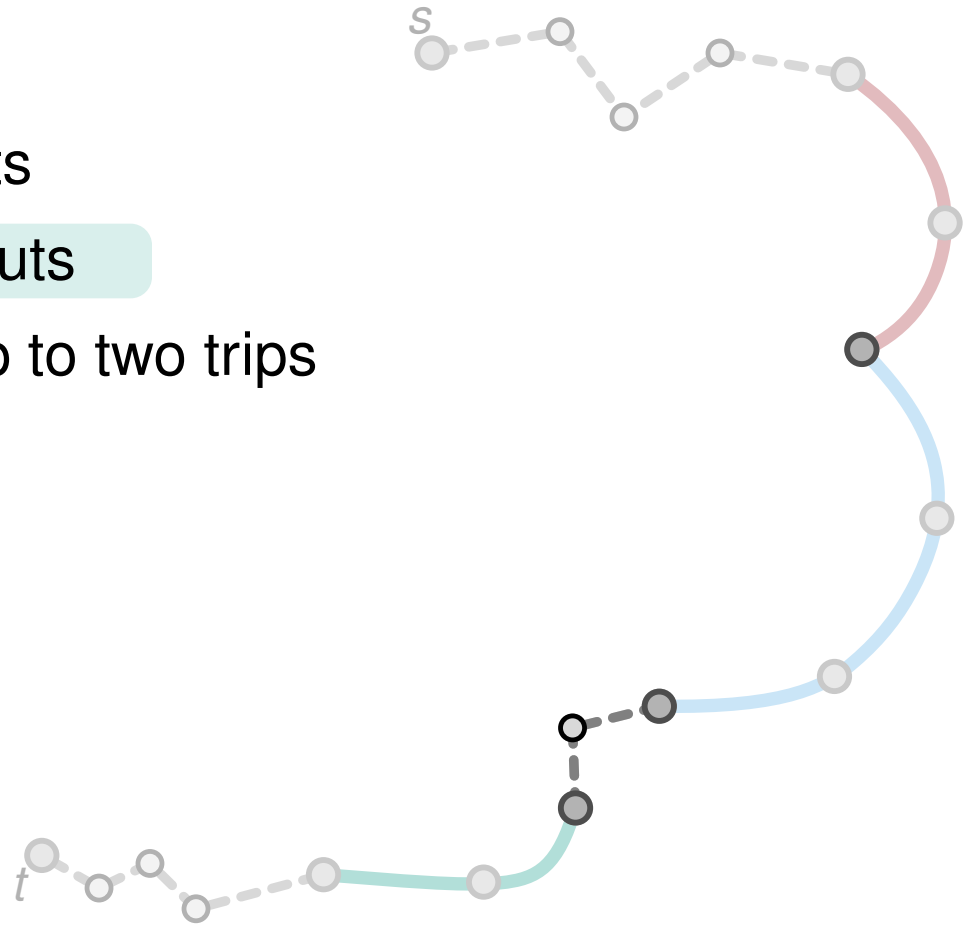
- Precomputed shortcuts represent transfers
  - ULTRA uses **time-independent** shortcuts
  - Trip-Based uses **time-dependent** shortcuts
- Preprocessing enumerates journeys with up to two trips



Integrating ULTRA and Trip-Based Preprocessing

Similarities and Differences:

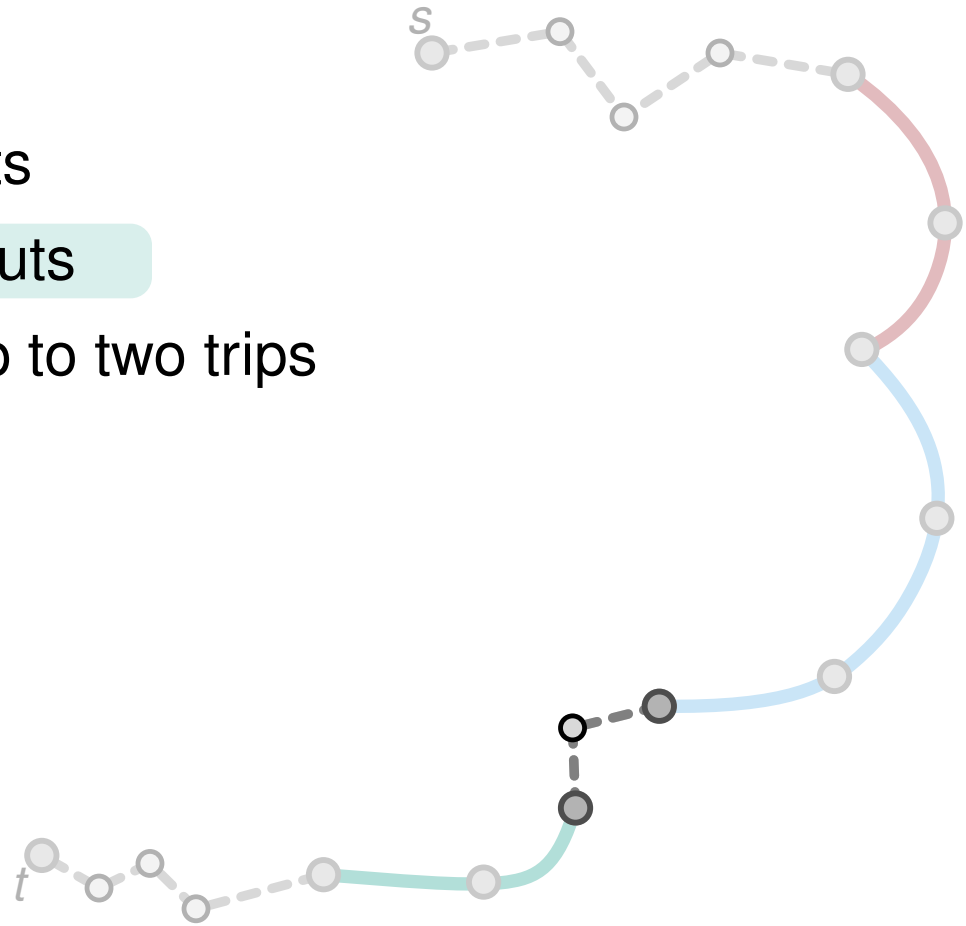
- Precomputed shortcuts represent transfers
  - ULTRA uses **time-independent** shortcuts
  - Trip-Based uses **time-dependent** shortcuts
- Preprocessing enumerates journeys with up to two trips



Integrating ULTRA and Trip-Based Preprocessing

Similarities and Differences:

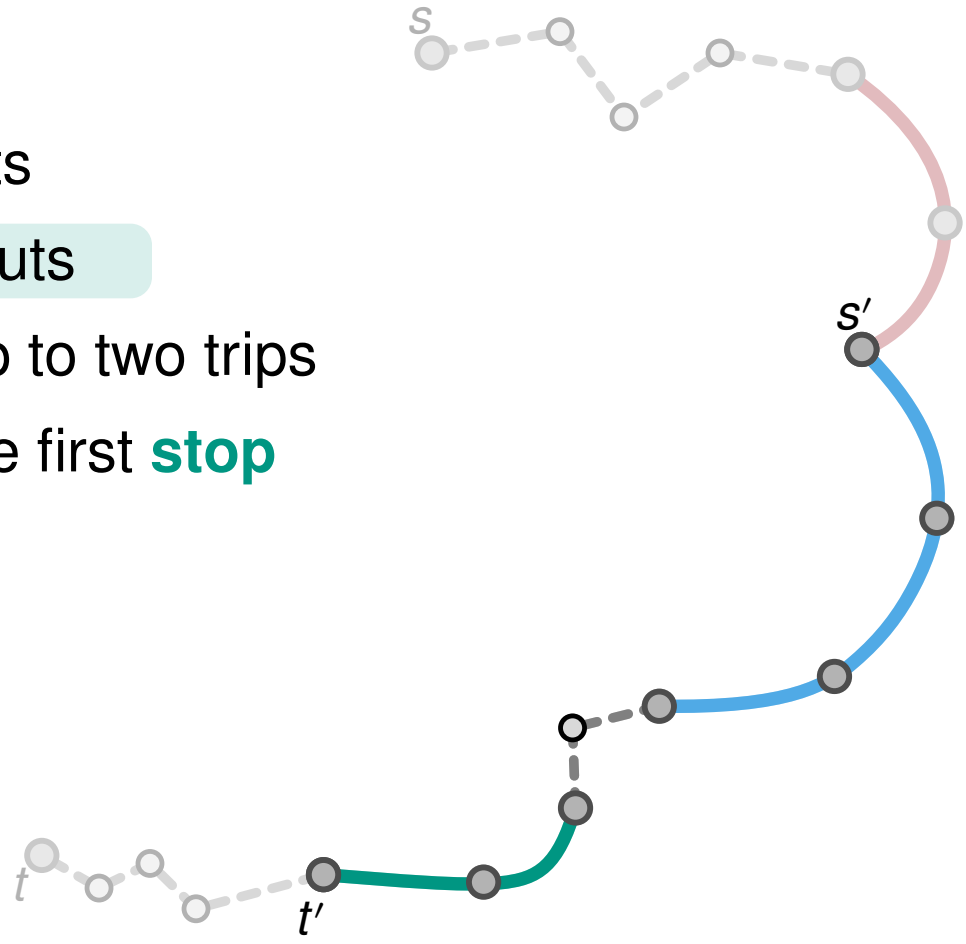
- Precomputed shortcuts represent transfers
  - ULTRA uses **time-independent** shortcuts
  - Trip-Based uses **time-dependent** shortcuts
- Preprocessing enumerates journeys with up to two trips



Integrating ULTRA and Trip-Based Preprocessing

Similarities and Differences:

- Precomputed shortcuts represent transfers
  - ULTRA uses **time-independent** shortcuts
  - Trip-Based uses **time-dependent** shortcuts
- Preprocessing enumerates journeys with up to two trips
  - ULTRA compares journeys with the same first **stop**

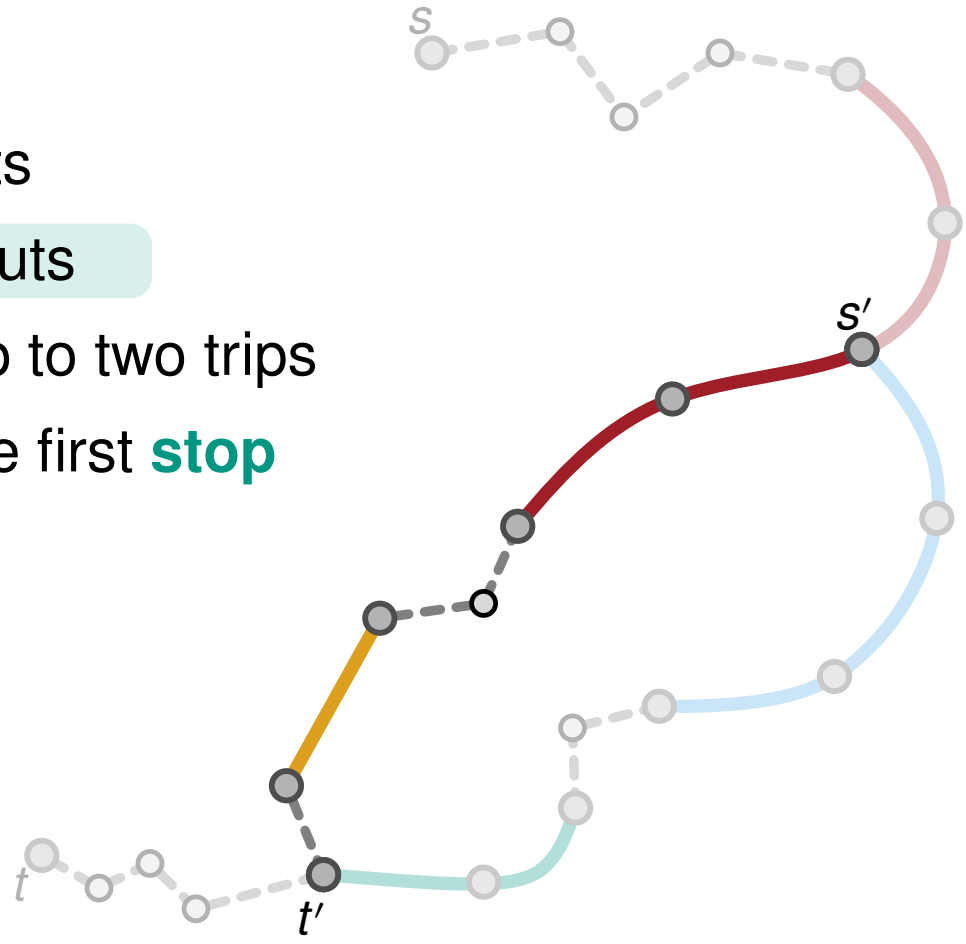


Institute of Theoretical Informatics
Algorithmics Group

Integrating ULTRA and Trip-Based Preprocessing

Similarities and Differences:

- Precomputed shortcuts represent transfers
  - ULTRA uses **time-independent** shortcuts
  - Trip-Based uses **time-dependent** shortcuts
- Preprocessing enumerates journeys with up to two trips
  - ULTRA compares journeys with the same first **stop**

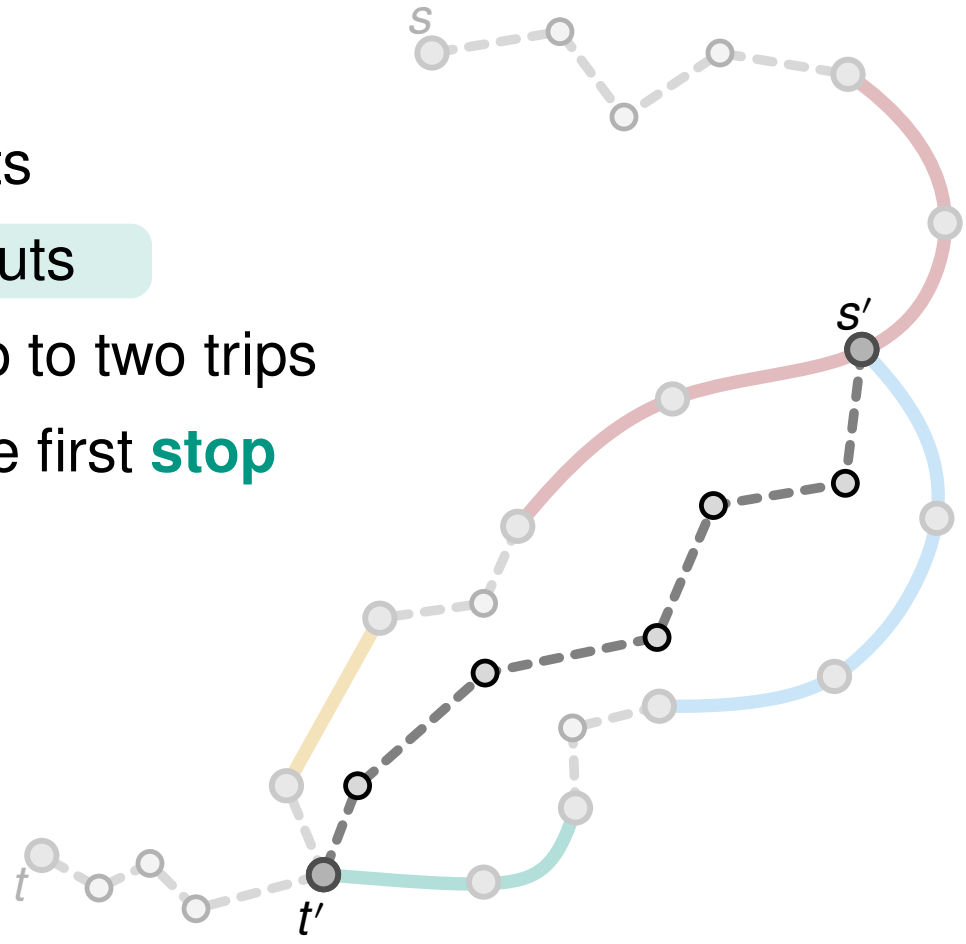


Institute of Theoretical Informatics
Algorithmics Group

Integrating ULTRA and Trip-Based Preprocessing

Similarities and Differences:

