Contextual bandit-based sequential transit route design under demand uncertainty

Gyugeun Yoon
Ph.D. Candidate
Department of Civil and Urban Engineering
New York University
15 MetroTech Center, Brooklyn, NY 11201
Email: ggyoon@nyu.edu
ORCiD: 0000-0003-1622-9021

Joseph Y. J. Chow
Deputy Director, C2SMART University Transportation Center
Assistant Professor, Department of Civil and Urban Engineering
New York University
15 MetroTech Center, Brooklyn, NY 11201
Email: joseph.chow@nyu.edu
ORCiD: 0000-0002-6471-3419

Word Count: 5,946 words + 4 tables = 6,946 words
(words in tables excluded)

Submitted February 9, 2020

ABSTRACT

While public transit network design has a wide literature, the study of line planning and route generation under uncertainty is not so well covered. Such uncertainty is present in planning for emerging transit technologies or operating models in which demand data is largely unavailable to make predictions on. In such circumstances, we propose a sequential route generation process in which an operator periodically expands the route set and receives ridership feedback. Using this sensor loop, we propose a reinforcement learning-based route generation methodology to support line planning for emerging technologies. The method makes use of contextual bandit problems to explore different routes to invest in while optimizing the operating cost or demand served. Two experiments are conducted. They (1) prove that the algorithm is better than random choice; and (2) show good performance with a gap of 3.7% relative to a heuristic solution to an oracle policy.

Keywords: Transit network design, line planning problem, reinforcement learning, route generation, contextual bandit problem

1 INTRODUCTION

Public transit network design (see *[1, 2]*) involves five general steps: route design, frequency setting, timetabling, vehicle scheduling, and crew scheduling. The collective problem is generally regarded as intractable and algorithms developed over the years addressed subgroups, such as the line planning problem (LPP) *[3]* to combine route design and frequency setting. Algorithms have advanced far (e.g. *[4]*) since the earliest efforts in line planning *[5, 6]*, but fundamental challenges remain. One challenge that continues to be actively studied is the route set construction problem *[7]*. Others include transfer optimization *[8]*, the NP-hardness of the problem *[9]*, and the complexity of user route choice involving hyperpaths and varying congestion effects *[10]*.

The underlying component of LPPs is the generation of the route set (e.g. *[7]*) which assumes a deterministic demand setting, i.e. the origin-destination (OD) demand are known in advance as constants or functions of system performance. However, uncertainty in demand is a very significant problem. Inaccuracies in transit project demand forecasts are well-known *[11]* with rail investments on average overpredicting demand by 40% *[12]*. With the high capital and operating costs of transit, designing routes under this uncertainty can be very costly. Even the cost of reducing this uncertainty through surveys can be an expensive undertaking. For example, Chicago Metropolitan Agency for Planning budgeted $2.7 million for their household travel survey, asking 12,000 households out of more than 3 million households within their region for 9 months, ending up sampling only 0.4% of the total population *[13, 14]*. The degree of uncertainty is further pronounced for emerging transit technologies; new types of services like microtransit *[15]* that can use crowdsourced data or shared autonomous vehicle fleets *[16]* have little to no data to begin with, and each new city deployment requires starting from scratch.

Costly information collection is an important problem. When data is limited, even knowledge of the distributions of the demand is limited. A bus company choosing to deploy a route over one of two regions would be able to not only serve that region but also learn important details about the demand in that region that they could have otherwise learned about the other region. Optimal information collection in a network setting is an emergent topic (e.g. *[17, 18]*) that involves deploying resources to proactively learn the system state/parameters to maximize cumulative rewards. Methods in this class of problems fall under reinforcement learning *[19]*, with one popular concept being the "multi-armed bandit" problem *[20]*.

We propose a new algorithm for the sequential route design problem. For the sake of better comparison of algorithms, we restrict the decision variables only to route design within the LPP and ignore frequency setting or timetables (although the method can work with those considerations as well). The objective of such an algorithm is to strategically sequence the operation of routes in a network over time such that earlier routes formed would act as sensors to learn the characteristics of the demand so that the cumulative reward over a finite horizon is optimized. This is the first such study to consider sequential route design under uncertainty, and the first to incorporate reinforcement learning theory to generate transit routes that can both serve passengers and learn about them at the same time. Applications of such work extend beyond transit line planning to general network design, mobile sensor location, emerging technology deployment, and towards self-organizing transit fleet growth for shared autonomous vehicles *[16]*. To be clear, this work targets two practical contributions:

1) In existing transit planning, agencies already adjust their routes (e.g. bus network redesigns, of which there have been numerous instances) and schedules every few months. In some cities

with private transit operations (e.g. minibuses in Hong Kong, or the Ford Chariot with its crowdsourcing of routes prior to shutting down), routes might be changed even more regularly. This work argues in favor of building in an optimal learning consideration to help agencies learn about their users much more efficiently (in the case of a new service or new nearby developments, for example).

