Multi-armed bandit on-time arrival algorithms for sequential reliable route selection under uncertainty

Jinkai Zhou
Department of Civil and Urban Engineering
Tandon School of Engineering
New York University
6 Metrotech Center, Brooklyn, NY 11201
Email: jz1476@nyu.edu

Xuebo Lai
Department of Computer Science
Courant Institute of Mathematical Sciences
New York University
251 Mercer Street, New York, NY 10012
Email: xl1638@nyu.edu

Joseph Y. J. Chow*
Deputy Director, C^{2}SMART University Transportation Center
Assistant Professor, Department of Civil and Urban Engineering
Tandon School of Engineering
New York University
6 Metrotech Center, Brooklyn, NY 11201
Email: joseph.chow@nyu.edu
ORCiD: 0000-0002-6471-3419

* Corresponding author

Words: 5738 – 136 (cover) – 239 (tables) = 5363
Tables: 4 x 250 = 1000
Total Word Count: 6363

ABSTRACT

Traditionally vehicles act only as servers in transporting passengers and goods. With increasing sensor equipment in vehicles, including automated vehicles, there is a need to test algorithms that consider the dual role of vehicles as both servers and sensors. We formulate a sequential route selection problem as a shortest path problem with on-time arrival reliability under a multi-armed bandit setting, a type of reinforcement learning model. A decision maker has to sequentially make a finite set of decisions on departure time and path between a fixed origin-destination pair such that on-time reliability is maximized while travel time is minimized. The Upper Confidence Bound algorithm is extended to handle this problem. We conduct several tests. First, simulated data successfully verifies the method. We then construct a real-data New York City scenario of a hotel shuttle service from midtown Manhattan providing hourly access to the John F Kennedy International Airport. Results suggest that route selection with multi-armed bandit learning algorithms can be effective but neglecting passenger scheduling constraints can negatively impact on-time arrival reliability by as much as 4.8% and combined reliability and travel time by 66.1%.

1. INTRODUCTION

Route selection under uncertainty involves selecting a route to get from an origin to a destination with incomplete information about the underlying network. It is a well-studied topic with an abundant literature (e.g. (*1*)) as travel times can be highly stochastic due to random fluctuations in travel demand and supply degradation (*2*). A deeper review is provided in Section 2. A subset of this literature deals with the sequential route selection problem, where routes are selected repeatedly between a common origin and destination and random travel time outcomes are observed. Applications of this problem can be found in long haul truck routing, flight navigation between designated airports, emergency medical services, and local airport shuttle services.

In the age of automation and ubiquitous sensors (*3*), mobile servers also act as sensors: vehicles can be equipped to provide operators with such information on their routes as travel time, local traffic densities, nearby pedestrians, local weather conditions, emergence of incidents, etc. Services like Waze already make use of travelers as sensors for updating network state information while most operator fleets (truck, transit, urban service vehicles, etc.) are equipped with devices like GPS and video equipment to provide information on the vehicle's traversed route. In most cases, fleet operators may just route vehicles to optimize service. However, if uncertainty in the network traffic state is significant or costly to obtain from external sources, there is value to using the vehicles as sensors as well. An example is to deploy a vehicle to deviate from a de facto route to gather more intelligence to be shared with subsequent trips serving that origin/destination (OD) pair, which some fleets already do if a traffic incident occurs.

The dual role of vehicles as both servers and sensors is especially relevant with automated vehicle (AV) fleets since much more information is continuously obtained as shown in **Figure 1**. AVs are rapidly becoming a reality and hold great promise for increasing efficiency and safety (*4*). A particularly attractive operational paradigm involves routing shared AV (SAV) fleets to provide service on-demand to customers. In recent years, there has been a surge of interest in testing such services like Uber in the U.S. and nuTonomy in Singapore *(5 – 6),* and more recently with deployments in Arizona, California, and Texas. There is clearly a need for algorithms to select routes for vehicles acting as both servers and sensors making repeated OD trips.

