Optimization Problems in Bus Route Network Design

Student Name: Saatvik Jain
Roll Number: 2016261

BTP report submitted in partial fulfillment of the requirements
for the Degree of B.Tech. in Computer Science & Applied Mathematics

on 10 July 2019

BTP Track: Research Track

BTP Advisor
Dr. Pravesh Biyani

Indraprastha Institute of Information Technology
New Delhi

Student's Declaration

I hereby declare that the work presented in the report entitled **Optimization problems in bus route network design** submitted by me for the partial fulfillment of the requirements for the degree of *Bachelor of Technology* in *Computer Science & Applied Mathematics* at Indraprastha Institute of Information Technology, Delhi, is an authentic record of my work carried out under guidance of **Dr. Pravesh Biyani**. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

....
...(student's name)...

Place & Date: ....

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

....
..(advisors' name)...

Place & Date: ....

Abstract

Ride-sharing has been one of the emerging solutions to the ever-increasing problem of congestion and air pollution in cities. However, the effectiveness of ride-sharing in reducing private vehicles is contingent on its efficiency while maintaining economic viability from the perspective of both the operator and the passenger. This work proposes a ride-sharing framework that jointly takes into account the quality of sharing as well as the operator revenue while fixing the consumer cost. It formulates a weighted graph coloring optimisation problem that has the flexibility to incorporate various factors that promote ride-sharing efficiency while maximizing the operator revenue. Application of the method on two separate datasets from two different geographies, Bangalore, India and New York City, USA provides promising results vis-a-vis the state-of-the-art approaches. The proposed framework has the potential to increase the occupancy of vehicles, enable sharing of bigger vehicles while assigning naturally shared rides and at the same time, the framework can easily be scaled to cater to dynamic ride sharing requests without much modification. Further, when compared to there being no ride sharing, on an average, the percentage of riders sharing rides and percentage of vehicles being shared is more than 85% and 75% respectively with over 60% reduction in the number of vehicles being used. The key insight obtained from this work is that incorporating quality and efficiency of shared rides has the potential to enhance operator revenue as well.

Keywords: Ride-sharing, graph coloring, Ride Matching and Reservation, Theory and Models for Optimization and Control, Commercial Fleet Management

Acknowledgments

I would like to thank my advisor, Dr. Pravesh Biyani who has been guiding me throughout the course of the project.

Contents

1 Introduction **1**

2 Related Work **3**

3 Problem Formulation **5**

3.1 RSP: A Closer Look 5

3.2 Dealing with Constraints: Admissibility Graph 6

3.3 An Optimisation Problem for RSP 6

4 Proposed Solution **8**

4.1 Stage 1: Graph Coloring 9

4.1.1 Standard Coloring 9

4.1.2 Weighted Vertex Coloring 11

4.2 Stage 2: Minimum Cost Vehicle Allocation 11

4.3 Stage 3: Least Distance Route Allocation 12

4.4 A note on dynamic ride sharing 13

5 Implementation Details **15**

6 Experimental Results **17**

6.1 Evaluation Parameters 17

6.2 Experimental Studies 19

6.3 Comparison with other Methods: A Note 20

7 Future Work **21**

8 Progress and Additional Comments **22**

Chapter 1

Introduction

The emergence of Mobility on Demand (MoD) systems has lead to a surge in app-based taxi and shuttle services all around the world. The two immediate consequences of this mobility revolution are: first, an addition of more cars on the streets of cities often leading to more congestion and air pollution and second, the rise of many taxi aggregators that organize individual traditional taxis under one app-umbrella. Consequently, on-demand ride-sharing as a product has emerged as a serious option thanks to the aggregation of taxis as well as real-time availability of passenger demand. Ride-sharing has tremendous potential not only in making the on-demand taxi services profitable but also environmentally sustainable [11, 16] by decreasing both congestion on roads and pollution in the air. For instance, authors in [2] report that ride-sharing can result in a decrease in as much as 70-80% of taxis in New York City. Finally, an efficient ride-sharing setup will also lead to a decrease in private car ownership, especially when driverless cars arise in the future.

The problem of matching taxis/shuttles with riders or the *Ride Sharing Problem (RSP)* hitherto called a dial-a-ride problem [7] has taken center stage in the intelligent transportation community. The ride-sharing problem is a problem of matching the available fleet with a given capacity with the real-time passenger demand that optimizes certain criteria subject to the various quality of service constraints. The objective function could be a function of distance travelled, operator revenue maximization or passenger cost minimization. Similarly, a possible example of constraints could be the distance covered by a taxi or the total travel time of individual passengers. Indeed most of the work, including recent works [1–3, 12, 18] have been devoted to one of the above-mentioned objective function along with the appropriate set of constraints.

One of the aspects that has not been studied in depth is the practicality of the solutions that emerge in the above works. For instance, a shared ride between two passengers where the actual sharing is only for a small fraction of the total distance travelled may not be practical in terms of implementation, both operationally and financially to the riders and the driver. Authors in [6] show how taxi aggregators like Uber have non-transparent pricing schemes which leads to unfairness in the pricing to the riders while using their systems. Hence, both, the cost of sharing the ride to the passengers and the profit of the driver needs to be factored in whether a potential set of trips should be combined.

This project aims to re-look at the ride-sharing problem by additionally incorporating both operator and driver economics and the efficiency of ride-sharing in a common metric. The efficiency of a shared ride is defined by *occupancy index* which is the ratio of the total shared distance travelled and the total distance travelled by a vehicle. A higher occupancy index leads to higher revenue for the driver as well as a more reasonable cost for the passengers. Further, the other constraints that define the practicality of sharing, for instance, gender (pertaining to safety), time of the day and even personal preferences for last mile, etc have not been incorporated, in the ride-sharing literature. The modelling of the efficiency of a shared ride as well as driver economics along with the above constraints requires a detour from the traditional optimisation framework.