- Precomputed shortcuts represent transfers
  - ULTRA uses **time-independent** shortcuts
  - Trip-Based uses **time-dependent** shortcuts
- Preprocessing enumerates journeys with up to two trips
  - ULTRA compares journeys with the same first **stop**

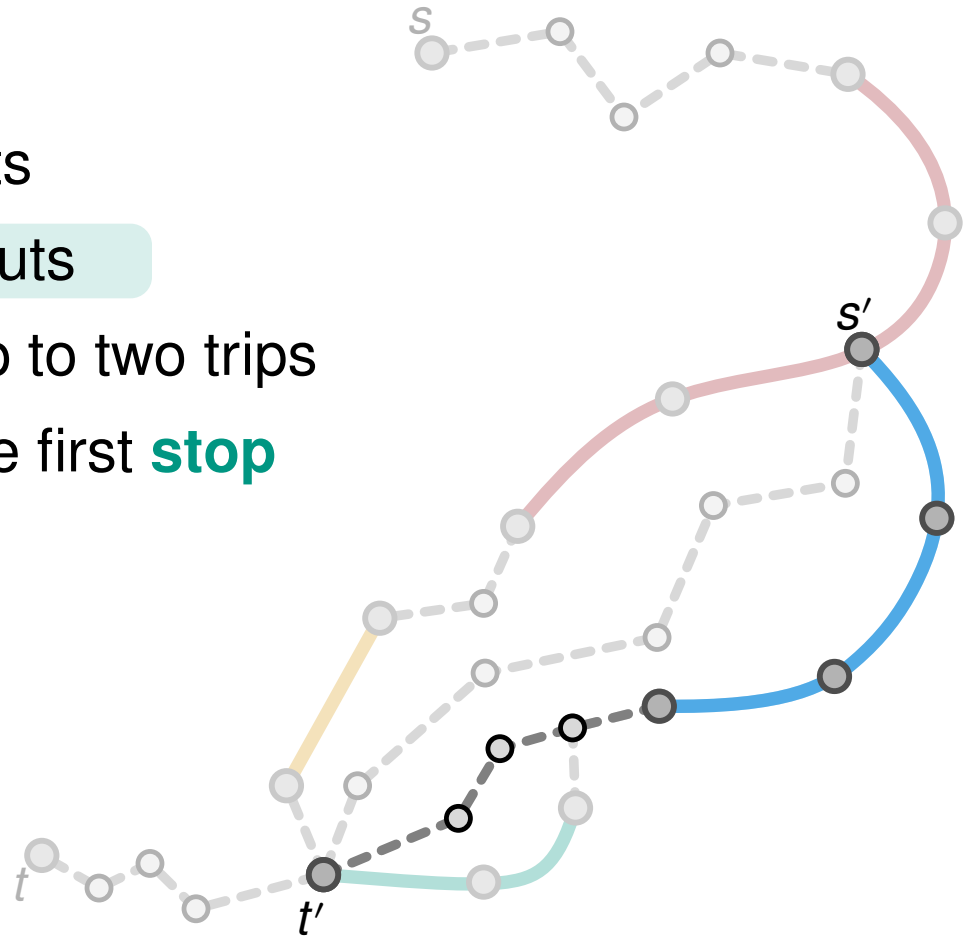


Institute of Theoretical Informatics
Algorithmics Group

Integrating ULTRA and Trip-Based Preprocessing

Similarities and Differences:

- Precomputed shortcuts represent transfers
  - ULTRA uses **time-independent** shortcuts
  - Trip-Based uses **time-dependent** shortcuts
- Preprocessing enumerates journeys with up to two trips
  - ULTRA compares journeys with the same first **stop**



Institute of Theoretical Informatics
Algorithmics Group

Integrating ULTRA and Trip-Based Preprocessing

Similarities and Differences:

- Precomputed shortcuts represent transfers
  - ULTRA uses **time-independent** shortcuts
  - Trip-Based uses **time-dependent** shortcuts
- Preprocessing enumerates journeys with up to two trips
  - ULTRA compares journeys with the same first **stop**

Institute of Theoretical Informatics
Algorithmics Group

Integrating ULTRA and Trip-Based Preprocessing

Similarities and Differences:

- Precomputed shortcuts represent transfers
  - ULTRA uses **time-independent** shortcuts
  - Trip-Based uses **time-dependent** shortcuts
- Preprocessing enumerates journeys with up to two trips
  - ULTRA compares journeys with the same first **stop**

Institute of Theoretical Informatics
Algorithmics Group

Integrating ULTRA and Trip-Based Preprocessing

Similarities and Differences:

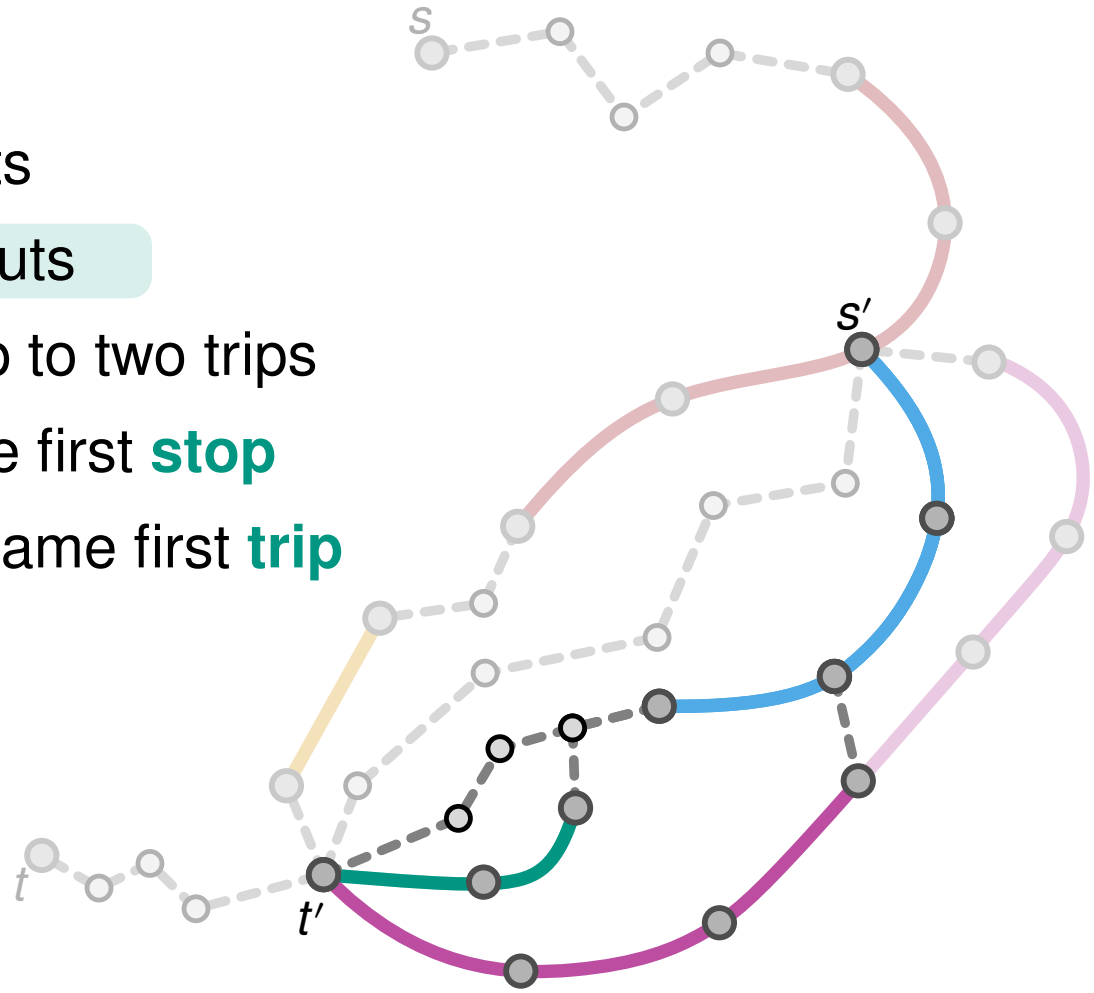
- Precomputed shortcuts represent transfers
  - ULTRA uses **time-independent** shortcuts
  - Trip-Based uses **time-dependent** shortcuts
- Preprocessing enumerates journeys with up to two trips
  - ULTRA compares journeys with the same first **stop**

Institute of Theoretical Informatics
Algorithmics Group

Integrating ULTRA and Trip-Based Preprocessing

Similarities and Differences:

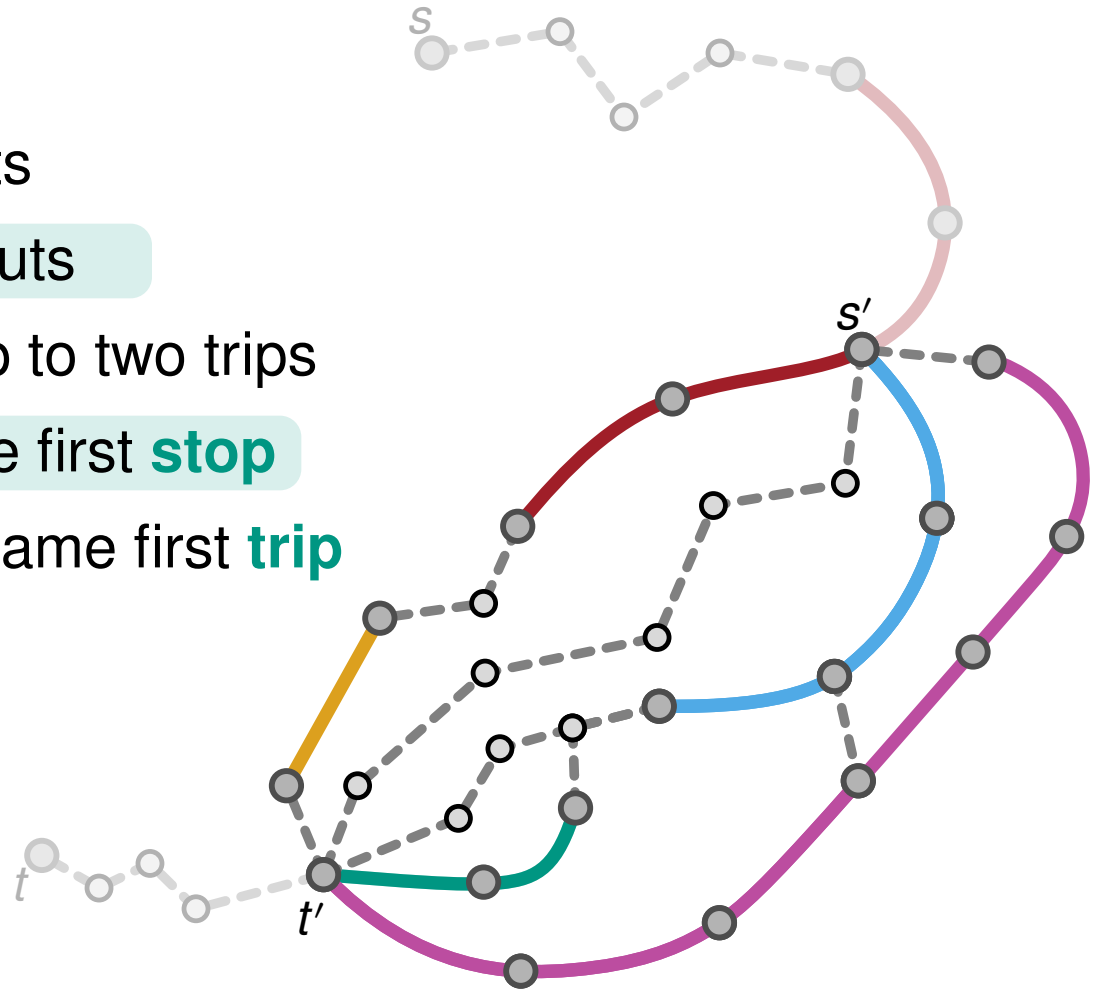
- Precomputed shortcuts represent transfers
  - ULTRA uses **time-independent** shortcuts
  - Trip-Based uses **time-dependent** shortcuts
- Preprocessing enumerates journeys with up to two trips
  - ULTRA compares journeys with the same first **stop**
  - Trip-Based compares journeys with the same first **trip**



Integrating ULTRA and Trip-Based Preprocessing

Similarities and Differences:

- Precomputed shortcuts represent transfers
  - ULTRA uses **time-independent** shortcuts
  - Trip-Based uses **time-dependent** shortcuts
- Preprocessing enumerates journeys with up to two trips
  - ULTRA compares journeys with the same first **stop**
  - Trip-Based compares journeys with the same first **trip**



Integrating ULTRA and Trip-Based Preprocessing

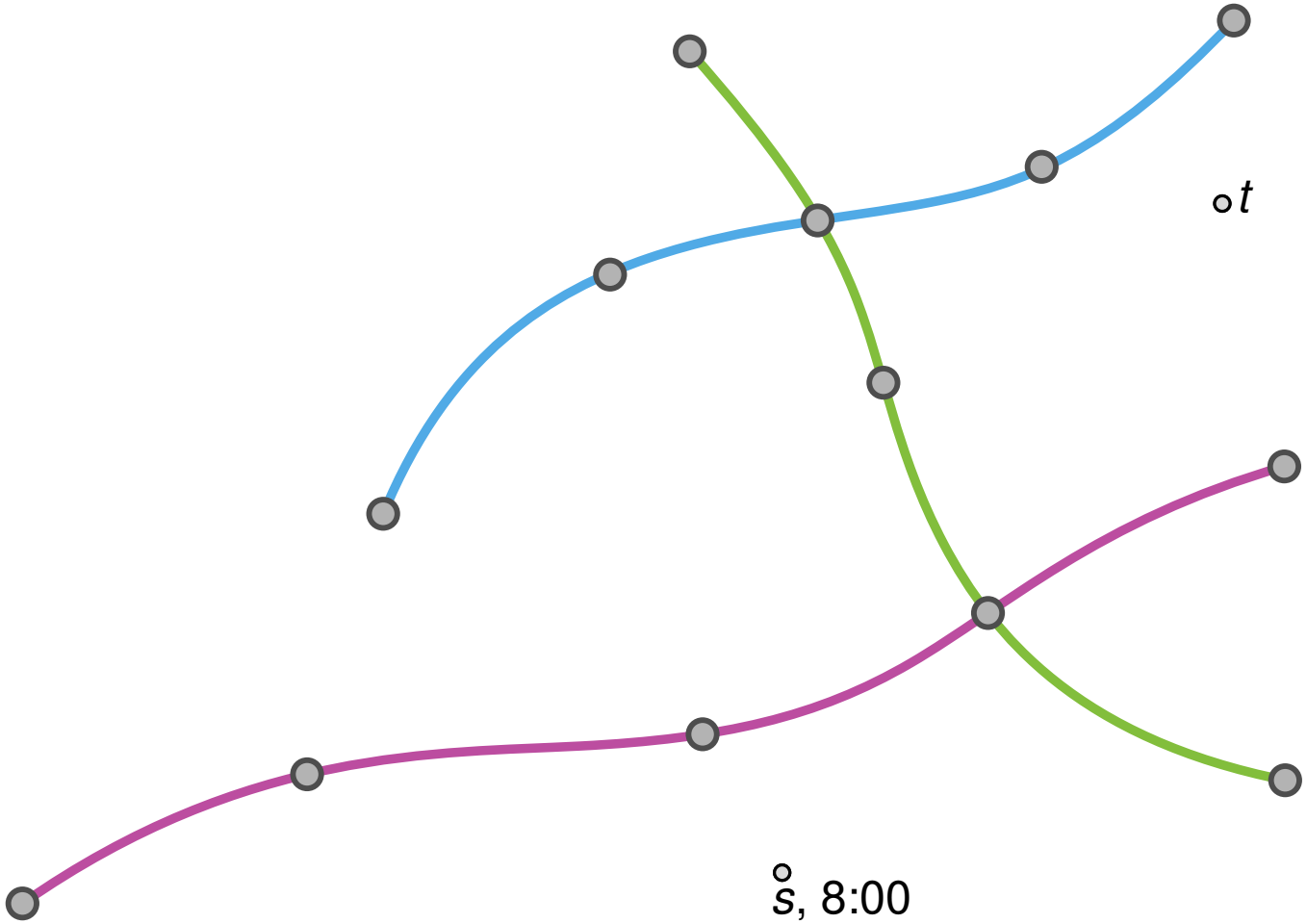
Similarities and Differences:

- Precomputed shortcuts represent transfers
  - ULTRA uses **time-independent** shortcuts
  - Trip-Based uses **time-dependent** shortcuts
- Preprocessing enumerates journeys with up to two trips
  - ULTRA compares journeys with the same first **stop**
  - Trip-Based compares journeys with the same first **trip**
- Shortcut is required if no better alternative exists
  - For a suitable definiton of “better”

Institute of Theoretical Informatics
Algorithmics Group

Integrated ULTRA-Trip-Based Query

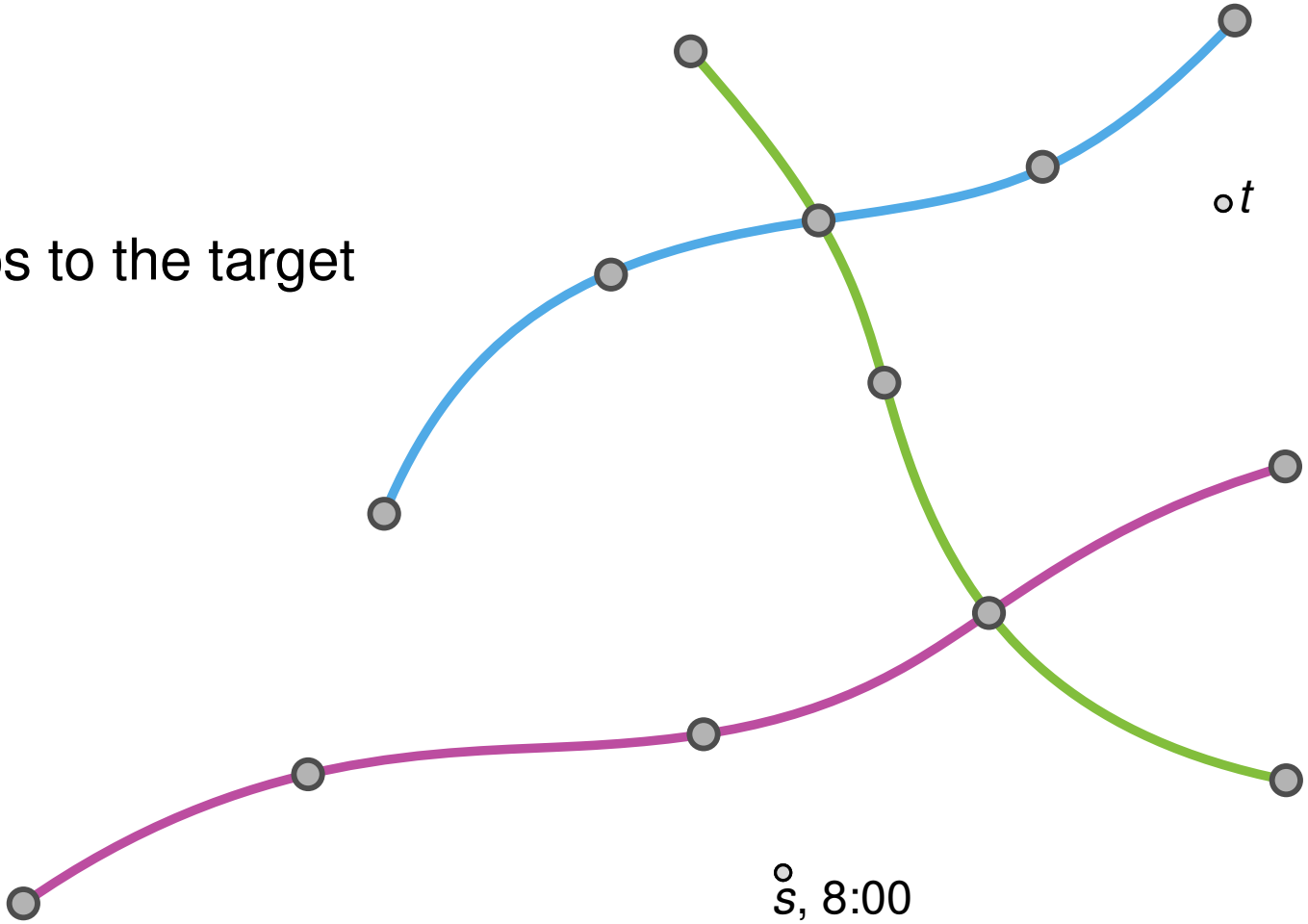
Query Algorithm Outline:



Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

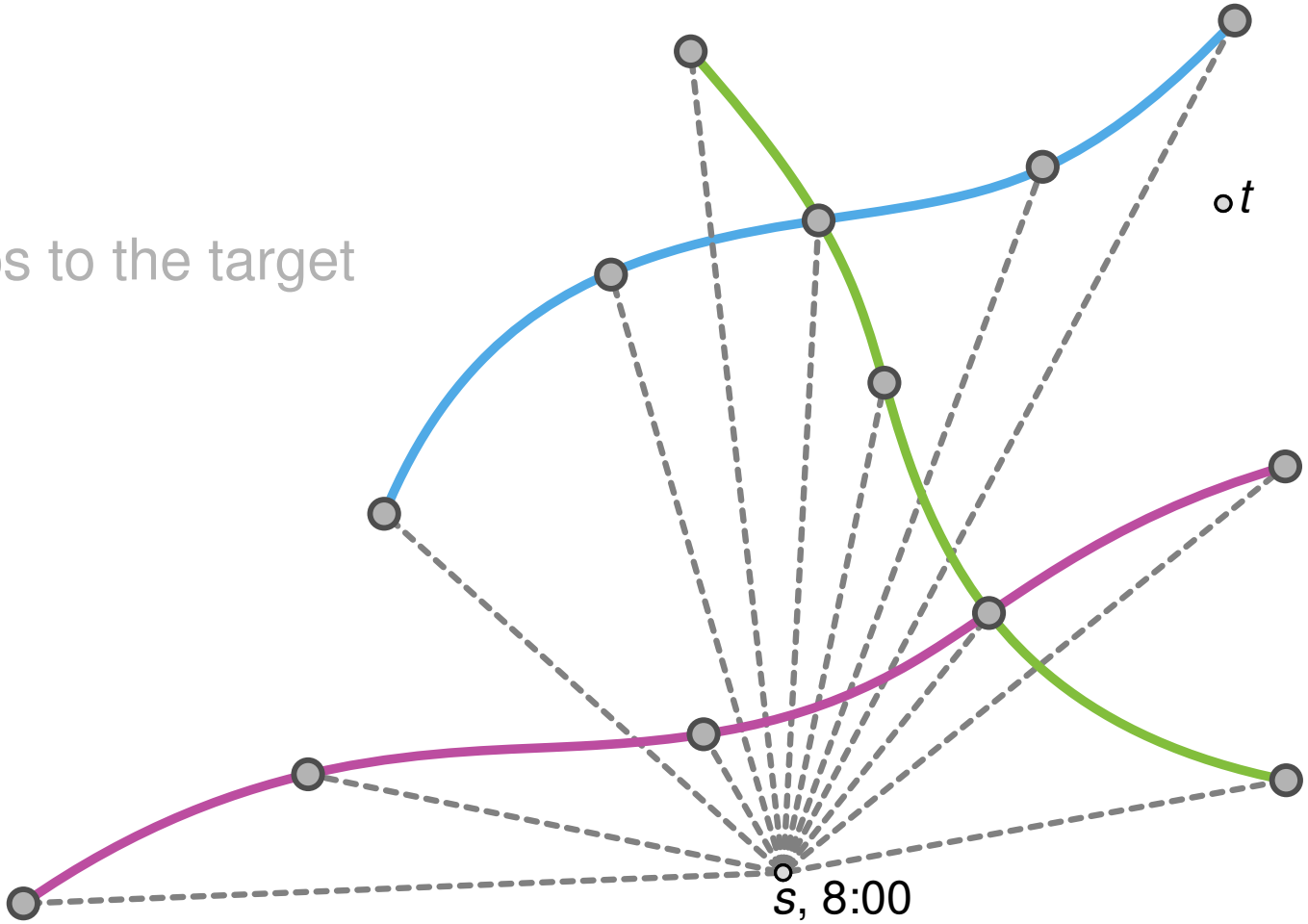
1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target



Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

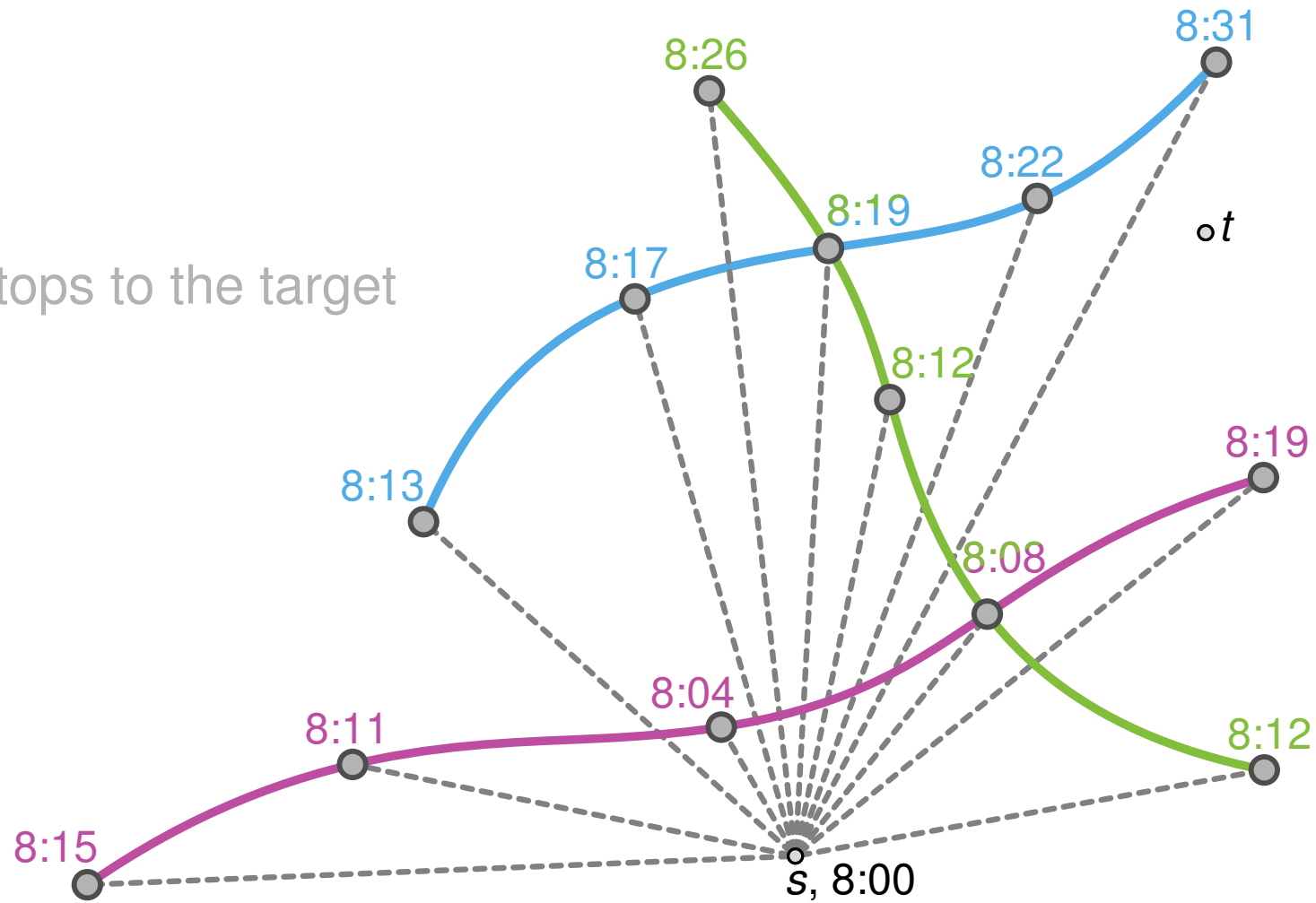
1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target



Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

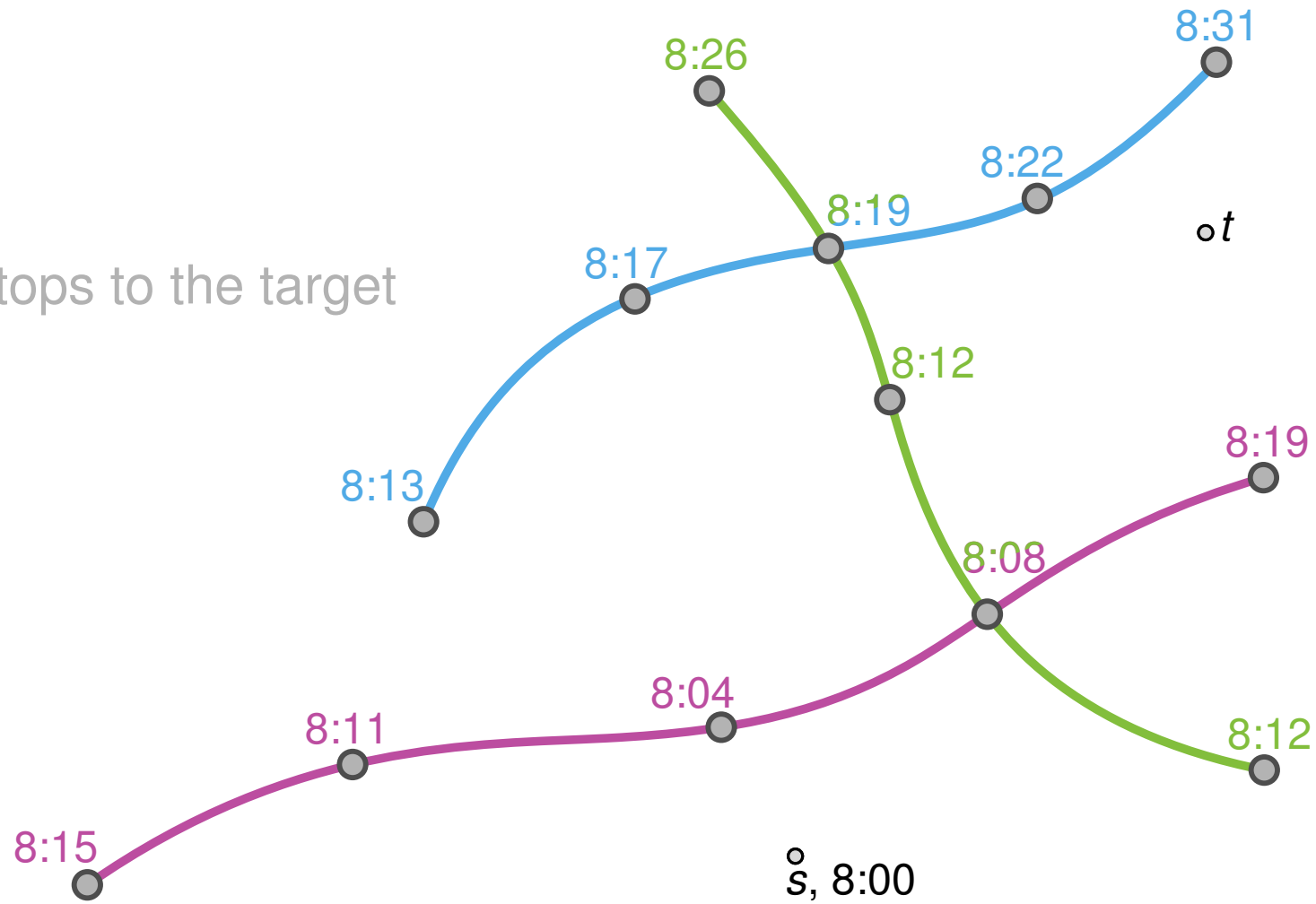
1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target



Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target

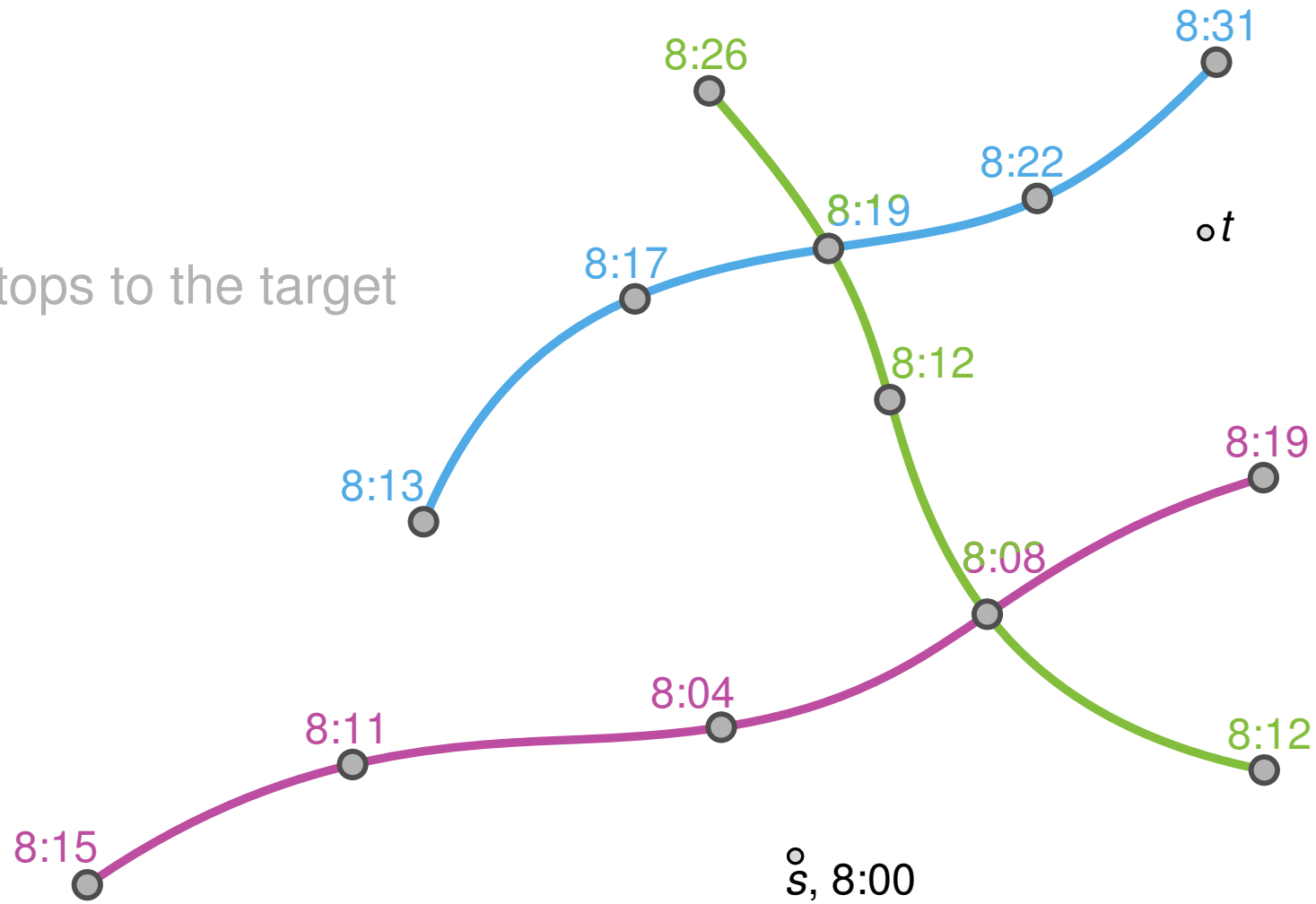


Institute of Theoretical Informatics
Algorithmics Group

Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

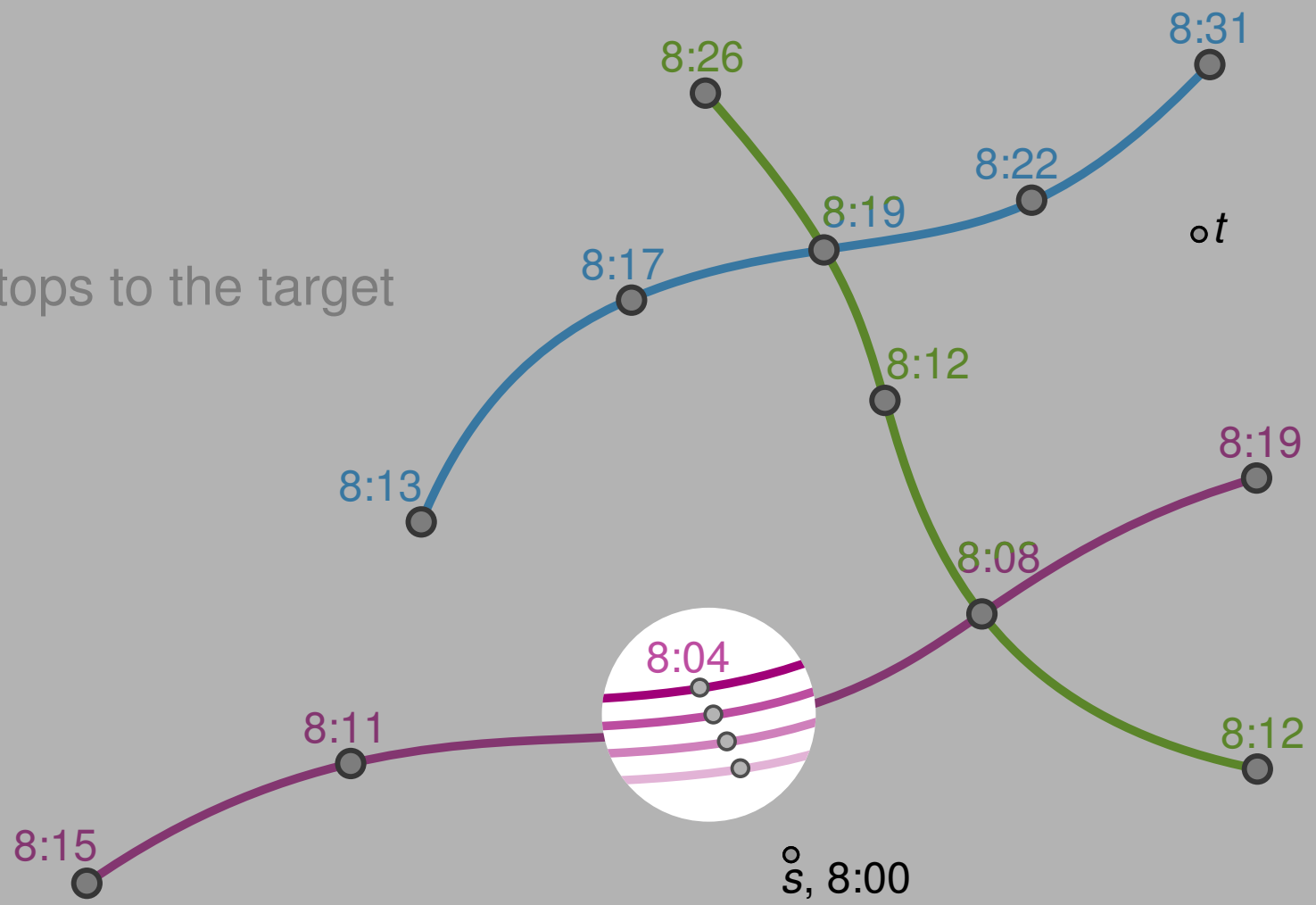
1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target
2. Trip-based query?



Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target
2. Trip-based query?