The decision of whether a vehicle should act more as a server or a sensor can be cast as a classic trade-off in reinforcement learning (*9*). A decision-maker makes a finite sequence (or an infinite sequence with nonstationary state variables) of repeated decisions where the underlying distribution of the reward for different options is initially unknown. Choosing an option is equivalent to sampling the reward, but if the reward is low, it is an opportunity cost if there is another option in which the expected reward is higher. This opportunity cost is measured as regret. The classic trade-off is whether to "exploit" known options or to "explore" other options to improve the potential for better exploitation in future decisions. The more the decision-maker samples from an alternative, the more they will learn the parameters of the distribution. On the other hand, if one alternative appears to be significantly better than the other after a few samples, it may make more sense to allocate the remaining choices to the first alternative. In a finite sequence, there is typically more exploration in the beginning and more exploitation later. Algorithms designed under such a setting are called multi-armed bandit (MAB) algorithms (*10*).

There are many different types of MAB algorithms. For the sake of brevity, we focus on the ones that have been developed for sequential path selection (e.g. (*11 – 13*)). These have been primarily applied to non-passenger networks, such as transmission networks, where only the shortest travel cost is desired over multiple samples.

Our contribution to this literature is based on the recognition that design of path selection algorithms for SAV fleets and urban service fleets handling a given OD pair needs to consider the schedule constraints of passengers and goods. Passengers care not only about the shortest travel time, but they also care about on-time arrival reliability and minimized schedule delay (*1, 14 – 17*). The empirical studies among this literature have found that travelers do place a fairly high value of time to schedule delay. For example, Small (*15*) found that work commuters delayed one minute would be equivalent to 2 to 11 minutes of travel time, depending on how far it is beyond the desired arrival time. Clearly, any MAB algorithm applied to finding shortest paths and learning about the paths should also consider the passenger objective of on-time arrival, especially when employed in a dynamic route choice setting with explicit departure time consideration. If they do not, how much worse can the results be? That is the research question we seek to address empirically in our experiment using real data.

We formulated and studied a Multi-Armed Bandit On-Time Arrival Problem (MABOTAP) that considers the dual role of vehicles as servers and sensors (of which SAVs are a major target technology) in repeated trips serving schedule-constrained passengers for a fixed OD pair. We empirically evaluated an algorithm customized to solve this problem. We designed a controlled experiment using real data collected from 504 real-time Google queries made over a 14-hour period (169 observations/path) to see how they would fare against conventional MAB algorithms based on shortest travel time only.

2. RELATED WORK

The shortest path problem on probabilistic graphs has been studied extensively under different considerations. Probabilistic path queries using weights as random variable following different distributions was first introduced by Frank (*18*). A Monte Carlo simulation was proposed to approximate the distribution of the shortest path. Loui (*19*) applied a utility function to define the preference among paths. More recently, adaptive routing has been studied as well. In these cases, decisions are made progressively within one trip to select which remaining links or subroutes to take based on updated information (e.g. (*20 – 21*)).

In *a priori* route selection problems under uncertainty, route optimization decisions that also consider optimizing learning efficiency for subsequent route selection decisions are multi-armed bandit problems. In the classical multi-armed bandit problem (MAB), a decision-maker must choose an arm from a list of arms to play. After playing an arm, the decision maker realizes a reward whose distribution is unknown. The objective is to maximize their total accumulated reward over a sequence of trials.

Thompson (*22*) first proposed an algorithm to address the MAB problem by playing the arm with the highest possibility of being the best arm. A widely used MAB algorithm

is the ε-greedy algorithm due to its simplicity. In each round t, the algorithm chooses with probability $1-\varepsilon$ the arm with the highest empirical mean and randomly selects any other arm with probability ε (*23*). Agrawal et al. (*24*) proved that the regret in that algorithm increases at a logarithmic rate as a worst case bound. Another popular algorithm is the upper confidence bound (UCB) method proposed by Auer et al. (*25*). UCB also achieves logarithmic regret with bounded support while satisfying the exploration and exploitation dilemma.

In recent years, bandit algorithms have gained increasing attention in solving shortest path problems. Each path is modeled as an arm. In the classical MAB setting, the arm rewards may be independent of each other (*26*). In the shortest path MAB, the arms are dependent through common links. Ignoring the dependence has led to poor regret performance. **Table 1** summarizes some of the major studies that have applied MAB to path selection problems. We note that none of these studies have considered passenger on-time arrival reliability as a measure for quantifying regret, nor have they considered joint route-departure time choice.

Signal routing has been studied extensively using multi-armed bandit algorithms as shown in Table 1. Both signal routing and vehicle route selection aim to reach a destination with highest payoff. However, the amount of information that a signal can send back is limited. In vehicle route selection, with improving emerging technologies like the 5G network and ubiquitous sensors, AVs and connected vehicles can send back much more information.