To incorporate above necessities, ride-sharing is modelled using a graph coloring framework. The nodes of the graph are the riders and the edges indicate the ability of two riders to share their rides together. In effect, the constraints mentioned above can be encoded in the edges of the graph. Further, weights are assigned to the nodes of this graph which pushes the algorithm to share rides with maximum overlap. Graph coloring algorithms and their variants are applied to get clusters of people who can travel together. Once that is obtained, vehicles are allocated to the obtained clusters. Finally, a route is provided to the vehicles to follow so that they can cater to the demands efficiently. It is made sure that the entire pipeline is completed while remaining unbiased to both the operators as well as the riders. It is observed that assigning weights helps in increasing the occupancy index and therefore the operator revenues. It also helps in a better assignment of vehicles and avoids forced sharing. Simulation experiments are performed at two different geographies, Bangalore, India and New York City, USA and encouraging results and new insights are encountered in the ride-sharing problem.

Chapter 2

Related Work

A lot of research has been going on in the area which attempts at tackling at least the bare bones of the ride sharing problem: taking a system of rides and some information about it and combining the rides in the system in such a way that the new obtained system is optimal in a pre-defined context. Most of the work done related to the problem thus far have approaches that range from graph theory, optimisation and to methods that provide a more geometric approach to RSP.

In their method, Paolo Santi et al [18] model the problem in terms of a hypergraph and then use maximum matchings in it as shared rides by maximising the number of shared rides in the system or equivalently, minimize the total time required to run all the trips. This method, however, can not be solved for matching more than 3 rides together and is not flexible. NAH Agatz et al [1] also use the idea of matching to get a solution of a slight variant of the problem in which they match riders with candidate rides on the basis of the cost of sharing. They solve weighted maximum matchings of the bipartite graph that maximizes the amount of distance saved by sharing rides. Another notable work, the work done by Yongzheng Jia et al [12] brings into consideration the angle of driver and passenger profit maximization and solves the original problem through the maximum node disjoint problem of graphs. This, so far, is one of the few works done which considers profit maximization, however, it only considers configurations where riders are picked up and dropped off by drivers and then the driver moves on to next rider, hence not justifying ride-sharing but favoring the profit aspect of it.

Alonso Mora et al [2] extend the strength of a graph-based approach by starting with a greedy allotment of riders to vehicles using maximal cliques and then optimize it using constrained optimisation with an objective to minimize the total delay in time over all rides being conducted. Their method also takes into consideration optimal vehicle routing on the basis of the passengers allotted to it and then comment about vehicle re-balancing based on demand, but, they work under the assumption that all these above factors will directly incorporate profit maximization into them and at the same time, the authors claim that their method reaches the optimal solution if given enough time to run, however, their method involves finding cliques in an input graph, which is an NP-Hard problem, hence, not aligning with their claims. More recent work, done by Xiaohui Bei et al [3] formulate the work as a combinatorial optimisation problem by optimizing the distance travelled, they, however only consider the combination of two trips at a time, leading to vulnerabilities in their system.

JF Cordeau et al [7] summarizes the algorithms and methods proposed so far for the dial-a-ride problem. Their review also lists down limitations and constraints for all approaches, which hints towards the limited nature of solutions to the problem. These approaches range from the above mentioned methods to methods using a geometric approximation of the localities, one such notable work, the one done by Shuo Ma et al [14] considers several aspects of RSP and give a dynamic solution which considers aspects like pricing and usability and attempts to minimize the total increase in travel distance for all riders.

As can be seen, all works done so far have a single focus on which their entire method is built. This focus ranges from maximization of profit, shared rides or minimization of extra distance travelled, extra time taken and then work under the assumption that this optimisation will lead to control over other parameters directly, which, however, is not the case.

Chapter 3

Problem Formulation

This chapter describes in detail the Ride Sharing Problem and introduces all the necessary tools required for getting to the solution of the Ride Sharing Problem. Section 3.1 first introduces the RSP formally along with constraints pertaining to its quality as mentioned earlier, Section 3.2 then introduces necessary tools required for the proposed framework in order to arrive at a solution to RSP and finally Section 3.3 presents RSP as an optimisation problem and comments about its complexity.

3.1 RSP: A Closer Look

RSP starts with a set of N riders, where each rider r_i is denoted by a pair of locations (s_i^r, e_i^r). Here, s_i^r and e_i^r are the source and the destination locations, respectively. It is assumed that the cost to the rider i in travelling from s_i^r to e_i^r depends only on the shortest distance $d_{i,min}$ between them. This is fixed in order to ensure that the model does not bias itself against riders by requiring riders to compensate for travelling extra distance that arises due to ride-sharing. Let the fleet of m vehicles be denoted by set $F = \{c_1, c_2 ... c_m\}$. The passenger carrying capacity of the j^{th} vehicle is given by κ_j. The cost incurred by the operator to run a taxi when it travels a distance d is a function $\theta(d, p_j)$, where, p_j is the per km cost of the j^{th} vehicle which in turn depends on its capacity and is known apriori.

RSP seeks an allocation of the set of riders to a given fleet of vehicles such that the overall cost incurred by the fleet is minimized while satisfying the following quality of service constraints to the passengers:

1. *Distance*: The distance travelled by each rider cannot exceed $(1 + \delta)\, d_{i,min}$.

2. *Time*: Likewise, the time taken for a rider to travel between her source and the destination should not be more than $(1 + \delta)$ times the time taken to navigate through the shortest path.

3. *Capacity*: The number of passengers assigned to a vehicle cannot be more than its capacity.

3.2 Dealing with Constraints: Admissibility Graph

Before discussing the ride-sharing optimisation problem, first a framework is defined to handle the above mentioned time and distance constraints. A notion of *admissibility* of two riders is introduced which is their ability to share their rides. For each pair of riders, it is calculated whether they are admissible or not based on the above-mentioned constraints. Then, an *admissibility graph*, $G = (V, E)$ is formed whose nodes represent riders and an edge joins two nodes if either of the riders in a pair can share their ride with the other rider without violating its time and distance constraints. It is worth noting that these constraints are flexible and the RSP problem can be extended to other conditions between two riders such as the gender of occupants, time of the day when a ride is to be shared and, age, etc. Further, for each node in the graph, another value is assigned to it, known as *Vertex Weight*. Vertex weight is a positive integer function $f : V \to \mathbb{N}$ such that $f(v_i) = w_i$. Vertex weight is set to the average distance of the source of the corresponding rider from all the other riders' sources. As it will be seen later that the weights in the admissibiity graph will play a crucial role in ascertaining the "quality" of ride-sharing from both operators and the passengers perspective.

