
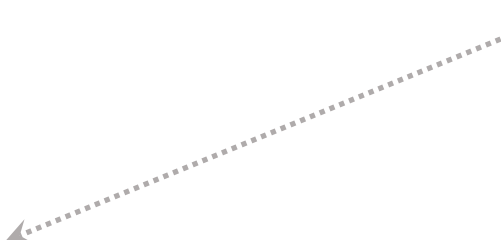
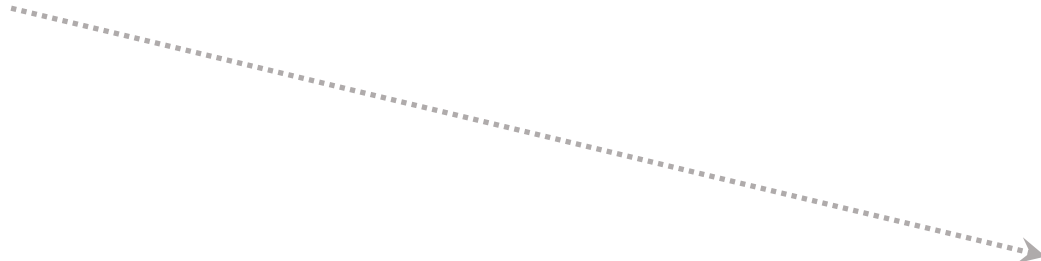



GeoPrune: Efficiently Matching Trips in Ride-sharing Through Geometric Properties

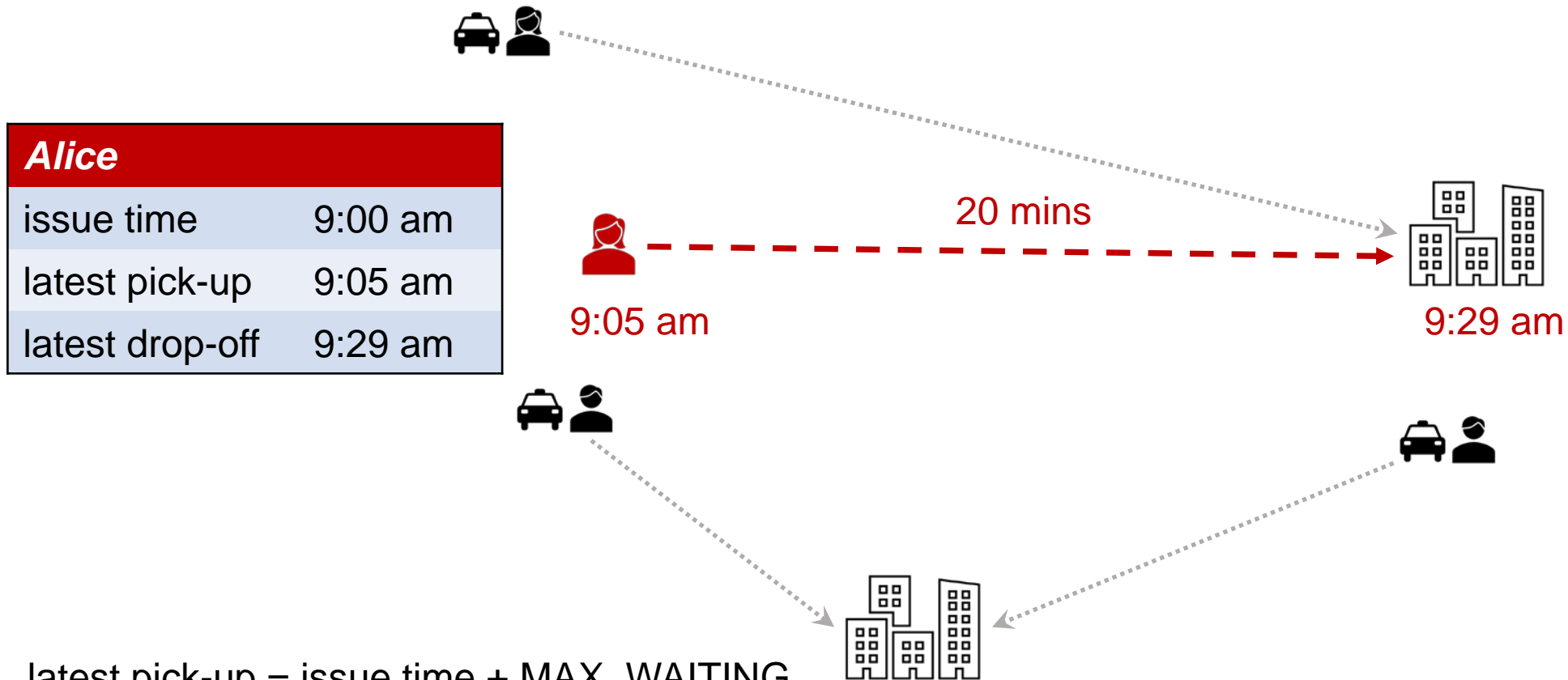
Yixin Xu, Jianzhong Qi, Renata Borovica-Gajic, Lars Kulik
The University of Melbourne

 9:00 am




 9:00 am

Ride-sharing matching: dispatch a feasible and optimal vehicle to Alice

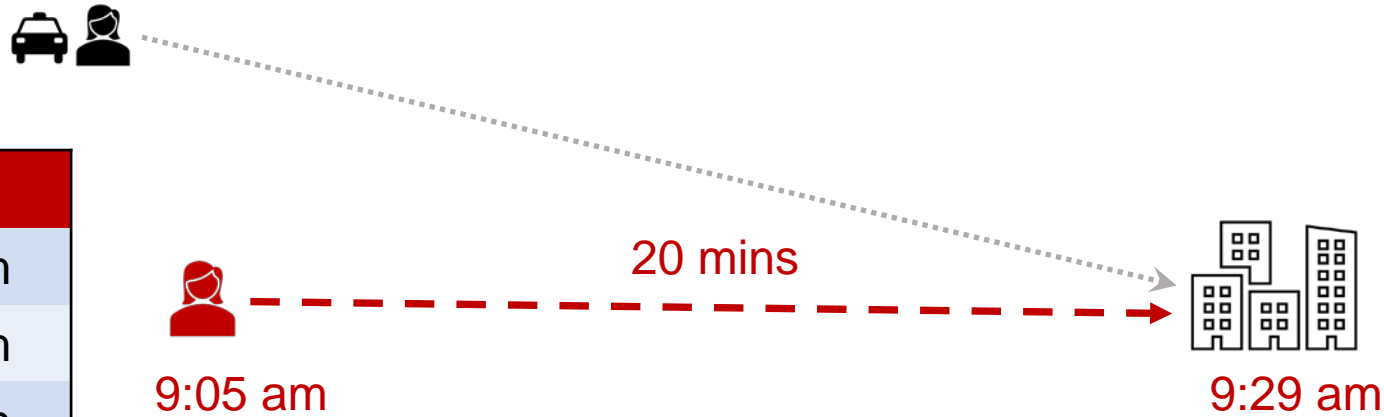


latest pick-up = issue time + MAX_WAITING

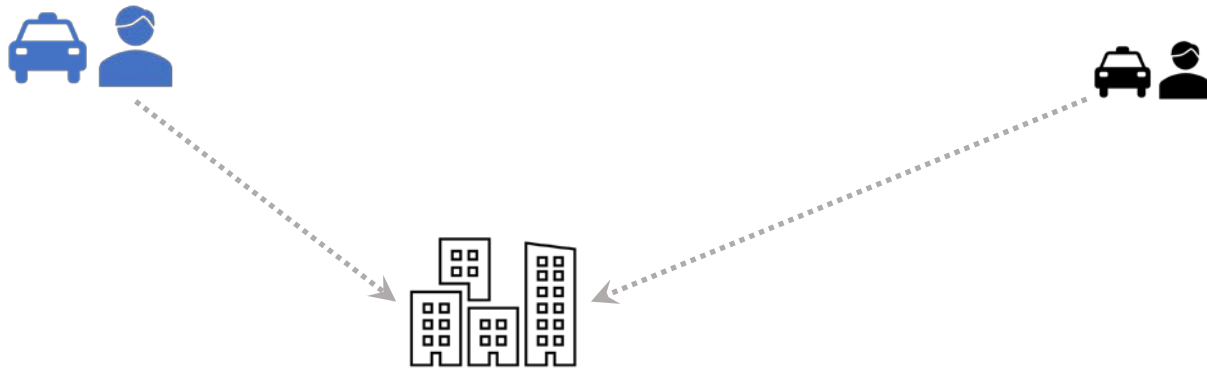
latest drop-off = latest pick-up + SP_TIME * (1 + DETOUR_RATIO)


 9:00 am

Ride-sharing matching: dispatch a feasible and optimal vehicle to Alice



<i>Alice</i>	
issue time	9:00 am
latest pick-up	9:05 am
latest drop-off	9:29 am



 9:00 am

Ride-sharing matching: dispatch a feasible and optimal vehicle to Alice



Alice	
issue time	9:00 am
latest pick-up	9:05 am
latest drop-off	9:29 am



9:05 am

20 mins




9:29 am



15 mins



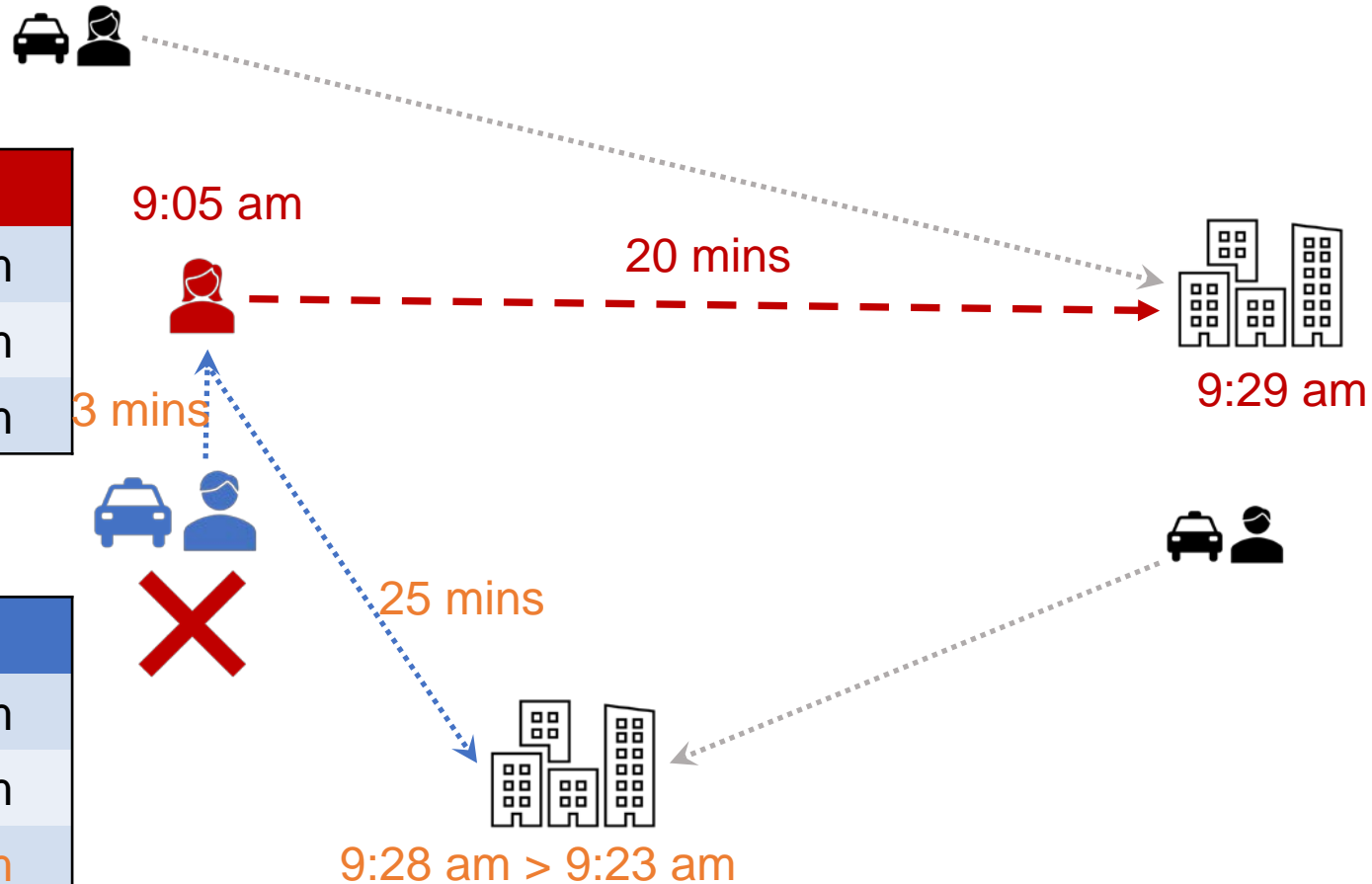
Bob	
issue time	9:00 am
latest pick-up	9:05 am
latest drop-off	9:23 am

