This demo presents Noah: a dynamic ridesharing system. Noah supports large-scale real-time ridesharing with service guarantees on road networks. Taxis and trip requests are dynamically matched. Different from traditional systems, a taxi can have more than one customer on board given that all waiting time and service time constraints of trips are satisfied. Noah's real-time response relies on three main components: (1) a fast shortest path algorithm with caching on road networks; (2) a fast dynamic matching algorithm to schedule ridesharing on the fly; (3) a spatial indexing method for fast retrieving moving taxis. Users will be able to submit requests from a smartphone, choose specific parameters such as number of taxis in the system, service constraints, and matching algorithms, to explore the internal functionalities and implementations of Noah. The system analyzer will show the system performance including average waiting time, average detour percentage, average response time, and average level of sharing. Taxis, routes, and requests will be animated and visualized through Google Maps API. The demo is based on trips of 17,000 Shanghai taxis for one day (May 29, 2009); the dataset contains 432,327 trips. Each trip includes the starting and destination coordinates.

Ridesharing Service

Despite high amounts of congestion and pollution in concentrated areas throughout the world, many private and public vehicles continue to travel with empty seats. The mean occupancy rate of personal vehicle trips in the United States is only 1.6 persons per vehicle mile. In this demonstration, we simulate and visualize the effects of taxi ridesharing on the city of Shanghai.

What is Ridesharing?

Real-time ridesharing, enabled by low-cost geo-locating devices, smartphones and wireless networks, is a service that dynamically matches each taxi to a potential multitude of taxi requests. Assuming two requests have similar initial and destination locations, it is more resource efficient for one taxi to fulfill both requests with a limited amount of waiting time and route detour, rather than have two taxis each fulfill one request. In our demonstration simulation, we use an optimized kinetic tree algorithm to evaluate each taxi request with the available taxis. A taxi will fulfill the request if it satisfies all user-defined constraints.

Benefits:

- Reduce global energy consumption
- Conserve exhaustible natural resources
- Decrease road congestion

Compromises:

- Increased average waiting time
- Potential route detours

Providing a large-scale urban ridesharing service is a non-trivial issue. At the core of the implementation is a real-time matching algorithm that can quickly determine the best taxi to satisfy incoming service requests. Traditional approaches such as branch-and-bound or mixed integer programming are inefficient when dealing with a problem of this magnitude. Furthermore, most previous solutions focus on scenarios where all requests are known ahead of time. These methodologies are incapable of performing well in stochastic and modern situations, where a new request can be made at any time.

iOS Application

Using an iOS application, demo attendees are able to create and queue a new taxi request in real time, choosing the user's pickup location and intended destination on a map view and sending this information to the server. Once the request has been made, it will be placed in a pickup queue, and a taxi will be allocated to pick up the passenger. The iOS application will allow also the user to view relevant information regarding their taxi.

Server Simulation

The backend simulator time is kept in sync with the web client time, and taxi locations are updated once every simulator second. The simulator writes all taxi information to a MySQL database table, which is then read and visualized by the web client. The server also implements a system for handing new requests sent by the iOS application, and adding that request to the queue.

Conclusion

This demonstration visualizes real-time ridesharing on the Shanghai road network, dynamically matching real-time trip requests with available taxis. Our kinetic tree algorithm implements slacks time and clustering optimizations, and is able to outperform common approaches to large-scale ridesharing, such as branch-and-bound and mixed-integer programming. Through our web client, users are able to control the ridesharing simulation in real-time, view live algorithm performance statistics, and observe animated request and taxi location information based on predetermined simulation parameters.

References


This work was partially supported by the National Science Foundation under Grant No. IIS-1019726.