3.3 An Optimisation Problem for RSP

Having build the admissibility graph G, the RSP is now written as an optimisation problem. Let x_{ij} be a binary assignment variable that is assigned 1 if the rider i is allotted vehicle j, and is set to 0 otherwise. Likewise, let z_j be a binary variable that is 1 if vehicle j is not empty and 0 otherwise. Let $M = max\{\kappa_j,\ \forall j = 1 \dots m\}$. Then RSP is stated as follows:

Problem P1

$$\text{minimize} \sum_{j=1}^{m} \theta(d, p_j)\, z_j \tag{3.1}$$

$$\text{s.t.} \sum_i x_{ij} \le \kappa_j \qquad \forall j = 1 \dots m \tag{3.2}$$

$$\sum_j x_{ij} = 1 \qquad \forall i = 1 \dots N \tag{3.3}$$

$$\sum_i \frac{x_{ij}}{M} \le z_j \qquad \forall j = 1 \dots m \tag{3.4}$$

$$x_{i_1 j} + x_{i_2 j} \le 1 \qquad (i_1, i_2) \notin E,\ \forall j = 1 \dots m \tag{3.5}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i = 1 \dots N, \forall j = 1 \dots m \tag{3.6}$$

$$z_j \in \{0, 1\} \qquad \forall j = 1 \dots m \tag{3.7}$$

The above formulation minimizes the total cost incurred by all the vehicles. Since the rider cost is fixed for each rider and is a function of only its source and destination, minimizing driver cost would lead to assurance of both maximum operator profit and minimum rider costs. The constraint (3.2) asserts that no vehicle is filled beyond capacity while constraint (3.3) ensures that each rider is allocated exactly one vehicle. (3.5) constraints that all pairs of passengers in a vehicle should be admissible. Finally, (3.4) together with (3.6) ensures that vehicle assignment variable is assigned 1 if there is a passenger allocated to it and (3.7) ensures that rider assignment variable only takes values 0 or 1.

Lemma 3.3.1. *The Ride Sharing Problem is NP-Hard.*

Proof. This lemma can be proved by reducing the *Vertex Coloring Problem (VCP)* to RSP.

An instance of VCP is reduced to an instance of RSP as follows: Given a graph $G = (V, E)$, i) Set the vertex set of G to be the set of all riders in the system, ii) Calculate admissibility of each pair of riders and add an edge between riders if they can not travel together based on the set constraints.

A vertex coloring of G will lead to division of the vertex set into color classes. Within a color class, no two vertices are joined by an edge, which means that for our case, no two riders are not admissible. This ensures that the color classes are clusters of people who can travel together, which is can be seen as a solution to the case of RSP when cost of running any vehicle is a fixed constant and each vehicle has infinite capacity.

If cost of running any vehicle is a fixed constant, the cost of allotment is not affected by the kind of vehicle being used in the allotment, and if a vehicle is of suitably large capacity, then any number of people can be fit in it. Hence, if a list of riders can be obtained who are admissible, allotment of a single vehicle to them would be a valid allotment of minimum cost, which is a solution to the RSP. Hence, it is only necessary to find a division of the set of riders into different clusters where each pair of riders is admissible which VCP provides through the admissibility graph.

The above construction completes the proof that RSP is a generalization of VCP. Since VCP is NP-Complete [10], therefore it can be concluded that RSP is NP-Hard. □

Chapter 4

Proposed Solution

Consider any vehicle in a feasible solution to problem P1. Either the vehicle is empty or it has riders that are admissible with each other. Since each passenger can be allocated only one vehicle, the allocation of passengers to vehicles can be seen as the partition of the nodes of the admissibility graph G. Further, if $\theta(d, p_j)$ is a constant and if the capacity of vehicles κ_j is a large value, then the ride-sharing optimisation problem is a problem of finding the minimum number of vehicles that can accommodate the riders. The above two observations motivate the connection between RSP and the graph coloring problem. The main objective of graph coloring is to partition the set of vertices into a minimum number of subsets called color classes such that in each color class, all pairs of vertices do not have an edge between them.

The proposed passenger to vehicle assignment algorithm for problem P1 works in three stages (Algorithm 1). In the first stage, a partition of the nodes of the admissibility graph (riders) is obtained into mutually exclusive and collectively exhaustive subsets. Next, vehicles are assigned to these subsets such that the total cost of running all the vehicles is minimized. Owing to fewer classes than the number of riders, the second stage problem is of much lower dimension. Once it is known which set of vehicles are assigned to which subset of riders, in the final stage, a route is computed for each of the vehicle such that the total distance travelled for each vehicle is minimized without violating the original passenger constraint. This chapter now discuss the three stages of the proposed algorithm and then concludes by describing how the algorithm is to be updated if riding requests are dynamic in nature (Algorithm 2).

Algorithm 1 Solution to RSP

Require: Solution to RSP from input S, the set of riders and F, the fleet of vehicles

$G \leftarrow$ Admissibility graph corresponding to S

$A \leftarrow$ Set of color classes of vertex coloring/WVC of $\overline{G}$

$V \leftarrow$ Vehicle and passenger allotment from F to all color classes of A corresponding to minimum running cost

for vehicle in V **do**
  $R \leftarrow$ Best route for riders in vehicle
  route(vehicle) $\leftarrow$ R
end for

return V with routes

4.1 Stage 1: Graph Coloring

The main aim of graph coloring is to partition the riders (nodes) into possible subsets (color classes) such that each subset consists of riders who can potentially share their rides. To this end, the complement $\overline{G}$ of the constructed admissibility graph G is considered, whose vertices v_i and v_j are connected by an edge if the riders r_i and r_j are *not* admissible. It implies that in the coloring of $\overline{G}$, riders r_i and r_j will be in the same class if they can share their ride. Recall that all the passenger constraints have been incorporated via the construction of G. Two coloring schemes are proposed - standard coloring and weighted vertex coloring for obtaining color classes of $\overline{G}$. It will be seen that the latter provides a better ride-sharing than the former in terms of the proposed optimisation cost function.