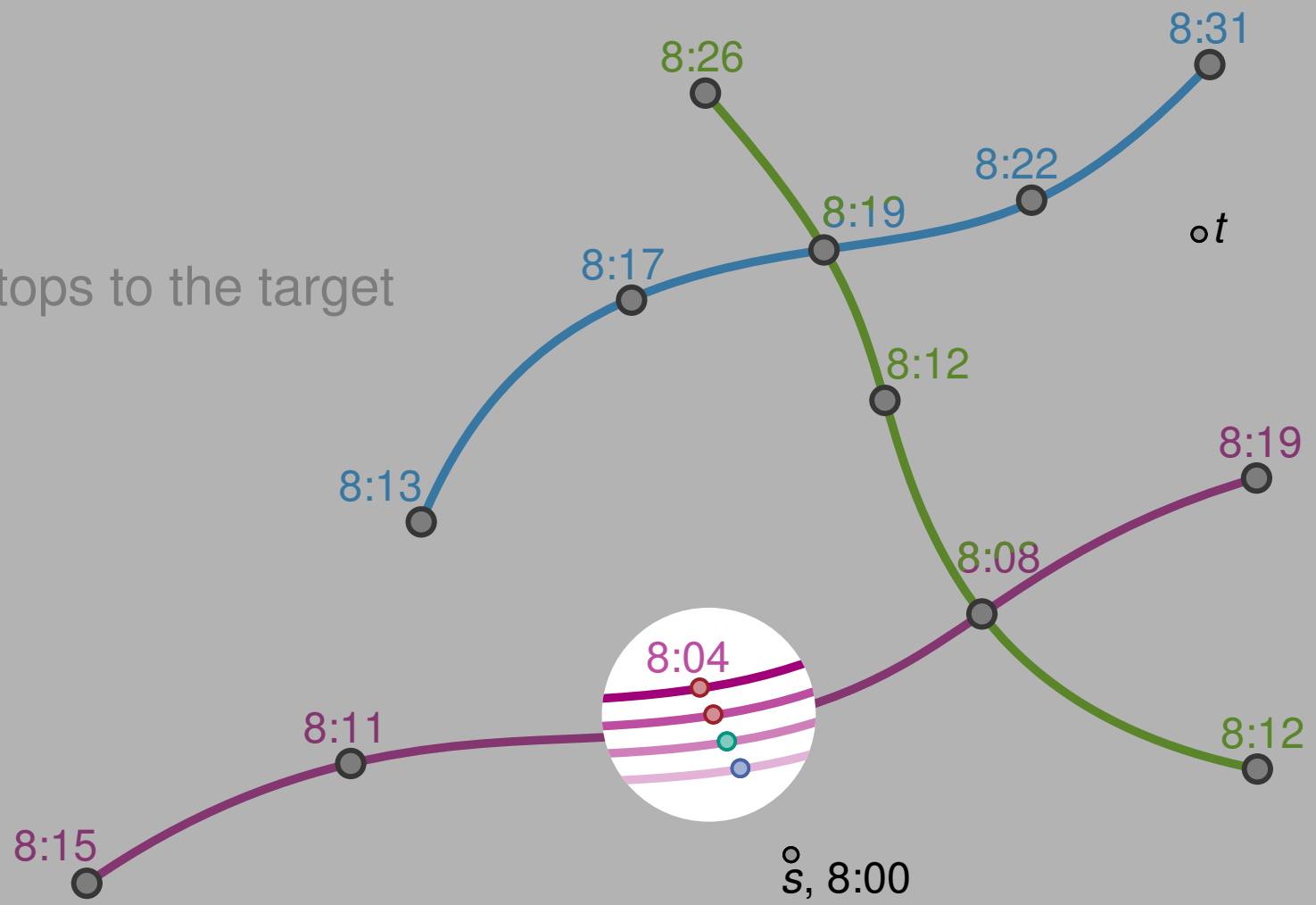


Institute of Theoretical Informatics
Algorithmics Group

Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

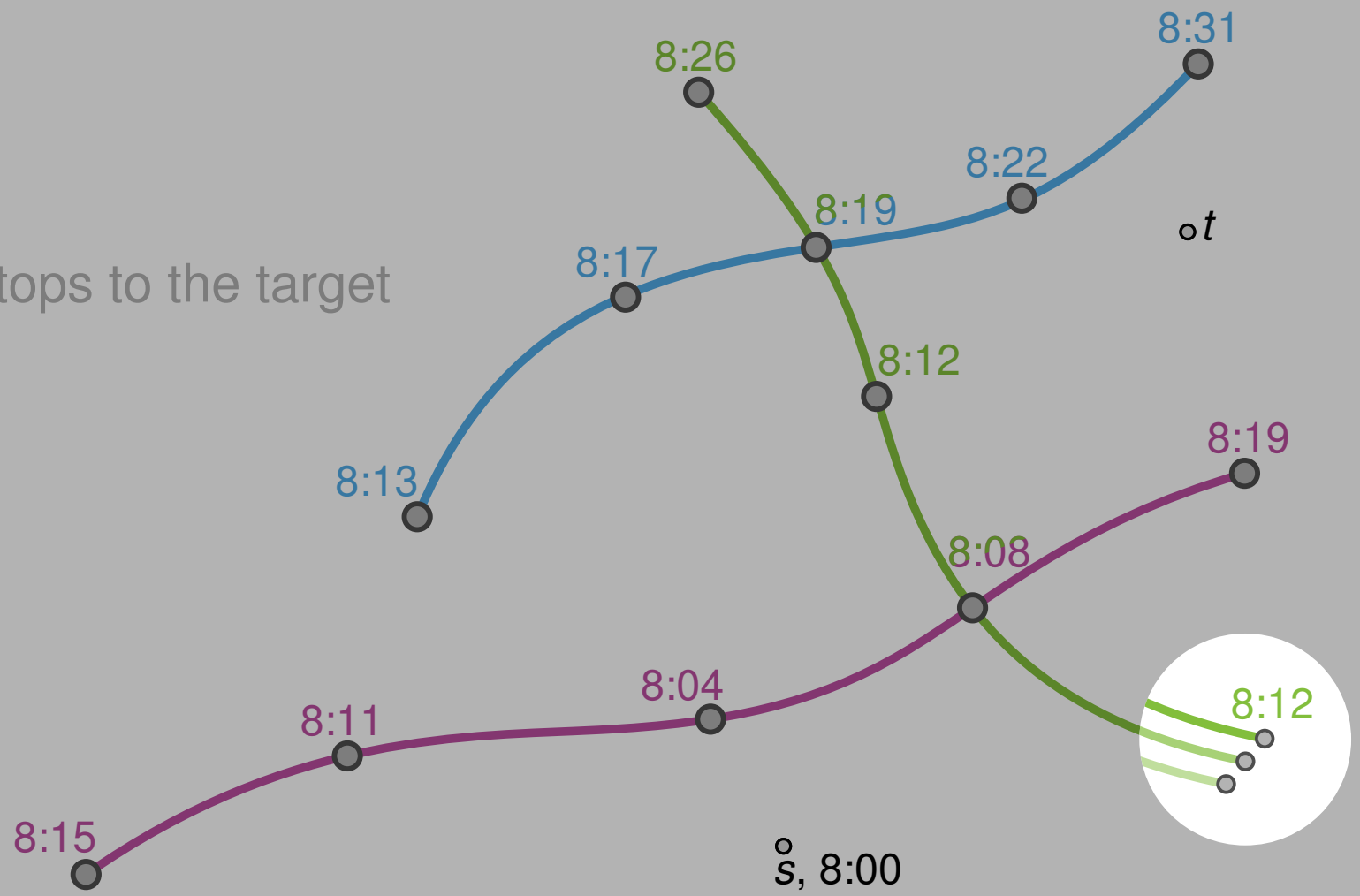
1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target
2. Trip-based query?



Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target
2. Trip-based query?

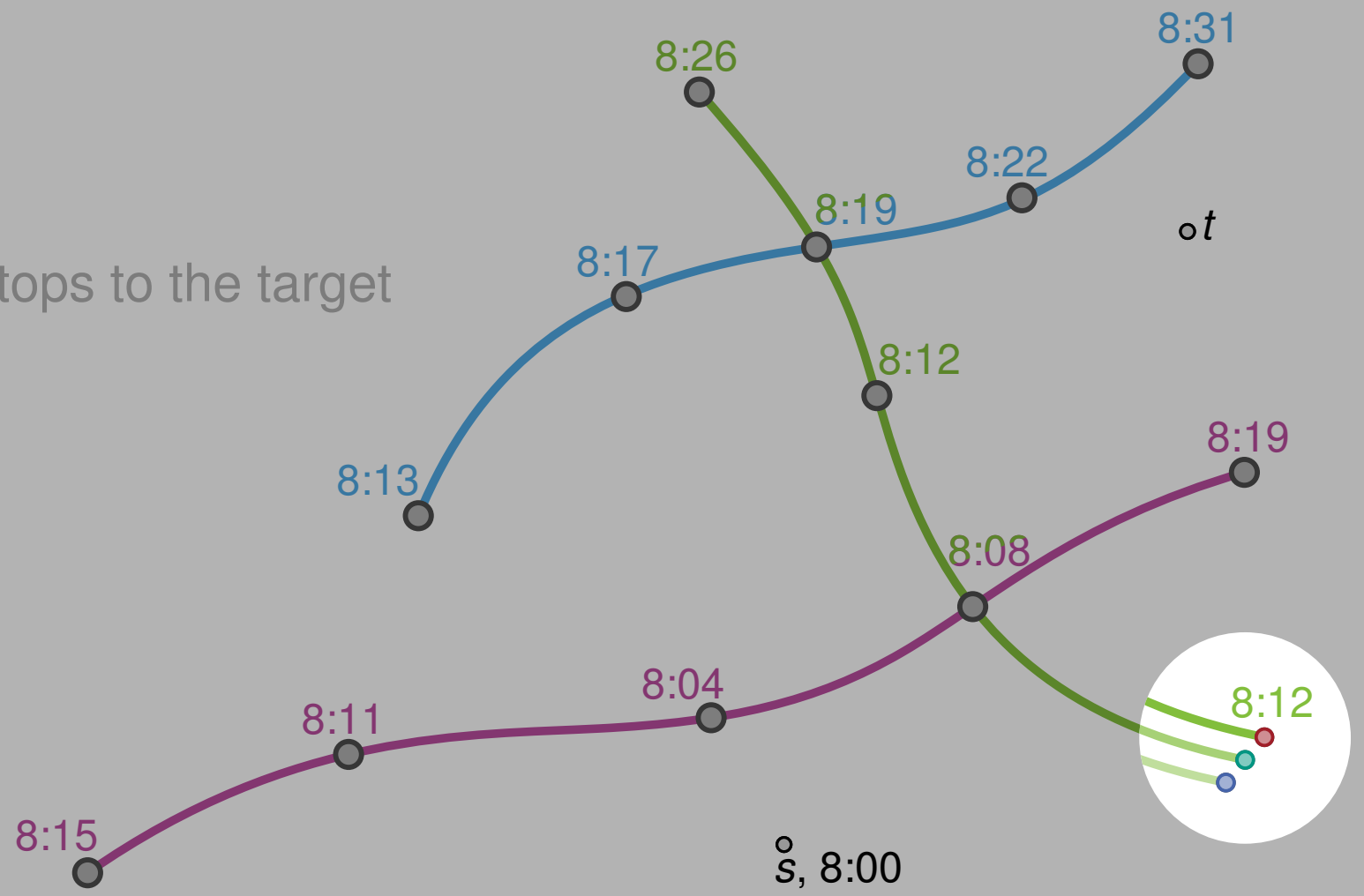


Institute of Theoretical Informatics
Algorithmics Group

Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target
2. Trip-based query?

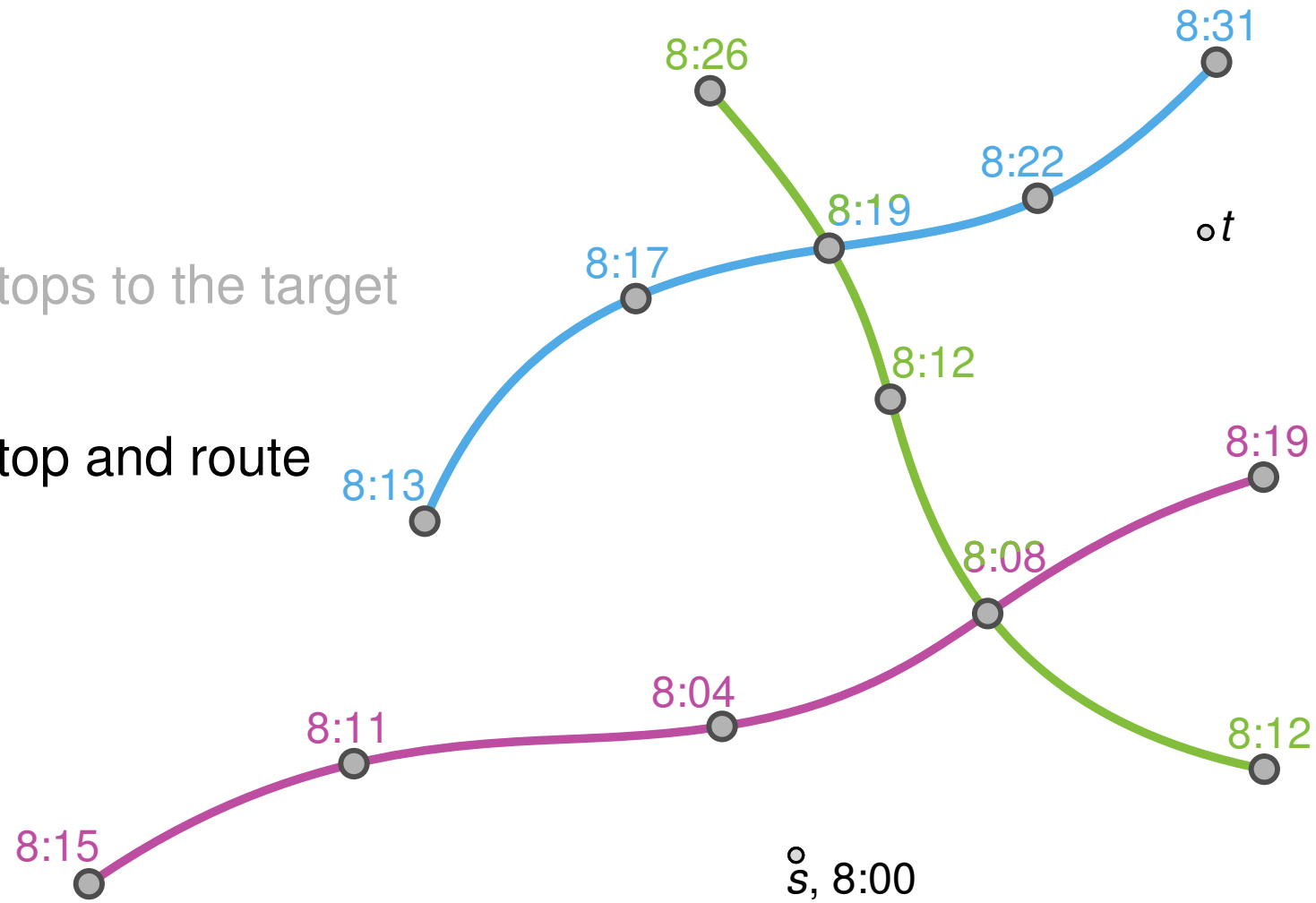


Institute of Theoretical Informatics
Algorithmics Group

Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

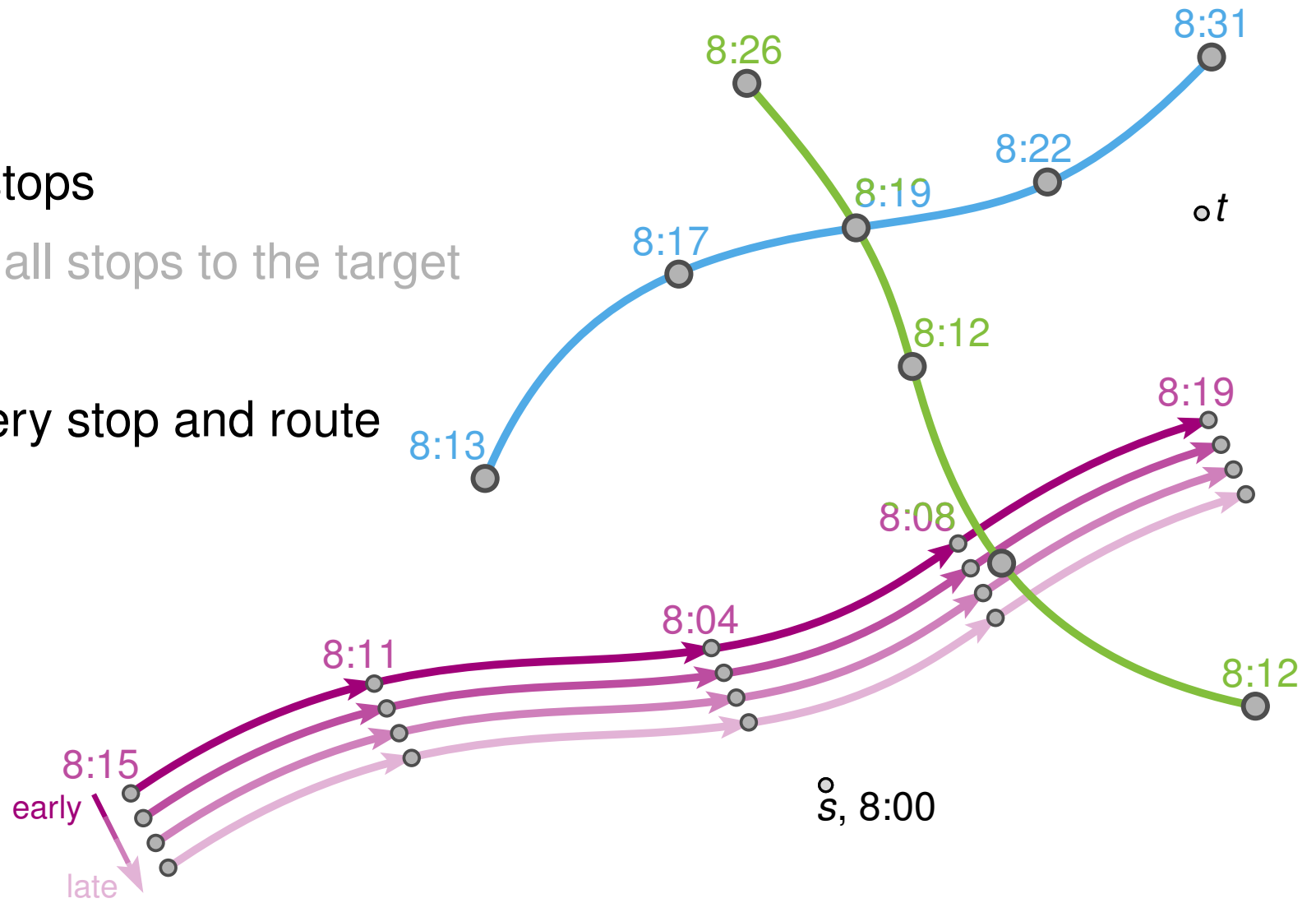
1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target
2. RAPTOR collect routes
   - Find the first trip for every stop and route



Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target
2. RAPTOR collect routes
   - Find the first trip for every stop and route

