# A New Algorithm for the Optimization of Autonomous Vehicle Routing

**Abstract**

In this paper I present a novel approach to the optimization of vehicle routing in an autonomous urban transportation service. In such future transportation systems, a fleet of robotic autonomous vehicles services passengers on demand within an urban setting in an efficient manner to maximize the quality of service (i.e. minimize waiting times). I propose a region-based model that divides the area of interest into distinct geographical regions and tracks the state of the system in a series of discrete timesteps using receding horizon control. Second, I present an algorithm that, at each timestep, optimizes not only for the current state but also for a number of timesteps in the future. Then, I analyze the complexity of the model and algorithm. I propose an approximation heuristic that can be solved in polynomial time, then I analytically prove the upper bound of its error. Finally, I show through simulation with real-world data that my algorithm significantly improves on both the state of the art and recent literature by reducing passenger waiting times by as much as 16% compared to algorithms in use by major dispatch companies.

# 1 Introduction

Personal mobility and congestion is an ever-growing issue in modern urban environments; in 2010, urban congestion caused a 4.8 billion hour increase in the collective travel time of Americans, with a total cost of $101 billion [11]. As recent research has led to the development of vehicles that can drive autonomously [14], possible applications of autonomous vehicles to rethink future urban transportation have been considered. One such idea has been to maintain a fleet of autonomous vehicles that can be called on demand in place of traditionally owned vehicles or taxi systems [16]. In this scenario, the vehicles must be autonomously routed to service passengers on request and rebalance as needed. The goal of the routing service is to optimally assign vehicles to passengers to minimize passenger waiting times.

## 1.1 State of the Art

Although no such service currently exists, as the vehicles themselves are still under development, the routing problem is similar to the taxi dispatch problem, in which passengers request a ride and a taxi is assigned to service him or her. The current state of the art for vehicle routing, that is, the algorithms used by leading taxi companies, typically involve some variant of the nearest neighbor algorithm, whereby each passenger is assigned to the nearest unassigned taxi on a first-come, first-served basis [9]. Market leaders such as Cordic, DDS Digital Dispatch, and EzyFleet utilize the nearest neighbor algorithm, with the addition of region-based queues to ensure fairness among drivers [2, 3, 4]. Although this might seem to be a reasonable strategy, it is a greedy approach, and there are several scenarios, such as the one depicted in figure 1, for which the nearest neighbor algorithm is clearly suboptimal [6].

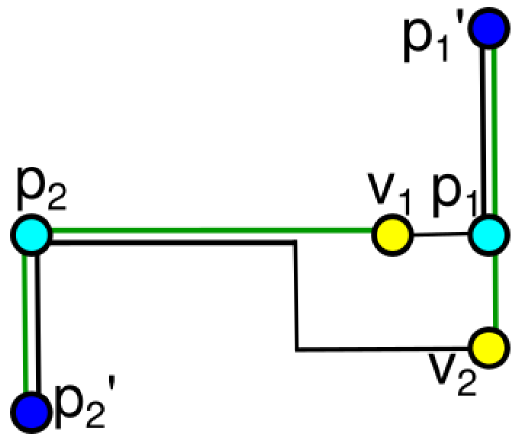


Figure 1: $v_i$ refers to vehicle i, $p_i$ refers to passenger i, and $p_i'$ refers to the destination of passenger i. $v_1$ is assigned to $p_1$ because it is closest, whereas the optimal strategy is for $v_1$ to service $p_2$, and for $v_2$ to service $p_1$. The black line indicates the path chosen by the nearest neighbor algorithm, and the green line indicates the optimal path.

### 1.2 Related Research

Due to the growing significance of these issues, with increasing urbanization worldwide [13], autonomous vehicle routing has become an active area of research in recent years. Recent publications have suggested algorithms to overcome the limitations of current implementations, although they are yet to be implemented by taxi companies. These include $P_{HM}$, a rebalancing policy based on Markov chains designed to achieve a desired stationary distribution [15] and $NTuCab$, a multiagent system whereby vehicles negotiate to achieve minimal driving distances [10].

While these algorithms improve on the state of the art by optimizing for the entire system as a whole instead of the greedy approach of optimizing for each individual, they also come with a number of flaws.

$NTuCab$, while slightly less greedy than nearest neighbor algorithms, only optimizes for the initial assignments. so it does not consider possible future assignments after each vehicle finishes servicing its first passenger. As a result, it fails to account for situations where the destination of one passenger is near the location of another. Figure 2 depicts a scenario for which it would return a suboptimal solution.

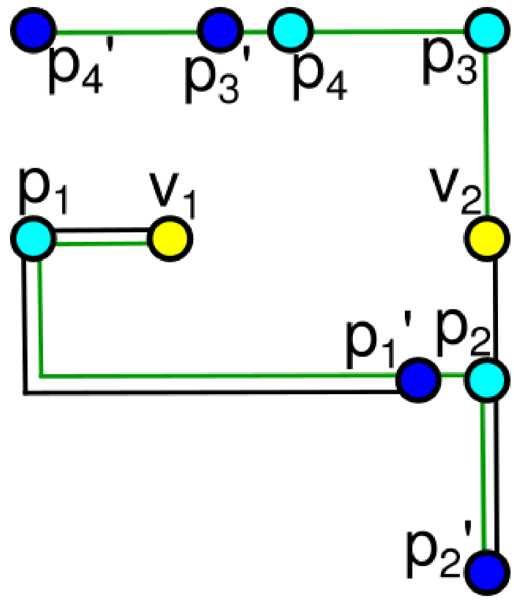


Figure 2: $NTuCab$ assigns $v_1$ to $p_1$ and $v_2$ to $p_2$, but the optimal strategy is for $v_1$ to service $p_1$ then $p_2$, and for $v_2$ to service $p_3$ then $p_4$. The black line indicates the path chosen by NTuCab, and the green line indicates the optimal path.