4.1.1 Standard Coloring

The objective of standard coloring is to minimize the number of color classes being used. Smaller the number of color classes, higher the number of riders in each class. This ensures that larger capacity vehicles can then be allocated to these color classes and truly achieve a high sharing quality. Since graph coloring is a NP-Hard problem [10], standard coloring can be solved only heuristically using greedy algorithms like DSATUR [5] or SEQ [13]. Notwithstanding the ability of standard coloring to obtain fewest vehicles required for RSP, the subsets so obtained may not be optimal for the operators. Figure 1 illustrates the above argument. Using the standard coloring procedure, red and blue riders are paired while the green rider goes alone. This, however, may not be a desirable allotment, since there exist alternative sharing possibilities in

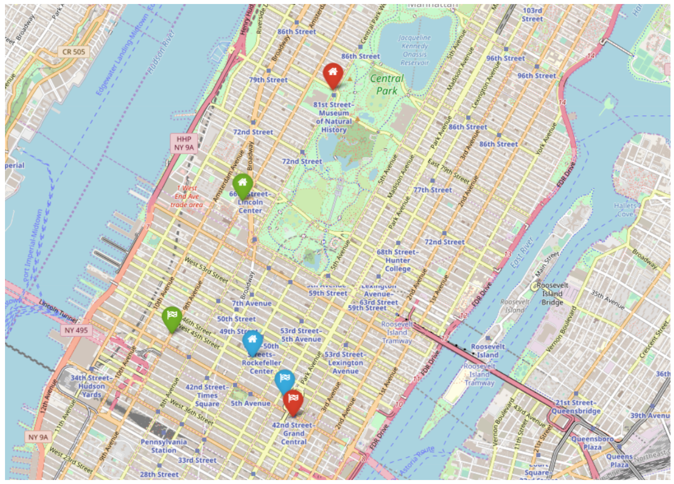

Figure 4.1: A Case Study in NYC: standard coloring vs weighted coloring

which total distance travelled by vehicles is lesser, hence being more feasible to the operator and at the same time, in the current allotment the red rider enjoys the monetary perks associated with ride-sharing in spite of sharing the ride for a small distance only.

One of the explanations why standard coloring may not fit the operators perspective is the fact that standard coloring does not take into account the operator cost function θ. In other words, the color classes should be formed in a way that vehicle allotment to them leads to maximum occupancy rather than minimum number of vehicles used to maximize revenue of operators. Since the rider cost (price) is already fixed and is only dependent on the shortest distance between her source and destination (and not the actual distance travelled), a higher occupancy implies higher revenue for the driver. In effect, seeking an allotment that would pair riders that are closer to each other and have more overlapping routes is desirable. To achieve this objective, weights of vertices can be used. If a coloring of vertices can be obtained that then takes into account both admissibility of riders and proximity of their corresponding vertex weights to color vertices of the graph, better occupancy rides can be obtained hence filling the void created by standard coloring. As shall be explained later, the weighted vertex coloring (WVC) scheme results in allocation that achieves the same. Consequently, in figure 4.1, after WVC is used, red and green riders are paired together to share their rides and blue rider goes alone, which reduces the total travel distance of the vehicles and justifies sharing of the red rider.

It can be observed that for any given vehicle, closer the riders' sources are to each other, more will be the average occupancy over the entire distance travelled by the vehicle. This is because now lesser distance is travelled in picking up riders as compared to when sources are not nearby. This indicates that weights should be modelled in a way such that they take into account the distance between sources of riders. Hence, weight of a vertex is set to be the average distance between the source of that rider and the sources of all the other riders in the system.

4.1.2 Weighted Vertex Coloring

To obtain an allocation of riders based on proximity of their weights, a variation of the graph coloring problem is used, namely the *Weighted Vertex Coloring Problem (WVC)*. In the weighted vertex coloring (WVC) problem there is an associated cost with each color class. Let $A = \{A_1 \ldots A_u\}$ be the set of color classes for any given vertex coloring. The cost of each color class is equal to the maximum of the vertex weights of vertices in the color class. Let f be the function mapping vertices to their weights. The objective of WVC is to minimize the sum of costs of all color classes, and this can be written as minimizing the following function g.

$$g = \sum_{i=1}^{u} \max f(v_j), \forall j \in A_i \tag{4.1}$$

It is easy to see that with this objective function of WVC in consideration, admissible riders that have similar weights would be preferred to be put in the same color class. Similar weights would lead to similar average distances between sources, which when combined with admissibility leads to rides with improved occupancy.

4.2 Stage 2: Minimum Cost Vehicle Allocation

The graph coloring methods in the previous stage partitions all the riders into clusters containing admissible riders. Once these clusters are obtained, the next step of allocating vehicles to these riders such that the total operator cost is minimized is to be carried out. Since in a given cluster, all riders are admissible, instead of allocating vehicles to riders, vehicles are allocated to clusters such that cost is minimized and then riders are assigned to these vehicles. Recall that set F contains all the vehicles. A new set $F^{'}$ is defined that maintains a record of the unique capacity vehicles available in F. Let $F^{'} = \{c_1^{'} \ldots c_l^{'}\}$ be such that there are n_j vehicles of type $c_j^{'}$ available and let y_{ij} be an integer assignment variable that counts the number of $c_j^{'}$ type vehicles allotted to color class A_i. Then, a minimum cost vehicle allocation to the above obtained color classes can be obtained by solving the following optimisation problem:

Problem P2

$$\text{minimize} \sum_{j=1}^{l} \sum_{i} \theta(d, p_j)\, y_{ij} \tag{4.2}$$

$$\text{s.t.} \sum_i y_{ij} \leq n_j \qquad \forall j = 1 \dots l \tag{4.3}$$

$$\sum_j y_{ij}\, \kappa_j \geq |A_i| \qquad \forall i = 1 \dots u \tag{4.4}$$

$$y_{ij} \in \mathbb{Z}^+ \qquad \forall i = 1 \dots u, \forall j = 1 \dots l \tag{4.5}$$