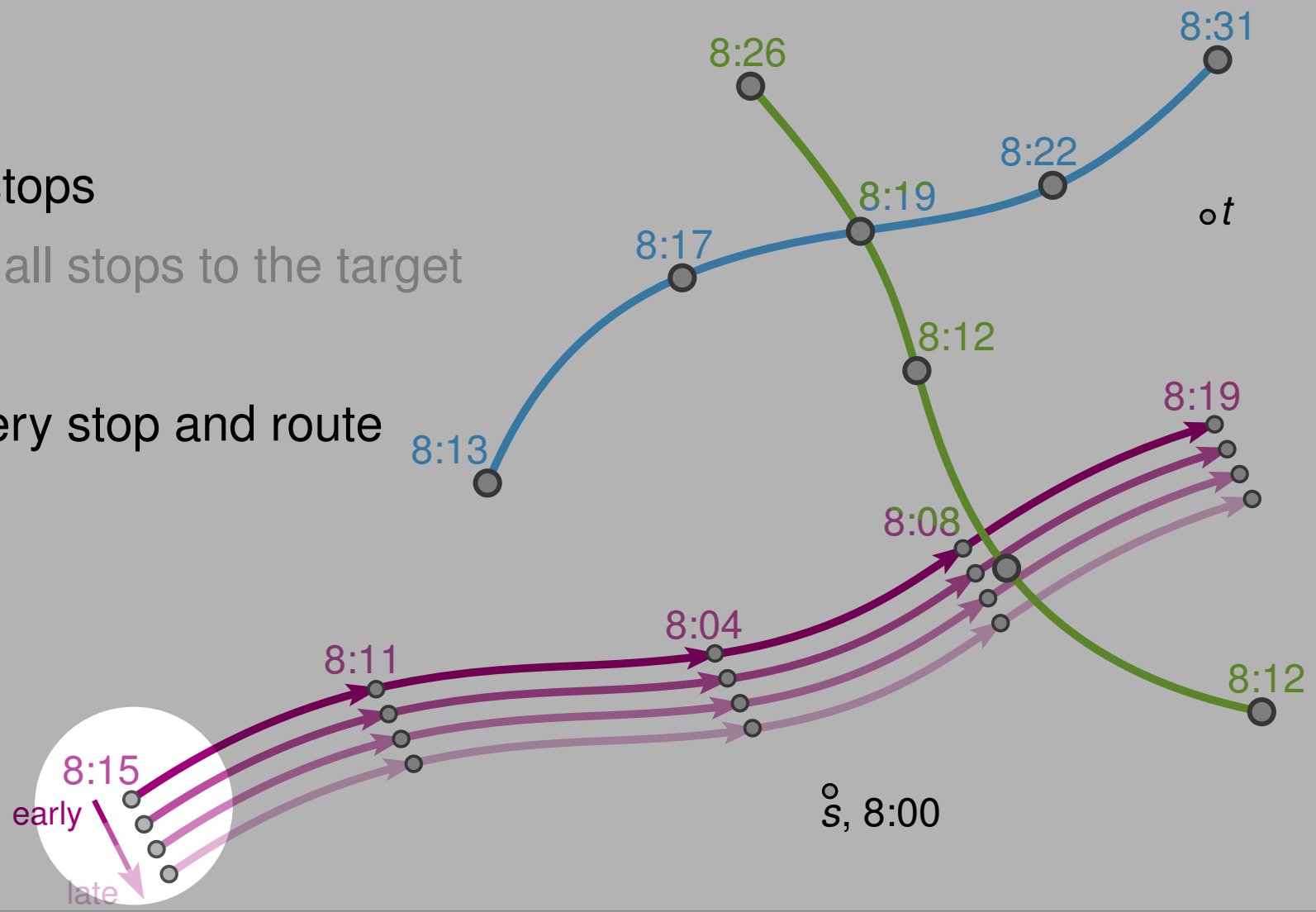


Institute of Theoretical Informatics
Algorithmics Group

Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target
2. RAPTOR collect routes
   - Find the first trip for every stop and route

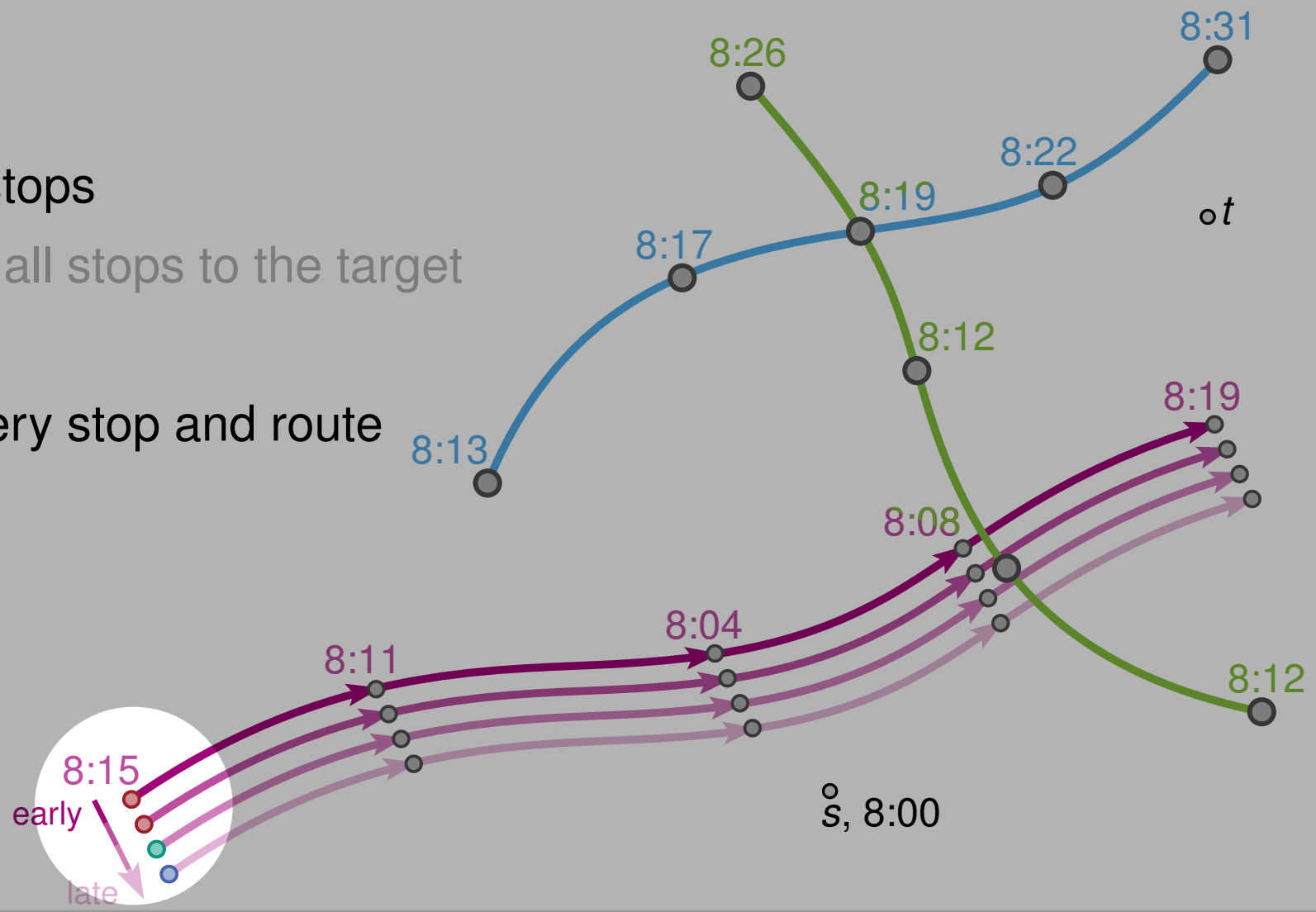


Institute of Theoretical Informatics
Algorithmics Group

Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target
2. RAPTOR collect routes
   - Find the first trip for every stop and route

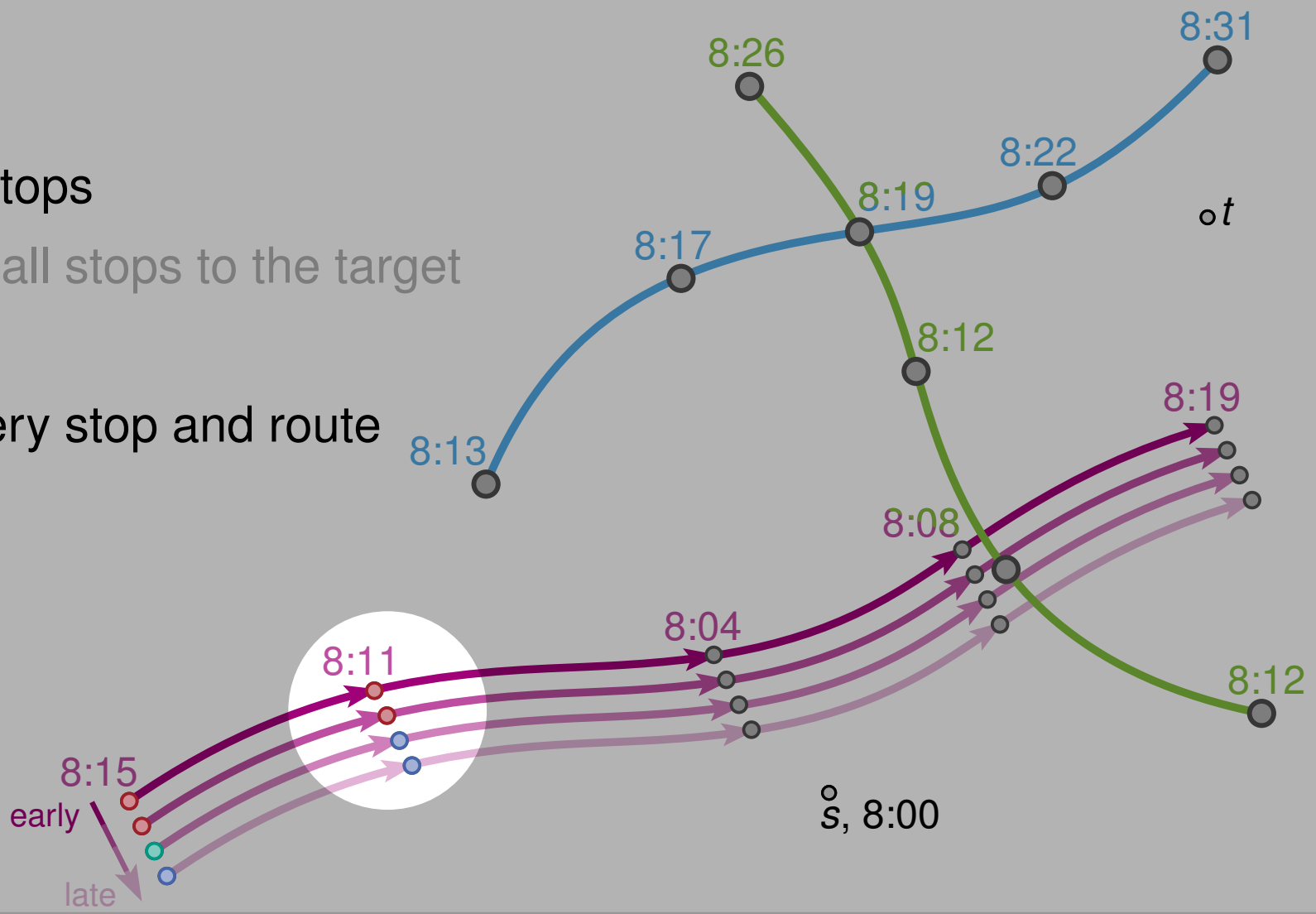


Institute of Theoretical Informatics
Algorithmics Group

Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target
2. RAPTOR collect routes
   - Find the first trip for every stop and route

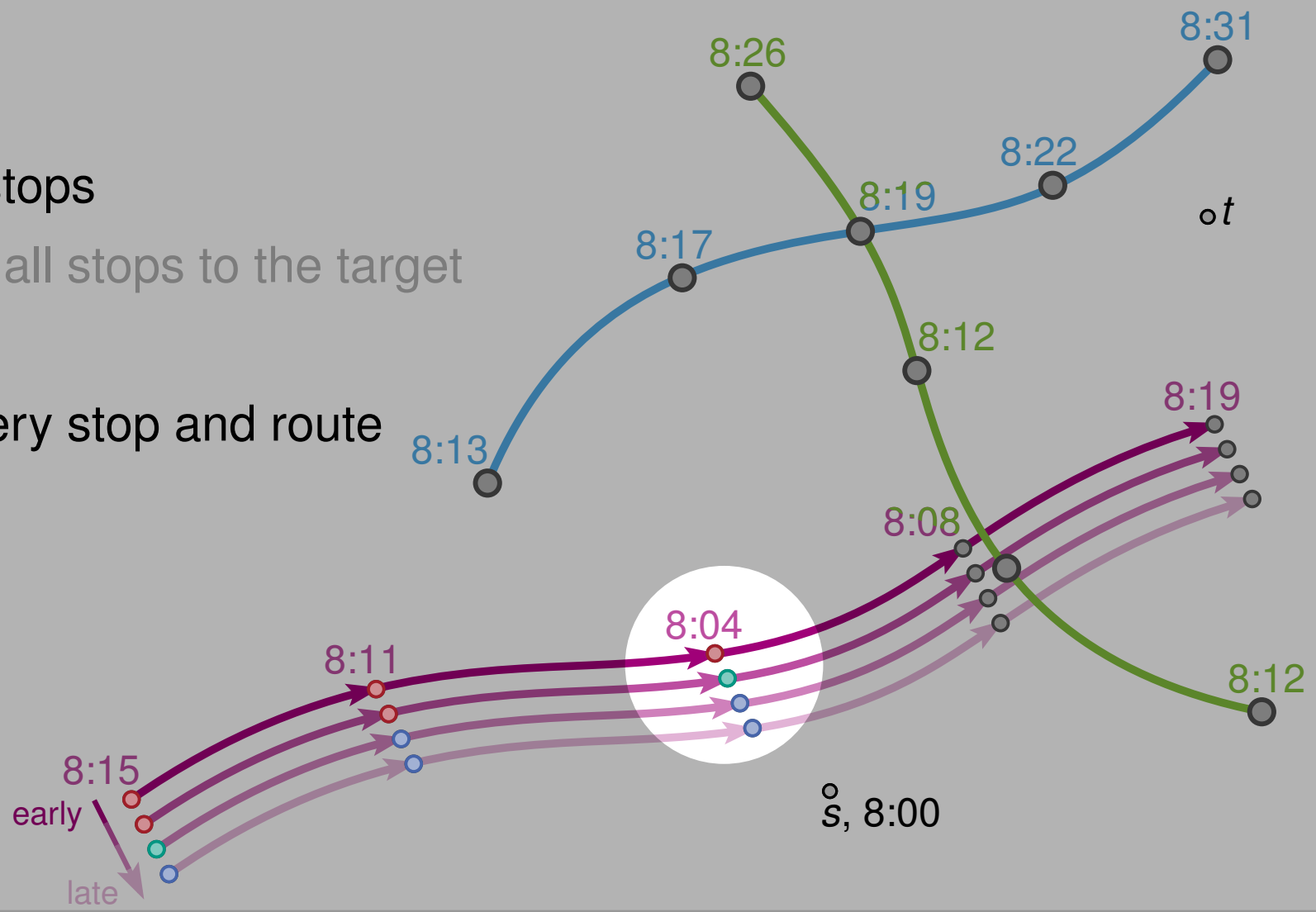


Institute of Theoretical Informatics
Algorithmics Group

Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target
2. RAPTOR collect routes
   - Find the first trip for every stop and route

