MATH4202 Project Proposal

T.Manderson,I.Rudge

August 25, 2017

Problem Definition

The paper selected for the project is Stable Matching for Dynamic Ride-Sharing Systems taken from Informs Transportation science published on the 16th of August 2017. This paper was selected as both students found it an interesting topic with plenty of room for potential work, while still presenting a challenge. The paper, in short , seeks to determine stable or nearly stable matches between riders and drivers in ride share apps such as Uber and lyft.This is done by considering a dynamic environment in which participants can determine driving and passenger claims within a close window to desired departure and arrival times. The paper then seeks to minimize the system wide cost of travel, which coincides with a reduction in distance travelled. This is seen as the comparison between the distance travelled together as Driver and Rider compared to if both were to transport individually. Initially the paper considers the stable case, that is no passenger or driver has a preference on matching, this is a reasonable assumption as the system will degrade as it becomes less stable, implying optimal solutions will be invalid or more difficult. The paper seeks to minimize the cost in these stable or near stable systems for individual passengers as this will in turn minimize the total system cost. An additional difficulty in this problem is that the end state is not known, it can continue on an infinite horizon when only a finite time frame of data is known.

Prior Solutions

There are several solutions discussed in the paper, focusing on solutions for stable and unstable environments, with different methods for each.

Stable solutions

Greedy Algorithm

The paper discusses the use of a greedy algorithm that will use a matching heuristic with the goal of minimizing total system costs, this works in the following manner

- determine for each passenger $r \in R$the driver announcement $d \in D$ (if any) that represents a feasible match with the largest savings per participant in the match
- Amoung these matches select (r_m, d_m) with the largest saving and add this match to the matching
- requests r_m and d_m are marked as completed and removed from the set of requests

Basic Stable Formulation

The basic formulation for a stable case can be viewed as a networking problem with the following formulation. First a node is created at $R \cup D$ with an arc connecting a node $i \in R$ to a node in the Bipartation $j\ inD$ only if it is a feasible match in both time and producing positive travel savings. For each Arc (i, j) we can assign a cost C_{ij} with the set A representing all feasible nodes.

Sets
Set of Riders $i \in R$
Set of Drivers $j \in D$
Set of Arcs $(i, j) \in A$
Data
Cost of (i, j) $C_{ij} \quad \forall (i, j) \in A$
Variables
$x_{i,j} = 1$ if (i, j) is used, 0 otherwise $\forall (i, j) \in A$
Objective

$$max(\sum_{i,j \in A} c_{i,j} x_{i,j})$$

Constraints

$$\sum_{j \in D} x_{i,j} \leq 1 \quad \forall i \in R$$

$$\sum_{i \in R} x_{i,j} \leq 1 \quad \forall j \in D$$

$$\sum_{j` \geq j_i} x_{i,j`} + \sum_{i` \geq i_j} x_{i`,j} + x_{ij} \geq 1 \quad \forall (i, j) \in A$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A$$

Near stable solution

The paper also considers a solution where only a small percentage of Riders and Drivers will have a preference, this is similar to the prior section, however with the added constraint that some pairs can not match, this is implemented simply with an additional constraint in the paper. This will increase the runtime of the code as the amount of preferences increases. Following this the system could be considered for multiple people

Suggested Solution

The Initial thoughts of both students is to implement the Basic stable case and apply any preferences as Lazy constraints, this should reduce the amount of constraints and allow the margin of solvable stability to increase. Following this we hope to start work on taking into account multiple Riders in the system, which may actually require delayed column generation (or this may be appropriate at least). These are the main goals of both students. We hope that by implementing Lazy contraints instead of the current methodology we are able to achieve similar results in similar or better running times.

Data

Getting the data for this paper may prove difficult, as of this point in time requests to the authors have been sent asking for data for testing, If this is not the case then the Raw data that they use from the Atlanta Metropolitan Council in 2009 is avialbale online which can be used to generate values for traffic. However the most likely option is to write code that will generate the start and goal locations of Riders and Drivers that will allow us to not only test our code but handle the data in a way that we desire.