 9:00 am

Ride-sharing matching: dispatch a feasible and optimal vehicle to Alice

Alice	
issue time	9:00 am
latest pick-up	9:05 am
latest drop-off	9:29 am

Bob	
issue time	9:00 am
latest pick-up	9:05 am
latest drop-off	9:23 am



Framework of existing ride-sharing algorithms

➤ Pruning

- Filter out infeasible vehicles

➤ Selection

- Select the optimal match among a few remaining vehicles

Our research focus: prune infeasible vehicles efficiently

Framework of existing ride-sharing algorithms

➤ Pruning

- Filter out infeasible vehicles

➤ Selection

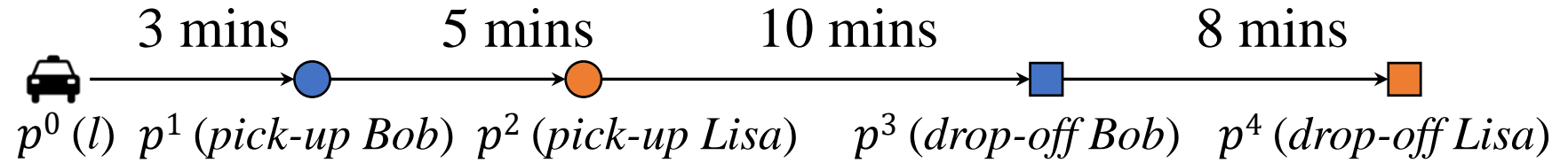
- Select the optimal match among a few remaining vehicles

Infeasible vehicles: violate the time constraint of the new request OR vehicles (committed requests)

Time constraints of a vehicle

Bob	
issue time	9:00 am
latest pick-up	9:05 am
latest drop-off	9:23 am

Lisa	
issue time	9:07 am
latest pick-up	9:12 am
latest drop-off	9:30 am

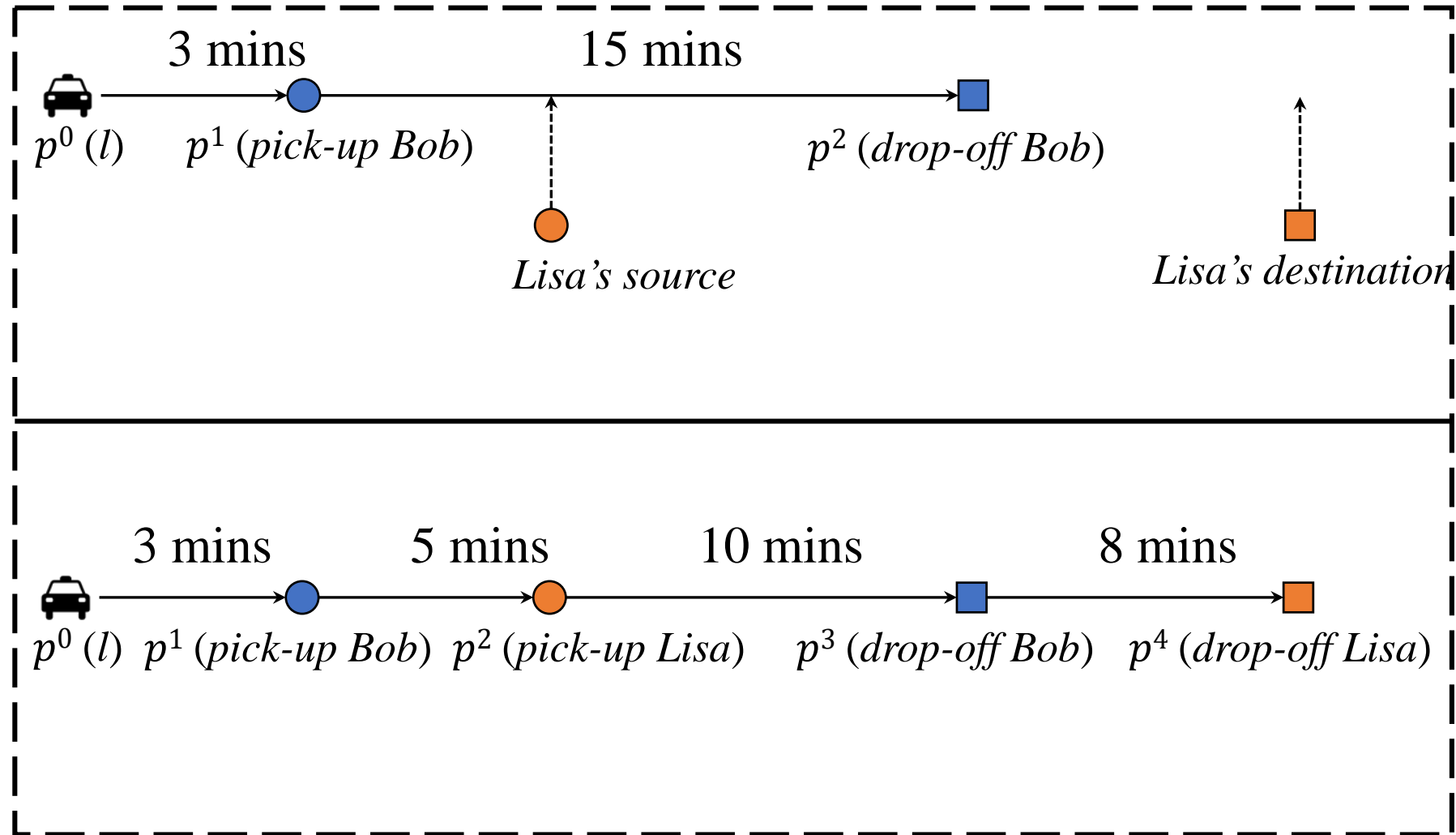


	p_0	p_1	p_2	p_3	p_4
Est arrival (Arr)	9:00 am	9:03 am	9:08 am	9:18 am	9:26 am
Lat arrival (Lat)	9:00 am	9:05 am	9:12 am	9:23 am	9:30 am

Vehicle routing -- insertion

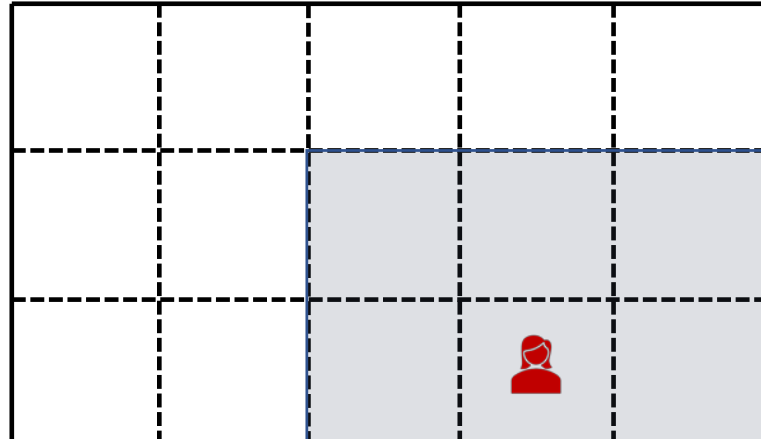
Bob	
issue time	9:00 am
latest pick-up	9:05 am
latest drop-off	9:23 am

Lisa	
issue time	9:07 am
latest pick-up	9:12 am
latest drop-off	9:30 am



State-of-the-art

➤ GreedyGrids



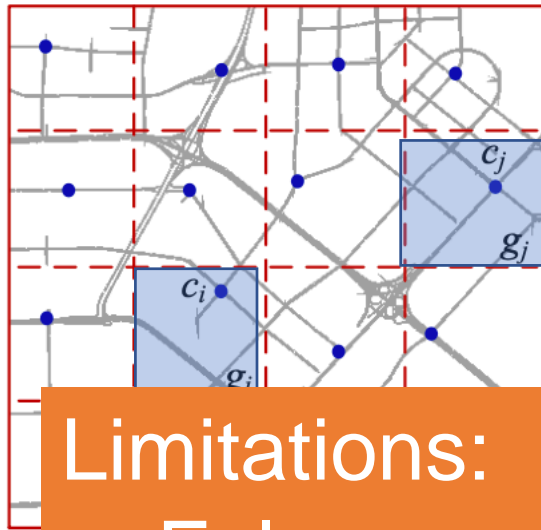
<i>Alice</i>	
issue time	9:00 am
latest pick-up	9:05 am
latest drop-off	9:29 am

Limitations:

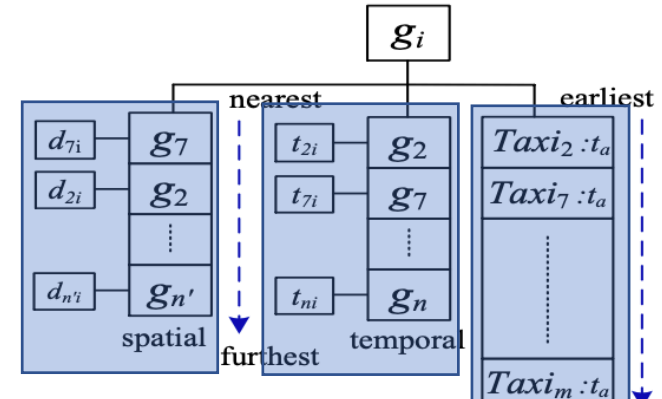
- Failed to consider the latest drop-off time of request
- Failed to consider the time constraints of vehicles

State-of-the-art

➤ T-share



$$M = \begin{matrix} & g_0 & g_1 & \dots & g_j & \dots & g_n \\ g_0 & \phi & D_{01} & \dots & D_{0j} & \dots & D_{0n} \\ g_1 & D_{10} & \phi & \dots & D_{1j} & \dots & D_{1n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ g_i & D_{i0} & D_{i1} & \dots & D_{ij} & \dots & D_{in} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \end{matrix}$$

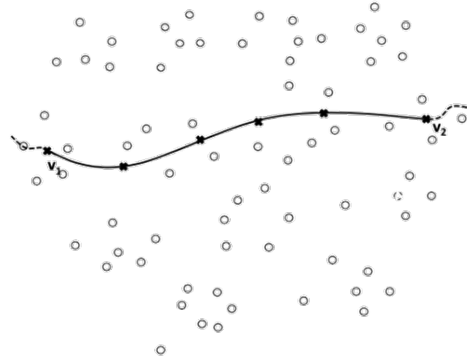


Limitations:

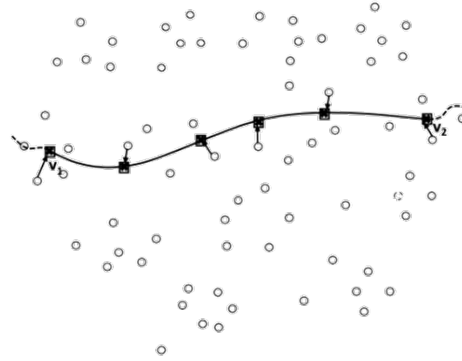
- False negatives
- Expensive update cost