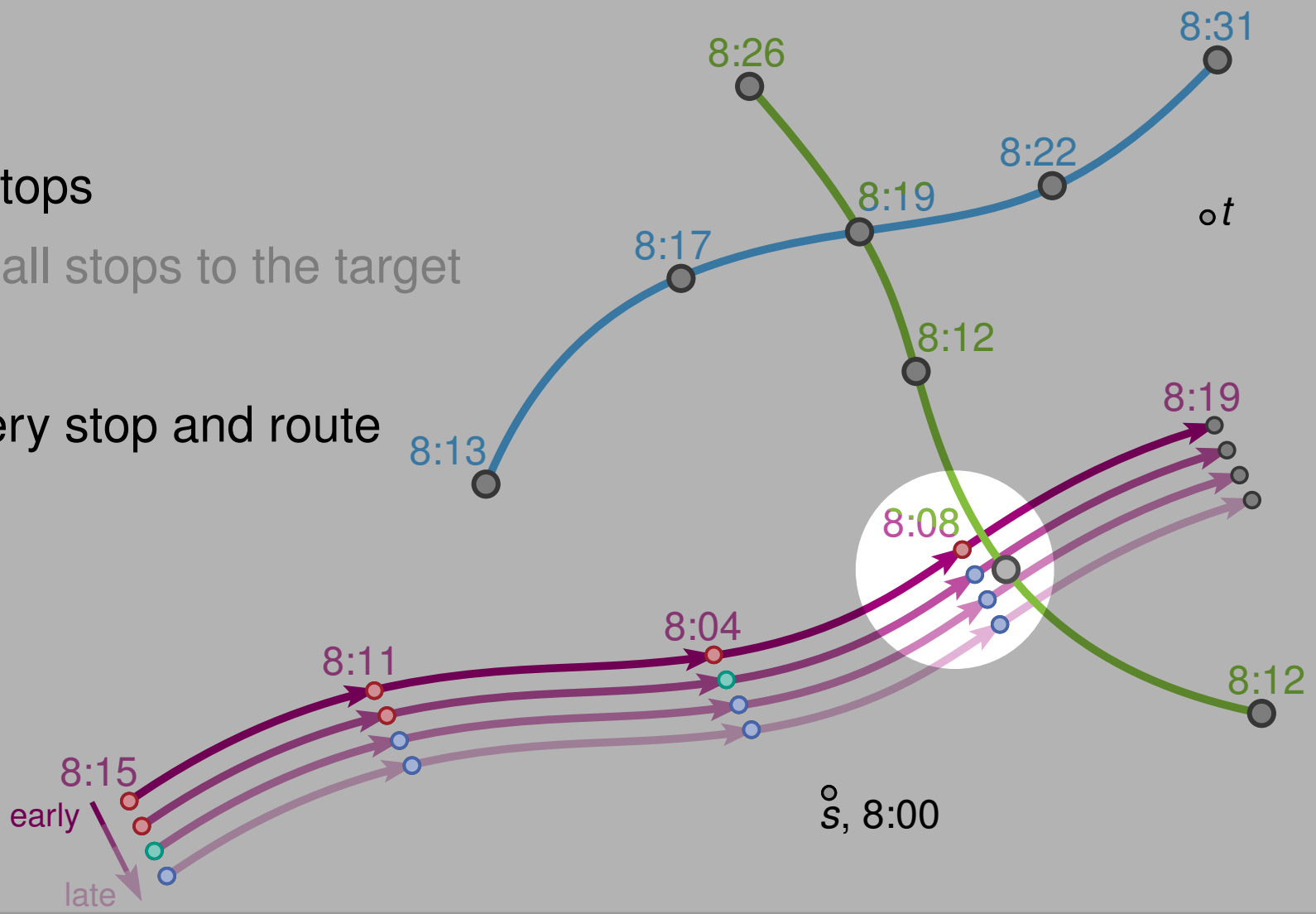


Institute of Theoretical Informatics
Algorithmics Group

Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target
2. RAPTOR collect routes
   - Find the first trip for every stop and route

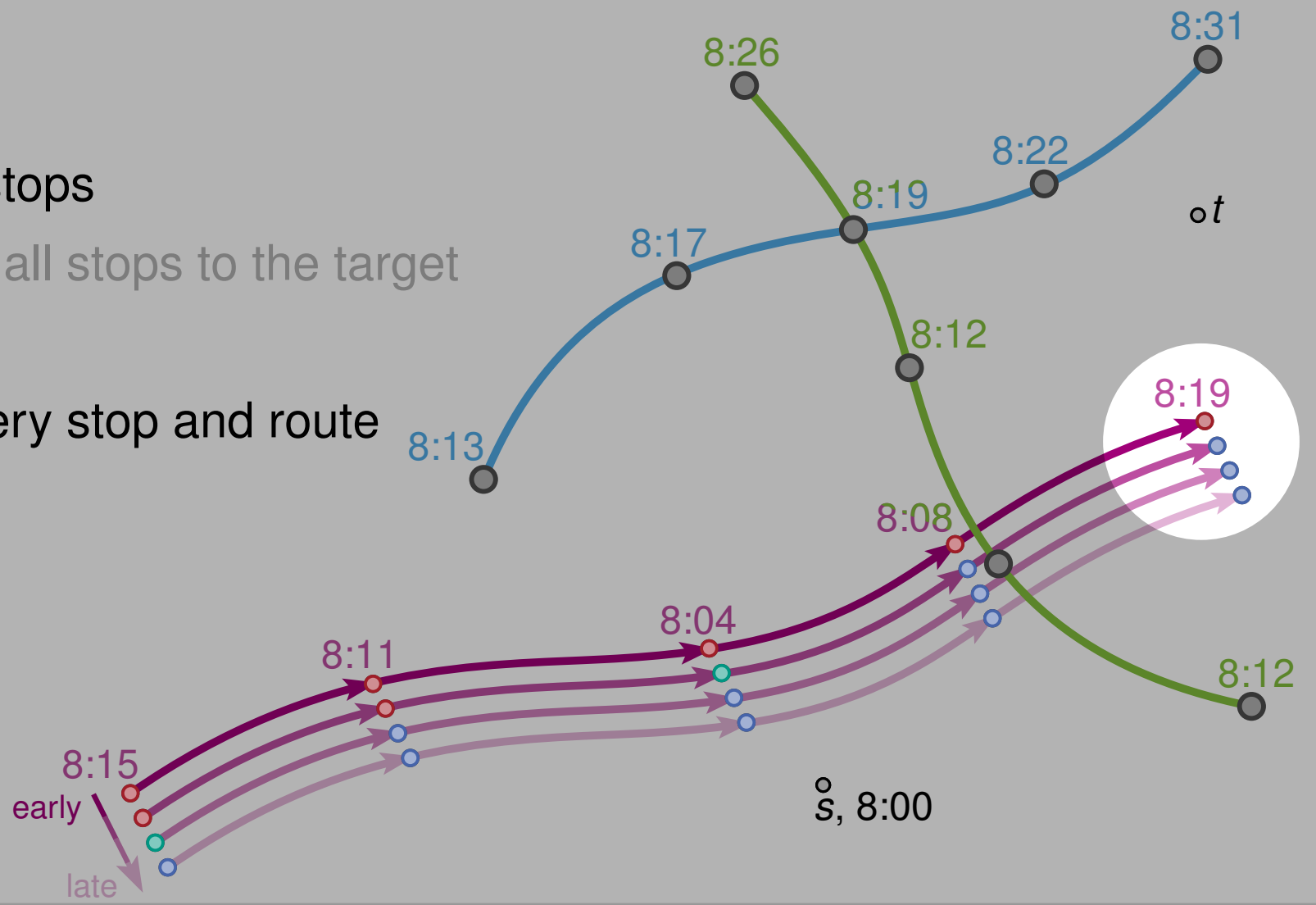


Institute of Theoretical Informatics
Algorithmics Group

Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target
2. RAPTOR collect routes
   - Find the first trip for every stop and route

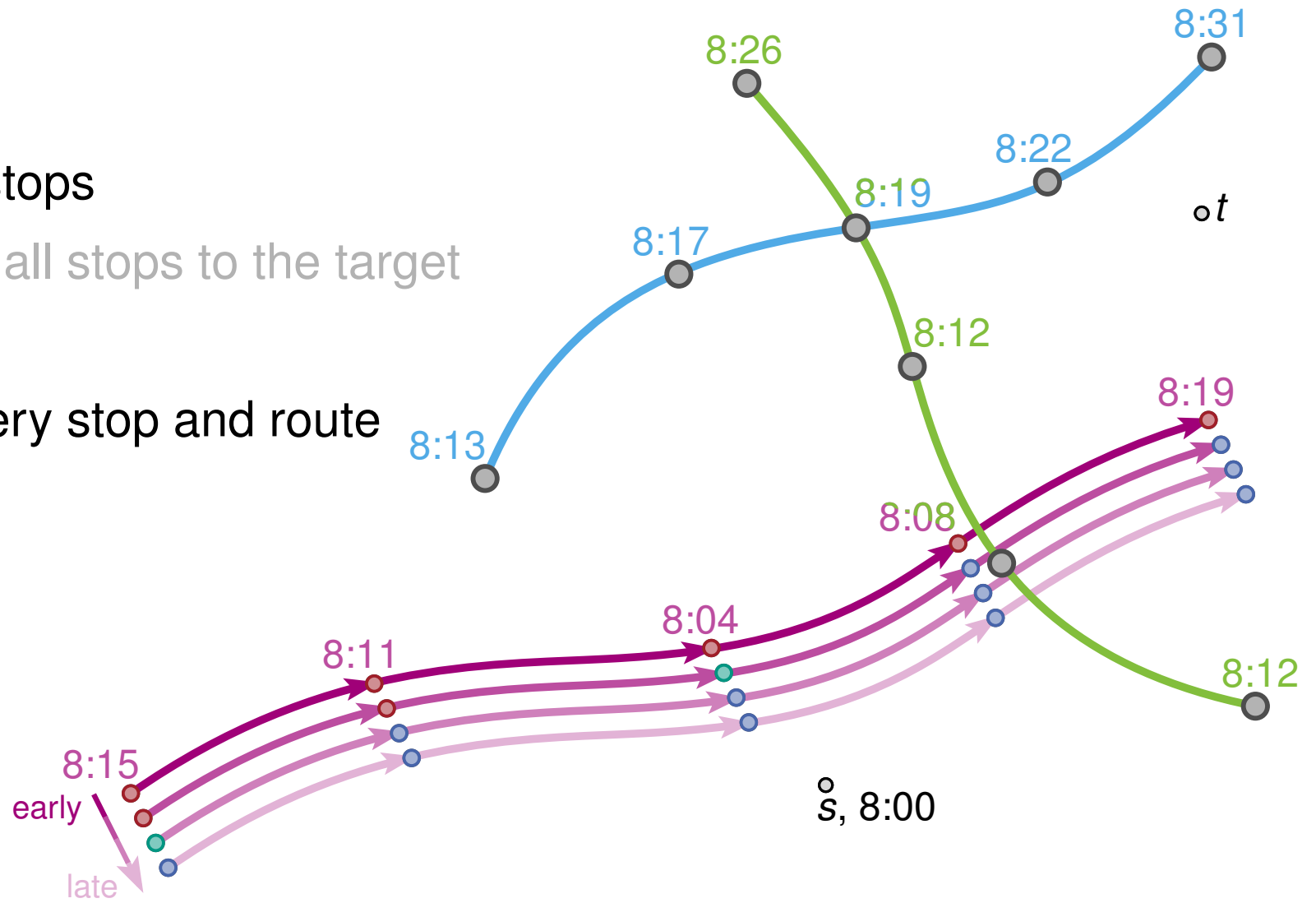


Institute of Theoretical Informatics
Algorithmics Group

Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target
2. RAPTOR collect routes
   - Find the first trip for every stop and route

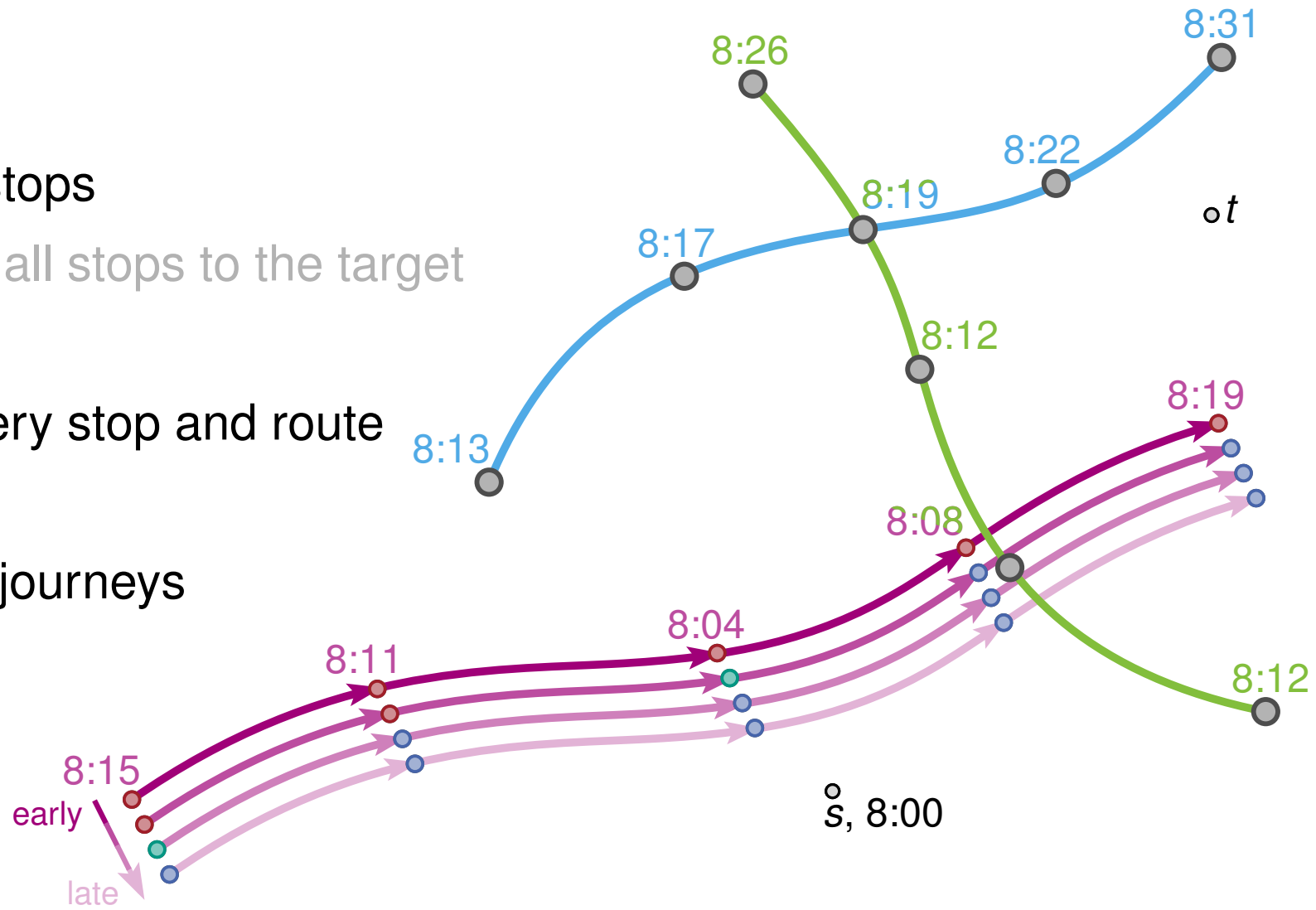


Institute of Theoretical Informatics
Algorithmics Group

Integrated ULTRA-Trip-Based Query

Query Algorithm Outline:

1. Bucket-CH query
   - Find arrival time at all stops
   - Find transfer time from all stops to the target
2. RAPTOR collect routes
   - Find the first trip for every stop and route
3. Trip-Based trip scanning
   - Find all Pareto optimal journeys



