Xhare-a-Ride

A Search optimized Peer-to-Peer Dynamic Ride Sharing System

Koyel Mukherjee

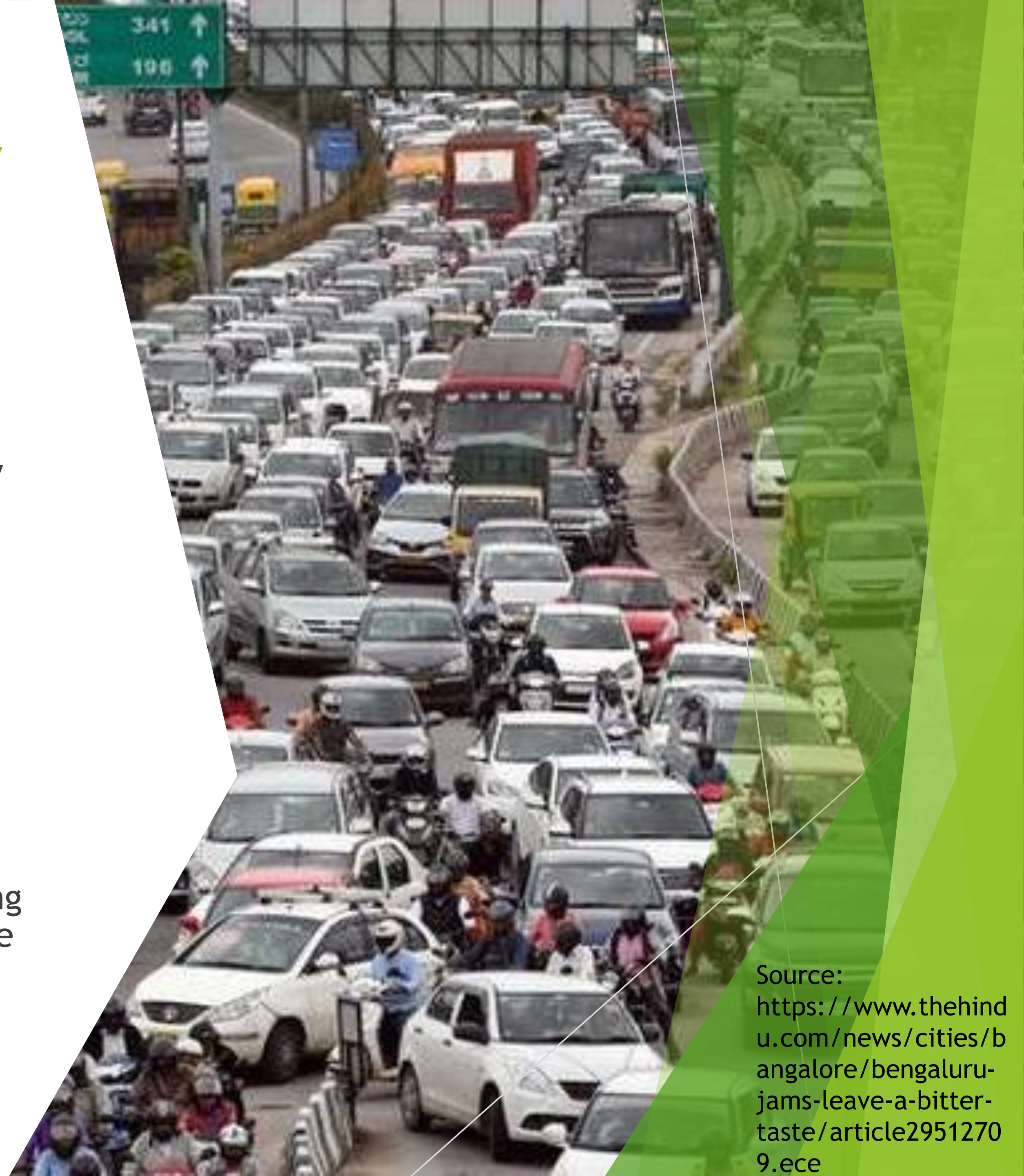
Advisory Research Scientist

IBM Research – India

(This work was done while the author was a Senior Research Scientist, in Xerox Research Center India, 2015-2016.)

Peer-to-Peer Ride Sharing

- Ride sharing: a sustainable, environmentally friendly mode of commute.
- Not many platforms for facilitating peer-to-peer ride sharing.
- Dynamic scenario?
- Integrated with multi-modal trip planners?
- Extend the friendly neighbourhood carpooling to a dynamic, large-scale system and platform.