Shortest path problems with on-time reliability find paths that maximize the probability that the path length does not exceed a specified threshold (*30 – 33*). It is known as the "stochastic on-time arrival problem" (SOTA). Based on the concept of on-time arrival, many algorithms have been developed to find reliable paths in a stochastic network. Chen and Ji (*33*) proposed solving the on-time arrival problem by using a simulation-based genetic algorithm. Nie and Wu (*1*) introduced a label-correcting algorithm to find the most reliable paths by generating all non-dominated paths under the first-order stochastic dominance (FSD). A case study was conducted in Chicago (*34*) and heterogeneous risk-taking behavior was considered in Wu and Nie (*35*). Ji et al. (*36*) proposed an α-reliable path finding method for networks with correlated stochastic link costs. Chen et al. (*37 – 39*) introduced new algorithms based on dominance conditions to find reliable shortest paths under various correlated or time-dependent scenarios. These studies demonstrate the need to consider path selection under uncertainty with on-time arrival reliability. However, none of these cited studies consider joint learning and optimization in a sequential route selection setting.

Our work differs as follows: we used MAB algorithms to find the most reliable paths given a certain threshold. Previous reliable path finding algorithms have focused only in finding the most reliable path, while using MAB we can add the benefit of exploration when routing a vehicle over the same set of route alternatives (e.g. a shuttle service from a hotel to the airport, or flexible route microtransit service) over multiple trials.

To illustrate the difference from using only shortest path, consider an example where there are two parallel paths P_1 and P_2 between A and B along with two departure times t_1 and t_2, where $t_1 > t_2$. These departure times are relative to the desired arrival time. There are four alternatives to choose from: $P_{1,t_1}, P_{1,t_2}, P_{2,t_1}, P_{2,t_2}$, with some unobserved distribution for each. The left side of **Figure 2** shows how the distributions for travel time

can differ from the distributions of on-time arrival reliability, while the right-hand side shows how the on-time reliability distributions as estimated at the nth trial may still differ from the true distributions.

As illustrated in **Figure 2**, at a certain iteration n one alternative may seem to be best (P_{1,t_2}) but with enough exploration one will realize that another (P_{1,t_1}) is better. There is a need to incorporate the on-time reliability factors into the reward function because the distribution of on-time arrivals is different at different departures. To our best knowledge, this problem is not addressed in previous bandit literature.

3. PROPOSED METHODOLOGY

3.1. Preliminaries

$G(V, E, T)$ is a simple directed probabilistic graph. V is the set of vertices, E is the set of edges, and T is the period/trial of interest. This is the typical setup of a road network where the vertices are intersections and edges are roads between intersections. The period of interest is defined as $n\Delta$, where n is an integer and Δ is the length of a time interval. Within a time interval, there are multiple departure times (DTs) $\tau \in \{1, \dots, s\}$. A trip is performed by selecting both a path $P \in \{1,2, \dots, k\}$, which is a sequence of time-dependent links connecting an OD pair, and a departure time, resulting in ks different path-time choices. This is illustrated in **Figure 3** where there are three paths and three departure times (DTs) with a single preferred arrival time (PA). Depending on departure time, a path might be better or worse than another path in travel time and in terms of on-time arrival. This problem can be naturally cast as a sequential decision problem over multiple trials; e.g. a hotel shuttle to the airport departing each hour needs to decide both departure time and which route to take. While departure time is conventionally a continuous variable, fleet operations may schedule departures so using discrete time options makes sense.

The traveler conducts multiple trials of route selection and observes only the costs corresponding to the chosen paths in those trials. We define the arms as $a_{k,s} \in \{(1,1), \dots, (k,s)\}$. At each trial $t = 1, \dots, n$, for a corresponding preferred arrival time PA_t the traveler picks an arm $a_{k,s}$ from a subset of all possible path-times $P_{t,k,s}$ where the departure times relative to the PA_t are fixed across trials. For example, a trial may occur with preferred arrival times at the start of every hour, and the arms may be defined as a set of paths that depart 70 minutes, 60 minutes, and 50 minutes prior to that preferred arrival time. The observed choice at each trial t is denoted as $I_t \in P_{t,k,s}$ with associated cost C_{I_t}. To facilitate performance analysis in the bandit setting, we define the reward as $g_{I_t} = 1/C_{I_t}$ for each path-time choice I_t made. This conversion is used as most bandit algorithms are based on maximizing the reward (*25*).

