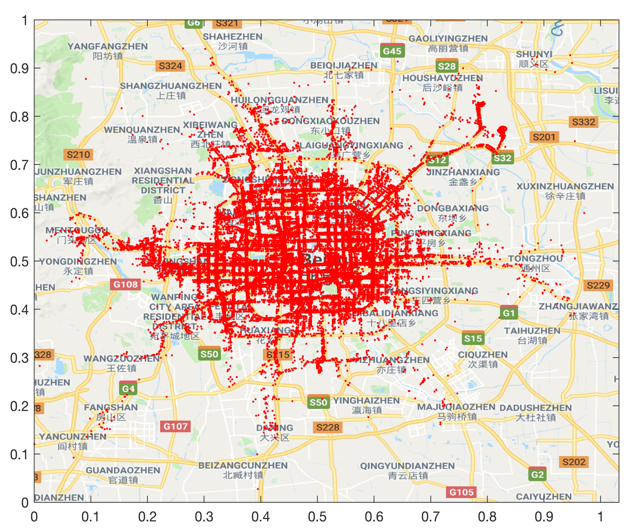
Supercomputing for the Multi-Driver Routing

Zeyang Ye and Yuefan Deng (Advisor)
Department of Applied Mathematics & Statistics, Stony Brook University, NY 11794-3600, United States

Abstract

Supercomputing is essential for routing traffic by providing drivers the optimal routes with minimal traveling distances or time. The unique challenges that require supercomputers to overcome are of multiple folds: numerous drivers, massive simultaneous requests, multiple locations, and needs of instant gratifications, etc. We developed two parallel methods, PSAD and PSAD-M, by using domain decomposition and state-mixing techniques. On the same computing platform with 96 cores, for the same problem, our PSAD methods outperform all published benchmarks by over a hundred times, while improving the solution quality. For the same routing problem on 384 cores, our PSAD-M reduced the elapsed time from the unbearable ten minutes to the reasonable 5 seconds, achieving a record-breaking speedup of 170. By providing instant routing solutions that enable online recommendations, our methods break the bottleneck of the widely adopted offline approaches.

Problem Statement



21,824 pick-up points in Beijing, China

Data Description:
12,481 cab mobility traces in one week in Beijing, China.

Parameters obtained:
The locations of a set of potential pickup points: $C = \{c_1, c_2, \dots, c_{|C|}\}$
The associated probabilities of successfully picking-ups: $P = \{P(c_1), P(c_2), \dots, P(c_{|C|})\}$

The multi-driver routing problem:
Output: The optimal driving route set $R_{opt} = \{\vec{r}_{opt_1}, \vec{r}_{opt_2}, \dots, \vec{r}_{opt_K}\}$ that minimizes the multi-user PTD (MPTD), before the next successful pickups.
PTD calculation:
$D(\vec{d}) = \langle D_{c_0,c_1}, (D_{c_0,c_1} + D_{c_1,c_2}), \dots, \sum_{i=1}^{L} D_{c_{i-1},c_i}, D_{max} \rangle$
$P(\vec{d}) = \langle P(c_1), \overline{P(c_1)} \cdot P(c_2), \dots, \prod_{i=1}^{L-1} \overline{P(c_i)} \cdot P(c_L), \prod_{i=1}^{L} \overline{P(c_i)} \rangle$
$\boldsymbol{PTD(\vec{r}) = D(\vec{r}) \cdot P(\vec{r})}$
where D_{c_{i-1},c_i} is the distance between c_{i-1} and c_i; D_{max} is the penalty for taxi drivers not picking up a passenger; $\overline{P(c_i)} = 1 - P(c_i)$
MPTD calculation:
$\boldsymbol{MPTD(K, R_i) = \sum_{j=1}^{K} PTD(c_{0_j}, \vec{r}_{i_j})}$
where $\forall l, m$ such that $1 \le l \le m \le K, \overrightarrow{r_{i_l}} \cap \overrightarrow{r_{i_m}} = \emptyset$

Significance and Challenges

Significance

Future transportation mode
The current modes of transportation such as private car, taxi, or bus can be increasing replaced by personalized, on-demand mobility systems, similar to those offered by Uber and Lyft.