Source: https://www.thehindu.com/news/cities/bangalore/bengaluru-jams-leave-a-bitter-taste/article29512709.ece

Xhare-a-Ride

- A **dynamic**, **scalable** ride sharing systems for **peer-to-peer ride sharing.**
- **Search optimized, ensures** accuracy and **quality considerations and constraints e.g.** detour and walking preferences.
- Integrated with Multi-Modal Trip Planner.
- Novel region discretization and indexing for scalable and dynamic ride searching and matching
- Learning and Optimizing commuter satisfaction in grouping.
- Fairness in Cost Sharing.

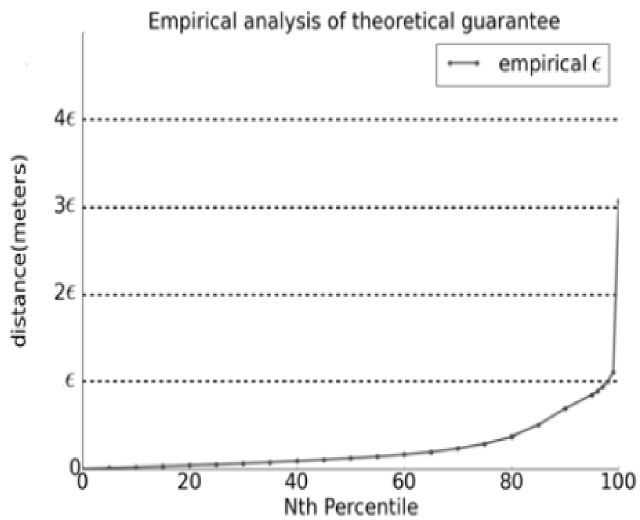
Hierarchical Region Discretization

- Three-tiered region discretization: clusters, landmarks and grids.
- Efficient in-memory indexing for maintaining, updating and efficiently retrieving ride/request spatio-temporal information.
- Optimized and scalable ride search in real-time
- Additive approximation guarantee on the total detour and walking distances.
- Any point location, given by a latitude and a longitude uniquely mapped to a grid, a landmark and finally a cluster, *without any ambiguity*.
- Hierarchy: region → clusters → landmarks → grids → point locations, with cross relations across the levels.

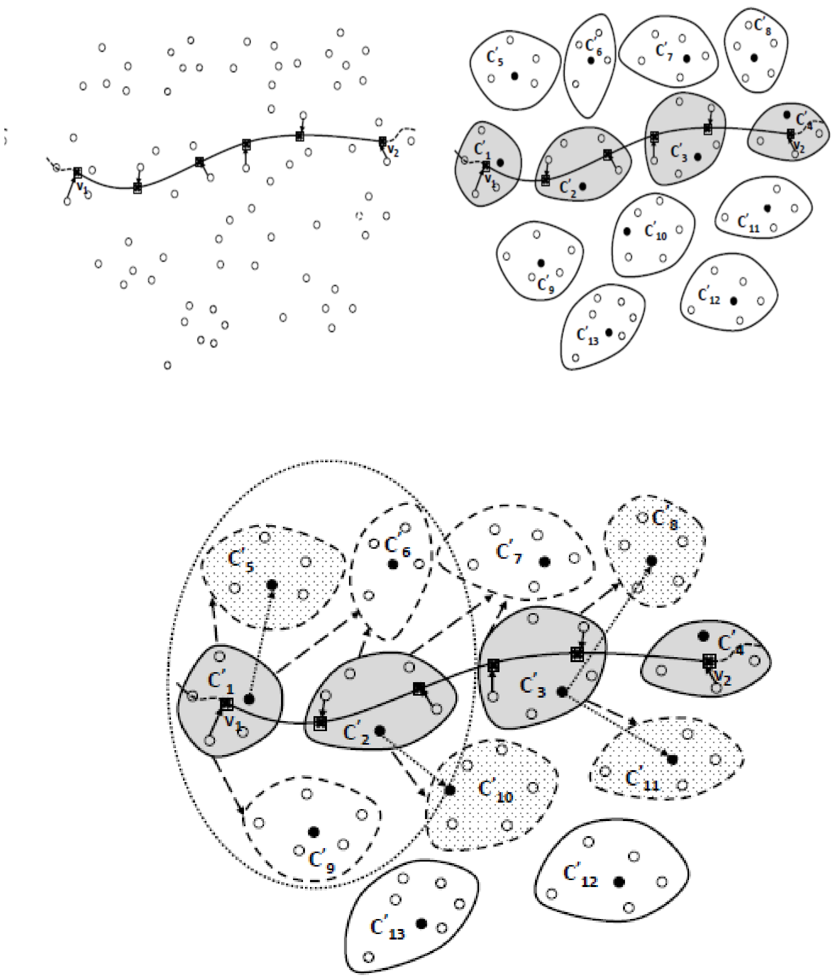
[Xhare-a Ride: A Search Optimized Dynamic Ride Sharing System with Approximation Guarantee, T. Rajasubramaniam, K. Mukherjee, G. Raravi, A. Metrewar, N. Annamaneni, K.Chattopadhyay, ICDE 2017]