3.2. Multi-Armed Bandit On-Time Arrival Problem (MABOTAP)

We introduce the Multi-Armed Bandit On-Time Arrival Problem (MABOTAP). The objective is to identify the arm with both the shortest travel time and most likely to arrive on time. In the traditional MAB road network setting, the objective is to find the arm leading to the largest reward G_{I_t}, where the reward is a function of travel time (*40*). Here we focus on the problem of finding the arm the minimizes the travel time and maximizes the on-time arrival reliability of the paths chosen over multiple trials.

To consider the early arrival time and later arrival time, we adopt Eq. (1) (see (*41*)).

$$C_{I_t} = tt_{I_t} + \alpha E + \beta D \quad (1)$$

Where
C_{I_t} is the realized weighted cost for path-time I_t
tt_{I_t} (minutes) is the travel time of chosen path-time I_t
τ_{I_t} (00:00) is the departure time of chosen path-time I_t
E is early arrival time defined as $E = \begin{cases} \tau_{I_t} + tt_{I_t} - PA, & if\ (PA_t - \tau_{I_t} - tt_{I_t}) \geq 0 \\ 0, & Otherwise \end{cases}$
D is the late arrival time defined as $D = \begin{cases} (PA_t - \tau_{I_t} - tt_{I_t}, & if\ (PA_t - \tau_{I_t} - tt_{I_t}) < 0 \\ 0, & Otherwise \end{cases}$
α is the penalty per unit of time for arriving early
β is the penalty per unit of time for arriving late
PA_t (00:00) is the preferred arrival time for the user in trial t

To measure the on-time reliability of different departure times of the paths, Pu (*42*) uses a reliability measure based on failure rate on road networks. However, Pu assumes a known distribution to find the failure rate while in our case the distributions of the path departure times are unknown. As such we measure the success and failure of arriving on time empirically as Q_t shown in Eq. (2).

$$Q_{t,k,s} = \frac{s_{t,k,s}}{s_{t,k,s} + f_{t,k,s}} \quad (2)$$

where
$s_{t,k,s}$ is the number of counts of on-time arrivals after trial t for arm $a_{k,s}$, which is updated each trial as Eq. (3).

$$s_{t,k,s} = \begin{cases} s_{t-1,k,s} + 1, & if\ P_{t,k,s} = I_t \text{ and } (PA_t - \tau_{I_t} - tt_{I_t}) \geq 0 \\ s_{t-1,k,s}, & Otherwise \end{cases} \quad (3)$$

$f_{t,k,s}$ is the number of counts of late arrivals after trial t for arm $a_{k,s}$, which is updated each trial as Eq. (4).

$$f_{t,k,s} = \begin{cases} f_{t-1,k,s} + 1, & if\ P_{t,k,s} = I_t \text{ and } (PA_t - \tau_{I_t} - tt_{I_t}) < 0 \\ f_{t-1,k,s}, & Otherwise \end{cases} \quad (4)$$

Eq. (2) for the on-time reliability measure decreases if a late arrival is recorded at time t and $a_{k,s}$ is chosen. For the unchosen arms, $Q_{t,k,s}$ stays the same because no new information is collected about the unchosen paths and the on-time reliability measure is not updated.

3.3. Proposed On-Time UCB Algorithm

We introduce an on-time reliability bandit algorithm called On-Time Confidence Bound (On-Time UCB) algorithm to achieve regret minimization of minimum travel time while achieving a good on-time reliability. Our algorithm uses the padding function derived from the original UCB (*25*) which is the logarithmic part of Eq. (5). This function is proven to help achieve a regret bounded to log order. The proposed **Algorithm 1** is presented here.