2) With the many deployments of Shared Autonomous Vehicle fleets currently (see EasyMile, Optimus Ride, and Navya, for example), there is very little information on local user demand for such modes. Our work would provide AI to these autonomous systems to grow more organically as they learn and adapt to their markets.

This study consists of the following remaining parts. First, we review related literature and previous works that investigated feasible route design problems or transit demand learning. Second, we summarize approaches and concepts of use to this study. Third, the research methodology and proposed algorithm are presented. Fourth, we bring sample cases to verify the proposed algorithm and evaluate the performance by benchmark comparison. Finally, the conclusion, introduction of future works, and expected applications are delivered.

2 LITERATURE REVIEW

The survey consists of two main areas – route design problems and reinforcement learning.

2.1 Route design problems

A feasible route is a route satisfying prerequisites regarding geometrical or operational attributes such as total length or mandatory visit to certain nodes. It is a crucial element of LPPs. Schöbel *[3]* classified LPPs into 4 types of models: cost-oriented models minimizing operational cost, passenger-oriented models maximizing direct travelers or minimizing traveling and riding cost, game-theoretic models dealing with delay and operators' profit, and location-based models concerning the access distance or network coverage.

This study falls under the passenger-oriented models as the objective involves maximizing ridership or providing more benefits to passengers. Passenger-oriented models employ several methodologies for route set construction: a heuristic route construction algorithm recommending feasible routes focused on serving demand directly between origins and destinations *[21]*, genetic algorithm *[22]*, column-generation algorithm *[9]*, ant colony optimization *[23]*, robust optimization *[24]*, two-phase stochastic program *[25]*, mobile phone trajectory process *[26]*, adaptive neighborhood search metaheuristic *[4]*, or particle swarm optimization *[27]*. These methods all assume OD demand is revealed and derivable.

As an example, the heuristic devised by Ceder and Wilson *[21]* is implemented on the small network in **Figure 1**. It requires the shortest path time and the passenger demand among nodes to verify whether enumerated routes are feasible or not. To make decisions, it uses several criteria including: 1) total route length should lie between an upper and lower bound designated by the policy, and 2) travel time between nodes cannot exceed a maximum allowable travel time for each OD pair. According to this heuristic, there are 470 feasible routes with 100 minutes of the maximum round trip time among all 644 physically available routes (235 of 322 after removing symmetric ones). The problem size increases to 2,580,785 candidate routes when the grid network simply increases from 3×3 to 5×5.

The latest studies have devised methodologies to avoid the route enumeration as it can consume much computational effort. On the other hand, our learning approach makes use of candidate routes so it makes sense to consider at least some degree of route enumeration.

2.2 Reinforcement learning

Line planning under uncertainty is not well studied. A few studies tackle line planning with two-stage stochastic programs *[25, 28]* or robust optimization *[24]*. One strategy to mitigating uncertainty over a time horizon is to consider the buildout over multiple stages and to adapt subsequent stages to prior outcomes *[29]*. Staged development is a natural approach to transit networks, resulting in observable evolutions over time (see *[30-33]*). This notion of flexibility leads to a sequential network design problem under uncertainty *[34]*, a more complex category of general Markov decision processes (see *[35]*) where decisions are made dynamically over multiple periods with information revealed over time. Approximate dynamic programming algorithms are typically used to optimize such problems, although no literature exists for sequential LPP under uncertainty.

Algorithms in reinforcement learning (and machine learning in general) have a wide literature; we restrict this to a more focused review on multi-armed bandit (MAB) algorithms as such. Ayyadevara *[36]* provides a good overview of different machine learning techniques, including linear and logistic regression, decision tree, random forest, gradient boosting machine, artificial neural network, convolutional neural network, recurrent neural network, clustering, and principal component analysis.

MAB problems are sequential decision processes that consider selection of one among multiple alternatives over multiple trials. Each alternative represents an "arm" whose selection provides a reward to the user. The decision-maker starts out not knowing the distribution of the rewards of each alternative; they are learned through exploration of the arms over multiple trials, where the reward each trial is a random outcome. This algorithm is widely employed in computer science and operations research including cognitive radio networks *[37]*, design of clinical trial *[38]*, web-based recommendation, advertising *[39]*, and transportation *[40]*.