On the other hand, $P_{HM}$ assumes constant passenger appearance rates, an assumption that incurs some error in the system as appearance rates change throughout the course of the day. Although these errors are slight, the algorithm only considers appearance rates, not the actual passenger distribution; any buildup of passengers in a particular region has no effect on the resulting rebalancing policy, so the system makes no attempt to adjust for it.

Both algorithms, as well as almost every other proposed solution, such as those discussed in [1, 7, 16, 17], enforce that a passenger that is picked up by a vehicle is serviced directly to its destination in a single trip by the same vehicle. However, this does not always result in optimal behavior. Figure 3 depicts a scenario for which the optimal solution does not conform to this constraint.

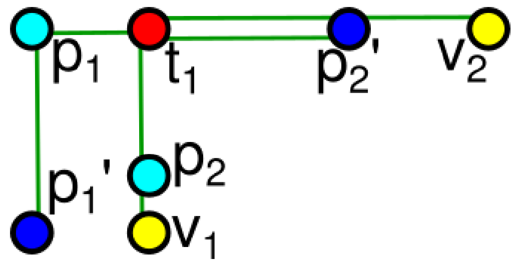


Figure 3: The optimal strategy is for $v_1$ to carry $p_2$ to $t_1$, then service $p_1$ while $v_2$ services $p_2$. The green line indicates this path.

### 1.3 This Paper

In this paper I propose an original algorithm to address these concerns. The rest of this paper is organized as follows: Section 2 defines a model to represent the system, Section 3 describes the steps taken to optimize the model, Section 4 evaluates and analyzes the computational complexity of the algorithm, Section 5 proposes an approximating heuristic that achieves a high quality solution in polynomial time, Section 6 describes the procedures and results for simulating the performance of the algorithm in comparison to the industry standard and state of the art algorithms, and Section 7 discusses conclusions and suggestions for future research.

## 2 Model

Consider a network flow model whereby there exists a set of regions, each with a quantity of vehicles and a quantity of passengers destined for each other region. The model accounts for the state of the system through a finite series of discrete timesteps. At each time step, a vehicle may service a passenger to any adjacent region, not necessarily the destination region of the passenger. Upon arrival, the vehicle might continue and service the passenger to another adjacent region, or it might drop the passenger off in order to free itself for other actions. A passenger that is dropped off in a region other than its destination region is then considered as part of the set of passengers waiting in that region. It may be assumed that the time required for a passenger to get in or out of a vehicle is negligible compared to the length of the time step. Therefore, a vehicle that is servicing a passenger through a region is considered as identical to a vehicle and passenger waiting in the region. Additionally, the flow of passengers is not guaranteed to be balanced, so to prevent buildup of vehicles in a single region, vehicles can rebalance to other regions without servicing passengers.

### 2.1 Model Constants

Suppose the area of interest is divided into $n$ regions where $N$ denotes the set of such regions and $|N| = n$. Let $w_{ij}$ $\forall i, j \in N$ equal the number of time steps required for a vehicle to travel from region $i$ to region $j$. Let $M$ represent the sparse directed graph of $n$ nodes such that node $i$ is connected to node $j$ if and only if a vehicle can travel from region $i$ to region $j$ in a single time step (i.e. $i, j \in M \Leftrightarrow w_{ij} \leq 1$). Note that the regions must be selected such that $M$ is a connected graph, that is, there exists a path between any two nodes in $M$. Then, let $d_{ij}$ $\forall i, j \in N$ represent the length of the shortest path from node $i$ to node $j$ in $M$. Also let $b$ denote the maximum number of regions any region is adjacent to, that is, the cardinality of the graph $M$.

For the purpose of modeling the future state, let $F$ represent the set of time steps to consider, where $|F| = f$, or the prediction depth.

### 2.2 State Variables

The model can be described by the number of vehicles and passengers in each region and the destination distributions of the passengers at each time step. The number of vehicles in region $i \in N$ at time $t \in F$ is given by $v_i^t$. There must be a nonnegative integral number of vehicles in each region, so

$$v_i^t \in \mathbb{Z} \text{ and } v_i^t \geq 0 \quad \forall\, t \in F,\ i \in N \tag{1}$$

The number of passengers that are waiting in region $i$ and destined for region $j$ at time $t$ is given by $c_{ij}^t$. There must also be a nonnegative integral number of passengers waiting in each region that are destined for each region, so

$$c_{ij}^t \in \mathbb{Z} \text{ and } c_{ij}^t \geq 0 \quad \forall t \in F,\ i, j \in N \tag{2}$$

The variables $V$ and $C$ form the complete state of the system at each time step.

### 2.3 Decision Variables

The control decisions of this model can be described by the passengers and vehicles at each region that transfer to each adjacent region. Let $p_{ijk}^t \ \forall i, j, k \in N, \ t \in F$ represent the number of passengers waiting in region $i$ and destined for region $j$ that are serviced to region $k$ at time $t$. The number of passengers leaving a particular region must equal the number of passengers in the region at the previous time step, so

$$\sum_k p_{ijk}^t = c_{ij}^{t-1} \quad \forall\, t \in F, \ i, j \in N \tag{3}$$

In the case of rebalancing, let $r_{ij}^t \ \forall i, j \in N, t \in F$ represent the number of vehicles that rebalance from region $i$ to region $j$ at time $t$. The total number of vehicles leaving a particular region (including both rebalancing vehicles and vehicles that are servicing a passenger) must equal the number of vehicles that exist in the region, so

$$\sum_j r_{ij}^t + \sum_{k \mid k \neq i} p_{ijk}^t = v_i^{t-1} \quad \forall\, t \in F, \ i \in N \tag{4}$$