Algorithm 1: On-time Upper Confidence Bound Algorithm

For each arm $k = 1, \dots N$ and $\tau \in \{1, \dots, s\}$
Play each arm at least once
  Compute $\rho_{t-1,k,s}$ for all arms $k = 1, \dots N$ and $\tau \in \{1, \dots, s\}$
For each $t = N + 1, N + 2 \dots n$ Do
  Play arm $I_t = argmax_{k,s}\, \rho_{t-1,k,s}$
  Receive reward $g_{t,k,s}$ based on Eq. (1)
  Update $Q_{t,k,s}$ using Eq. (2)
  Update $\rho_{t,k,s}$ using Eq. (5)
  Update $G_{t,k,s}$ using Eq. (6)
  Update $T_{t,k,s} = T_{t-1,k,s} + 1$
End
Compute regret using Eq. (7)

The reliability score is shown in Eq. (5).

$$\rho_{t,k,s} = \begin{cases} Q_{t,k,s} * G_{t,k,s} + \sqrt{\dfrac{2\log(n)}{T_{t,k,s}}}, & if\ P_{t,k,s} = I_t \\ \rho_{t-1,k,s}, & otherwise \end{cases} \quad (5)$$

$T_{t,k,s}$ is the cumulative number of times arm $a_{k,s}$ is chosen up through trial *t*. $G_{t,k,s}$ is the mean reward of arm $a_{k,s}$ based on all trials up to *t* and is updated each trial using Eq. (6).

$$G_{t,k,s} = \begin{cases} g_{t-1,k,s} + \dfrac{1}{T_{t,k,s}}(g_{t,k,s} - g_{t-1,k,s}), & if\ P_{t,k,s} = I_t \\ g_{t-1,k,s}, & otherwise \end{cases} \quad (6)$$

The variable $g_{t,k,s} = 1/C_{I_t}$ is the reward received from choosing I_t and C_{I_t} is obtained from Eq. (1). To compute the regret, we compare the cumulative reward of the action taken at each time t with the overall best reward observed thus far, which is summarized in Eq. (7).

$$R = nG^*_{t,k,s} - \sum_{t=1}^{n} G_{t,k,s} \quad (7)$$

where G^* denotes the true optimal mean reward of all observed actions at all times (*23*). This optimal mean reward is obtained by assuming that in a hypothetical scenario, we can

pull each arm $a_{k,s}$ at each t. After n, we observe the mean reward of all arms and use that as the true optimal mean reward.

3.4. Numerical Example

We illustrate three scenarios using our proposed methodology. We randomly generated 3 paths with normally distributed travel times (in which we hide the parameters from the algorithm) for $n = 200$ rounds in each scenario. Each interval t is 5 minutes. The characteristics in each scenario are described in **Table 2**.

In this example we have 9 arms in our MABOTAP setting at each trial t. The traveler is faced with the decision to choose between three paths at three different departure times. For example, if t = 1 and a traveler's preferred arrival time (PA_1) is 8:45, then the choice may be to depart between 8:00, 8:05 or 8:10 as displayed in **Figure 4**. At t = 2, PA_2 is 8:50 and the departure time choices are 8:05, 8:10 or 8:15 for each path.

What should we expect to see? First, all the scenarios should see the average regret reduce with the number of trials until some bottom range. Second, the algorithm should sift through the options over time to eventually prefer the option that has the best combined minimum travel time (option A) and on-time arrival reliability (DT1 options). Third, there should be more fluctuation in the learning algorithm when standard deviation is higher (for example, comparing Scenario 3 to Scenario 2). Lastly, running the learning algorithm assuming only shortest path and ignoring on-time reliability should end up having a worst regret when that regret is measured in terms of on-time reliability as well (Eq. (6)).

For the scenarios shown in **Table 2**, we used our methodology to compute the regret and compared with the algorithm presented by György et al. (*12*), which is one of the most famous algorithms on selecting shortest path in a bandit setting. The results of the two different regrets are displayed in **Figure 5**.

The numerical results of the different scenarios are shown in **Table 3**. After running *n* iterations, the On-Time UCB algorithm performs better in terms of regret, empirical mean rewards and on-time arrivals. The regret difference between the SP and On-Time UCB was quite significant in Scenario 1, which can be explained by the higher variance in the rewards and because the proposed algorithm is more efficient in selecting more reliable arms. In Scenarios 2 and 3, when the variation between arms increase together uniformly, the differences in regret between the two algorithms are smaller. This experiment illustrates the effectiveness of the proposed algorithm in capturing on-time reliability.

4. EMPIRICAL EXPERIMENTATION

4.1. Experiment design and data

In this section, we empirically analyze the performance of our method. A case study in New York City is presented to validate our methodology. We setup the experiment to represent an artificial intelligence used in navigating a hotel shuttle from midtown Manhattan to the JFK International Airport. Real time route travel time data was collected using Google Maps API. The area of analysis is limited to a small section of New York City. We collected travel times of edges between the center of New York City and JFK Airport because we consider these as origin and destination respectively. The top 3 paths are shown in **Figure 6**.