State-of-the-art

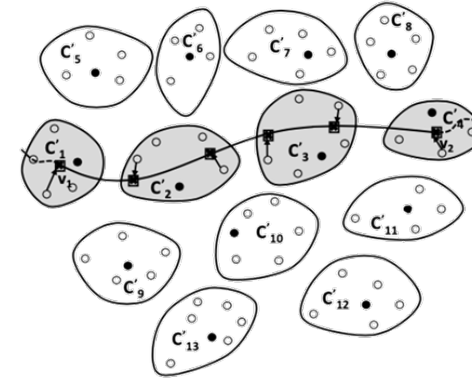
➤ Xhare



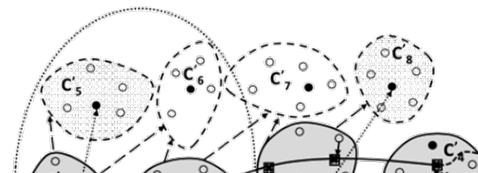
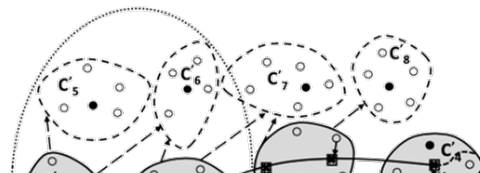
(a) Segment of a route (of a ride) represented as a set of way-points



(b) A few grids that a segment of a route (represented as a set of way-points) passes through and the mapping of those grids to nearby landmarks



(c) Clustering of landmarks; pass through clusters of the ride can be seen

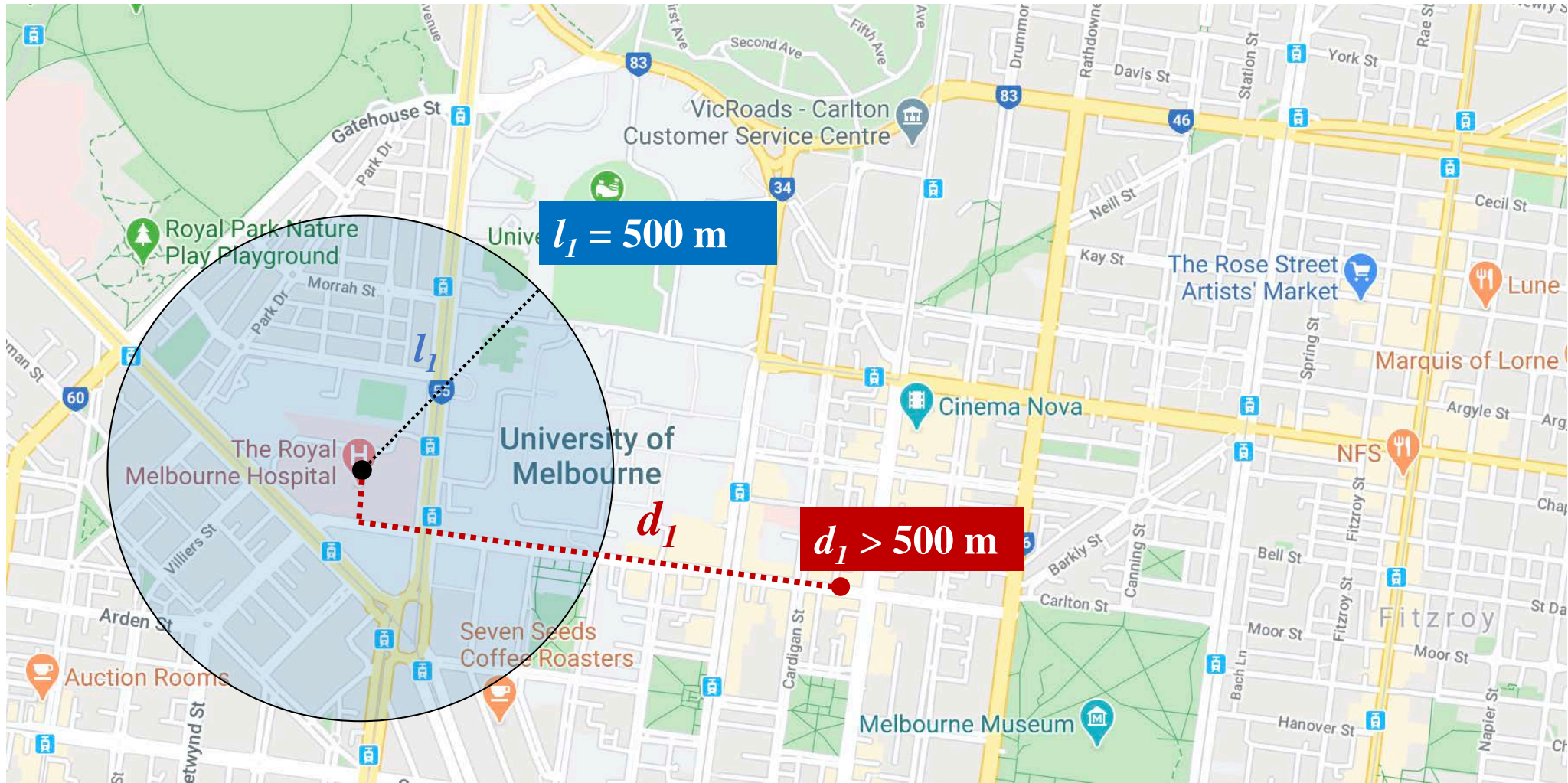


- ✱ Way-points
- v_1 Via-point of the beginning of the segment
- v_2 Via-point of the end of the segment
- Grids of way-points; other grids on the segment are not shown
- Centers of clusters
- Landmarks
- Clusters

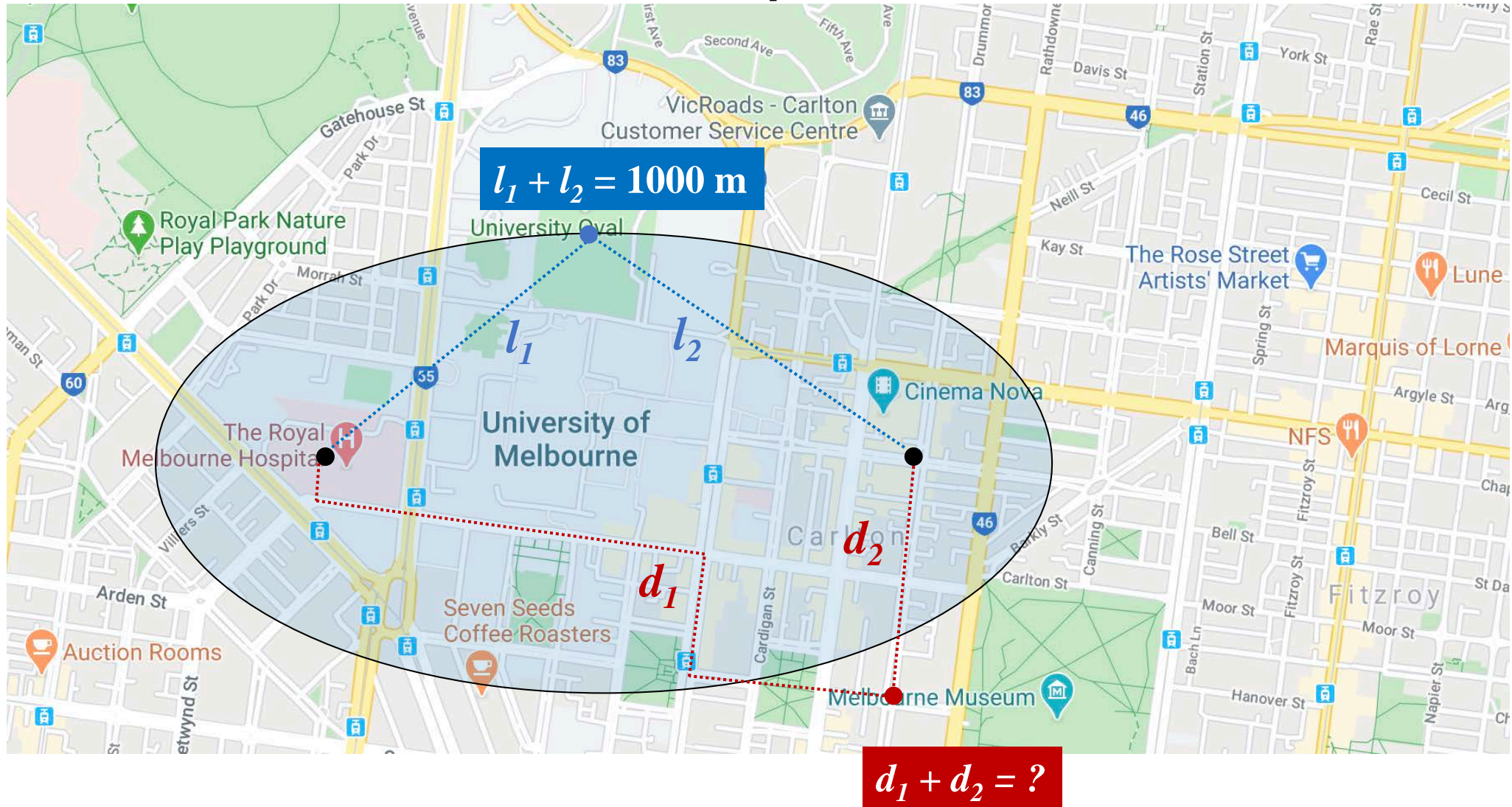
Limitations:

- False negatives
- Expensive update cost

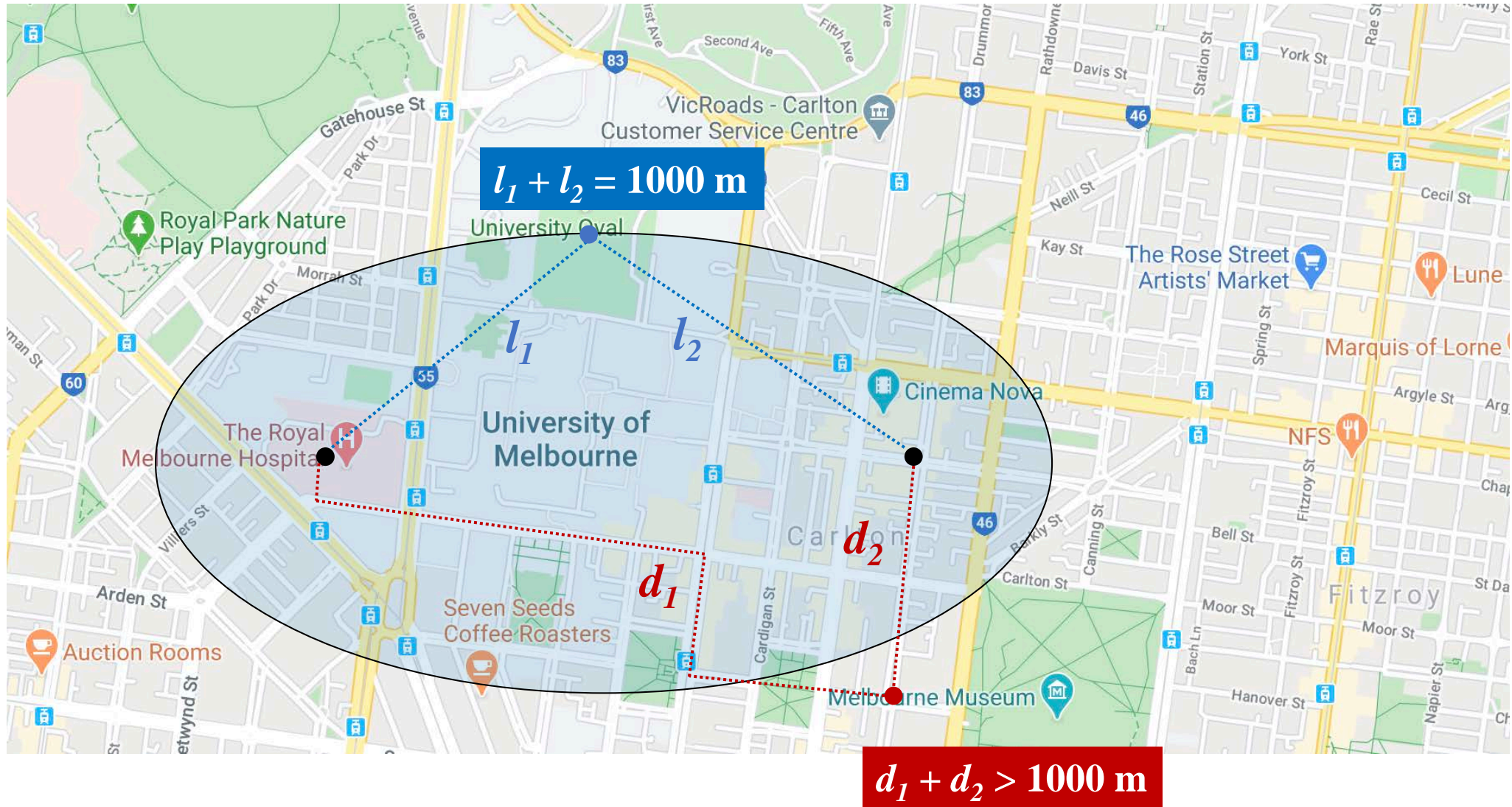
GeoPrune - waiting circle



GeoPrune – detour ellipse



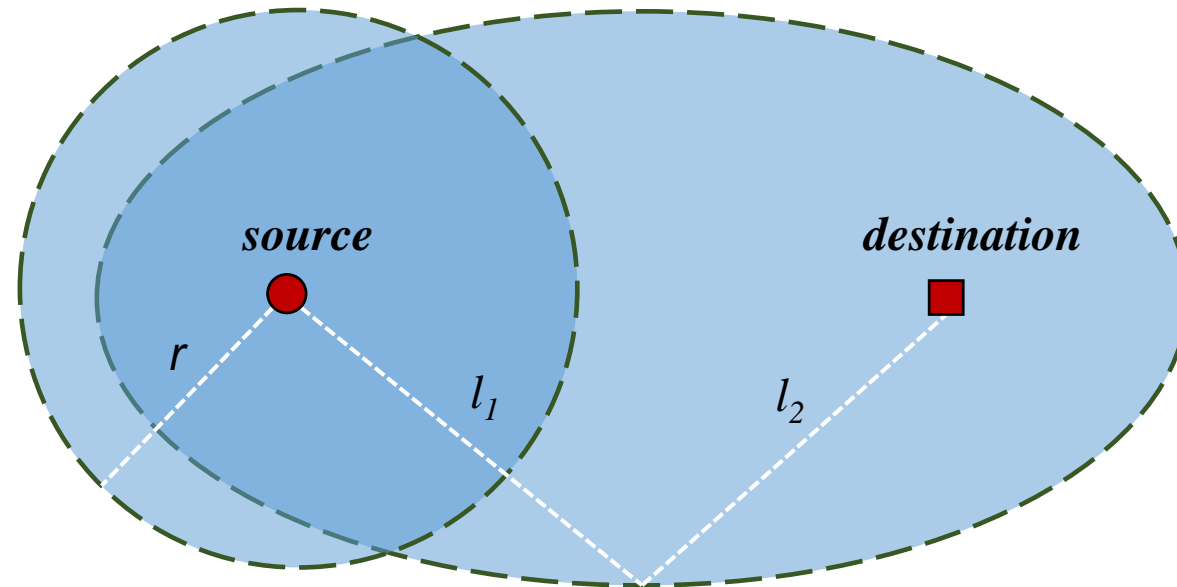
GeoPrune – detour ellipse



GeoPrune

- Request's time constraint

waiting circle
 $r = 5 \text{ mins}$



Alice

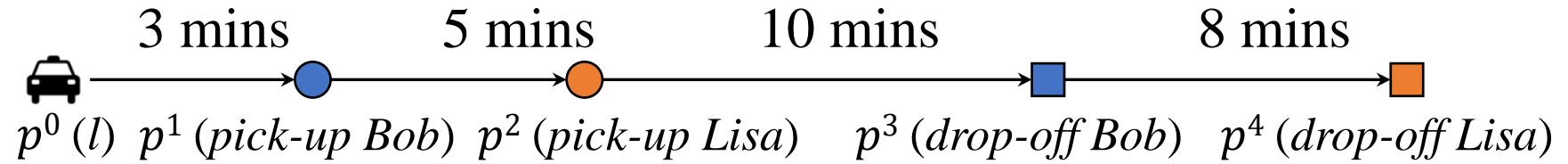
issue time	9:00 am
latest pick-up	9:05 am
latest drop-off	9:29 am

detour ellipse
 $l_1 + l_2 = 29 \text{ mins}$

Time constraints of a vehicle

Bob	
issue time	9:00 am
latest pick-up	9:05 am
latest drop-off	9:23 am

Lisa	
issue time	9:07 am
latest pick-up	9:12 am
latest drop-off	9:30 am

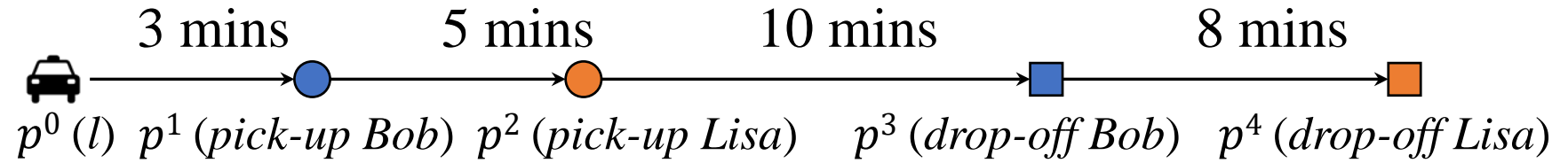


	p_0	p_1	p_2	p_3	p_4
Est arrival (Arr)	9:00 am	9:03 am	9:08 am	9:18 am	9:26 am
Lat arrival (Lat)	9:00 am	9:05 am	9:12 am	9:23 am	9:30 am

Time constraints of a vehicle

Bob	
issue time	9:00 am
latest pick-up	9:05 am
latest drop-off	9:23 am

Lisa	
issue time	9:07 am
latest pick-up	9:12 am
latest drop-off	9:30 am

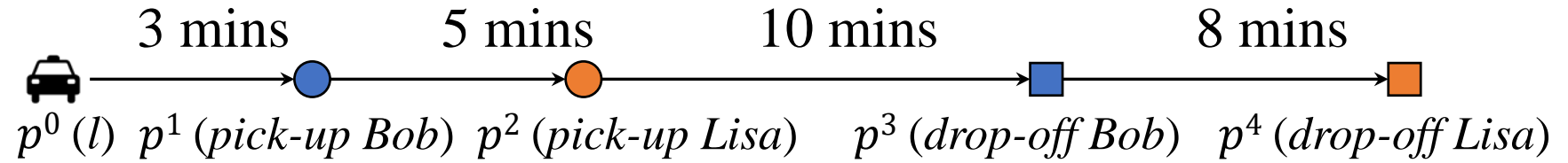


	p_0	p_1	p_2	p_3	p_4
Est arrival (Arr)	9:00 am	9:03 am	9:08 am	9:18 am	9:26 am
Lat arrival (Lat)	9:00 am	9:05 am	9:12 am	9:23 am	9:30 am
Lat - Arr	0 mins	2 mins	4 mins	5 mins	4 mins

Time constraints of a vehicle

Bob	
issue time	9:00 am
latest pick-up	9:05 am
latest drop-off	9:23 am

Lisa	
issue time	9:07 am
latest pick-up	9:12 am
latest drop-off	9:30 am

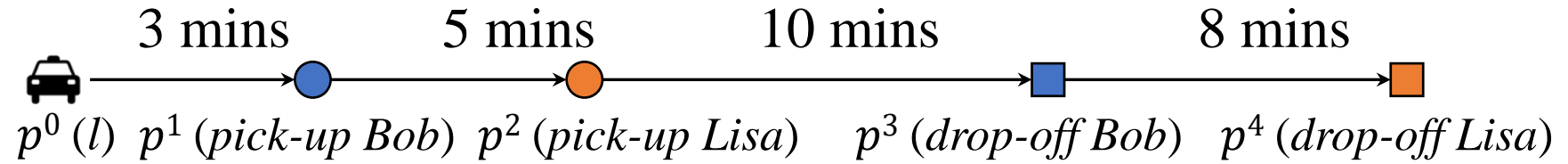


	p_0	p_1	p_2	p_3	p_4
Est arrival (Arr)	9:00 am	9:03 am	9:08 am	9:18 am	9:26 am
Lat arrival (Lat)	9:00 am	9:05 am	9:12 am	9:23 am	9:30 am
Lat - Arr	0 mins	2 mins	4 mins	5 mins	4 mins
Slack time	0 mins	2 mins	4 mins	4 mins	4 mins

Time constraints of a vehicle

Bob	
issue time	9:00 am
latest pick-up	9:05 am
latest drop-off	9:23 am

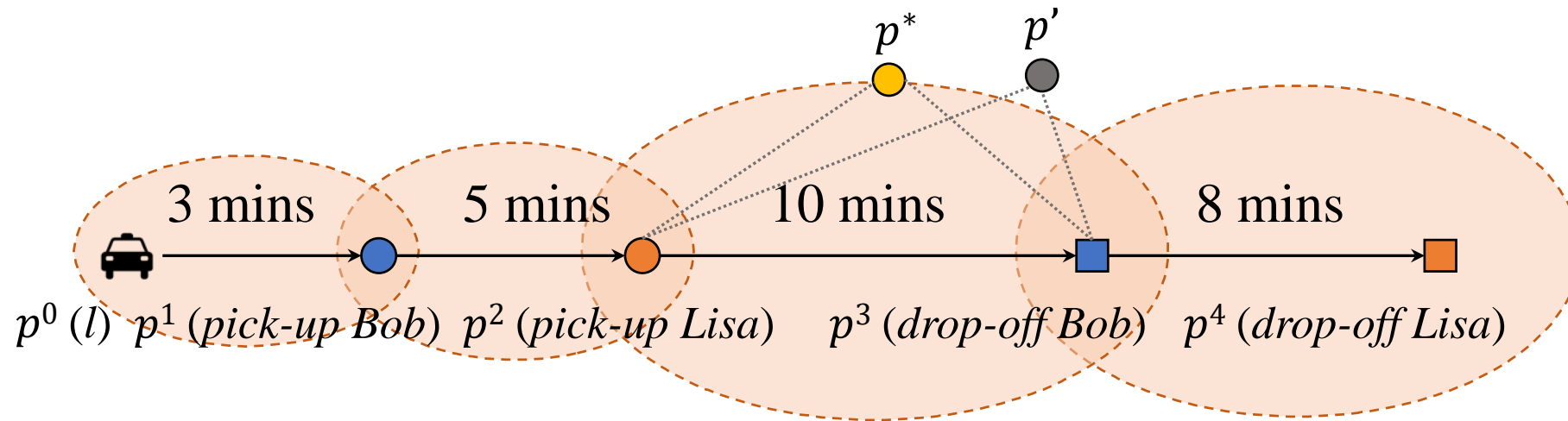
Lisa	
issue time	9:07 am
latest pick-up	9:12 am
latest drop-off	9:30 am



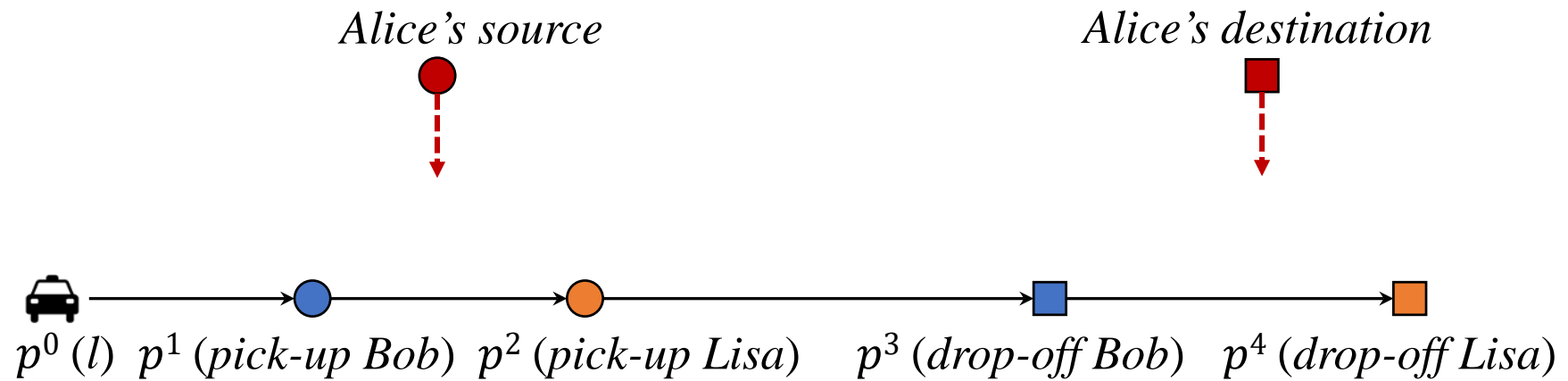
	p_0	p_1	p_2	p_3	p_4
Est arrival (Arr)	9:00 am	9:03 am	9:08 am	9:18 am	9:26 am
Lat arrival (Lat)	9:00 am	9:05 am	9:12 am	9:23 am	9:30 am
Lat - Arr	0 mins	2 mins	4 mins	5 mins	4 mins
Slack time	0 mins	2 mins	4 mins	4 mins	4 mins
Max time	--	5 mins	9 mins	14 mins	12 mins