Note that these variables contain elements that represent the vehicles and passengers that "travel" to the same region that they start at. These denote the vehicles and passengers that remain in place. Therefore, since a passenger that does not travel does not require a vehicle, the number of passengers that remain in place does not contribute to the total number of vehicles leaving the region, hence the "$k \neq i$" in (4). However, a vehicle that remains in place is considered to be "leaving" the region because it is unavailable to perform any other action, so it is included in (4). Vehicles and passengers may only transfer to adjacent regions, so

$$p_{ijk}^t = 0 \quad \forall t \in F, \ i, j, k \in N, \ \{i, k\} \notin M \tag{5}$$

$$r_{ij}^t = 0 \quad \forall t \in F, \ i, j \in N, \ \{i, j\} \notin M \tag{6}$$

and there must be a nonnegative integral number of vehicles and passengers that transfer between any two regions, so

$$p_{ijk}^t \in \mathbb{Z} \text{ and } p_{ijk}^t \geq 0 \quad \forall t \in F, \ i, j, k \in N \tag{7}$$

$$r_{ij}^t \in \mathbb{Z} \text{ and } r_{ij}^t \geq 0 \quad \forall t \in F, \ i, j \in N \tag{8}$$

### 2.4 Propagation of the State

The state collections $V := \{v_i^t \mid t \in F,\ i \in N\}$ and $C := \{c_{ij}^t \mid t \in F,\ i,j \in N\}$ propagate solely based on the decision collections $P := \{p_{ijk}^t \mid t \in F,\ i,j,k \in N\}$ and $R := \{r_{ij}^t \mid t \in F,\ i,j \in N\}$. The number of vehicles in a particular region (i.e. $v_i^t$) is equivalent to the number of vehicles in that region at the previous time step (i.e. $v_i^{t-1}$) minus the number of vehicles that left plus the vehicles that arrived. Define $L_v(t,i)$ as the number of vehicles that leave region $i$ at time $t$ and $A_v(t,i)$ as the number of vehicles that arrive at region $i$ at time $t$.

$$L_v(t,i) = \sum_{jk} p_{ijk}^t + \sum_j r_{ij}^t \qquad \forall t \in F,\ i \in N \tag{9}$$

$$A_v(t,i) = \sum_{jk} p_{kji}^t + \sum_j r_{ji}^t \qquad \forall t \in F,\ i \in N \tag{10}$$

$$v_i^t = v_i^{t-1} - L_v(t,i) + A_v(t,i) \qquad \forall t \in F,\ i \in N \tag{11}$$

Similarly, the number of passengers in a particular region destined for a particular region (i.e. $c_{ij}^t$) is equivalent to the number of passengers in that region with the same destination at the previous time step (i.e. $c_{ij}^{t-1}$) minus the number of passengers that left plus the passengers that arrived. Define $L_c(t,i,j)$ as the number of passengers destined for region $j$ that leave region $i$ at time $t$ and $A_c(t,i,j)$ as the number of passengers destined for region $j$ that arrive at region $i$ at time $t$.

$$L_c(t,i,j) = \sum_k p_{ijk}^t \qquad \forall t \in F,\ i,j \in N \tag{12}$$

$$A_c(t,i,j) = \sum_k p_{kji}^t \qquad \forall t \in F,\ i,j \in N \tag{13}$$

$$c_{ij}^t = c_{ij}^{t-1} - L_c(t,i,j) + A_c(t,i,j) \qquad \forall t \in F,\ i,j \in N \tag{14}$$

## 3 Optimization

The goal of this algorithm is to optimize this model for a set of objectives. Note that the resulting decisions are only carried out for the first timestep; at the next timestep, the optimization is called again. As a result, due to the receding horizon, any suboptimal behavior resulting from the limited prediction depth would have plenty of time to be corrected at later timesteps.

### 3.1 Feasible Sets

Here, define as $Q$ the set of all feasible sets of $[V, C, P, R]$, satisfying all of the above constraints.

$$Q := \left\{ S = [V, C, P, R] \;\middle|\; \begin{array}{c} S \text{ satisfies (1), (2), (3), (4)} \\ \text{(5), (6), (7), (8), (11), and (14)} \end{array} \right\} \tag{15}$$

### 3.2 Objectives

1. Minimize the average time a passenger has to wait until it reaches its destination.

$$U_t(C) = \sum_{ijt \mid i \neq j} c_{ij}^t \tag{16}$$

2. Minimize the distance left for passengers to travel after the final timestep.

$$U_d(C) = \sum_{ij} c_{ij}^f d_{ij} \tag{17}$$

3. Minimize vehicles' wasted time. Any time a vehicle is not transporting a passenger is considered to be wasted.

$$U_w(R) = \sum_{ijt} r_{ij}^t \tag{18}$$

### 3.3 Integer Linear Program

Therefore, the optimal solution can be computed by optimizing the objective functions constrained to the feasible set. Because all constraints and objective functions are linear and the variables are constrained to integer values, the algorithm takes the form of an integer linear program.

$$
\begin{aligned}
&A_m: \\
&\text{Given} \quad && v_i^0,\ c_{ij}^0 \quad \forall\, i,j \in N \\
&\underset{V,C,P,R}{\text{minimize}} \quad && U_t(C) + \lambda_1 U_d(C) + \lambda_2 U_w(R) \\
&\text{subject to} \quad && S \in Q
\end{aligned}
$$

where $\lambda_1$ and $\lambda_2$ are small constants that correlate to the degree to which final passenger distances and wasted time, respectively, should factor into the optimization.

Note that there will always be at least one feasible set. Consider the system whereby no vehicles or passengers move, so the state remains constant. Because this is guaranteed to follow the above constraints, it must be a feasible set (albeit a generally undesirable one). Therefore, the algorithm will never fail to produce a solution.