We collected data for 30 days over 5-minute intervals from 7 am to 9 am. We chose a typical weekday (September 21st, 2017) among this data set to test the algorithm. We ran the SP and On-Time UCB algorithms on the data in a simulated online setting. For this experiment we had n = 169 intervals. We used 9 arms as in the previous example and the first iteration $t = 1$ was set to a $PA_1 = 8{:}05\ am$ with departure times of each path as 7:00am, 7:05am or 7:10am. The Google data is highly stochastic with variation in standard deviation and distribution throughout the day as shown in **Figure 7**. The data will be made available on https://github.com/BUILTNYU upon publication of this work.

4.2. Results

The average regret curve is shown in **Figure 8**. The On-Time UCB outperforms SP in terms of regret difference shown in **Figure 8** and in terms of on-time arrivals and mean rewards in **Table 4**. The on-time arrivals between the two algorithms are very close because in some cases the traveler will arrive late no matter which path and departure times they choose. As shown in the travel time distribution of paths in **Figure 7**, there were about 30 iterations (~18% of iterations) which produced late arrivals. This decreased the on-time arrivals of our On-Time UCB.

The results suggest that multi-armed bandit algorithms can be effective in learning which routes to select but applying them without considering behavioral preferences of travelers can be counterproductive. Testing the algorithms on the real data experiment shows that the proposed algorithm was able to improve on the arrival time percentage by 4.8% and total reward value by 66.1%.

5. CONCLUSION

We studied the use of bandit algorithms to solve sequential on-time arrival route selection problems. One primary application is the operation of autonomous vehicle fleets. Under the MABOTAP setting, a traveler must choose a path that maximizes their on-time arrival reliability while minimizing their expected travel time. An Upper Confidence Bound algorithm from Auer et al. (*25*) was extended to incorporate the on-time arrival criterion. Reward and regret measures appropriate to this criterion are introduced to guide the algorithm.

We conducted two sets of experiments. The first were computational experiments used on simulated data to demonstrate and verify the algorithm. The second set of experiments were performed using real time Google Maps API queries for three routes from a hotel in midtown Manhattan to the JFK International Airport. There are 169 intervals with three routes queried, resulting in 507 different route query observations used. We simulated an online environment of observing these routes based on applying a benchmark MAB algorithm assuming only shortest path and one using our proposed algorithm. The experimental results showed operating MAB algorithms using only shortest path without considering traveler scheduling preferences could negatively impact the on-time arrival probability (4.8% in our experiments) and total reward considering travel time as well (66.1%).

For future work, we will explore other bandit algorithms that can better address the stochasticity in road networks. Testing the algorithm in field settings would enable a fleet of AVs to effectively learn the road conditions used. MAB learning for routing to serve customers using vehicle routing problems can be another extension.

ACKNOWLEDGMENTS
This research was conducted with the support of NSF CAREER grant CMMI-1652735.

AUTHOR CONTRIBUTION STATEMENT
The authors confirm contribution to the paper as follows: study conception and design, analysis and interpretation of results, draft manuscript preparation: J. Zhou, J. Y. J. Chow; data collection: J. Zhou, X. Lai. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