GeoPrune

➤ Vehicle's time constraint

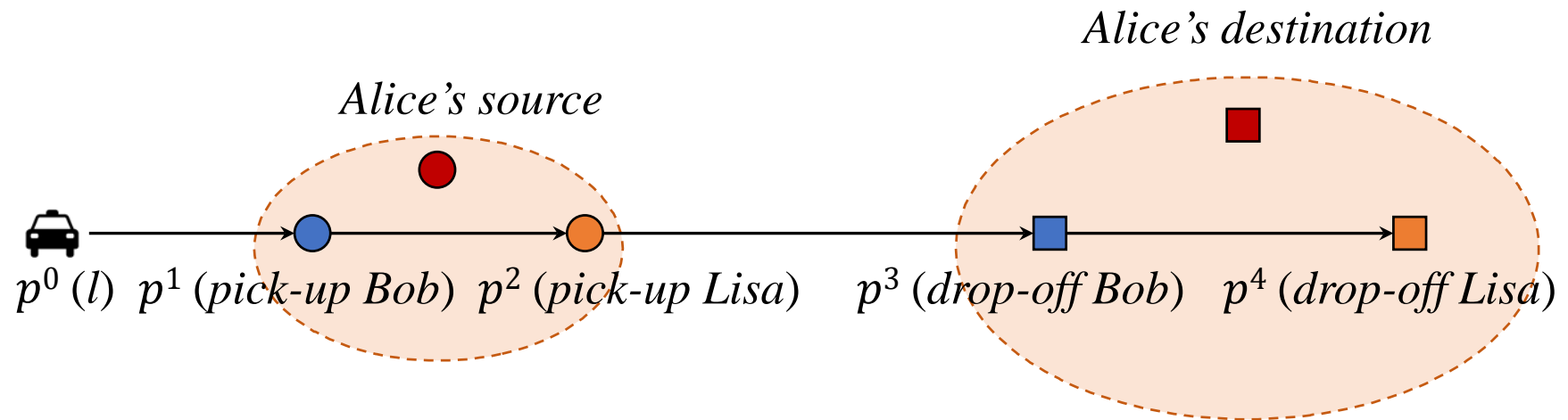


Insert-insert

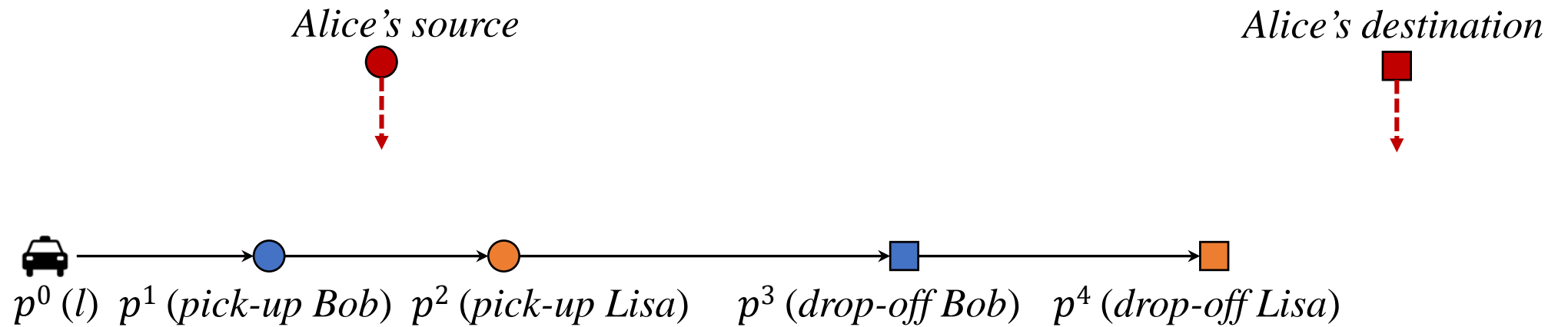


Insert-insert

One detour ellipse covers Alice's source
One detour ellipse covers Alice's destination

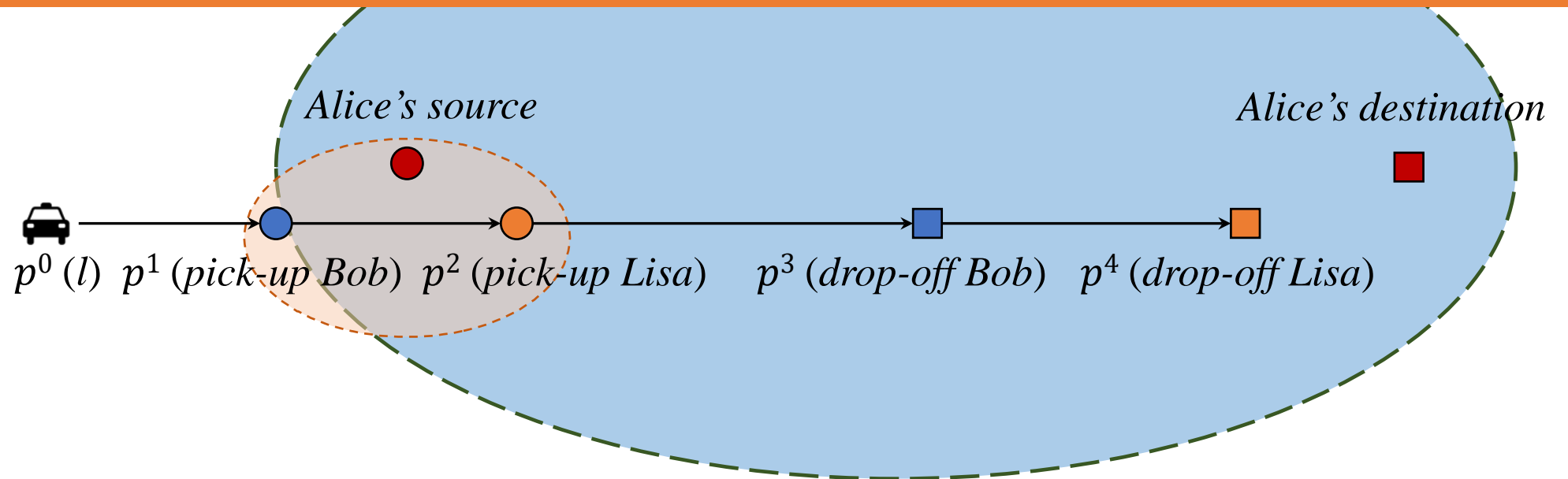


Insert-append

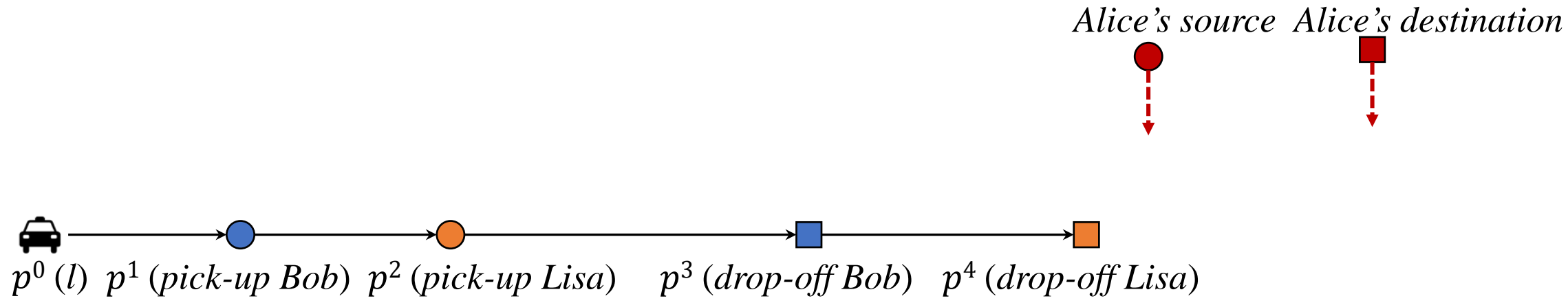


Insert-append

At least one detour ellipse covers Alice's source
Alice's detour ellipse covers the last stop