23% greenhouse gas emissions
30% reduction to the fleet size
The carbon footprint of traffic accounts for about 23% of global greenhouse gas emissions. Taxi is an important transportation mode to look into. For New York City for example, the near-optimal service can lead to 30% reduction to the fleet size compared to the current taxi operation. Our methods aim to reduce the drivers' cruising time in order to make the service optimal.

Driver - higher profit
Passenger - more convenient
Taxi demands are usually much higher than the number of taxis in peak hours of major cities.

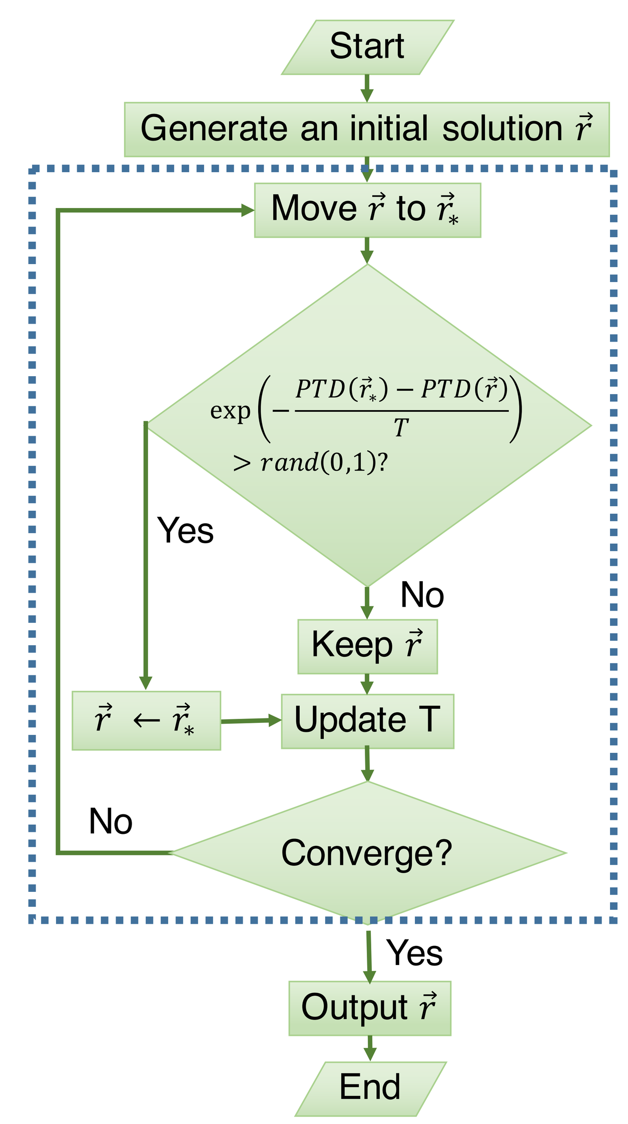
Challenges

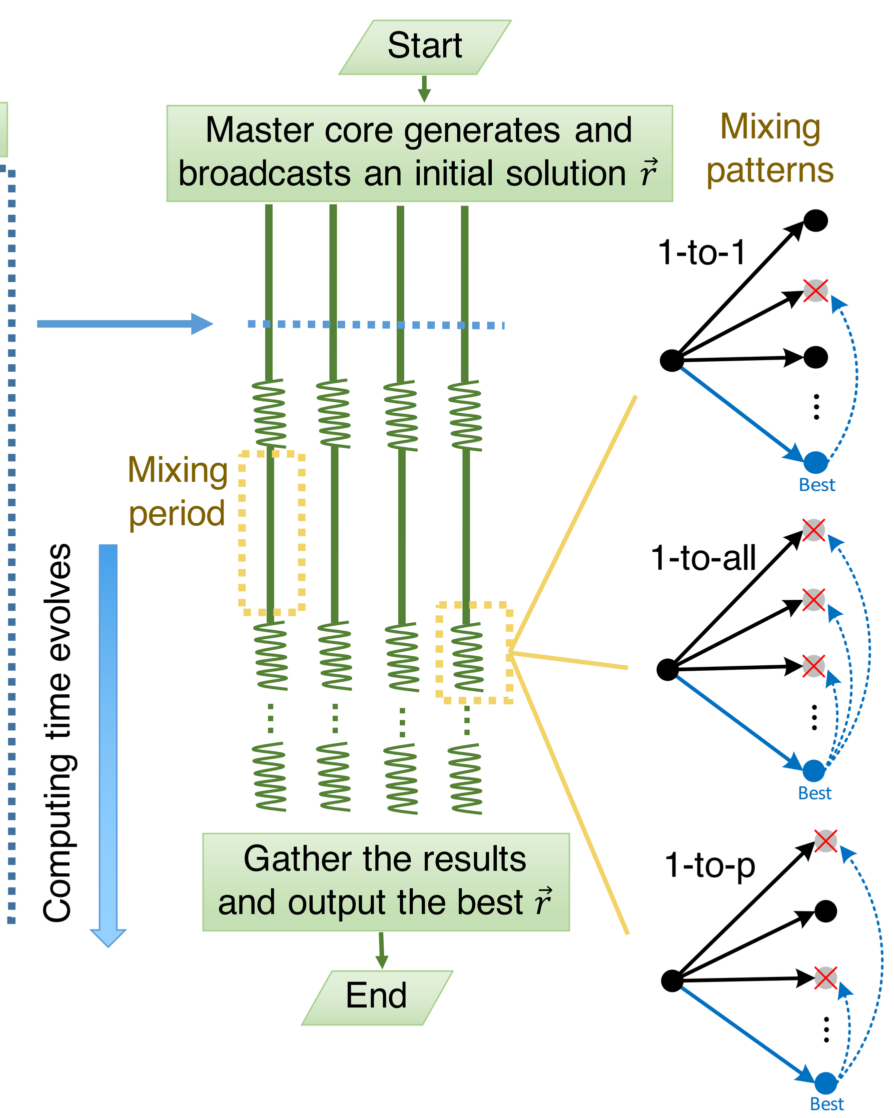
Numerous cabs and trips
For example, in New York City streets, at the end of 2016, there are over 60,000 taxi drivers and over 600,000 daily trips including those from yellow cabs, Uber, and Lyft.
(Source: https://ny.curbed.com/2017/1/17/14296892/yellow-taxi-nyc-uber-lyft-via-numbers)

Great needs for the online recommendation: The taxi driver asks for a route at a random location and needs the answer instantly. On the other hand, the traffic changes rapidly and the recommendation should be made at the same pace.