# 4 Complexity

## 4.1 Number of Variables

Let $S$ represent the combined system $[V, C, P, R]$, and $E(X)$ equal the number of elements in variable $X$. Since $E(P)$ and $E(R)$ depend on the number of regions each region is connected to, let $k$ denote the average number of adjacent regions.

$$k = \frac{|M|}{N} \tag{19}$$

$$E(V) = fn \tag{20}$$

$$E(C) = fn^2 \tag{21}$$

$$E(P) = fn^2k \tag{22}$$

$$E(R) = fnk \tag{23}$$

The total number of variables in the system is represented by

$$
\begin{aligned}
E(S) &= f(n^2k + n^2 + nk + n) \\
&= O(fn^2) \text{ as } f, n \to \infty
\end{aligned}
\tag{24}
$$

so the representation of $E(S)$ has polynomial size. Note that $k$ is constant with respect to $f$ and $n$ because typically, the time step is selected based on the distances between regions such that the number of connections for each region is small (around 2 to 6).

### 4.2 Integer Linear Programming

Because of constraints (1), (2), (7), and (8), the elements in $S$ are all constrained to integral values, so the resulting optimization problem becomes an Integer Linear Program (ILP). The complexity of ILP is NP-Hard with respect to the number of variables [12], and, by (24), the number of variables scales polynomially with $n$ and $f$, so this optimization is NP-Hard with respect to both $n$ and $f$. As a result, the algorithm has an infeasible computation time for systems with many regions, and the maximum feasible prediction depth is low.

## 5 Approximation Heuristic

To solve this issue, consider a heuristic to approximate the optimal solution for the system. By relaxing constraints (1), (2), (7), and (8) to

$$v_i'^t \geq 0 \qquad \forall\, it \tag{25}$$

$$c_{ij}'^t \geq 0 \qquad \forall\, ijt \tag{26}$$

$$p_{ijk}'^t \geq 0 \qquad \forall\, ijkt \tag{27}$$

$$r_{ij}'^t \geq 0 \qquad \forall\, ijt \tag{28}$$

the problem becomes a Continuous Linear Program (LP), which can be solved in polynomial time [8].

However, it is impossible for a fraction of a vehicle or passenger to transfer to another region, so it is necessary to transform the solution to the LP to satisfy the constraints of the ILP. Denote by $e(x)$ the error between the variable $x$ in the solution to the LP ($S'$) and the constrained solution ($S$). The errors are given by

$$e(v_i^t) = v_i^t - v_i'^t \qquad \forall\, it \tag{29}$$

$$e(c_{ij}^t) = c_{ij}^t - c_{ij}'^t \qquad \forall\, ijt \tag{30}$$

$$e(p_{ijk}^t) = p_{ijk}^t - p_{ijk}'^t \qquad \forall\, ijkt \tag{31}$$

$$e(r_{ij}^t) = r_{ij}^t - r_{ij}'^t \qquad \forall\, ijt \tag{32}$$

A feasible solution for the decision variables at each timestep can be found by distributing the errors of the state variables at the previous timestep, then taking the floor. The number of vehicles and passengers that remain in place should be adjusted to compensate for the decrease in transfers and satisfy (3) and (4).

$$p_{ijk}^t = \lfloor p_{ijk}'^t + \min(\frac{1}{b}e(c_{ij}^{t-1}), \frac{1}{nb}e(v_i^{t-1}))\rfloor \qquad \forall\, ijkt \mid i \neq k \tag{33}$$

$$r_{ij}^t = \lfloor r_{ij}'^t + \frac{1}{b}(e(v_i^{t-1}) - \sum_{mk \mid k \neq i} e(p_{imk}^t))\rfloor \qquad \forall\, ijt \mid i \neq j \tag{34}$$

(33) generates feasible values by determining the errors in the quantities of both available vehicles and available passengers, choosing the error that is most limiting, and allocating the loss or gain evenly to each possible destination. (34) generates feasible values by determining the error in the number of available vehicles and allocating the loss or gain evenly to each possible destination. Taking the floor is guaranteed to be feasible because it requires less vehicles and passengers than $S'$ does. Therefore, the errors of the decision variables are bounded by

$$\min(\frac{1}{b}e(c_{ij}^{t-1}), \frac{1}{nb}e(v_i^{t-1})) - 1 < e(p_{ijk}^t) \leq \min(\frac{1}{b}e(c_{ij}^{t-1}), \frac{1}{nb}e(v_i^{t-1})) \qquad \forall\, ijt \mid i \neq k \tag{35}$$

$$\frac{1}{b}(e(v_i^{t-1}) - \sum_{mk \mid k \neq i} e(p_{imk}^t)) - 1 < e(r_{ij}^t) \leq \frac{1}{b}(e(v_i^{t-1}) - \sum_{mk \mid k \neq i} e(p_{imk}^t)) \qquad \forall\, it \mid i \neq j \tag{36}$$

$$e(p_{iji}^t) = e(c_{ij}^{t-1}) - \sum_{k \mid k \neq i} e(p_{ijk}^t) \qquad \forall\, ijt \tag{37}$$

$$e(r_{ii}^t) = e(v_i^{t-1}) - \sum_{j \mid j \neq i} e(r_{ij}^t) - \sum_{jk \mid k \neq i} e(p_{ijk}^t) \qquad \forall\, it \tag{38}$$

The errors in the decision variables at each time step can propagated to determine the errors in the state variables. By (11) and (14), the errors in the state variables at each time step are given by

$$e(v_i^t) = \sum_j e(r_{ji}^t) + \sum_{jk \mid k \neq i} e(p_{kji}^t) \tag{39}$$

$$e(c_{ij}^t) = \sum_k e(p_{kji}^t) \tag{40}$$

### 5.1 Optimality of the Approximation

In this section, I analytically prove the upper bound on the error of the heuristic for the primary objective, that is, $e(U_t(C))$.