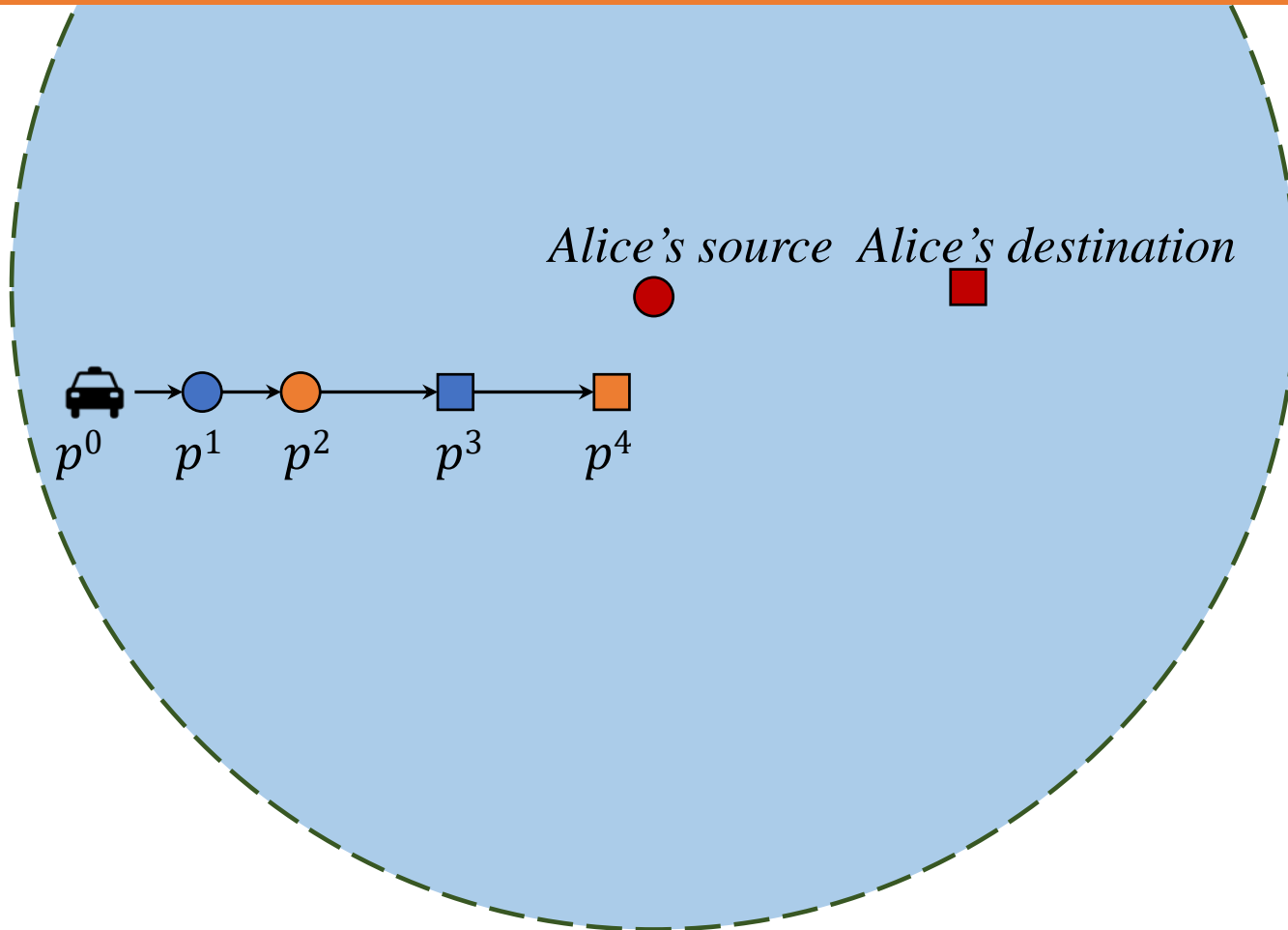


Append-append



Append-append

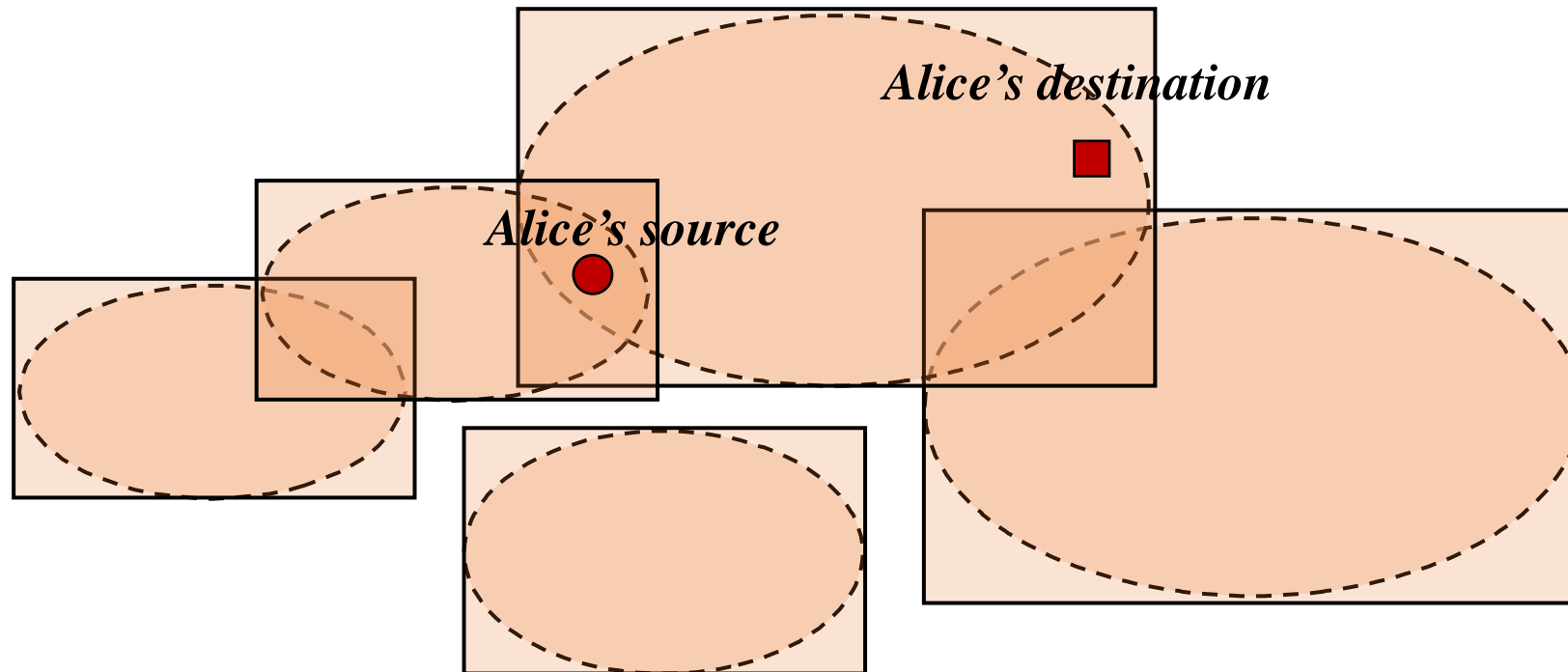
Alice's waiting circle covers the last stop



Vehicle candidates

➤ Insert-insert

- $T_{\text{seg}}.\text{pointQuery}(\text{source}) \cap T_{\text{seg}}.\text{pointQuery}(\text{destination})$



Vehicle candidates

- Insert-insert
 - $T_{\text{seg}}.\text{pointQuery}(\text{source}) \cap T_{\text{seg}}.\text{pointQuery}(\text{destination})$
- Insert-append
 - $T_{\text{seg}}.\text{pointQuery}(\text{source}) \cap T_{\text{end}}.\text{rangeQuery}(\text{ellipse}(\text{source}, \text{destination}))$
- Append-append
 - $T_{\text{end}}.\text{rangeQuery}(\text{circle}(\text{source}))$

Experimental settings

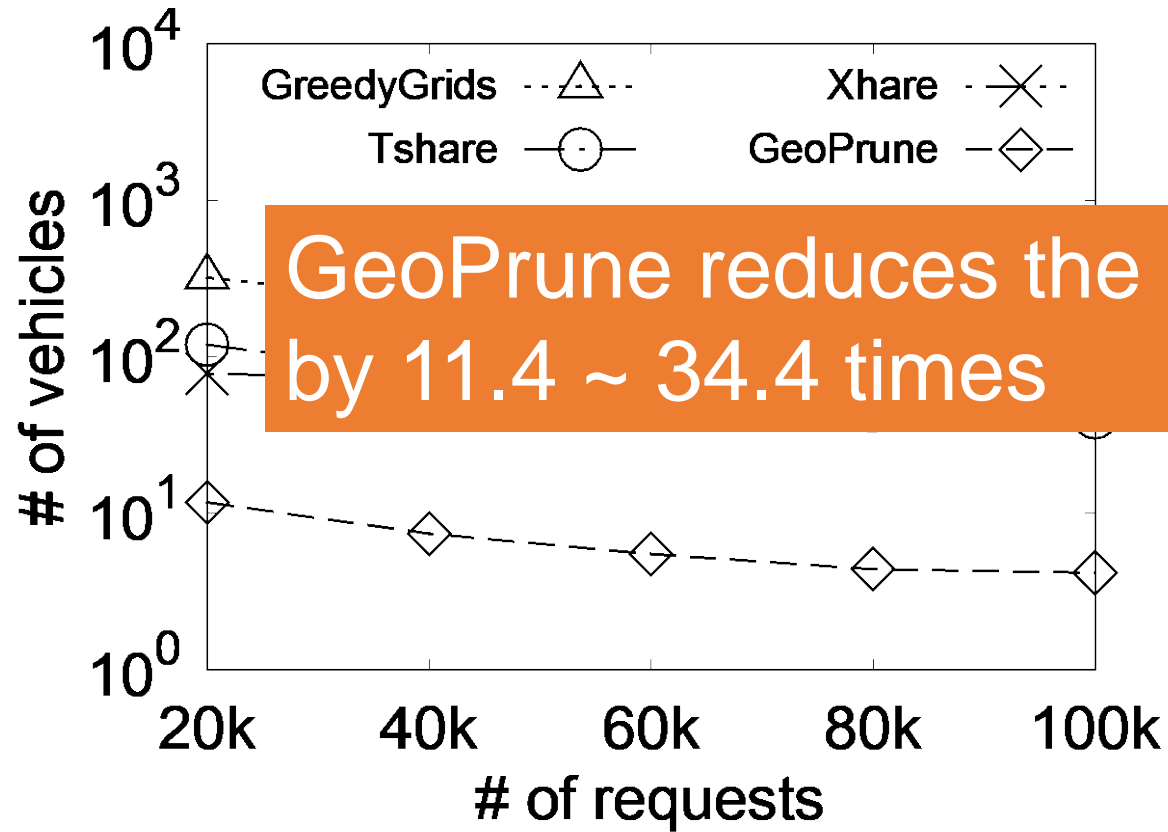
➤ **Two real-word datasets**

- *New York City (NYC): 254,423 nodes, 467,773 edges*
- Chengdu (CD): 166,296 nodes, 405,460 edges

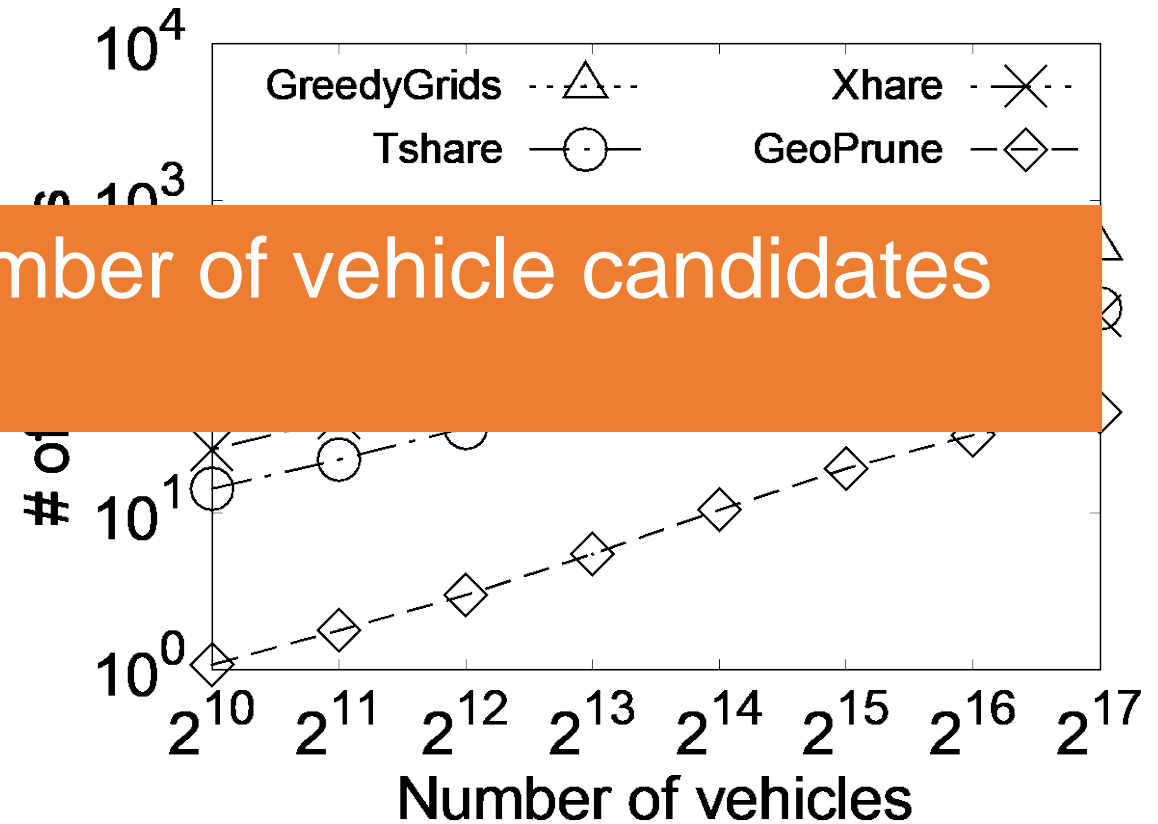
➤ **Default settings**

- # requests: 60000
- # vehicles: 8192
- Waiting time: 4 mins
- Detour ratio: 0.2
- Minimize the total travel distance

