Optimizing Routes of Public Transportation Systems by Analyzing the Data of Taxi Rides

Keven Richly
Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam, Germany
keven.richly@hpi.de

Ralf Teusner
Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam, Germany
ralf.teusner@hpi.de

Alexander Immer
Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam, Germany
alexander.immer
@student.hpi.de

Fabian Windheuser
Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam, Germany
fabian.windheuser
@student.hpi.de

Lennard Wolf
Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam, Germany
lennard.wolf
@student.hpi.de

ABSTRACT

Public transportation systems are flexible and affordable for the passengers. In contrast, the operation and construction of the necessary infrastructure is cost-intensive and requires extensive planning. Decisions about the scheduling, capacities and the location of stations are dependent on various economic, social, and environmental factors and have a major impact on the structure of a city. In this context, information about the starting points and destinations of potential passengers is highly relevant for operators. Unfortunately, the collection of this data is not trivial and often based on time intensive and expensive studies.

In this paper we present a novel approach to gain knowledge for transportation system optimization based on the data of taxi rides, which have been recorded for documentation purposes. This data can be analyzed and offers an insight into the fine-grained travel intentions of millions of people. We introduce an interactive web application, which enables the analysis of about 700 millions taxi rides in New York City. Additionally to the exploration of the most frequent travel routes, the application can automatically suggest useful extensions of the exciting transportation system or suggest an optimized route map, which can be used to evaluate the existing one. With this functionality, the presented software effectively supports the decision processes of operators and enables the continuous evaluation of the existing systems.

UrbanGIS'15 November 03-06, 2015, Bellevue, WA, USA
 ISBN 978-1-4503-3973-5/15/11…$15.00
DOI: http://dx.doi.org/10.1145/2835022.2835035

1. INTRODUCTION

Public transportation sytems influence the structure of cities and their livability, economic, social, and environmental characteristics [7]. Since transit has great possibilities to reduce traffic overloads and exhaust emissions, the optimization of the existing public transportation networks is a worthwhile goal. Based on a data set of about 700 million taxi rides in New York, the approach aspires to make the information explorable and relate it to the city's public transport system. This would allow taxi companies as well as operators of public transportation systems – like the MTA[1] – to improve their existing infrastructure. When trying to improve public transportation systems, the most important information is where people come from, and where they want to go. Through this it is possible to understand the needs of the customers and thus expand the system's infrastructure efficiently and conveniently. In some parts of the world, for instance New York City, such data is being recorded by taxi drivers and their companies on a daily base, which yields vast opportunities for local public transport. Analysing such financial, spatial, and temporal data could therefore help companies such as the MTA to make predictions about occupancy rates and potential revenue, in case changes are made.

The main goal of this work is to use the data of taxi trips to enable local transportation authorities to optimize the scheduling, capacities and the location of stations with the consequence to reduce the necessity of taxis. The described approach tries to reveal, whether this is possible at all. Therefore the leading question is: *Can improvements of the public transportation system based on taxi data make more taxi rides dispensable?* Thus, data relating to New York's public transportation infrastructure was incorporated additionally. For clarity's sake, only the city's subway network is used in the presented prototype.

[1]MTA: Metropolitan Transportation Authority, the corporation in charge of New York's public transportation

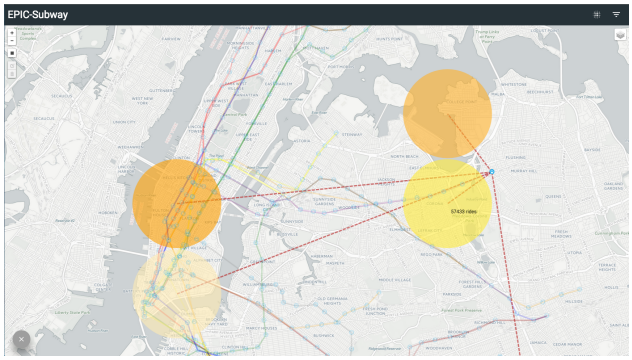

Figure 1: A screenshot of the application in Exploration mode, with the cluster view and the settings menu opened