Here, the objective function is the total cost incurred by the operator on the assignment of vehicles. Constraints in (4.3) ensures that no more than available vehicles are ever used. Constraints in (4.4) require the allotment to be such that each color class gets sufficient vehicles to accommodate all riders in the color class. Finally, constraint (4.5) makes sure that y_{ij} is a positive integer

Problem P2 is an Integer Linear Programming problem (ILP). To solve the above, the dual simplex algorithm is used that applies the simplex algorithm on the dual of a given problem in order to solve the problem, in this case, problem P2 [4, p.157]. With an allotment of the fleet of vehicles to the color classes, each rider is greedily assigned to the first available vehicle that is not yet full from the list of vehicles allotted to the rider's color class. Since problem P2 ensured that there are enough vehicles in each color class for assignment of passengers of it, it is guaranteed that this greedy allotment strategy will work as required.

4.3 Stage 3: Least Distance Route Allocation

Once the riders for a vehicle are decided, a route is wished to be fixed for the vehicle such that it travels the minimum distance while maintaining the constraint corresponding to each of the riders. Given a graph, it is known that the traditional *Travelling Salesman Problem (TSP)* requires a person (salesman) to visit all nodes of the graph by travelling minimum distance. This idea of a minimum distance route can be extended to a more general problem that includes constraints over certain nodes being visited before others.

Then the above criterion can be modelled to enforce the constraint that for a given rider, the source node should be visited before the destination node. This is known as the *Vehicle Routing Problem with Pickup and Delivery (VRPPD)* [9]. Since Travelling Salesman Problem is not in P, VRPPD is also at least NP-Hard. In VRPPD, the objective is for a set of vehicles currently at some given positions need to satisfy a set of transportation requests. VRPPD can be modelled to obtain the vehicle route for the given vehicle by having a single vehicle with start position as equidistant from all the sources and having source and destination of riders as the transportation requests. A solution to the problem is set to be the route of the corresponding vehicle. Further, it can be seen that due to the smaller problem size, i.e vehicles of not large capacities, the solutions obtained will be closer to optimal. Especially as is the case with vehicles being small capacity cars, where the solution can be obtained via direct enumeration of all the possible cases.

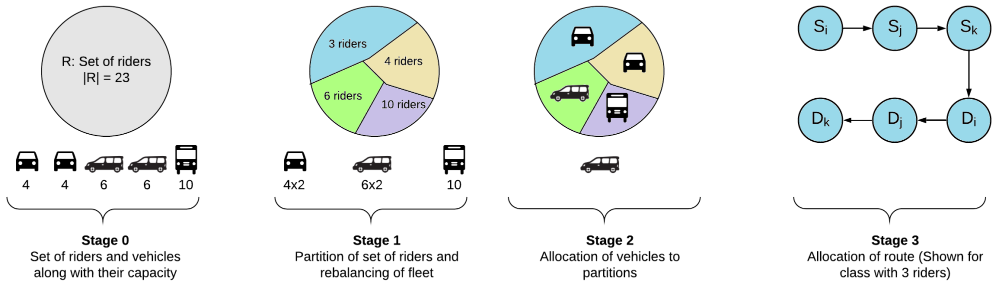

Figure 4.2: A representation of the proposed algorithm

Figure 4.2 shows a general workflow of the proposed framework for all the above discussed steps.

4.4 A note on dynamic ride sharing

This section explains briefly, how stages 1-3 will need to be modified in order to arrive at a solution to the dynamic RSP. The dynamic RSP is a case where instead of all details about rides known beforehand, new rides come as requests one by one over time. With requests coming over time, it is necessary to maintain the dynamic positions of vehicles.

The admissibility graph has now two kinds of nodes, riders and vehicles. A vehicle is said to be admissible with a given rider if all riders of the vehicle are admissible with the given rider. For any given rider who is currently in a vehicle, admissibility with another rider at a given point of time is calculated by checking if total distance and time travelled so far added with extra distance and time from current position does not exceed the previous set constraints. Weight of a vehicle is taken to be the average of the weights of all the riders in it, and is updated dynamically each time the vehicle position is updated by putting source of rider in the vehicle as the current position of the vehicle.

Once admissibility graph is obtained, stage 2 and 3 majorly remain the same. In stage 2, instead of solving problem P2 directly, it is first checked if the obtained color class has any vehicles or not, if it does, then the same greedy strategy is used to fill all the vehicles with riders of the color class till the time they are full and for the remaining riders, problem P2 is solved to allot them fresh vehicles. Stage 3 remains the same with the only update being that once more riders have been allotted to a vehicle already serving riders, route is re-calculated by giving current position of the vehicle as starting point and updating the sources of all the previous riders in the vehicle as the current position of the vehicle. These modifications pave the way for the algorithm to be run periodically at a set frequency to cater to the dynamic demands.

Algorithm 2 Solution to Dynamic RSP

Require: Solution to dynamic RSP from dynamic input R

```
while true do
    Update vehicle positions since last time
    Update active riders data according to new positions

    R ← Set of new requests since last time
    S ← Set of riders still in vehicles
    V ← Set of remaining non-empty vehicles
    G ← Graph with vehicle nodes from V and rider nodes from R + S
    A ← Set of color classes of G

    for class in A do
        if class has vehicle nodes and class has rider nodes then
            Fill vehicles with riders till possible
        end if
        V' ← Vehicles required to satisfy remaining riders of class
        V ← V + V'
    end for

    for vehicle in V do
        R' ← Best route for vehicle
        route[vehicle] ← R'
    end for

    Wait for more requests to arrive
end while
```

Chapter 5

Implementation Details

This chapter describes what steps are used in order to get the desired functionality from each step of the framework after the admissibility graph has been created to obtain the final results in Chapter 6.