Supercomputing for data mining

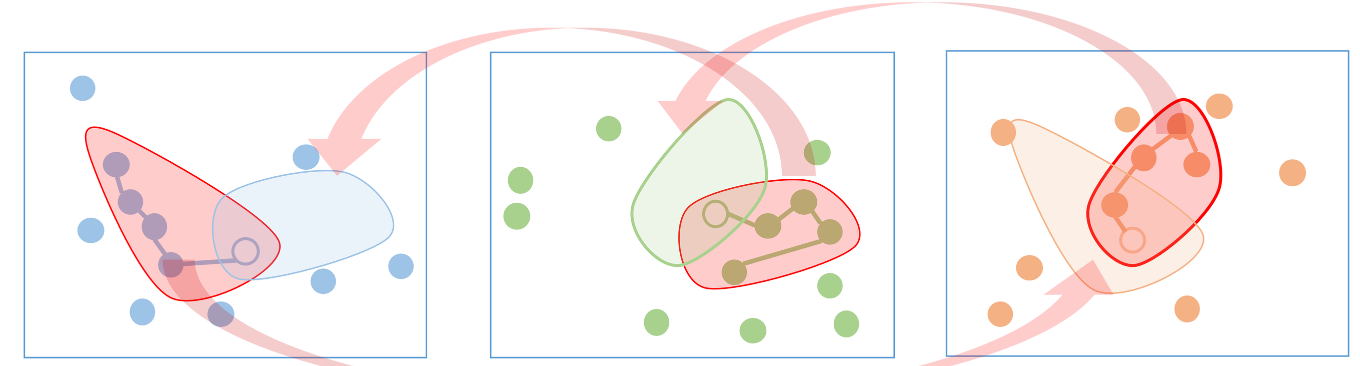
Novel Methodologies



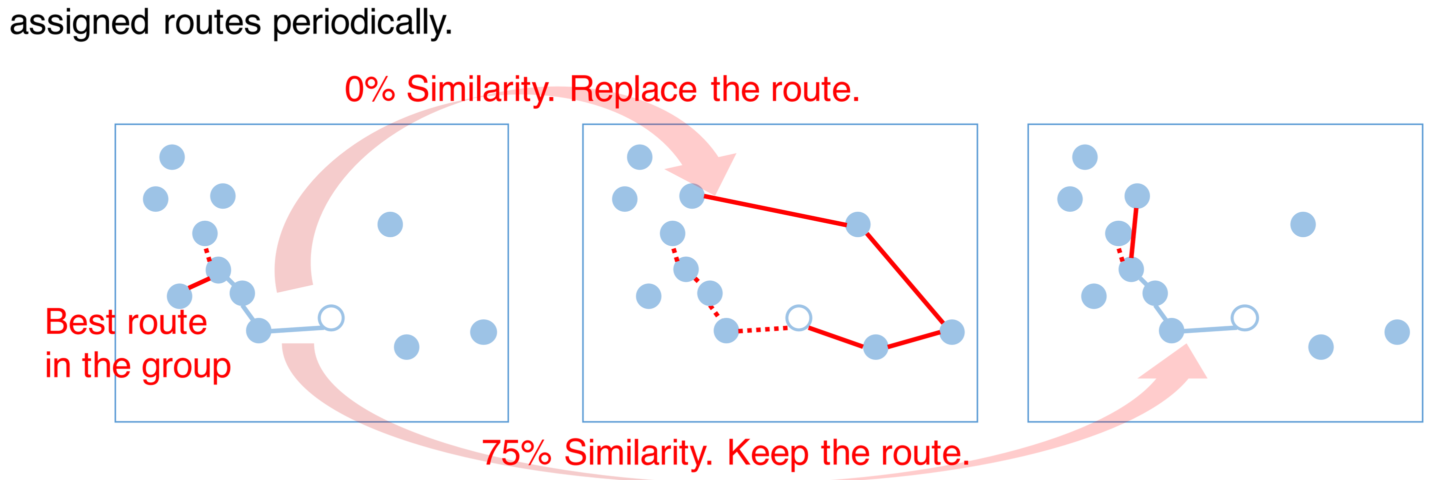


For each core in parallel SA, it runs serial SA in each independent step. When the number of steps reaches mixing period, the cores communicate with each other based on the mixing patterns. Then each core runs SA again based on their current solution.