Our work aims to provide an interactive application for optimizing public transportation networks. In this paper, we exemplarily demonstrate the concept with the subway network of New York. The same approach can be applied to bus, ferry, tram, or railway networks of various cities, as well as combinations arbitrary combinations. Therefore the focus is on the following three abstract use cases: First, making the taxi rides geographically explorable with respect to Manhattan's subway network. This enables the user to find out where most taxi rides are going from a selected station. Thus overlaps and gaps in the subway network can be detected. Second, revealing the most travelled routes and those that are not covered by public transportation, which could potentially expose gaps in the infrastructure. Third, proposing completely new subway lines, that would substitute as many taxi rides as possible. This can be done by aggregating the spacial information of the rides and combining them with the distribution of the subway stations. This process involves benchmarking existing as well as suggested lines.

2. RELATED WORK

There are different research projects, which focus either on analyzing the data of taxi trips or on the optimization of the location of transit stations. In contrast to existing projects, the approach presented in this paper combines both topics and aims to support the decision processes of public transportation system operators. Yang et al. [8] presented a method to find typical demand patterns in a data set of taxi rides in Boston. For this purpose, they take a comprehensive look at the time and location as well as demand fluctuations to identify events and important geographical positions (e.g. community centers etc.) and correlations between these. They also incorporated weather data into their pattern findings, to create additional knowledge for taxi vendors. To achieve this they used neural networks and machine learning techniques with ridge regression. A comparable approach was developed by Ganeshpillai [3]. He focused on the analysis of twitter information to extract knowledge about the demand of taxis in specific locations. HubCab[2] is an interactive application that visualizes the ways of over 170 million taxi trips. The application enables the user to analyze the pickup and drop off points of the passengers and additionally analyze potential taxi sharing benefits between locations. A similar application was developed by Toole [5], but he focused more on the analysis of the routes of the taxis. Another interactive application in this direction was developed by Kellermeier, Kesar, and Paul[3]. The interactive web application visualizes the user on the one hand, where the most profit is made at a specific time and, on the other hand, at which point in the week the most profit is made in a specific region. Chien et al. [2] formulated a mathematical model, which improves the accessibility of bus stations on a segment of a bus route. In this case the optimization of the locations of the stations is limited to the given segment of the route. Furthermore there are different publications about analytical models, which use historical and statistical data to optimize the existing transportation network [1] [4]. These approaches are often limited to the data that was recorded by the operators of public transportation systems.

[2]http://hubcab.org/#13.00/40.7219/-73.9484

3. APPROACH

The focus of the implemented application is to support three use cases, which are further examined in this section. Firstly, a way to explore the existing subway network with respect to the taxi rides is proposed (see Section 3.3). It should be possible to find out where taxi rides, that started from a selected station, most commonly headed. Furthermore, it should then also be possible to find out where taxi rides, that end in a selected station, are most frequently coming from.The second part of our work covers the problems of exploring and displaying the vast amount of spatial taxi ride data (Section 3.4). In this use case, the main investigation is how to find routes that are frequently travelled with taxis in New York City. Regarding the enormous dataset with almost 700,000,000 rides, it is necessary to find a way to aggregate this data efficiently and make real time analysis possible. Finally, based on the newly found information, it is attempted to suggest one or more new subway lines that would extend the existing subway network of New York best (see Section 3.5). This includes benchmarking the existing, as well as the newly generated lines. This way it can also be discovered, how well existing lines match the needs of New York citizens.

3.1 The Data Set

Because of its keen enforcement of the American Freedom of Information Act, New York is one of the leading cities in the area of open data [6]. This leads to the disclosure of an immense volume of all different kinds of information, which also entails data on NYC's taxi & limousine operations.