1. **Standard Coloring:** Like mentioned earlier, a coloring of the vertices of a graph can be obtained using greedy algorithms, the most commonly used being the DSATUR [5] algorithm, which is used in our algorithm as well. At each step in the iterative algorithm, the vertex with the best *saturation degree* is colored greedily. It is coded using NetworkX, a Python library used for graphs.

2. **Weighted Vertex Coloring:** Malaguti et al. [15] suggest a combination of standard vertex coloring algorithms along with linear programming in order to solve WVC. At each step in their iterative algorithm, they require an ordering of the set of vertices on which methods of standard vertex coloring are applied and are optimized until the solution converges. Instead of a truly random ordering of vertices at each iteration, we use the reordering heuristics as suggested by authors in [8] at the end of each step which puts color classes in a certain specified order instead of vertices directly to maximize the chances of getting a better coloring at the next step.

3. **Vehicle Allocation:** Problem P2 is an Integer Linear Programming problem (ILP). To solve the above, we use the dual simplex algorithm that applies the simplex algorithm on the dual of a given problem in order to solve the problem, in our case, problem P2 [4, p.157]. It is coded in Gurobi, a linear program solver available for Python.

4. **Route Allocation:** The methods described by Schrimpf et al. [19] and Pisinger et al. [17] are used to solve VRPPD heuristically. It is coded using JSprit, a Java library developed for the same.

It is to be noted that all the above mentioned implementation has been done using Python 2.7.15 and is available as open source code on GitHub and all the above computations can also be visualized by visiting this website, the website, developed during the course of the semester provides an interactive way to simulate the algorithm. Once the user enters basic details required to run the algorithm, the algorithm runs in the backend and then the user can visualize all results on interactive maps and view both summarized as well as detailed statistics about the results of the algorithm. Figures 5.1, 5.2 and 5.3 show the website in action.

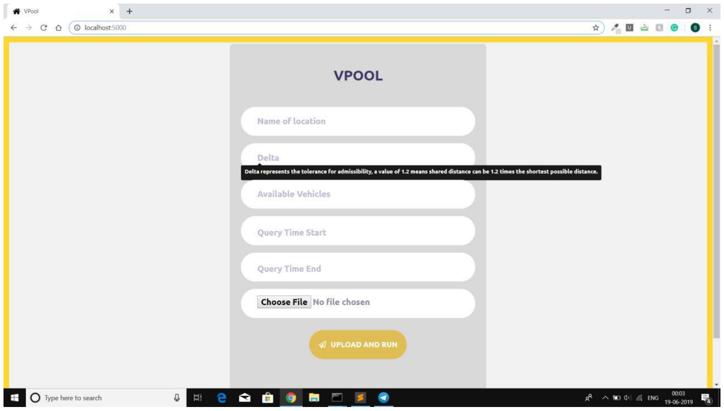

Figure 5.1: Form for website

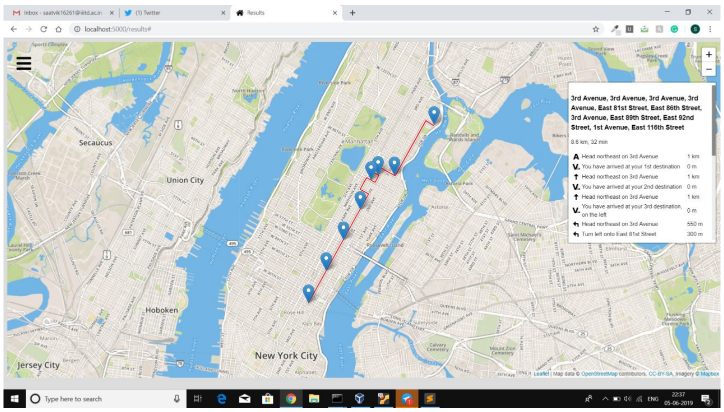

Figure 5.2: Route showed in website

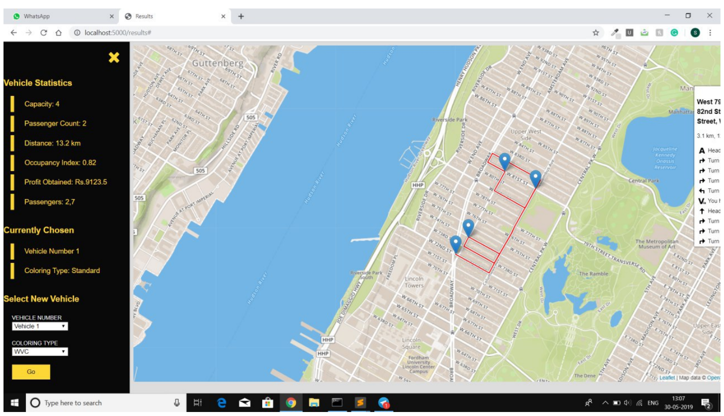

Figure 5.3: Statistics for vehicle

Chapter 6

Experimental Results

This chapter describes the performance of the two proposed graph coloring based algorithms for the RSP. The objective of the experiments are two fold:

- To evaluate the effectiveness of graph coloring based methods for the ride-sharing problem for the case of both cabs as well as higher capacity shuttles.
- To understand the impact of putting operator economics and its dependency on the quality of shared rides obtained through putting weights on the vertices of admissibility graph.

Section 6.1 describes the statistics being used to test the performed experiments, while Section 6.2 describes the experiments carried out and results obtained in them with Section 6.3 concluding with a few comparisons with state-of-the-art methods.

6.1 Evaluation Parameters

With all the tests performed, the value of δ is fixed as $\delta = 0.2$. This means that no rider travels more than 1.2 times of the shortest distance. The fleet set F is assumed to contain vehicles of 5 different capacities. The vehicle capacities are 4, 6, 12, 35 and 41 with Table 6.1 showing the operating costs of the vehicles. The price that is being charged to the rider is fixed to be 15 INR/km if the rider shares a vehicle and 20 INR/km otherwise. Based on these rates and cost slabs, the results obtained by standard coloring (VC) and WVC are quantified through four main sharing criteria:

1. *Occupancy Index*: The occupancy index for a vehicle quantifies the amount of distance on an average each seat of a vehicle is occupied. For a vehicle having n passengers, the occupancy index is calculated below:

$$o = \frac{\sum_{i=1}^{n} d_i}{n\, d} \tag{6.1}$$

Where d_i is the distance for which rider i was in the vehicle and d represents the total distance travelled by the vehicle. Figure 6.3 shows an illustration of Occupancy Index for a vehicle with 3 riders.