PSA: Parallelization of SA

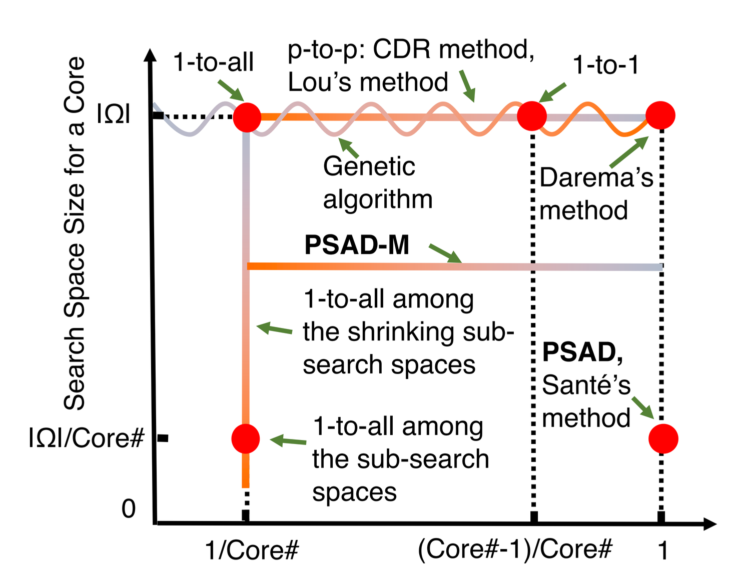


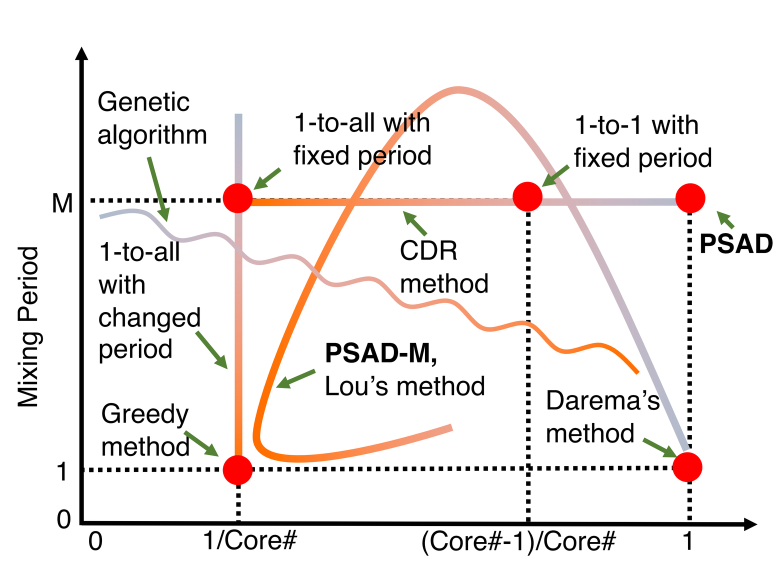
PSAD: each computing core optimizes a route by a subset of pick-up points and rotate the assigned routes periodically.



PSAD-M: a group of cores optimize the same route for some steps and communicate based on the similarity rate between the current best route and their own routes.

Framework of PSA



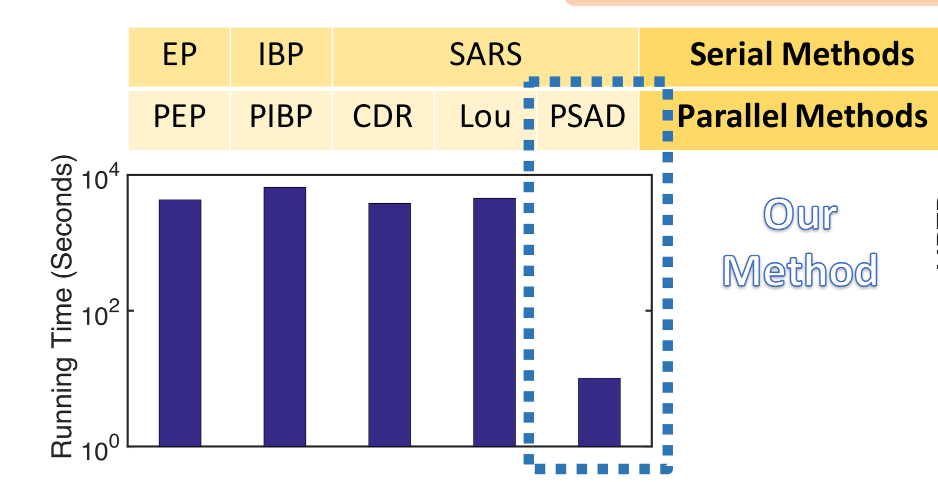


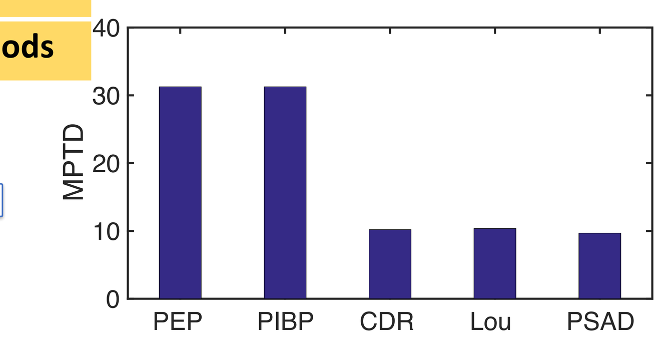
Speedup

Computing Environment: SeaWulf computing system from the Institute for Advanced Computational Science at Stony Brook University