1. Nie, Y., & Wu, X. (2009). Shortest path problem considering on-time arrival probability. *Transportation Research Part B* 43(6), 597–613.
2. Ottomanelli, M., & Wong, C.K. (2011). Modelling uncertainty in traffic and transportation systems. *Transportmetrica*, 7(1), 1–3.
3. Chow, J.Y.J. (2018). *Informed Urban Transport Systems: Classic and Emerging Mobility Methods toward Smart Cities*. Elsevier, Amsterdam, The Netherlands.
4. Mahmassani, H.S., 2016. 50th Anniversary Invited Article—Autonomous Vehicles and Connected Vehicle Systems: Flow and Operations Considerations. *Transportation Science* 50(4), 1140-1162.
5. Aupperlee, A. (2018, July 24). Uber's self-driving fleet back on Pittsburgh roads ... but drivers are in control. Retrieved November 6, 2018, from https://triblive.com/local/allegheny/13897453-74/ubers-self-driving-fleet-back-on-pittsburgh-roads-but-drivers-are-in
6. NuTonomy. (2017). nuTonomy launches world's first public trial of self-driving car service and ride-hailing app. Retrieved November 6, 2018, from https://www.nutonomy.com/press-release/singapore-public-trial-launch/
7. Goebel K. & Agogino A. M. Sensor validation and fusion for automated vehicle control using fuzzy techniques. *Journal of Dynamic Systems, Measurement, and Control*, 123(1):145--146, 2001.
8. NVIDIA, 2017. Accelerating AI with GPUs: a New Computing Model. https://blogs.nvidia.com/blog/2016/01/12/accelerating-ai-artificial-intelligence-gpus/, last accessed July 31, 2017.
9. Powell, W.B. and Ryzhov, I.O., 2012. Optimal learning (Vol. 841). John Wiley & Sons.
10. Li, L., Chu, W., Langford, J. and Schapire, R.E., 2010, April. A contextual-bandit approach to personalized news article recommendation. In Proceedings of the 19th international conference on World wide web (pp. 661-670). ACM.
11. Awerbuch, B. and Kleinberg, R.D., 2004, June. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In Proceedings of the thirty-sixth annual ACM symposium on Theory of computing (pp. 45-53). ACM.

12. György, A., Linder, T., Lugosi, G. and Ottucsák, G., 2007. The on-line shortest path problem under partial monitoring. *Journal of Machine Learning Research* 8(Oct), pp.2369-2403.
13. Liu, K. and Zhao, Q., 2012, May. Adaptive shortest-path routing under unknown and stochastically varying link states. In Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2012 10th International Symposium on (pp. 232-237). IEEE.
14. Hendrickson, C. and Kocur, G., 1981. Schedule delay and departure time decisions in a deterministic model. *Transportation science*, 15(1), pp.62-77.
15. Small, K.A., 1982. The scheduling of consumer activities: work trips. The American Economic Review, 72(3), pp.467-479.
16. Arnott, R., De Palma, A. and Lindsey, R., 1990. Economics of a bottleneck. Journal of urban economics, 27(1), pp.111-130.
17. Kang, J.E., Chow, J.Y.J., Recker, W.W., 2013. On activity-based network design problems. Transportation Research Part B 57, 398-418.
18. Frank. H. (1969) Shortest paths in probabilistic graphs. *Operations Research*, 17(4):583—599.
19. Loui, R. P. (1983). Optimal paths in graphs with stochastic or multidimensional weights. *Communications of the ACM*, *26*(9), 670-676.
20. Hall, R.W., 1986. The fastest path through a network with random time-dependent travel times. *Transportation Science*, 20(3), 182-188.
21. Fu, L., 2001. An adaptive routing algorithm for in-vehicle route guidance systems with real-time information. *Transportation Research Part B* 35(8), 749-765.
22. Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, *25*(3/4), 285-294.
23. Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction* (Vol. 1, No. 1). Cambridge: MIT press.
24. Agrawal, S., & Goyal, N. (2012, June). Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory* (pp. 39-1).
25. Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, *47*(2-3), 235-256.
26. Lai, T. L., & Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, *6*(1), 4-22.
27. Zou, Z., Proutiere, A., & Johansson, M. (2014, June). Online shortest path routing: The value of information. In *American Control Conference (ACC), 2014* (pp. 2142-2147). IEEE.
28. Talebi, M.S., Zou, Z., Combes, R., Proutiere, A. and Johansson, M., 2017. Stochastic online shortest path routing: The value of feedback. *IEEE Transactions on Automatic Control*.
29. Chorus, C.G., 2012. Regret theory-based route choices and traffic equilibria. *Transportmetrica*, 8(4), pp.291-305.
30. Fan, Y. Y., Kalaba, R. E., & Moore, J. E. (2005a). Arriving on time. *Journal of Optimization Theory and Applications*, *127*(3), 497-513.
31. Fan, Y. Y., Kalaba, R. E., & Moore, J. E. (2005b). Shortest paths in stochastic networks with correlated link costs. *Computers & Mathematics with Applications*, *49*(9), 1549-1564.