Cluster Optimization

- Cluster: a collection of landmarks, such that no pair of landmarks in a cluster are more than a specified driving distance away.
- Cluster Minimization: NP-hard
- Greedy Search: A bicriteria algorithm that finds a solution $(k_{OPT}, 4\delta)$ corresponding to OPT (k_{OPT}, δ).
- Algorithm used Greedy 2-approx for Metric k-Center algorithm as a subroutine.
- Potential Rides <r,t>: associated with each cluster.



Ride Representation



- *Ride*: *<source, destination, departure time, seats>*
- *Route*: shortest path (unless alternate route specified)
- *via-points*: the point locations through which a ride passes;
- Segments: the portion of the route between a pair of via-points.
- *Pass through clusters*: the clusters through which the ride passes in a segment
- *Reachable clusters*: the clusters that the ride can reach without violating the detour limit.

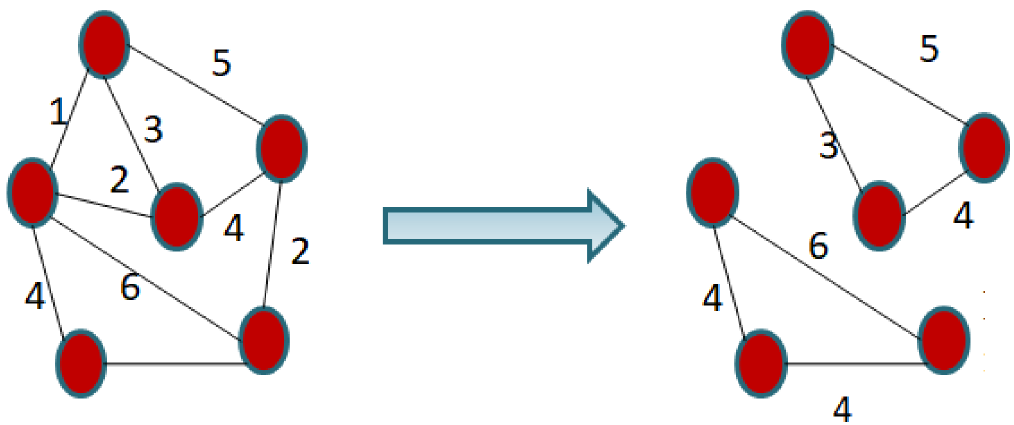
Dynamic Search and Update

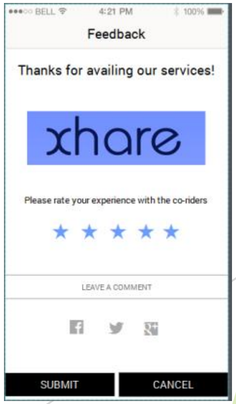
- Ride Search:
  - Identify the grid of src.
  - Find set of *walkable clusters* associated with this grid.
  - For each cluster from the list of *potential rides*, return those with estimated time of arrival in C = departure time.
  - Do the same for dest.
- Ride Tracking:
  - Once ride crosses a pass-through cluster, mark that cluster and associated reachable clusters as obsolete for ride;
  - Remove the ride from the potential rides of those clusters;
  - Delete obsolete clusters from list of pass-through clusters.
- Ride Booking:
  - Update route.
  - Update constraints.
  - Update pass-through clusters and potential rides of clusters.

Commuter Group Satisfaction

- Different commuters have different pairwise compatibility.
- For matching rides and requests, it becomes a resource allocation problem: Resources need to be partitioned into k groups of size $\leq m$.
- We would like to create happy groups!
- How do we learn Compatibility and Ensure Maximum Group wise happiness?