These records of cab rides were released by the University of Illinois[4] in 2014 and form the basis of this work. This data set contains a total number of 697,622,435 cab rides by 54,990 different taxi drivers over the years from 2010 to 2013. The data, as used, is split up into the two tables *Trip* and *Fare*. The *Trip* table contains info regarding every taxi ride. It contains the driver ID, the amount of passengers, the start and end points, as well as start and end times, and the overall time and distance travelled. The *Fare* table stores additional financial information for each trip. This includes the total payment, the tip, and the payment method.

[3]https://github.com/mockingbird2/EPIC-ness

[4]`http://publish.illinois.edu/dbwork/open-data`

Trip		Fare	
Medallion	INTEGER	Medallion	INTEGER
Driver	INTEGER	Driver	INTEGER
Vendor	STRING	Vendor	STRING
Pickup TIME	timestamp	Pickup TIME	timestamp
Dropoff TIME	timestamp	Payment Type	char
Rate	INTEGER	Fare	REAL
Flag	char	Surcharge	REAL
Passenger	INTEGER	Tax	REAL
Triptime	REAL	Tip	REAL
Distance	REAL	Tolls	REAL
Pickup Lng	REAL	Total	REAL
Pickup Lat	REAL		
Dropoff Lng	REAL		
Dropoff Lat	REAL		

Table 1: Both tables in the stored structure with identifiers and their SQL data types

The table structure is represented in Table 1. In addition to the taxi ride data, info on the MTA subway network is used. This is based on a data set from *NYC Open Data*[5], which contains information on the name, the position, and the related lines for each subway station. Table 2 shows how the stations are stored in our SQL based database.

3.1.1 Data Quality

Before actually using the data in the application, we examined it by visualizing the different values in bar charts with the x-axis always being the time. This showed that not all entries were accurate or technically possible. Some rides went to impossibly far away places, took way too long, or were charged with immensely high fares going up to millions of dollars, resulting in sometimes extreme aberrations in the charts. These faulty entries, as well as duplicates, can be removed from future results using a set of filters that exclude negative fare payments and all other corrupt entries. Since for the scope of the application the only used entries were *Pickup TIME*, *Dropoff TIME*, *Pickup Lng*, *Pickup Lat*, *Dropoff Lng*, *Dropoff Lat*, and *Passengers* only these have to go through the filtering process and thus the amount of entries that have to be analyzed is manageable.

3.1.2 Subway Data Preparation

In order to make the data on New York's subway stations combined with their corresponding lines usable and displayable on a map, it has to be processed. It misses information about the order of stations in a line which is essential for displaying a coherent network of lines.

ID	Name	Lines	Latitude	Longitude
INTEGER	STRING	STRING	REAL	REAL

Table 2: SQL structure of the Table containing the subway stations

To achieve this, the following algorithm was used:
First, a starting station is set for each line. This information is taken directly from the Metropolitan Transportation Authority[6]. The closest station of the same line then is added to the line and acts as a new starting point. This is repeated, until there are no more stations of the same line left. It should be mentioned that the data structure to store stations, as well as the line-generating algorithm can be applied to all public transport data of this kind.

3.2 System Architecture

The following approaches are implemented as an interactive application with a *node.js*[7] back end, which provides an API to encapsulate the data processing.
The API is used by the front end in order to visualize the results on an interactive geographical map. The front end was built using *AngularJS*[8], which enables us to build the real time interactive experience. To render the map, the *leaflet*[9] library is used. In comparison to other map libraries such as *Google Maps*, leaflet has the advantage of being open source and easily expandable. In addition to leaflet, *d3.js*[10] allows displaying custom graphs for the exploration mode. Our implementation is publicly available on Github[11]. It also contains a short instruction for setting up the app and running it.

The final application mainly consists of a small menu for filter customization and the big interactive geographical map – centered on New York City – on which a station icons is displayed for each subway station, as well as visualizations of all the subway lines connecting these stations (for a screenshot see Figure 1). It features two different usage modes: *Exploration* and *Optimization*. Each mode, described in the following subsections, let the user interact with the data in a distinct way and can give different insights.