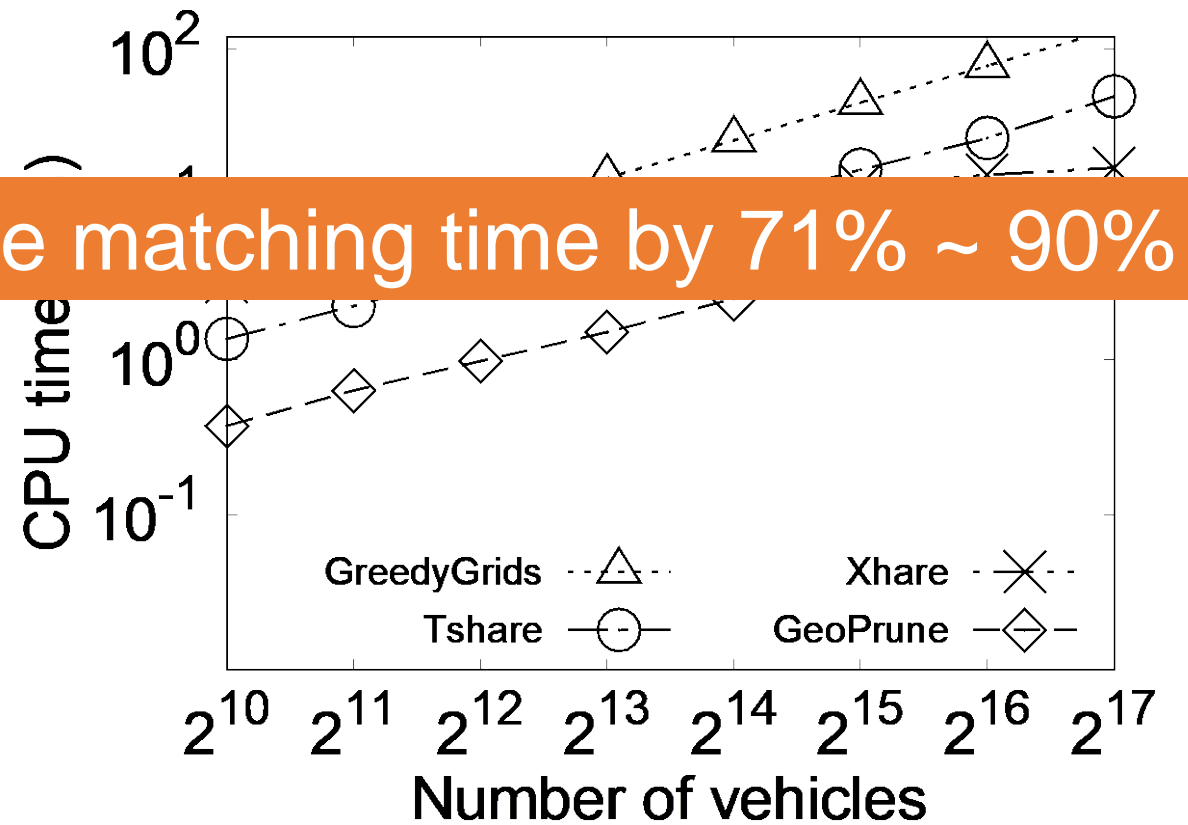
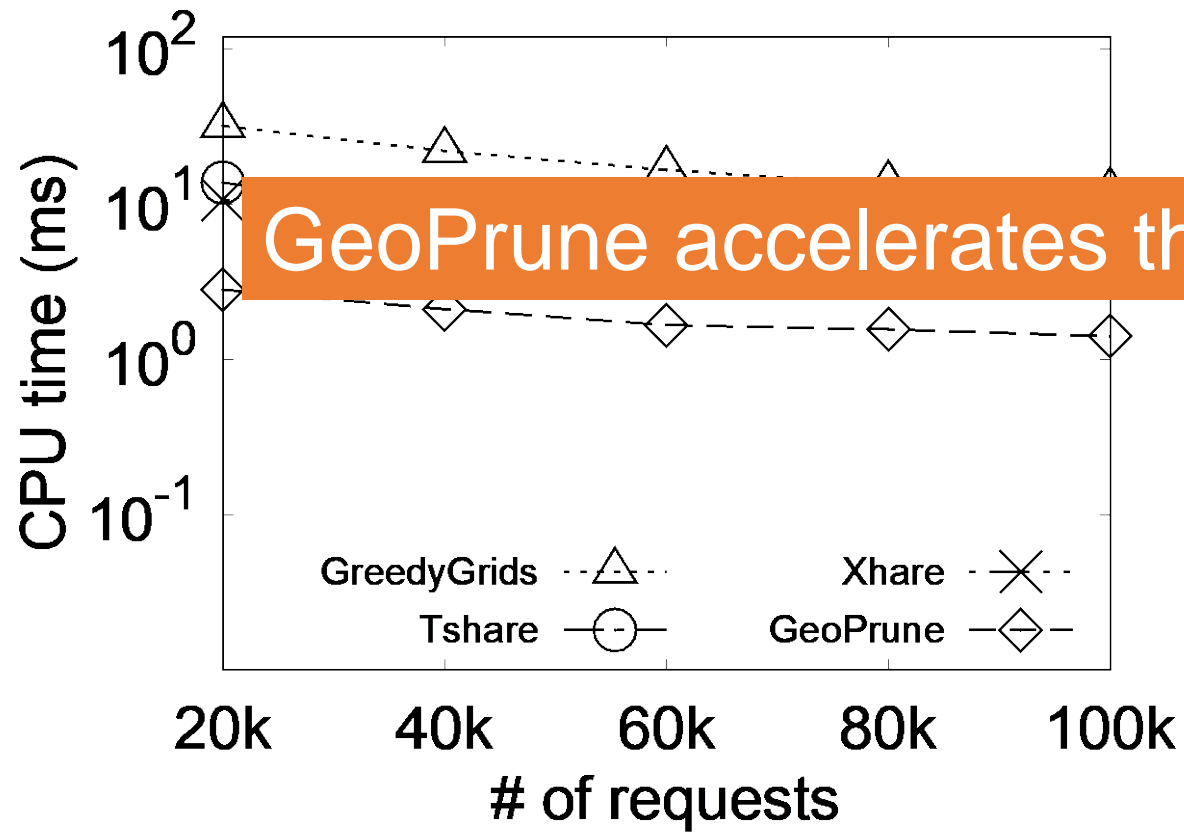
Number of candidate vehicles



GeoPrune reduces the number of vehicle candidates by 11.4 ~ 34.4 times

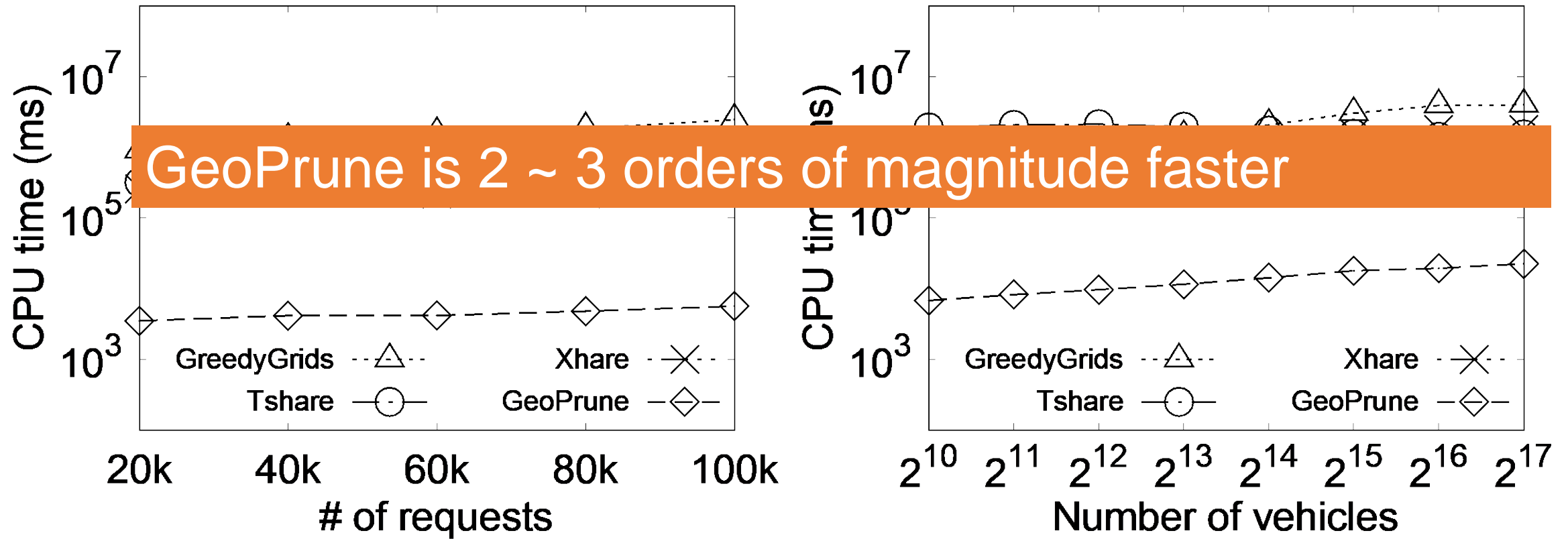


Match time



GeoPrune accelerates the matching time by 71% ~ 90%

Update time



Memory (MB)

	GreedyGrids	Tshare	Xhare	GeoPrune
NYC	0.38	100.34	1546.40	6.56
Chengdu	1.67	9965.37	21282.46	6.43