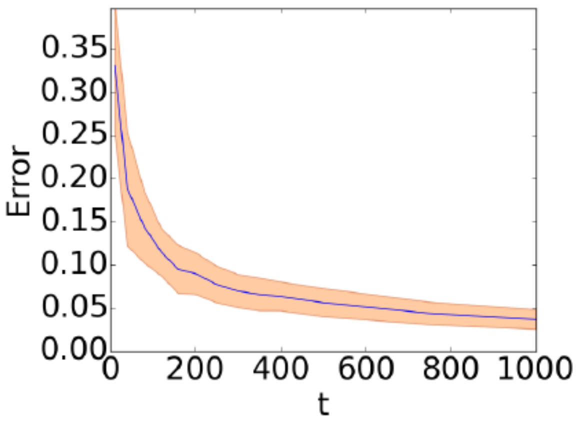
[A. Rajkumar, K. Mukherjee and T. Tulabandhula, Learning to Partition using Score Based Compatibilities, AAMAS 2017]

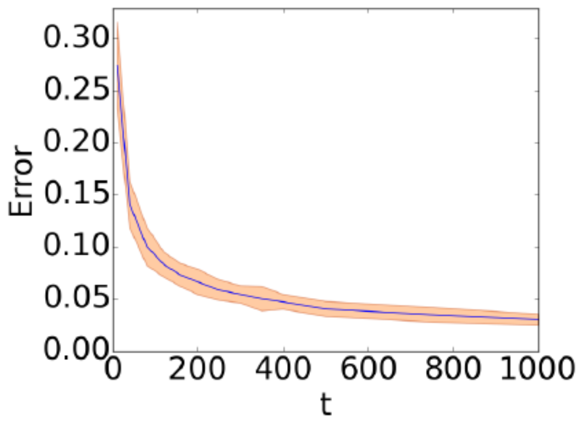




Commuter Group Satisfaction

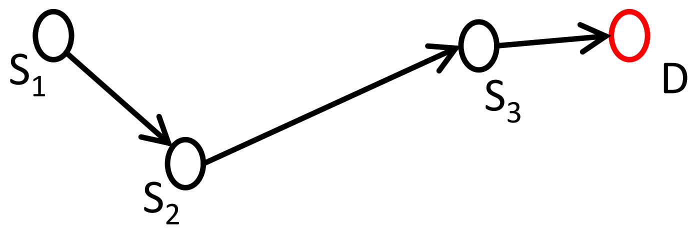
- Partitioning, maximizing different notions of happiness: NP-hard.
- We show hardness of approximation results.
- Under a score-based pairwise structure:
  - P-time algorithms and a ½ approximation algorithm for different objectives.
  - Algorithm LearnOrder: adaptively learns the ordering of the score vector, sufficient for grouping.





Peformance of LearnOrder on Random Graph and sampled from Facebook friendship graph using Jaccard similarity coefficient of features as scores. .

Fair Cost Sharing and Routing



Total cost = $(S_1S_2 + S_2S_3 + S_3D)$

Split cost evenly in every leg? $\left\{\left(S_1S_2 + \frac{S_2S_3}{2} + \frac{S_3D}{3}\right), \left(\frac{S_2S_3}{2} + \frac{S_3D}{3}\right), \left(\frac{S_3D}{3}\right)\right\}$