MAB algorithms aim to minimize regret R_n, the discrepancy between a maximally achievable reward and the acquired reward as shown in Eq. (1). For $K \geq 2$ arms, the rewards regarding each arm i at time step t, $X_{i,t}$, follow unknown distribution. When a user chooses an arm I_t, the reward is $X_{I_t,t}$ *[41]*.

$$R_n = \max_{i=1,\cdots,K} \sum_{t=1}^{n} X_{i,t} - \sum_{t=1}^{n} X_{I_t,t} \tag{1}$$

A MAB algorithm consists of T time steps of which an initialization period τ and a learning period $(T - \tau)$. While an algorithm focuses on collecting information about rewards from arms during τ, it uses that information for the remaining $T - \tau$ time steps to maximize the cumulative reward by choosing the best arm evaluated by the acquired knowledge. If optimally predicted, R_n can be 0, meaning that the user always chooses the best option. In general, true values of regret cannot be obtained because distributions of rewards are not known *a priori*. Instead, observable measures are used as proxies. For instance, "click-through ratio" (CTR) is popular in the area of

online content providers and advertisers, a binary indicator which becomes 1 when a link gets a click and 0 otherwise.

Some studies examined different MAB algorithms and found that the differences in algorithms primarily deal with how to choose sets of arms to learn the prevailing environment (e.g. *[41, 42]*). A variant of the MAB is the contextual bandit problem, which associates the arms with feature vectors, and stochastic rewards are determined by the model of which parameters are obtained and estimated from previous choices. Li et al. *[39]* introduced a generalized linear contextual bandit algorithm which is compatible with such generalized linear models (GLMs) as logistic regression. Eq. (2) describes how the relationship between the feature of the chosen arm, X, and the corresponding reward, Y, can be explained by the link function μ with the parameter θ^*. X' is the transpose of X.

$$E[Y|X] = \mu(X'\theta^*) \tag{2}$$

Applications exist in areas of mobility. Researchers have studied the demand management of electric vehicle charging stations by changing charging prices and recommending alternative stations when one is congested *[43]*. Zhou et al. *[40]* developed a recommender system for sequential departure time and path choice with on-time arrival reliability. Huang et al. *[44]* proposed a learning policy to efficiently allocate delivery vehicles in an urban area to minimize the expected cumulative operational cost.

In transit planning, Zolfpour-Arokhlo et al. *[45]* developed a route planning model using multi-agent reinforcement learning in a Malaysian intercity road network to reduce travel time between cities. Guo et al. *[15]* and Li et al. *[31]* designed optimal timing policies in transportation operational planning using real options theory. Khadilkar *[46]* proposed a reinforcement learning algorithm for scheduling railway lines. Tsai et al. *[47]* proposed a deep neural network approach to predict bus passengers. While not a reinforcement learning application, Cats and West *[48]* proposed a day-to-day learning procedure for updating transit passenger demand in dynamic transit assignment. MABs, and contextual bandit algorithms in particular, have not yet been applied to sequential LPPs.

3 METHODOLOGY

The sequential route design framework differs from conventional route design problems in LPPs; it is described in **Figure 2**. In conventional route design problems, the route pool is generated from identified information covering the potential network of the region. The constructed line planning problem given that route pool is solved by either an analytical model or heuristic algorithm. The route set solution would be deployed, leading to the information feedback to the demand estimation from collected demand data while operating the system.

On the contrary, we assume a setting where the demand information in the dashed border is incomplete. Thus, operators would generate an initial route pool only relying on other information as input to the reinforcement learning-based route planning problem. In this procedure, demand learning from the actual environment is implemented to acquire demand information more efficiently. A user behavior model is specified and updated over multiple deployments and information feedback. Potential passengers react to the system resulting in flows along chosen

routes in the network, which is monitored by the system to update the demand information and increase the accuracy of the model estimation.

We address whether (a) solving this sequential route design problem with a MAB-based algorithm is more effective than solving it without MAB (or under what conditions), and whether (b) that is a more effective transit planning strategy than to deploy with minimally collected data at the start with a fully committed design.