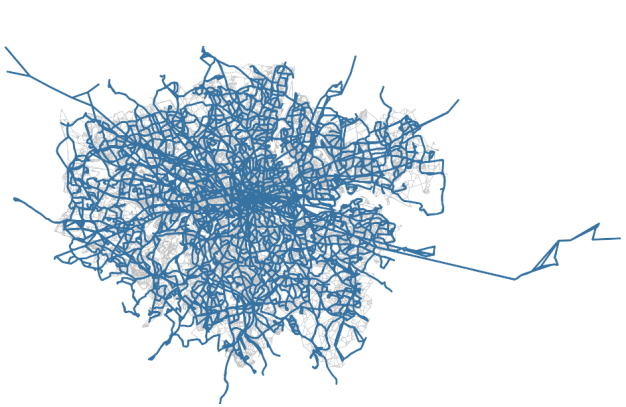
Institute of Theoretical Informatics
Algorithmics Group

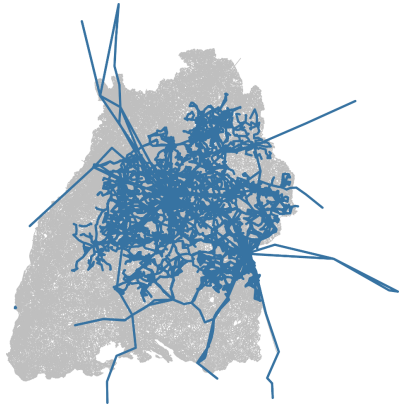
Experimental Evaluation

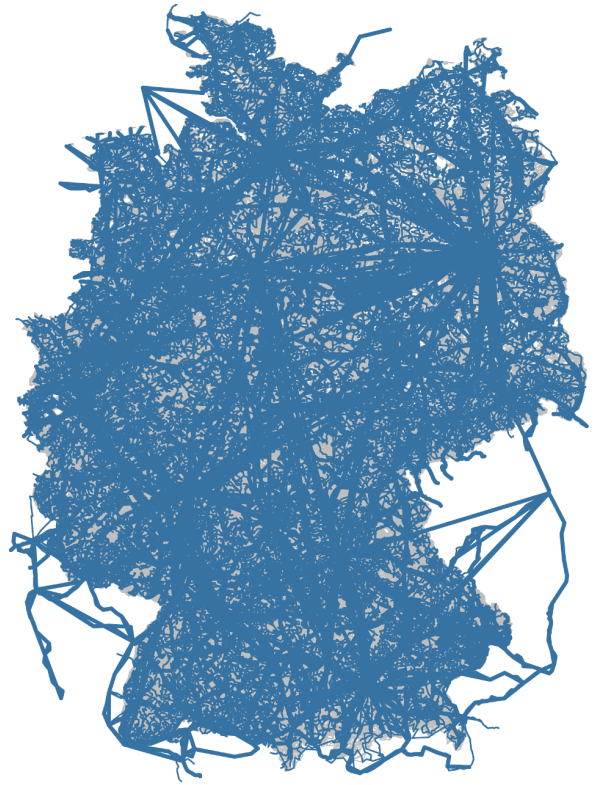
Instances:

- London, Stuttgart, Switzerland, and Germany
- Timetables comprising two days from TfL, GTFS-CH, and DB

Network	Stops	Routes	Trips	Vertices	Edges
London	20 595	2 107	125 436	183 k	579 k
Stuttgart	13 583	12 350	91 298	1 166 k	3 680 k
Switzerland	25 426	13 934	369 006	604 k	1 847 k
Germany	244 055	231 089	2 387 292	6 872 k	21 372 k







Experimental Evaluation – Preprocessing

Comparing Sequential and Integrated Preprocessing:

- Using ULTRA shortcuts as input for Trip-Based Routing is a bit faster
- The Integrated approach yields significantly fewer shortcuts

	Stuttgart	London	Switzerland	Germany
Time (sequential)	4:40	19:15	9:16	7:54:13
Time (integrated)	5:11	22:24	10:04	9:16:15
Shortcuts (sequential)	25 865 892	58 301 120	58 807 528	1 072 750 574
Shortcuts (integrated)	3 900 258	19 856 062	11 646 572	121 676 520

Experimental Evaluation – Preprocessing

Comparing Sequential and Integrated Preprocessing:

- Using ULTRA shortcuts as input for Trip-Based Routing is a bit faster
- The Integrated approach yields significantly fewer shortcuts

	Stuttgart	London	Switzerland	Germany
Time (sequential)	4:40	19:15	9:16	7:54:13
Time (integrated)	5:11	22:24	10:04	9:16:15
Increase	1.11	1.16	1.09	1.17
Shortcuts (sequential)	25 865 892	58 301 120	58 807 528	1 072 750 574
Shortcuts (integrated)	3 900 258	19 856 062	11 646 572	121 676 520

Experimental Evaluation – Preprocessing

Comparing Sequential and Integrated Preprocessing:

- Using ULTRA shortcuts as input for Trip-Based Routing is a bit faster
- The Integrated approach yields significantly fewer shortcuts

	Stuttgart	London	Switzerland	Germany
Time (sequential)	4:40	19:15	9:16	7:54:13
Time (integrated)	5:11	22:24	10:04	9:16:15
Increase	1.11	1.16	1.09	1.17
Shortcuts (sequential)	25 865 892	58 301 120	58 807 528	1 072 750 574
Shortcuts (integrated)	3 900 258	19 856 062	11 646 572	121 676 520
Reduction	6.63	2.94	5.05	8.82

Experimental Evaluation – Query

Query Overview for London and Germany:

- ULTRA-TB (int.) is the fastest algorithm (even faster than TB on a transitive graph)

Network	Algorithm	Full graph	Scans [k]		Time [ms]			
			Trips	Shortcuts	B-CH	R Collect	TB Scan	Total
London	Trip-Based*	○	22.75	1 376.26	0.01	0.05	6.10	6.16
	ULTRA-TB (seq.)	●	34.09	1 545.15	0.91	0.80	7.47	9.19
	ULTRA-TB (int.)	●	24.69	450.50	0.90	0.70	4.05	5.66
	ULTRA-RAPTOR	●	–	–	0.93	–	–	7.55
Germany	Trip-Based*	○	337.49	16 116.64	0.01	0.05	116.14	116.21
	ULTRA-TB (seq.)	●	439.35	38 092.34	25.34	18.96	151.35	195.67
	ULTRA-TB (int.)	●	204.23	3 149.87	26.12	19.13	46.38	91.65
	ULTRA-RAPTOR	●	–	–	25.68	–	–	415.17

Experimental Evaluation – Query

Query Overview for London and Germany:

- ULTRA-TB (int.) is the fastest algorithm (even faster than TB on a transitive graph)

Network	Algorithm	Full graph	Scans [k]		Time [ms]			
			Trips	Shortcuts	B-CH	R Collect	TB Scan	Total
London	Trip-Based*	○	22.75	1 376.26	0.01	0.05	6.10	6.16
	ULTRA-TB (seq.)	●	34.09	1 545.15	0.91	0.80	7.47	9.19
	ULTRA-TB (int.)	●	24.69	450.50	0.90	0.70	4.05	5.66
	ULTRA-RAPTOR	●	–	–	0.93	–	–	7.55
Germany	Trip-Based*	○	337.49	16 116.64	0.01	0.05	116.14	116.21
	ULTRA-TB (seq.)	●	439.35	38 092.34	25.34	18.96	151.35	195.67
	ULTRA-TB (int.)	●	204.23	3 149.87	26.12	19.13	46.38	91.65
	ULTRA-RAPTOR	●	–	–	25.68	–	–	415.17

Experimental Evaluation – Query

Query Overview for London and Germany:

- ULTRA-TB (int.) is the fastest algorithm (even faster than TB on a transitive graph)

Network	Algorithm	Full graph	Scans [k]		Time [ms]			
			Trips	Shortcuts	B-CH	R Collect	TB Scan	Total
London	Trip-Based*	○	22.75	1 376.26	0.01	0.05	6.10	6.16
	ULTRA-TB (seq.)	●	34.09	1 545.15	0.91	0.80	7.47	9.19
	ULTRA-TB (int.)	●	24.69	450.50	0.90	0.70	4.05	5.66
	ULTRA-RAPTOR	●	–	–	0.93	–	–	7.55
Germany	Trip-Based*	○	337.49	16 116.64	0.01	0.05	116.14	116.21
	ULTRA-TB (seq.)	●	439.35	38 092.34	25.34	18.96	151.35	195.67
	ULTRA-TB (int.)	●	204.23	3 149.87	26.12	19.13	46.38	91.65
	ULTRA-RAPTOR	●	–	–	25.68	–	–	415.17

Experimental Evaluation – Query

Query Overview for London and Germany:

- ULTRA-TB (int.) is the fastest algorithm (even faster than TB on a transitive graph)

Network	Algorithm	Full graph	Scans [k]		Time [ms]			
			Trips	Shortcuts	B-CH	R Collect	TB Scan	Total
London	Trip-Based*	○	22.75	1 376.26	0.01	0.05	6.10	6.16
	ULTRA-TB (seq.)	●	34.09	1 545.15	0.91	0.80	7.47	9.19
	ULTRA-TB (int.)	●	24.69	450.50	0.90	0.70	4.05	5.66
	ULTRA-RAPTOR	●	–	–	0.93	–	–	7.55
Germany	Trip-Based*	○	337.49	16 116.64	0.01	0.05	116.14	116.21
	ULTRA-TB (seq.)	●	439.35	38 092.34	25.34	18.96	151.35	195.67
	ULTRA-TB (int.)	●	204.23	3 149.87	26.12	19.13	46.38	91.65
	ULTRA-RAPTOR	●	–	–	25.68	–	–	415.17

Experimental Evaluation – Query

Query Overview for London and Germany:

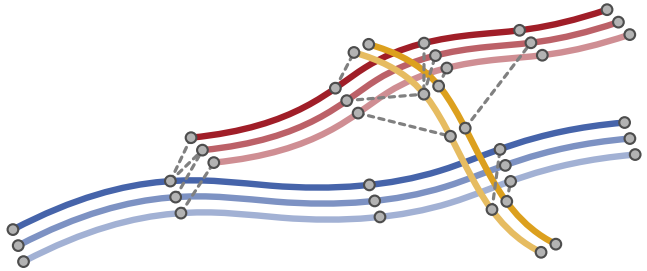
- ULTRA-TB (int.) is the fastest algorithm (even faster than TB on a transitive graph)

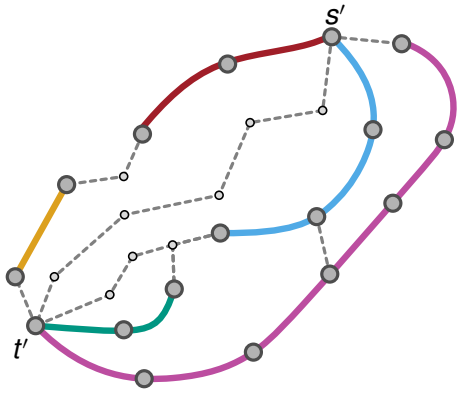
Network	Algorithm	Full graph	Scans [k]		Time [ms]			
			Trips	Shortcuts	B-CH	R Collect	TB Scan	Total
London	Trip-Based*	○	22.75	1 376.26	0.01	0.05	6.10	6.16
	ULTRA-TB (seq.)	●	34.09	1 545.15	0.91	0.80	7.47	9.19
	ULTRA-TB (int.)	●	24.69	450.50	0.90	0.70	4.05	5.66
	ULTRA-RAPTOR	●	–	–	0.93	–	–	7.55
Germany	Trip-Based*	○	337.49	16 116.64	0.01	0.05	116.14	116.21
	ULTRA-TB (seq.)	●	439.35	38 092.34	25.34	18.96	151.35	195.67
	ULTRA-TB (int.)	●	204.23	3 149.87	26.12	19.13	46.38	91.65
	ULTRA-RAPTOR	●	–	–	25.68	–	–	415.17

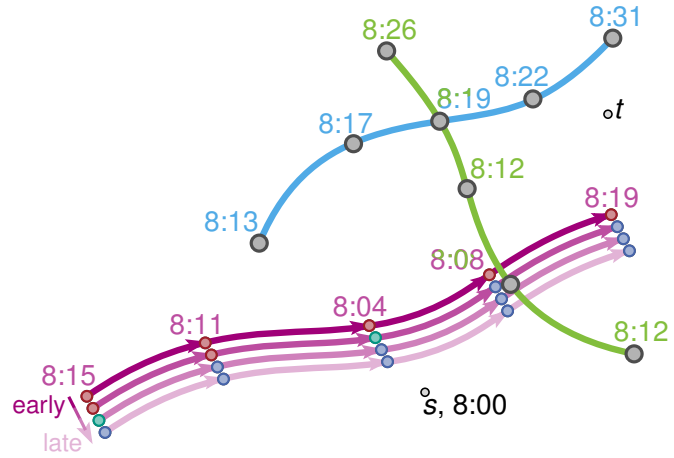
Conclusion

Our Contribution:

- We proposed the ULTRA-Trip-Based algorithm
  - About 4 times faster than best previous algorithm
  - Trip-Based shortcuts
  - ULTRA journey enumeration
  - Integrated query



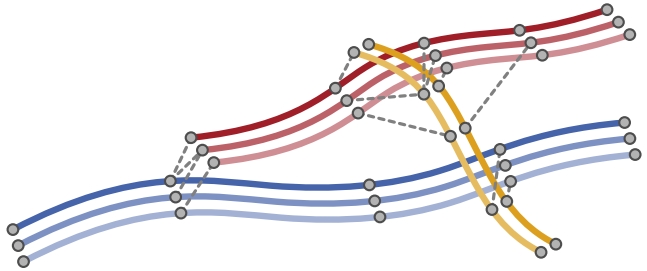


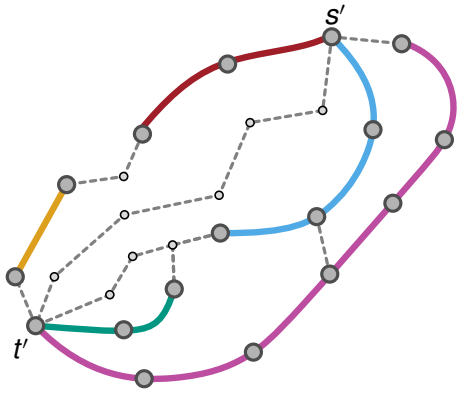


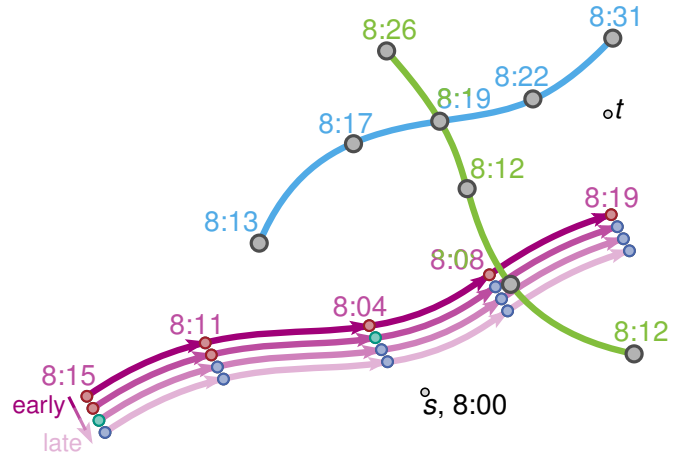
Conclusion

Our Contribution:

- We proposed the ULTRA-Trip-Based algorithm
  - About 4 times faster than best previous algorithm
  - Trip-Based shortcuts
  - ULTRA journey enumeration
  - Integrated query







Future Work:

- Optimize more criteria
- Handle delays

Institute of Theoretical Informatics
Algorithmics Group