**Theorem 1.**

$$e(U_t(C)) \leq 5^{2f} n^f b^{f+2} f \tag{41}$$

*Proof.* First note that since any solution to the ILP is also feasible for the LP, $U_t(C')$ is at least as good as that for the ILP. $e(c_{ij}^0) = 0$ and $e(v_i^0) = 0$ for all $i \in N$, so by (37) and (38), the error of the initial decision variables are given by

$$-1 < e(p_{ijk}^1) \leq 0 \qquad \forall\, ijk \mid i \neq k \tag{42}$$

$$-1 < e(r_{ij}^1) \leq 0 \qquad \forall\, ij \mid i \neq j \tag{43}$$

$$0 < e(p_{iji}^1) \leq b - 1 \qquad \forall\, ij \tag{44}$$

$$0 < e(r_{ii}^1) \leq b^2 - 1 \qquad \forall\, i \tag{45}$$

By (3) and (4) the error in the number of vehicles and passengers leaving each region is given by

$$\sum_{jk \mid k \neq i} e(p_{ijk}^t) + \sum_j e(r_{ij}^t) = e(v_i^{t-1}) \qquad \forall\, it \tag{46}$$

$$\sum_k e(p_{ijk}^t) = e(c_{ij}^{t-1}) \qquad \forall\, ijt \tag{47}$$

Denote by $\max x$ the maximum possible value of expression $x$, and by $\min x$ the minimum possible value of expression $x$. Then let $\operatorname{range} x = \max x - \min x$, or the range of possible values of expression $x$. By (35) and (37), the range of possible errors in passenger transfers is given by the following. Note that the sum of a set of elements is at most the number of elements times the maximum element, since the mean is at most the maximum. The result of (48) can be substituted into (49) to compute the bound in terms of the range in errors of the state variables.

$$\begin{aligned}
\operatorname*{range}_{ijk \mid i \neq k} e(p^t_{ijk}) &\leq \max(\frac{1}{b} \operatorname*{range}_{ij} e(c^{t-1}_{ij}), \frac{1}{nb} \operatorname*{range}_{i} e(v^{t-1}_{i})) + 1 \\
&\leq \frac{1}{b} \operatorname*{range}_{ij} e(c^{t-1}_{ij}) + \frac{1}{nb} \operatorname*{range}_{i} e(v^{t-1}_{i}) + 1
\end{aligned} \quad \forall\, t \tag{48}$$

$$\begin{aligned}
\operatorname*{range}_{ij} e(p^t_{iji}) &\leq \operatorname*{range}_{ij}(e(c^{t-1}_{ij}) + \sum_{k \mid k \neq i} e(p^t_{ijk})) \\
&\leq \operatorname*{range}_{ij} e(c^{t-1}_{ij}) + (b-1) \operatorname*{range}_{ijk \mid i \neq k} e(p^t_{ijk}) \\
&\leq \frac{2b-1}{b} \operatorname*{range}_{ij} e(c^{t-1}_{ij}) + \frac{b-1}{nb} \operatorname*{range}_{i} e(v^{t-1}_{i}) + b - 1 \\
&\leq 2 \operatorname*{range}_{ij} e(c^{t-1}_{ij}) + \frac{1}{n} \operatorname*{range}_{i} e(v^{t-1}_{i}) + b
\end{aligned} \quad \forall\, t \tag{49}$$

Similarly, by (36) and (38), the range of possible errors in rebalancing is given by the following. (48) can be substituted into (50), and (48) and (50) can be substituted into (51).

$$\begin{aligned}
\operatorname*{range}_{ij \mid i \neq j} e(r^t_{ij}) &\leq \frac{1}{b} \operatorname*{range}_{i}(e(v^{t-1}_{i}) - \sum_{jk \mid k \neq i} e(p^t_{ijk})) + 1 \\
&\leq \frac{1}{b} \operatorname*{range}_{i} e(v^{t-1}_{i}) + \frac{n(b-1)}{b} \operatorname*{range}_{ijk \mid i \neq k} e(p^t_{ijk}) + 1 \\
&\leq \frac{n(b-1)}{b^2} \operatorname*{range}_{ij} e(c^{t-1}_{ij}) + \frac{b-1}{b^2} \operatorname*{range}_{i} e(v^{t-1}_{i}) + 1 \\
&\leq \frac{n}{b} \operatorname*{range}_{ij} e(c^{t-1}_{ij}) + \frac{1}{b} \operatorname*{range}_{i} e(v^{t-1}_{i}) + 1
\end{aligned} \quad \forall\, t \tag{50}$$

$$\begin{aligned}
\operatorname*{range}_{i} e(r^t_{ii}) &\leq \operatorname*{range}_{i}(e(v^{t-1}_{i}) - \sum_{j \mid j \neq i} e(r^t_{ij}) - \sum_{jk \mid k \neq i} e(p^t_{ijk})) \\
&\leq \operatorname*{range}_{i} e(v^{t-1}_{i}) + (b-1) \operatorname*{range}_{ij \mid i \neq j} e(r^t_{ij}) + n(b-1) \operatorname*{range}_{ijk \mid i \neq k} e(p^t_{ijk}) \\
&\leq \frac{2n(b-1)}{b} \operatorname*{range}_{ij} e(c^{t-1}_{ij}) + \frac{3b-2}{b} \operatorname*{range}_{i} e(v^{t-1}_{i}) + n(b-1) + (b-1) \\
&\leq 2n \operatorname*{range}_{ij} e(c^{t-1}_{ij}) + 3 \operatorname*{range}_{i} e(v^{t-1}_{i}) + (n+1)b
\end{aligned} \quad \forall\, t \tag{51}$$