3.3 Exploration Mode: Exploring the Travel Routes of Taxi Passengers

In the first mode, the user can explore the correlations between the taxi rides and the subway stations. She can click on a station icon on the map and then easily see where most taxi rides are either heading or arriving from at the selected station. Preferences such as this one can be set in the aforementioned menu. The user can refine the results by setting multiple filters (incoming or outgoing rides, only rides from a specific year or time interval, and size of the underlying grid) and choose between visualizing his results using a hexbin or a cluster view (see Figure 2). The hexbin display shows the results in the manner of a heatmap, while the cluster view only presents the five most commonly used route clusters, which may sometimes be more helpful, depending on the user's goals. He can also select a subway route and see the number of taxi rides which overlap with the route. This gives the user a first impression of the general distribution of taxi rides in NYC, as well as a rough overview on how many taxi rides could be substituted by an additional subway route. The goal of the exploration mode is to give users the possibility to explore the subway network and to reveal possible weaknesses in its structure. If most

[5] `https://data.cityofnewyork.us/Transportation/Subway-Entrances/drex-xx56`

[6] `http://web.mta.info/nyct/service/`

[7] `https://nodejs.org`

[8] `https://angularjs.org/`

[9] `http://leafletjs.com`

[10] `http://d3js.org`

[11] `https://github.com/fawind/nyc-subway-optimization`

rides starting from a station are heading to an area where no station exists, adding a station there might be beneficial for a big group of people. This step is also really important as a prerequisite for the later optimization. In order to suggest new public transport lines, it is necessary to know how well the already existing transport system performs. An indicator for a possible enhancement of the network for example is given by frequent taxi drives between two stations, which are not directly connected.

In order to find the areas where most taxis are heading or starting given a station, the pickup and drop off points have to be clustered. However, the first step is to apply the filters, which yields the positive effect of drastically shrinking the data size, which in turn makes the clustering possible in real time. The first filter consists of taxi rides which either started or ended close to the selected station, based on whether the user chose to inspect incoming or outgoing rides in the settings. In a second step, the other filters get applied. This time only taxi rides, that belong to the selected time interval, are considered. As a result we want to group and count taxi rides which start or end in a similar area. In order to achieve this, the map of New York City is split up into a grid. For each field in the grid, the taxi rides that end or begin here, are summed up. Accordingly, each field in the grid yields the number of taxi rides that connect this field with the station. The size of the fields depends on the user's preference. A smaller grid size induces a longer calculation time, but the results are much more precise.

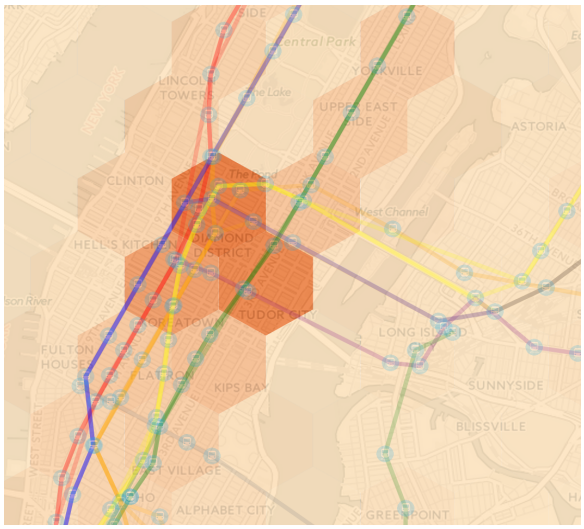

Figure 2: A screenshot of the Exloration mode using the hexbin view

In the end, when in cluster view, the top five clusters are visualized on the map for an intuitive overview. The user can clearly see the most traveled routes from a selected station. In order to calculate the amount of rides covered by a subway route, a similar approach is used. A subway route is defined by multiple connected stations. For each station all rides starting near the given station and ending near another station of the line are getting summed up. This calculation can be computed on the raw data or on the graph data structure proposed in Section 3.4, which results in a faster calculation speed, since the taxi rides are already pre-aggregated.