- 100 compute nodes from Cray, each with two Intel Xeon E5-2690v3 CPUs and each has 128 Gigabytes of DDR4 Memory.
- CPUs are codenamed "Haswell" and offer 12 cores each, and operate at a base speed of 2.6 Gigahertz.
- The nodes are interconnected via a high-speed InfiniBand® network by Mellanox® Technologies operating at 56 Gigabits per second, allowing transfer of ~7 Gigabytes of data each second.
- We use MPI for parallel implementations.

Compare with Benchmarks

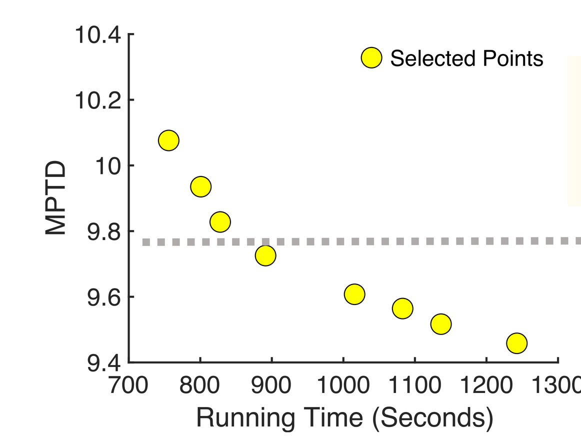




Problem size: 21,824 pickup points (BJ dataset)
Routes number: 96 different routes
Running time obtained: **10.1 seconds**
Computing core used: 96 cores

Compare with the Serial Method

Measure the **speedup**



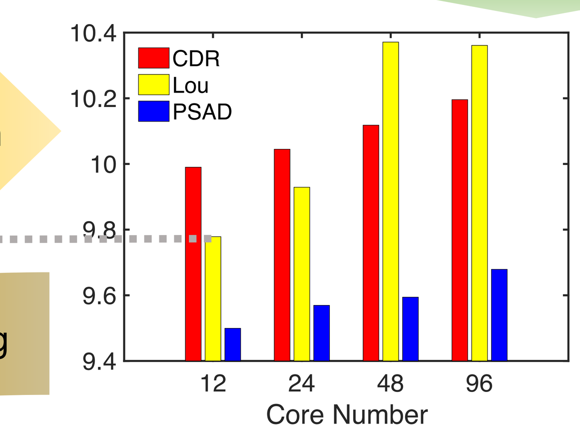
MPTDs obtained from the different cooling rates of serial method

Compare MPTDs between parallel and serial results in the same scale

Find corresponding MPTDs

Locate serial running time

Calculate parallel speedup based on serial and parallel running time



MPTDs from different parallel strategies

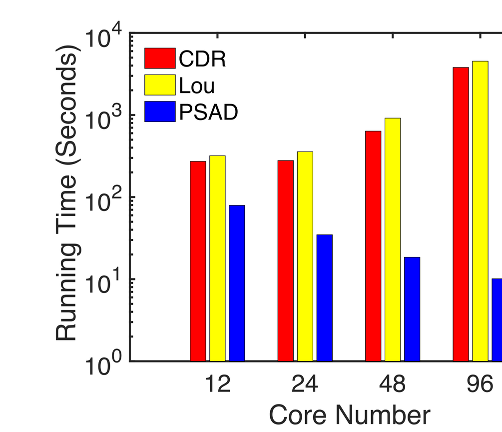
Problem size: 21,824 pickup points (BJ dataset)