By (39) and (40), the ranges in errors of the state variables are given by

$$
\begin{aligned}
\operatorname*{range}_{i} e(v_i^t) &\leq \max_{i} e(v_i^t) - \min_{i} e(v_i^t) \\
&\leq (\max_{i} e(r_{ii}^t) + (b-1) \max_{ij \mid i \neq j} e(r_{ji}^t) + n(b-1) \max_{ijk \mid i \neq k} e(p_{ijk}^t)) \\
&\quad - (\min_{i} e(r_{ii}^t) + (b-1) \min_{ij \mid i \neq j} e(r_{ij}^t) + n(b-1) \min_{ijk \mid i \neq k} e(p_{ijk}^t)) \qquad \forall\, t \qquad (52) \\
&\leq \operatorname*{range}_{i} e(r_{ii}^t) + (b-1) \operatorname*{range}_{ij \mid i \neq j} e(r_{ij}^t) + n(b-1) \operatorname*{range}_{ijk \mid i \neq k} e(p_{ijk}^t) \\
&\leq \operatorname*{range}_{i} e(r_{ii}^t) + b \operatorname*{range}_{ij \mid i \neq j} e(r_{ij}^t) + nb \operatorname*{range}_{ijk \mid i \neq k} e(p_{ijk}^t) \\
\operatorname*{range}_{ij} e(c_{ij}^t) &\leq \operatorname*{range}_{ij} \sum_{k} e(p_{kji}^t) \\
&\leq \operatorname*{range}_{ij} e(p_{iji}^t) + (b-1) \operatorname*{range}_{ijk \mid i \neq k} e(p_{ijk}^t) \qquad \forall\, t \qquad (53) \\
&\leq \operatorname*{range}_{ij} e(p_{iji}^t) + b \operatorname*{range}_{ijk \mid i \neq k} e(p_{ijk}^t)
\end{aligned}
$$

(52) and (53) can be rewritten as a a linear transformation over the range of errors in the decision variables.

$$
\begin{pmatrix} \operatorname{range}_{ij} e(c_{ij}^t) \\ \operatorname{range}_{i} e(v_i^t) \\ 1 \end{pmatrix} \leq \begin{pmatrix} b & 1 & 0 & 0 & 0 \\ nb & 0 & b & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \operatorname{range}_{ijk \mid i \neq k} e(p_{ijk}^t) \\ \operatorname{range}_{ij} e(p_{iji}^t) \\ \operatorname{range}_{ij \mid i \neq j} e(r_{ij}^t) \\ \operatorname{range}_{i} e(r_{ii}^t) \\ 1 \end{pmatrix} \quad \forall\, t \qquad (54)
$$

Similarly, (48), (49), (50), and (51) can be rewritten as a a linear transformation over the range of errors in the state variables at the previous time step. (54) can be substituted in to represent them as the result of a matrix power operation.

$$
\begin{aligned}
\begin{pmatrix} \underset{ijk \mid i\neq k}{\operatorname{range}}\, e(p_{ijk}^t) \\ \underset{ij}{\operatorname{range}}\, e(p_{iji}^t) \\ \underset{ij \mid i\neq j}{\operatorname{range}}\, e(r_{ij}^t) \\ \underset{i}{\operatorname{range}}\, e(r_{ii}^t) \\ 1 \end{pmatrix}
&\leq \begin{pmatrix} \frac{1}{b} & \frac{1}{nb} & 1 \\ 2 & \frac{1}{n} & b \\ \frac{n}{b} & \frac{1}{b} & 1 \\ 2n & 3 & (n+1)b \\ 0 & 0 & 1 \end{pmatrix}
\begin{pmatrix} \underset{ij}{\operatorname{range}}\, e(c_{ij}^{t-1}) \\ \underset{i}{\operatorname{range}}\, e(v_i^{t-1}) \\ 1 \end{pmatrix} \\
&\leq \begin{pmatrix} 2 & \frac{1}{b} & \frac{1}{n} & \frac{1}{nb} & 1 \\ 3b & 2 & \frac{b}{n} & \frac{1}{n} & b \\ 2n & \frac{n}{b} & 1 & \frac{1}{b} & 1 \\ 5nb & 2n & 3b & 3 & (n+1)b \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}
\begin{pmatrix} \underset{ijk \mid i\neq k}{\operatorname{range}}\, e(p_{ijk}^{t-1}) \\ \underset{ij}{\operatorname{range}}\, e(p_{iji}^{t-1}) \\ \underset{ij \mid i\neq j}{\operatorname{range}}\, e(r_{ij}^{t-1}) \\ \underset{i}{\operatorname{range}}\, e(r_{ii}^{t-1}) \\ 1 \end{pmatrix} \\
&\leq \begin{pmatrix} 2 & \frac{1}{b} & \frac{1}{n} & \frac{1}{nb} & 1 \\ 3b & 2 & \frac{b}{n} & \frac{1}{n} & b \\ 2n & \frac{n}{b} & 1 & \frac{1}{b} & 1 \\ 5nb & 2n & 3b & 3 & (n+1)b \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}^{t-1}
\begin{pmatrix} \underset{ijk \mid i\neq k}{\operatorname{range}}\, e(p_{ijk}^{1}) \\ \underset{ij}{\operatorname{range}}\, e(p_{iji}^{1}) \\ \underset{ij \mid i\neq j}{\operatorname{range}}\, e(r_{ij}^{1}) \\ \underset{i}{\operatorname{range}}\, e(r_{ii}^{1}) \\ 1 \end{pmatrix} \qquad \forall\, t \qquad (55) \\
&\leq \begin{pmatrix} 2 & \frac{1}{b} & \frac{1}{n} & \frac{1}{nb} & 1 \\ 3b & 2 & \frac{b}{n} & \frac{1}{n} & b \\ 2n & \frac{n}{b} & 1 & \frac{1}{b} & 1 \\ 5nb & 2n & 3b & 3 & (n+1)b \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}^{t-1}
\begin{pmatrix} 1 \\ b \\ 1 \\ b^2 \\ 1 \end{pmatrix} \\
&\leq (5 \cdot 5nb)^t b^2 = 5^{2t} n^t b^{t+2}
\end{aligned}
$$