32. Nikolova, E., Kelner, J., Brand, M., & Mitzenmacher, M. (2006). Stochastic shortest paths via quasi-convex maximization. *Algorithms–ESA 2006*, 552-563.
33. Chen, A., & Ji, Z.W. (2005). Path finding under uncertainty. *Journal of Advanced Transportation*, 39(1), 19–37.
34. Nie, Y.M., Wu, X., Dillenburg, J.F. and Nelson, P.C., 2012. Reliable route guidance: A case study from Chicago. *Transportation Research Part A* 46(2), 403-419.
35. Wu, X. and Nie, Y.M., 2011. Modeling heterogeneous risk-taking behavior in route choice: A stochastic dominance approach. Procedia-Social and Behavioral Sciences, 17, pp.382-404.
36. Ji, Z., Kim, Y.S. and Chen, A., 2011. Multi-objective α-reliable path finding in stochastic networks with correlated link costs: A simulation-based multi-objective genetic algorithm approach (SMOGA). Expert Systems with Applications, 38(3), pp.1515-1528.
37. Chen, B.Y., Lam, W.H., Sumalee, A. and Li, Z.L., 2012. Reliable shortest path finding in stochastic networks with spatial correlated link travel times. *International Journal of Geographical Information Science* 26(2), 365-386.
38. Chen, B.Y., Lam, W.H., Sumalee, A., Li, Q., Shao, H. and Fang, Z., 2013. Finding reliable shortest paths in road networks under uncertainty. *Networks and spatial economics* 13(2), 123-148.
39. Chen, B.Y., Lam, W.H., Sumalee, A., Li, Q. and Tam, M.L., 2014. Reliable shortest path problems in stochastic time-dependent networks. *Journal of Intelligent Transportation Systems* 18(2), 177-189.
40. Ramos, G. D. O., da Silva, B. C., & Bazzan, A. L. (2017, May). Learning to minimise regret in route choice. *In Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems* (pp. 846-855). International Foundation for Autonomous Agents and Multiagent Systems.
41. De Palma, A., Hansen, P., & Labbé, M. (1990). Commuters' paths with penalties for early or late arrival time. *Transportation Science*, *24*(4), 276-28
42. Pu, W. (2011). Analytic relationships between travel time reliability measures. *Transportation Research Record: Journal of the Transportation Research Board*, (2254), 122-130.

List of Figures

Table 1. Summary of studies involving MAB applications for path selection

Author(s)	**Applications**	**Objective**
Awerbuch & Kleinberg (*11*) Chen et al. (*33*) György et al. (*12*) Liu and Zhao (*13*) Zou et al. (*27*) Talebi et al (*28*)	Network signal routing	Shortest Path Selection
Chorus (*29*) Ramos et al. (*40*)	Traffic Route Planning	Shortest Path Selection

Table 2. Synthetic data scenarios used to test proposed algorithm

Preferred Arrival Time PA_t (00:00)	**Departure Times**	**Mean (minutes)**	**Scenario**	**Standard Deviation (%mean)**	**On Time Probability**		
					DT1	DT2	DT3
$PA_1 = 08{:}45$ am $PA_2 = 08{:}50$ am ... $PA_{n=200} = 01{:}20$ am	t=1, $DT_1 = 8{:}00$ am, $DT_2 = 8{:}05$ am, $DT_3 = 8{:}10$ am;	A. 30 B. 35 C. 33	1	A. 33	0.94	0.84	0.69
				B. 17	0.95	0.80	0.50
				C. 24	0.94	0.81	0.60
	t=2, $DT_1 = 8{:}05$ am, $DT_2 = 8{:}10$ am, $DT_3 = 8{:}15$ am;		2	A. 10	1.00	0.99	0.95
				B. 10	0.99	0.92	0.50
	...			C.10	0.99	0.98	0.73
	t=n, $DT_1 = 00{:}45$ am, $DT_2 = 00{:}50$ am, $DT_2 = 00{:}55$ am;		3	A. 20	0.99	0.95	0.80
				B. 20	0.92	0.76	0.50
				C. 20	0.97	0.86	0.62

Table 3. Numerical results after running the 3 scenarios.

Scenario	SP On time Arrivals	On-Time UCB On-time Arrivals	SP Mean rewards	On-Time UCB Mean rewards
1	47%	75%	0.0347	0.0394
2	87%	91%	0.0325	0.0351
3	70%	77%	0.0305	0.0320

Table 4. Numerical results of the algorithms

Scenario	SP On time Arrivals	On-Time UCB On-time Arrivals	SP Mean rewards	On-Time UCB Mean rewards
Google data (Sept 21, 2018)	62%	65%	0.180	0.299