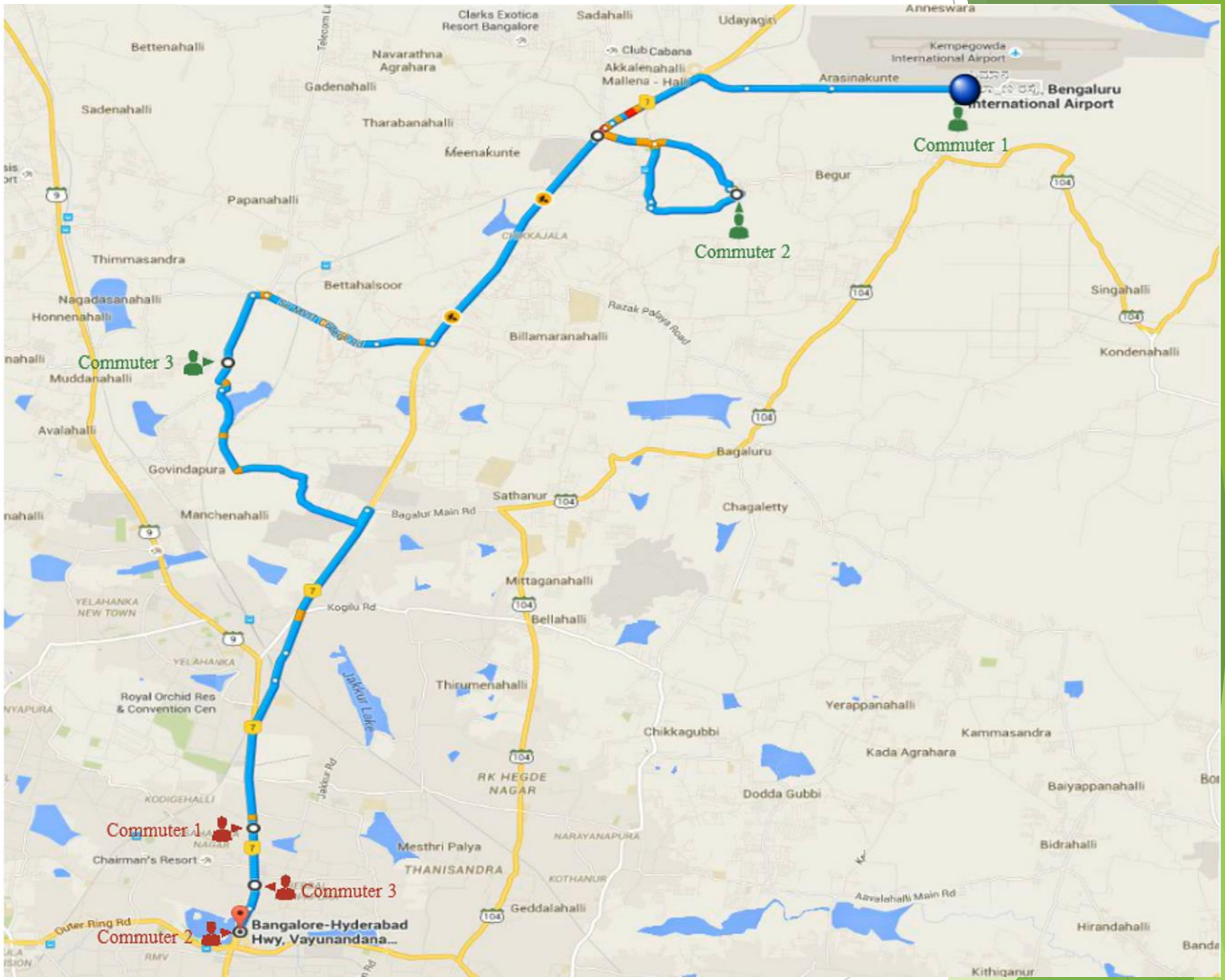
$\boldsymbol{S_1}$ taking a huge detour and potentially paying for it!

- Passengers **compensate** each other for the detour caused inconveniences.
- **Fair** and **rational:** estimated costs decrease every time there is new pick-up.

$$S_{i-1}S_i + S_iD - S_{i-1}D \leq \frac{S_iD}{i} \qquad \text{for} \quad i = 2,3,\dots,C$$

These constraints reinforce intuition: The permissible detour keeps decreasing with the number of passengers, and the closer you are to the destination, the smaller it is.

Fair Cost Sharing and Routing



Modes of Interaction with Multi-modal Trip Planner

Multimodal Trip Planner (MMTP)

<src, dest, time_interval, user_prefs, sys_params>

Ridesharing suggestions

Xhare-A-Ride (XAR)

Aider Mode

- Makes an infeasible trip plan feasible.

Enhancer Mode

- Makes a trip plan “better”

Bootstrapping

- Getting started

Conclusion

- Xhare-a-Ride: dynamic peer-to-peer ride sharing system integrated with a multi-modal trip planner.
- Novel, hierarchical geographical representation and in-memory indexing for dynamic update and accurate, optimized search.
- Several associated research threads: learning and optimizing commuter satisfaction, fair cost splitting, shared location tracking, approximation guarantees with layered system design.
- 4 Top-Tier Publications, 5 USPTO patents.
- Collaboration with external commercial partners.

Thank You!

kmukherj@in.ibm.com