3.4 Optimisation Mode: Suggesting Optimizations for Existing Networks

In this second mode, the user can create custom optimizations for the subway network. By adjusting the settings, the user can retrieve proposed subway lines and can see how many potentially taxi rides these lines cover. To achieve this, the whole dataset had to be made spatially analyzable by converting all taxi rides into a graph structure. The outcome of this procedure is a directed weighted graph. It is comprised of a vertex for each cluster and edges for all taxi connections between two clusters. In this specific case, the weight of each edge is the amount of connections between two clusters[12].
In comparison to the exploration approach, almost no filters can be applied to the data set when building the graph, except for data cleaning filters to remove duplicate and invalid trips (as discussed in Data Quality). In the Optimization, the data set is aggregated dynamically, which means that the aggregation parameters can be changed based on the user's requirements.
Additionally, a link to the subway network data is created in the aggregation. Each edge has two attributes that declare its vertices as either *in reach of a subway station* or *not in reach of a subway station*. Applying filters on the resulting graph allows a decrease of the amount of edges. For example edges that only cover less than a few hundred rides in four years are not considered to be relevant by our algorithm.

Figure 3: The application in Optimization mode, showing a collapsed graph of the 300 heaviest edges with a maximum length of 3 km per edge, generated with 1 km² squares[14]

As a result, the graph can be displayed and explored. In-

[12]Other weighting possibilities are proposed in the Future Work section

stead of finding hotspots via the hex bin view, this view allows to visualize the travel directions and likely profit margins. For example visualizing the profit margin is achieved by using the average fare as a weighting function for the graph edges (see Figure 3). The graph has further information on the underlying subway network. This allows us to get all edges that are not covered by New York's subway infrastructure. Figure 4 shows a snapshot of the developed tool. It depicts a filtered set of the generated edges displayed on the map.

In the following we will build on the approach, described in the exploration mode section. To calculate the weighted edges, we analyze the destination of the rides from a certain pickup area. A grid with a grid size of 1 km^2 is laid over the area that should be analysed. It is precise enough since a radius of around 500 meters around a station is in reach by foot. Additionally, for each cell we determine whether there is a subway station inside or not.

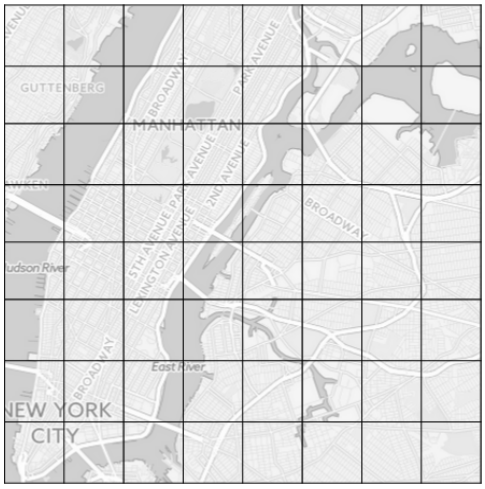

Figure 4: A sample cluster with big squares and thus only little precision mapped on New York

The aggregation steps are thus:

1. Cluster the whole given area in 1 km^2 squares
2. Mark each cluster as either `containing subway station` or `not containing subway station`
3. Collect all outgoing rides for each square
4. Group all outgoing rides by their destination point[15]
5. Sum up all rides that start and end in the same cluster. This count corresponds to the weight of the vertex

[15]It is possible to add the distance, the summed up taxi fare or other existing data in the given tables to the edges

The steps result in a data structure that can be saved in the following format.

```
TABLE "RIDE_EDGES" (
    LAT_OUT FLOAT,
    LNG_OUT FLOAT,
    STATION_OUT BOOLEAN,
    COUNTS INTEGER,
    LAT_IN FLOAT,
    LNG_IN FLOAT,
    STATION_IN BOOLEAN,
    DISTANCE FLOAT
)
```

In this example the distance is stored with the edges and the basic approach with the number of rides as weight is realized.