Therefore, the range in errors for transfers is given by

$$
\underset{ijk}{\operatorname{range}}\, e(p_{ijk}^t) = \max\big(\underset{ijk \mid i\neq k}{\operatorname{range}}\, e(p_{ijk}^t), \underset{ij}{\operatorname{range}}\, e(p_{iji}^t)\big) \leq 5^{2t} n^t b^{t+2} \quad \forall\, t \qquad (56)
$$

By (16), (40), (56), and the fact that the total number of passengers in the system is constant, the error of the primary objective is given by

$$
\begin{aligned}
e(U_t(C)) &\le \sum_{ijt \mid i \neq j} e(c_{ij}^t) = \sum_{ijt \mid j \neq k} e(p_{ijk}^t) = \sum_{ijkt} e(p_{ijk}^t) - \sum_{ijt} e(p_{ijj}^t) = -\sum_{ijt} e(p_{ijj}^t) \\
&\le -nbf \min_{ijk} e(p_{ijk}^f) \le nbf \operatorname*{range}_{ijk} e(p_{ijk}^f) \le 5^{2f} n^f b^{f+2} f
\end{aligned} \tag{57}
$$

□

This demonstrates that the error the heuristic incurs on the primary objective has a finite bound that scales polynomially with respect to the number of regions, but exponentially with respect to prediction depth. This is acceptable because the prediction depth does not need to be expanded to accomadate larger systems, and a large prediction depth is unnecessary because as the prediction depth increases, the increase in performance diminishes. It should be stressed that this is an absolute worst-case bound, and that on average, the heuristic is significantly more accurate; a better measure of the heuristic's average performance may be determined through simulation with real-world data. Note that this approximation performs best when there are many vehicles and passengers per region, as the error does not depend on the total number of vehicles or passengers, so the error has less of an impact on the effectiveness of the routing. Also note that, as with $k$, $b$ can be considered a constant value.

# 6 Comparison

## 6.1 Experimental Design

In order to test the performance of $A_m$, I compare it through simulation both to algorithms used by leading taxi dispatch services and to algorithms described in recent publications. Simulations were conducted comparing $A_m$ with the following algorithms:

1. Nearest Neighbor: $A_{nn}$ is a simple heuristic that assigns a passenger to the nearest vehicle that is not already in the process of servicing another passenger. A passenger that is picked up is serviced directly to its destination. This is similar to what is commonly implemented by leading taxi companies [2, 3, 4].

2. Collaborative Dispatch [10]: $NTuCab$ is a recently developed (2010) multiagent algorithm that improves on the Nearest Neighbor algorithm by having each vehicle

negotiate with other vehicles over passenger assignment. The result is a set of assignments that minimizes the distance for all vehicles as a group rather than each individually. Each passenger that is assigned is serviced directly to its destination.

3. Markov-based Redistribution [15]: $P_{HM}$ is a recently developed (2012) algorithm that attempts to achieve a stationary distribution through Markov transitions via rebalancing. A passenger in the same region as an idle car is immediately serviced directly to its destination, and leftover cars in regions with no passengers rebalance according to a rebalancing policy in the form of a Markov chain.

For each algorithm, simulations were run measuring wait times of passengers over the course of a 24 hour period, where the wait time is defined as the amount of time between when a passenger initially notifies the system of its desire to be serviced and when the passenger arrives at its destination. I choose to use this definition rather than the conventional measure, the time until a passenger is picked up, because in $A_m$, a passenger that is picked up is not guaranteed to be serviced directly to its destination.

The simulations use synthetically generated passenger requests from real world taxi distributions from within the financial district of Manhattan on March 1, 2012, as described in [17]. The area was divided into 10 regions for the relevant algorithms ($A_m$ and $P_{HM}$). Modifications were made to the simulation environment described in [17] to adapt to the specifications required for each algorithm. In particular, the constraint that a passenger be directly serviced to its destination was removed for $A_m$, and the region system was removed for $NTuCab$ and $A_{nn}$.

## 6.2 Results

Here, I present the average wait time with respect to time, with 20 vehicles and a discretization of an hour. I also present the average wait time between 8:00 and 11:00 A.M. (i.e. during the peak traffic period) with respect to the number of vehicles, where the number of vehicles is in the range $[15, 30]$ in increments of 3.

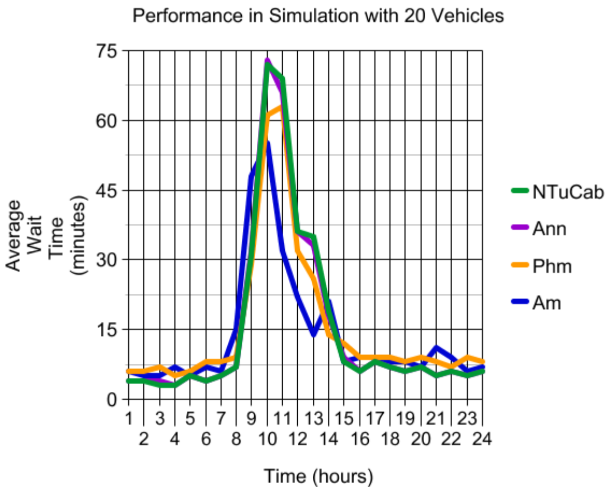


Figure 4: Each algorithm simulated over the course of an entire day.

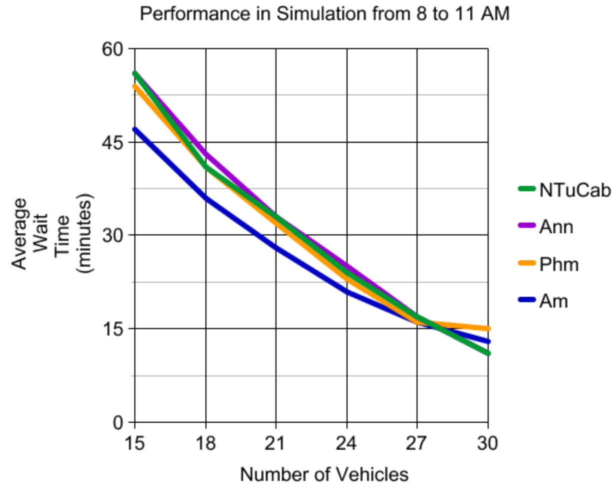


Figure 5: The peak traffic period of the day with varying fleet sizes.