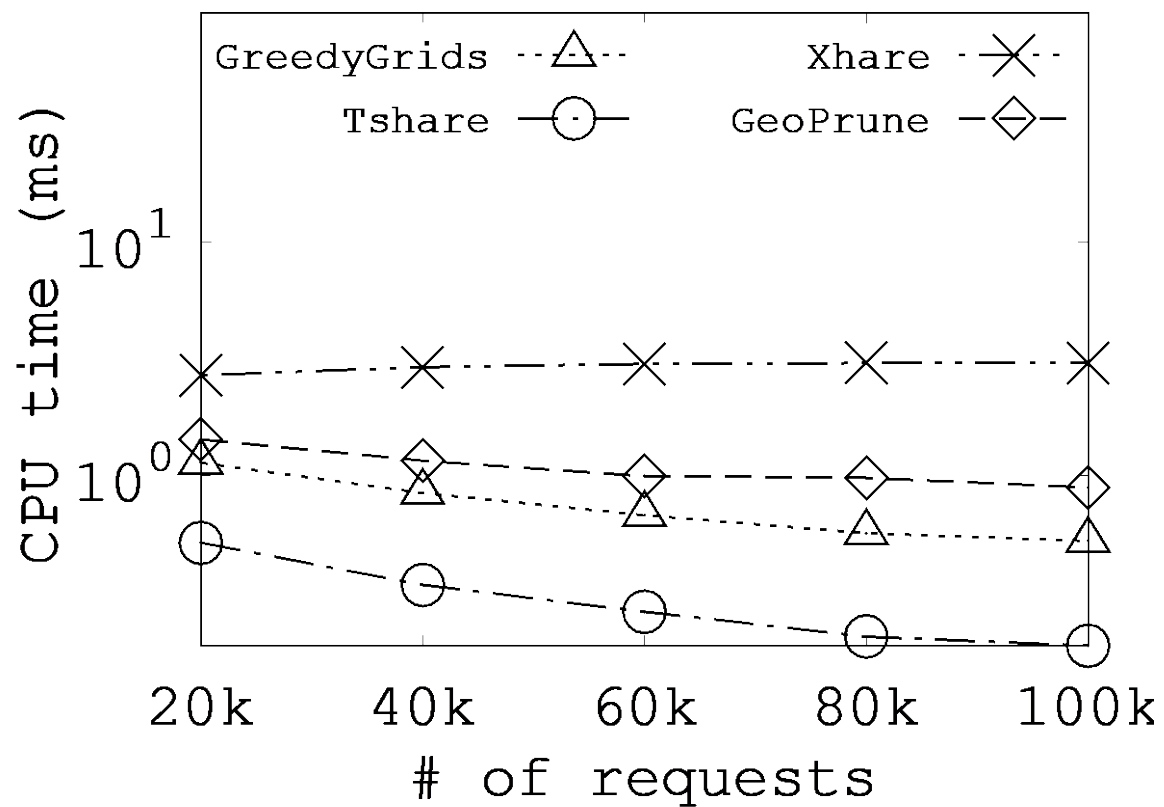
GeoPrune is scalable to large networks

Conclusion

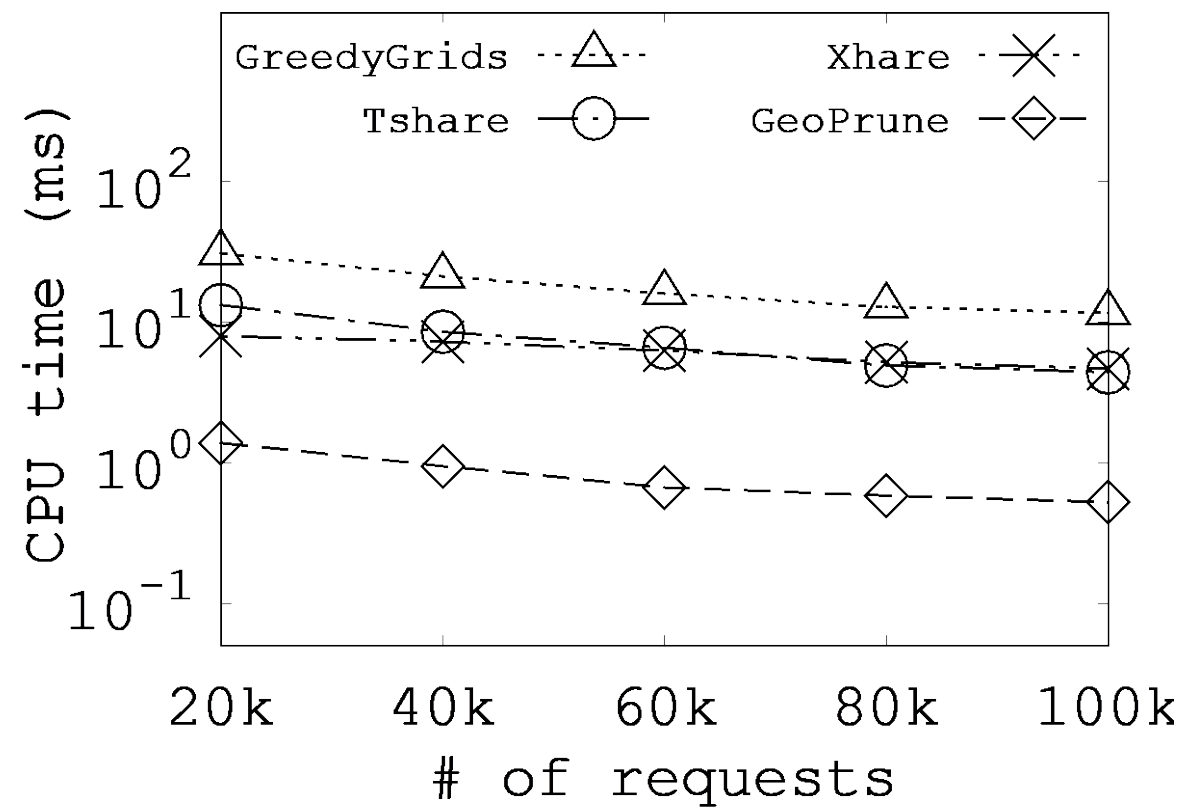
- Matching requests is a crucial problem in ride-sharing
- Effective and efficient pruning is the key to reduce the matching time
- Advantages of GeoPrune:
 - Gurantees no false negatives
 - Efficient and scalable
 - Low update cost
- Future: apply GeoPrune to other ride-sharing problems

Match time breakdown

Prune time

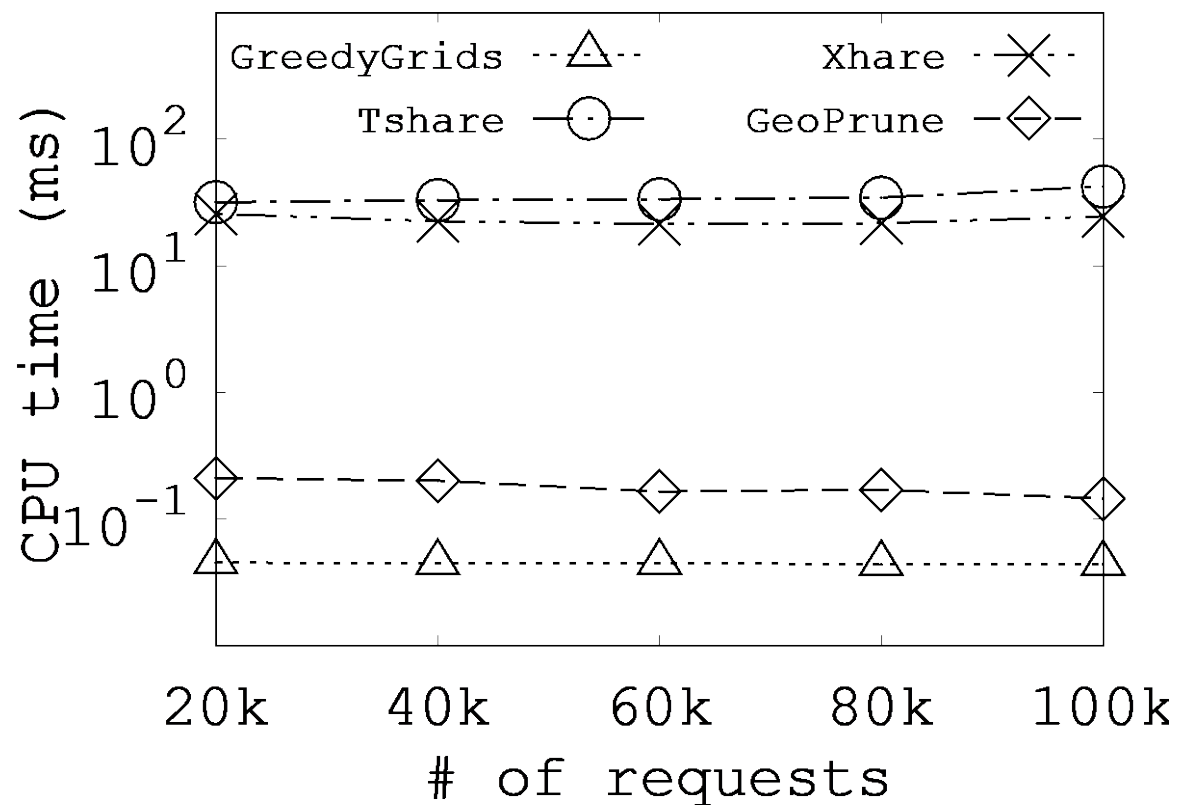


Selection time

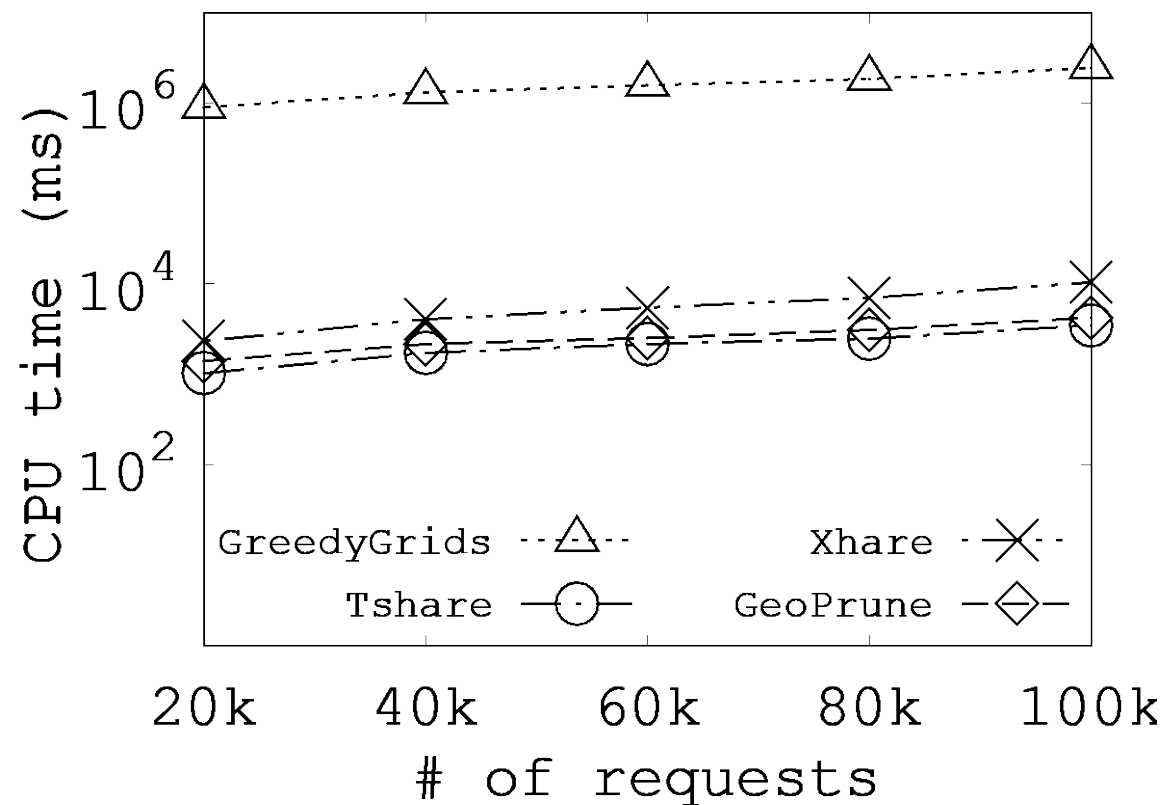


Update time breakdown

Match update

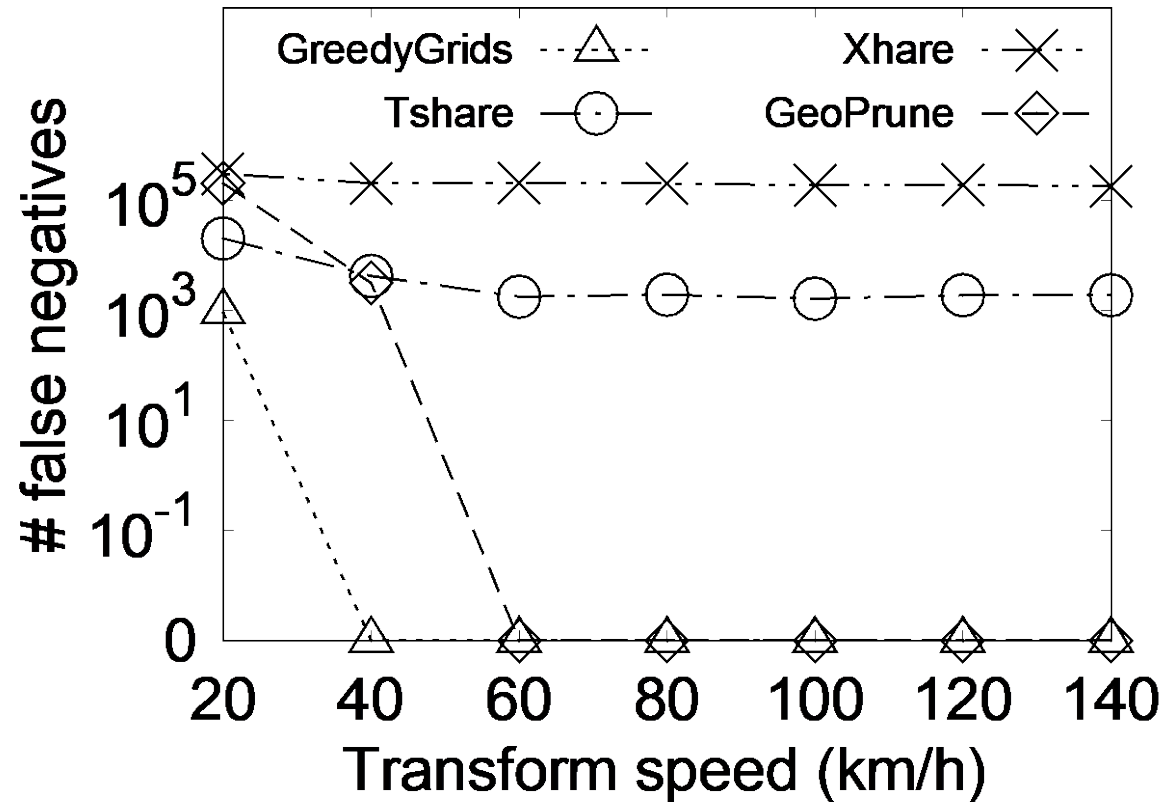


Move update



Effect of the transforming speed

false negatives



Match time