3.5 Finding Optimized Routes

With the previously proposed graph data structure it is possible to analyse the taxi ride distribution in real time. Depending on the chosen structure[16], different analyses are possible. In our case the graph data structure is used to generate subway lines that cover as many taxi rides as possible, that are not already covered by another subway line. Figure 5 depicts an example of a proposed line. This line is based on found gaps in public transportation on the eastern side of Manhattan.

Figure 5: Part of the proposed line 1, which is also included in Table 3

In order to suggest a subway line optimisation, the taxi data has to be correlate with information on the public transport infrastructure. This is done by building the graph structure as explained in the optimization section. The resulting graph contains three different kinds of edges:

[16]As described in the optimization part, the weight can be defined differently based on the requirements

1. Edges both starting and ending in areas already served by public transport
2. Edges starting in an area already served by public transport and ending in an unserved area or vice verse
3. Edges both starting and ending in areas unserved by public transport

The first kind is not useful when it comes to extending public transportation system, as there already is an existing route. The second and third kind cover routes or connections that can be used as starting points to propose new lines. A modified path finding algorithm can be applied to find the most promising new lines. Several constraints have to be taken into account when designing such an algorithm. Firstly a line is not supposed to wind itself, which means it has to go into one direction and not go back or form circles. Secondly, a line has to perform as well as possible, which will be measured in terms of ride coverage. Lastly, vertices have to be pulled together if they are within a given range. This implicitly changes the graph and improves the performance of lines. The designed algorithm is explaind in pseudo code. Given is the defined graph, a range to search for vertices, and the way to determine the weight of an edge (either the pure weight as e.g. the amount of rides is used, or it is put into correlation with the edge's length, simulating a higher price for long line segments).

```
get_heaviest_edge();
while vertices in range:
    vertices = get_vertices_in_range()
    heaviest_vertex = max(vertices)
    add_vertex_to_graph(heaviest_vertex)

return line
```

The algorithm starts by selecting the edge with the highest weight as the starting point. Starting from this edge, all vertices in the given range are collected. From these vertices the vertex with the highest weight gets added to the line. The newly added vertex acts as a new starting point and the process is repeated. The procedure ends when no more vertices are found in the given range.

4. RESULTS

An approach to visualize and explore all given taxi rides dependent on New York's subway network has been designed and developed. Furthermore, it was made possible to reveal often travelled routes with the weighted directed graph, which made it possible to make a prototypical suggestion for a new public transportation line, that would improve the existing infrastructure best. Using an in-memory database enables most of the analyses to be done in real time, which permits the user to directly explore the dataset in an interactive way. By analysing the New York City subway infrastructure with the described exploration tool, it can be observed that it already performs very well, in the sense that the most commonly traveled taxi routes are already heavily covered by the network. But there are still multiple connections, where an additional subway line could replace a lot of taxi rides. Using the previously described approach, it was possible to create a new line (suggested line 1, see Table 3), that covers a total number of close to *112,895,000* taxi rides during the given four years, which conforms to an average of about *77,325* taxi rides a day. Table 3 shows two proposed lines (including the aforementioned one) using default filtering (not more than 500 edges and edges shorter than 2200 meters) on the given graph. These newly suggested lines perform better than most existing lines, but not as well as the best lines of the existing subway infrastructure (lines 1 and 6).

Line	Ride Coverage	Length	Rides / km
Line 1	120,902,450	23.41 km	5,164,564
Line 6	126,527,867	23.27 km	5,437,381
Line 7	13,385,743	14.62 km	915,577
Proposed 1	112,894,793	24.40 km	4,626,835
Proposed 2	94,385,442	23.44 km	4,026,682

Table 3: Taxi ride coverage over 4 years and length of existing subway lines in comparison to subway lines proposed by the application. The rides per km column describes the average coverage efficiency of one km of a line