In general, $A_m$ outperformed both the state of the art ($A_{nn}$) and recently published algorithms ($NTuCab$ and $P_{HM}$). It resulted in a lower average wait time for the passengers at most times and numbers of vehicles. During the peak traffic period, $A_m$ achieved an as much as 16%, 15%, and 12% lower average waiting time with the same number of vehicles than $A_{nn}$, $NTuCab$, and $P_{HM}$, respectively; it required about 16% fewer vehicles to achieve a similar average waiting time. It should be noted that, while $A_m$ achieves significantly lower waiting times than the others when the system is under stress, (i.e. there are not enough vehicles to efficiently satisfy passenger demand), the margins diminish as the fleet's ability to handle the demand increases.

# 7 Conclusion, Ramifications, and Future Work

## 7.1 Conclusion

In this paper I proposed a novel algorithm to optimize vehicle routing and assignment. By eliminating the assumption that it is always optimal to travel directly to the final destination, our approach can address cases for which state of the art and recently published algorithms produce suboptimal solutions. The time-dependent model allows the algorithm to avoid greedy decisions, and the use of continuous linear programming as an approximation heuristic reduces the complexity to polynomial time. I proved a worst-case upper bound on the error of the heuristic that is independent of the number of vehicles or passengers in the system, and through simulations, it was determined that this heuristic outperforms both existing and academic vehicle routing algorithms by significant margins.

### 7.2 Significance

Optimal vehicle routing will have significant ramifications on the future of urban mobility and transportation. These results suggest that the implementation of the algorithm described in this paper can simultaneously increase passenger service rates and reduce the number of vehicles on the road, thereby reducing both traffic congestion and carbon emissions. This benefits not only the passenger, but also other drivers on the road, as well as the environment. This is becoming more essential as society increasingly becomes more urban [13], and as human caused global warming and climate change are on the rise [5].

### 7.3 Future Research

This project opens several avenues for future research. The simulations described in this paper were limited to the financial district of Manhattan over the course of a single day; in the future, I would like to evaluate the algorithm's performance over a larger area with a greater amount of real world passenger data. Additionally, I would like to further analyze the tradeoffs between numerical efficiency and performance as a result of varying region sizes and timestep lengths. Finally, I would be interested in testing the algorithm's effectiveness in a real-world taxi or autonomous vehicle based transportation system.

# References


[1] A. Alshamsi, S. Abdallah, and I. Rahwan. Multiagent Self-Organization for a Taxi Dispatch System. 8th International Conference on Autonomous Agents and Multiagent Systems, Budapest, Hungary, 2009.

[2] Cordic. Booking and Despatch. `www.cordic.com/booking_and_despatch.html`. 2015.

[3] DDS Digital Dispatch. All-in-One Taxi Dispatch Solution. `www.digital-dispatch.com/taxibook-overview/taxibook-all-in-one-taxi-dispatch-solution/`. 2015.

[4] Dispatch Algorithm Overview. EzyFleet. `ezyfleet.com/dispatch-algorithm-overview-2/`. 2015.

[5] Field, C.B, et al. IPCC, 2014: Summary for policymakers. *Climate Change 2014: Impacts, Adaptation, and Vulnerability. Part A: Global and Sectoral Aspects. Contribution of Working Group II to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. 2014.

[6] G. Gutin, A. Yeo, and A. Zverovick. Traveling Salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. *Discrete Applied Mathematics*, 117: 81 - 86, 2002.

[7] Jung, Jayakrishnan, and Park. Design and Modeling of Real-Time Shared-Taxi Dispatch Algorithms. *TRB 2013 Annual Meeting*, 2013.

[8] N. Karmarkar. A New Polynomial-Time Algorithm for Linear Programming. *Combinatorica*, 4 (4): 373 - 395, 1984.

[9] D. Lee, H. Wang, R. Cheu, and S. Teo. A Taxi Dispatch System Based on Current Demands and Real-Time Traffic Conditions. *Transportation Research Record*, 1882: 193 - 200, 2004.

[10] K. Seow, N. Dang, and D. Lee. A Collaborative Multiagent Taxi-Dispatch System. *IEEE Transactions on Automation Science and Engineering*, 7 (3): 607 - 616, 2010.

[11] D. Schrank, T. Lomax, B. Eisele. TTI's 2011 Urban Mobility Report. Texas A&M Transportation Institute. 2011.

[12] S. Skiena. Integer Programming. The Algorithm Design Manual. *Springer-Verlag*: New York, 1997.

[13] United Nations, Department of Economic and Social Affairs, Population Division. World Urbanization Prospects: The 2014 Revision, Highlights. 2014.

[14] C. Urmson. Green Lights for our Self-Driving Vehicle Prototypes. *Official Google Blog*. `googleblog.blogspot.com/2015/05/self-driving-vehicle-prototypes-on-road.html`. 2015.

[15] M. Volkov, J. Aslam, and D. Rus. Markov-based Redistribution Policy Model for Future Urban Mobility Networks. emphInternational IEEE Conference on Intelligent Transportation Systems, Anchorage, AK, 2012.

[16] R. Zhang and M. Pavone. Control of Robotic Mobility-On-Demand Systems: a Queueing-Theoretical Perspective. *Robotics: Science and Systems Conference*, 2014.

[17] R. Zhang, F. Rossi, and M. Pavone. Model Predictive Control of Autonomous Mobility-on-Demand Systems. *Robotics and Automation (ICRA), 2016 IEEE International Conference*.