Routes number: 96 different routes

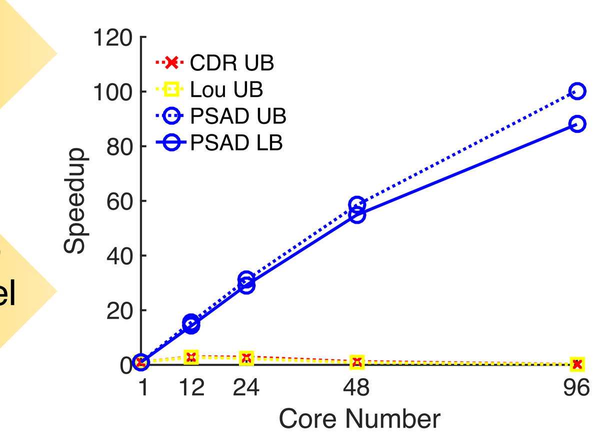
Running time **before acceleration**: $\boldsymbol{\geq 800.4}$ **seconds**

Running time **after acceleration**: **10.1 seconds**

Computing core used: 96 cores



Running time from different parallel strategies

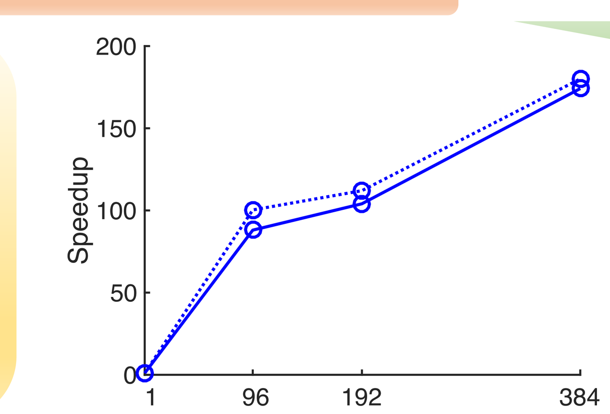


Upper and lower bounds of the speedup due to different MPTD level achieved

Additional Parallel Acceleration

Can we go even **further**?

Overall settings:
Dataset: BJ data set with 21,824 pickup points
Number of routes to recommend: 96
Serial method: SA global search
Parallel method: PSAD-M
Number of computing core in total /per route: 384/4



Upper and lower bounds of the speedup for PSAD-M

Running time **after further acceleration**: **4.6 seconds**

Computing core used: 384 cores

Conclusions

We formulated, and designed the parallel methods PSAD and PSAD-M to solve, a new multi-driver routing problem for recommending multiple optimal routes to multiple drivers at multiple locations, simultaneously. Efficiently solving the high-dimensional multi-driver problems, our PSAD method significantly outperforms all published benchmarks for the application. As a further improvement confirmed by our performance results to PSAD, PSAD-M enhances the utilization of computing cores for achieving higher speedup.

References

[1] Vazifeh, M. M., Santi, P., Resta, G., Strogatz, S. H., & Ratti, C. (2018). Addressing the minimum fleet problem in on-demand urban mobility. *Nature*, *557*(7706), 534.
[2] Ge, Y., Xiong, H., Tuzhilin, A., Xiao, K., Gruteser, M., & Pazzani, M. (2010, July). An energy-efficient mobile recommender system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 899-908). ACM.
[3] **Ye, Z.**, Xiao, K., Ge, Y., & Deng, Y. (2018). Applying Simulated Annealing and Parallel Computing to the Mobile Sequential Recommendation. *IEEE Transactions on Knowledge and Data Engineering*.
[4] **Ye, Z.**, Zhang, L., Xiao, K., Zhou, W., Ge, Y., & Deng, Y. (2018, July). Multi-User Mobile Sequential Recommendation: An Efficient Parallel Computing Paradigm. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2624-2633). ACM.
[5] **Ye, Z.**, Xiao, K., & Deng, Y. (2018, Nov). A Unified Theory and the Solution of the Mobile Sequential Recommendation Problem. In *Proceedings of 2018 IEEE International Conference on Data Mining.*
[6] Huang, J., Huangfu, X., Sun, H., Li, H., Zhao, P., Cheng, H., & Song, Q. (2015). Backward Path Growth for Efficient Mobile Sequential Recommendation. *IEEE Trans. Knowl. Data Eng.*, *27*(1), 46-60.

Presenter: Zeyang Ye
Email: zeyang.ye@stonybrook.edu
Phone#: 631-682-8240 (USA)