If we assume that all the people using the taxi rides would instead have used the new subway line to get to their destination, the new line would have reduced all taxi rides by *15.94%*. With an average gas usage of $^{11}/_{13\text{km}}$ and an average trip length of 4.2 km, this would also have saved a total amount of *24,981.92* liters of fuel per day, or *9,118,401.92* liters in only one year. As shown in the approach section, the application can propose new subway lines that meet the expectations in making many taxi rides obsolete.
When only investigating the raw numbers, the potential of our generated lines seems to be immense. However it is questionable, whether all citizens would then use the new line and pass up taxi rides altogether. During our analyses we found out that a lot of taxi rides are already covered by a direct subway connection. When looking at the history of New York City, the question whether our new subway line can really make that many taxi rides obsolete remains open. New York citizens are known for prefering the cab, possibly because of the public transportation's bad reputation.

5. FUTURE WORK

This paper describes a prototypical approach to make a huge data set visually accessible and combining the underlying data with other sources. The possible use cases for such a concept are not restricted to the approach presented in this work.

A possible addition would be to build multiple optimization graphs for various time intervals. This would allow the user to find the best extension for any given time or date. For example different usage patterns are be expected during day and night time. It is also possible to include other public transport means such as busses or trains. As the algorithms work in a very generic manner and only require a collection of stations, it would be a manageable task to also add the station data of the busses, ferries, and trams of New York. This would open the possibilities to find new stations for these transit methods, which may be a more frequent and cost sensitive use case than building an entire new subway line.

As already mentioned earlier, it would be also conceivable to incorporate data generated by other transportation means, in addition to solely using taxi data. The necessary information could originate from car and bicycle sharing services, or private car rides. Since the current data set it restricted to New York City, new knowledge can only be gained on that city's transit system, but especially growing cities in developing countries could benefit from such an approach. China's economic expansion for example also generates big demands for larger housing and thus middle sized cities are right now inflating to metropolises, creating an immense need for efficient public transportation systems.

Beside the optimization of the public transit network, other potential use cases are touched in this paper. It would be interesting to see whether higher demands for taxi rides and the expected profit could be predicted in real time. This would enable giving guidance to idle taxi drivers in order to maximize their profits and minimize their waiting time. In order to gain these insights, the graph structure can be used with the average revenue and the amount of rides per edge. The taxi driver could receive a recommendation based on the frequently travelled routes.

6. CONCLUSION

The main goal, whether local taxi ride data could yield knowledge for improving the public transportation system's infrastructure could be answered positively. The results showed, that the suggested lines and station connections based on the taxi ride data can definitely yield substantial improvements to reduce the need for taxi. The best example for this was the newly proposed line 1, which would ideally make *77,325* taxi rides dispensable every day. Carrying out such infrastructural improvements could of course bring about less ride reductions than calculated, but this does not negate the original proposition. Expanding this approach at the entire public transport system yields huge benefits for modern city planning and is suited to optimize transit infrastructure efficiency by reducing traffic, gas usage, and financial expenses.

7. REFERENCES

[1] S. K. Chang and P. M. Schonfeld. Multiple period optimization of bus transit systems. *Transportation Research Part B: Methodological*, 25(6):453–478, 1991.

[2] S. I. Chien* and Z. Qin. Optimization of bus stop locations for improving transit accessibility. *Transportation planning and Technology*, 27(3):211–227, 2004.

[3] G. Ganeshapillai, J. Brooks, and J. Guttag. Rapid data exploration and visual data mining on relational data. 2014.

[4] C. E. Mandl. Evaluation and optimization of urban public transportation networks. *European Journal of Operational Research*, 5(6):396–404, 1980.

[5] J. Toole. Visualizing road usage of taxi cabs. *MIT Big Data Challenge: Transportation in the City of Boston*, 2014.

[6] B. Tucker. Putting data into practice: Lessons from new york city. *Education Sector Reports*, 2010.

[7] V. R. Vuchic. *Urban transit: operations, planning, and economics*. 2005.

[8] Y. Yang, S. Colak, J. Toole, S. Desu, and L. Alexander. Finding patterns in taxi demand. *MIT Big Data Challenge: Transportation in the City of Boston*, 2014.