2. *Number of vehicles used*: The number of vehicles used counts how many vehicles were required to carry out all the rides.

3. *Ratio of shared rides*: The ratio of shared rides indicates that out of all the riders, how many riders were not going alone in their vehicles.

4. *Ratio of vehicles with shared rides*: The ratio of shared rides indicates that out of the total obtained vehicles, how many vehicles cater to shared rides instead of individual rides.

Table 6.1: Vehicle operating costs

Capacity	**Cost**
4	15/km
6	25/km
>6	30/km

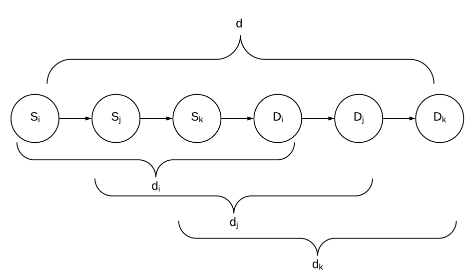

Figure 6.3: Occupancy Index for 3 rider vehicle
$o = \frac{d_i+d_j+d_k}{3d}$

A direct inference can be made from the case study earlier that passengers travelling smaller distances do not suffer severely in terms of cost when they go alone as compared to when they go in a shared ride. WVC uses this property in a way that if a shorter distance rider is replaced in a vehicle with a rider who will share for a longer distance, the overall effect is advantageous to both operators and riders. Table 6.2 presents a comparison with standard vertex coloring and weighted vertex coloring on sharing criteria mentioned above for all the conducted experiments.

Table 6.2: Vertex Coloring (VC) v/s WVC

Metric	NYC: Size 100		NYC: Size 200		NYC: Size 500		Bangalore	
	VC	WVC	VC	WVC	VC	WVC	VC	WVC
Vehicles Used	57	57	94	94	186	188	23	23
Occupancy Index	0.67	0.71	0.63	0.66	0.58	0.59	0.6	0.64
Ratio of shared rides	0.78	0.75	0.89	0.875	0.95	0.95	0.98	0.97
Ratio of shared vehicles	0.61	0.58	0.76	0.73	0.88	0.88	0.94	0.87

6.2 Experimental Studies

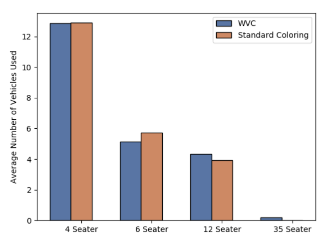

Figure 6.4: Average usage of vehicles (Bangalore)

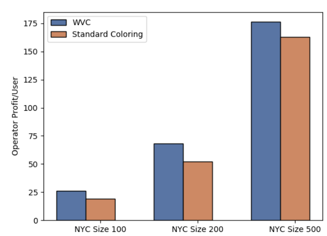

Figure 6.5: Operator Profits

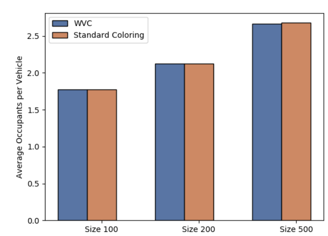

Figure 6.6: Average Occupants per Vehicle (NYC)

For the experiments being conducted two cities with entirely different geographies and ride-sharing use cases are considered. First is a static case of office commuters in Bangalore, where all the riders have a common destination and the ride-sharing problem is to design shuttle routes for them. While the second experimental set up is of the New York City, where the yellow cabs trips dataset is used to test the ride-sharing algorithms.

Office Commuters: Bangalore

As mentioned before, in Bangalore, India, riders with common destination are considered. This is a typical use case for office commuters travelling to a common location wanting to reach at the same time. To generate the source of the riders, random points are generated within a 30km radius centred at the city of Bangalore and select an office in the city for the destination.

An increase in occupancy index from 60% to 64% is observed, while the average profit obtained from WVC is almost 10% more than average profit obtained from VC (Figure 6.5). This shows that while still maintaining a rate for riders that is not unfair to them, an operator can arrive at a good quality sharing strategy using WVC. A higher occupancy index also shows that on an average the vehicle is filled with passengers for longer. Figure 6.4 shows that the presence of a vehicle fleet with larger capacity vehicles is better utilised as now, in WVC more vehicles of larger capacity are filled on an average which leads to a significant increase in profit. These observations about operator profit, coupled with an above 95% sharing ratio for the passengers and above 85% ratio of shared vehicles shows that WVC gives a solution better in terms of both sharing quality, and operator profits.

Yellow Taxi Cabs: New York City

For the dataset of yellow taxi rides in New York City, the algorithm is tested on cases of three different input sizes, 100, 200 and 500. These sizes depict the amount of riders that are considered for the ride-sharing tests. For each of the sizes, multiple cases are run to test the algorithm. Each case is generated by querying the dataset at a different starting time and then extracting rider details for the required number of rides.

It is noticed that both WVC and VC give a reduction in the number of vehicles required for carrying out all the rides. This reduction increases as number of rides increases and indicates that for a driver, the idle time will decrease. This is because now, fewer vehicles are on the road than before and as a result, a single driver spends more time serving rides than looking for them. It is also seen that with an increase in the number of rides, the average number of occupants per vehicle increases steadily (Figure 6.6), hence, enhancing the profit gained per vehicle.

As inferred earlier, WVC puts riders travelling smaller distances into individual, non shared rides in the interest of better overall ride-sharing. While this points at more number of vehicles being used and a reduction in the ratio of shared rides, it is observed that WVC gives on an average a comparable number of vehicles, a comparable ratio of shared vehicles and an average of over 26% increased profit. Figure 6.5 also shows that with an increase in number of riders considered, there is a constant increase in the average profit gained per rider. The main factor that governs this trend is the increased occupancy index given by WVC in all cases. Thus, it can be concluded that WVC is providing a better quality ride-sharing having higher occupancy index and improved operator profits but at the same time, does not compromise on parameters like the ratio of shared rides and number of vehicles used.

6.3 Comparison with other Methods: A Note

This section briefly discuss the comparison of our method with the other state-of-the-art methods. It is seen that when the proposed algorithm is applied to the dataset used in [2, 18] the number of shared rides increases with an increase in the number of riders. This increase goes to as high as 95% and averages above 85% when compared to the 60-70% obtained in [2, 18]. The algorithm also efficiently allocates riders to vehicles in a way that a vehicle on an average has more number of people occupying the seats of the vehicle. In [18], ride-sharing is limited to a maximum of 3 people per vehicle, while in [2] it is seen that for larger capacity vehicles, like vehicles with capacity 10, on an average more than 6 seats go empty. Additionally, including operator profit and optimal rider cost does not in any way compromise on the quality of ride-sharing, as it can be seen by the high occupancy index of 65% and 75% vehicles being shared.

Chapter 7

Future Work

While this marks the end of the BTP, there are several questions that can still be looked upon in future with respect to the framework. These include the aspect of waiting time for riders, the ability to have different tolerances in admissibility for different riders based on their travel distances and even inclusion of factors apart from cost in quality of ride-sharing. All these, questions are going to be tackled in due time and be sent in for publication accordingly.

Chapter 8

Progress and Additional Comments

This report provides a summary of all the work done as part of the project titled "Optimization Problems in Bus Route Network Design". It is to be noted that some of the work mentioned as part of the report was carried out as part of the Monsoon 2018 and Winter 2019 semester. The following parts of the proposed algorithm and results have been performed during the course of Summer 2019 semester-

1. **Financial Angle**: Earlier, the operating costs of vehicles were fixed at INR 15/km, however, now, the algorithm was run and tested with more realistic and varying costs of operating vehicles as depicted in Table 6.1.

2. **Website**: The entire website, https://vpool.chartr.in was developed from scratch in order to provide a better visualization of the algorithm in action. The website enables a user to first fill a basic form with respect to details required for the algorithm and then the user can see both summarized results of the algorithm and detailed vehicle by vehicle results on a separate interactive map. This map contains the vehicle route along with named directions and all details required for the vehicle.

3. **Updates for dynamic RSP**: Earlier, the framework was designed to cater for the use case of the static RSP, where all details about rides is known beforehand. As part of the proposed future work earlier, it was suggested to extend the algorithm for the dynamic RSP where requests come over time. A formal solution was developed for the same by suggesting small updates required in the algorithm.

4. **Experimental Results**: The framework has now been extensively tested on both the static and dynamic case of the RSP and shows encouraging results that align with both claims made about earlier systems and the advantages and robustness of the proposed algorithm.

The work done in the previous semester was submitted to and got accepted in the IEEE Intelligent Transportation Systems Conference - ITSC 2019. The work along with dynamic RSP will now be submitted to the annual meeting of Transport Research Board, 2020.

Bibliography

[1] Agatz, N., Erera, A. L., Savelsbergh, M. W., and Wang, X. Dynamic ride-sharing: A simulation study in metro atlanta. *Procedia-Social and Behavioral Sciences 17* (2011), 532–550.

[2] Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., and Rus, D. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences 114*, 3 (2017), 462–467.

[3] Bei, X., and Zhang, S. Algorithms for trip-vehicle assignment in ride-sharing. In *Thirty-Second AAAI Conference on Artificial Intelligence* (2018).

[4] Bertsimas, D., and Tsitsiklis, J. N. *Introduction to linear optimization*, vol. 6. Athena Scientific Belmont, MA, 1997.

[5] Brélaz, D. New methods to color the vertices of a graph. *Communications of the ACM 22*, 4 (1979), 251–256.

[6] Chen, L., Mislove, A., and Wilson, C. Peeking beneath the hood of uber. In *Proceedings of the 2015 Internet Measurement Conference* (2015), ACM, pp. 495–508.

[7] Cordeau, J.-F., and Laporte, G. The dial-a-ride problem: models and algorithms. *Annals of operations research 153*, 1 (2007), 29–46.

[8] Culberson, J. Iterated greedy graph coloring and the difficulty landscape.

[9] Desaulniers, G. *The VRP with pickup and delivery.*

[10] Garey, M. R., Johnson, D. S., and Stockmeyer, L. Some simplified np-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing* (1974), ACM, pp. 47–63.

[11] Hennessy, D. A., and Wiesenthal, D. L. Traffic congestion, driver stress, and driver aggression. *Aggressive Behavior: Official Journal of the International Society for Research on Aggression 25*, 6 (1999), 409–423.

[12] Jia, Y., Xu, W., and Liu, X. An optimization framework for online ride-sharing markets. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)* (2017), IEEE, pp. 826–835.

[13] Johnson, D. S., Aragon, C. R., McGeoch, L. A., and Schevon, C. Optimization by simulated annealing: an experimental evaluation; part ii, graph coloring and number partitioning. *Operations research 39*, 3 (1991), 378–406.

[14] Ma, S., Zheng, Y., and Wolfson, O. T-share: A large-scale dynamic taxi ridesharing service. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)* (2013), IEEE, pp. 410–421.

[15] Malaguti, E., Monaci, M., and Toth, P. Models and heuristic algorithms for a weighted vertex coloring problem. *Journal of Heuristics 15*, 5 (2009), 503–526.

[16] Pant, P., and Harrison, R. M. Estimation of the contribution of road traffic emissions to particulate matter concentrations from field measurements: a review. *Atmospheric environment 77* (2013), 78–97.

[17] Pisinger, D., and Ropke, S. A general heuristic for vehicle routing problems. *Computers & operations research 34*, 8 (2007), 2403–2435.

[18] Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S. H., and Ratti, C. Quantifying the benefits of vehicle pooling with shareability networks. *Proceedings of the National Academy of Sciences 111*, 37 (2014), 13290–13294.

[19] Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., and Dueck, G. Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics 159*, 2 (2000), 139–171.