3.1 Route planning problem

A conventional mathematical program for the route planning problem is presented in Eq. (3) – (8). The objective function represents a total passenger travel cost to be minimized.

$$\min \sum_{(r,s)\in D} \sum_{k\in K_{rs}} c_{rs}^k x_{rs}^k \tag{3}$$

s.t.

$$\sum_{k\in K_{rs}} x_{rs}^k = d_{rs}, \ \ \forall (r,s) \in D \tag{4}$$

$$\sum_{k\in K} \sum_{r\in P_l^k, s\in Q_l^k} x_{rs}^k \le u_l, \ \ \forall l \in A \tag{5}$$

$$x_{rs}^k \le M w_k, \ \ \forall k, (r,s) \in D \tag{6}$$

$$x_{rs}^k \ge 0, \ \ \forall k, (r,s) \in D \tag{7}$$

$$w_k \in \{0,1\}, \ \ \forall k \tag{8}$$

where,
c_{rs}^k: travel cost between origin r and destination s using route k
x_{rs}^k: passenger flow between origin r and destination s using route k
D: OD pair set
K_{rs}: set of routes providing service between origin r and destination s,
where $K = \bigcup_{(r,s)} K_{rs}$ is the set of all routes
d_{rs}: demand between origin r to destination s
P_l^k: set of nodes on route k located before link l,
Q_l^k: set of nodes on route k located after link l
u_l: capacity of link l
A: set of all links
c_l: travel cost of link l, where $c_{rs}^k = \sum_{l\in L_{rs}^k} c_l \,, \ \ \forall k, (r,s) \in D$
L_{rs}^k: set of links included in route k between origin r and destination s
w_k: 1 if route k is used, otherwise 0

Eq. (4) is a demand conservation constraint. Eq. (5) describes the link capacity. Eq. (6) is a fixed charge constraint allowing flow only if the route is selected, $w_k = 1$. Eq. (7) is the nonnegativity condition of x_{rs}^k. Eq. (8) is the binary condition of w_k, and the action space is a binary vector with a size equivalent to the number of available routes.

In the sequential setting, as shown in **Figure 2**, we still have a candidate route set. However, we are given up to a route size budget of T route investments. Instead of investing in all T routes right away, we interject a period of time η after each route investment to observe the ridership for all OD pairs served by all existing routes. The length of η can be days, months, or years, depending on the type of system and the type of decision-making (tactical versus strategic). For example, with $T = 60$ and $\eta = 1$ month, the operator will take 60 months (assuming each route can start instantaneously) to build out 60 routes, taking one month after each investment to learn from the data. The objective of this problem is to collect the best accumulated performance over the 60 months and decision variables will be binary variables that indicate whether the corresponding route is chosen or not.

In practice, the sequential planning of transit routes may be less effective for some agencies due to the process of public discussion and approval. These steps may include community public hearings, expert meetings, or legislative procedures to successfully initiate a new design. Nonetheless, route redesigns occur quite frequently and are even more prevalent in private operator systems (new transit business models like crowdsourcing routes with Ford Chariot or new shared autonomous vehicle fleet deployments) that can benefit even more from this model framework.

3.2 Upper Confidence Bound-Generalized Linear Model algorithm (UCB-GLM)

The UCB-GLM *[39]* is a contextual bandit algorithm where the upper bound of regret has been analytically derived, making it a reliable reinforcement learning algorithm. We consider using this algorithm together with the route planning problem. However, because the l_2-norms of feature vectors are larger than 1 and are not independently distributed, they violate the condition for the bound. As a result, our problem cannot guarantee this stochastic upper bound of regret. Although the normalization of feature vectors could convert them to fulfill this criterion, feature vectors of alternatives in our study are highly intertwined. They are not independent and identically distributed (IID) as they are substantially correlated. For example, if Alternative route A and B share a huge portion of their route, it can be hardly said that they are independent. Nevertheless, the study can borrow the concept of UCB-GLM because it can estimate the parameters of the link function μ, the generalized linear function.

During the learning period τ, the algorithm has opportunities to randomly choose an arm from the set of available alternatives $[K]$, and observe a response from the unknown system. X_t, the feature vector of the arm chosen at the time step t, can contain some significant information including either quantitative measures (generalized travel cost, fare) or qualitative ones (service reliability, quality). In the subsequent time steps prior to reaching T, the expected reward is estimated by applying updated estimators. A tuning parameter α controls the extent of exploration after τ.

After estimating $\hat{\theta}_t$ each trial t, the algorithm recommends choosing the arm a_t to maximize the corresponding arm value that consists of an estimated reward, the linear combination of feature vector and estimated coefficients, and exploration term. This last term is the product of exploration factor α and the norm vector $\|x\|_A \coloneqq xAx'$. It also keeps observing responses by selecting the arm which predicts the highest return. Note that Eq. (1) regarding the regret of choosing suboptimal

routes does not explicitly appear in **Algorithm 1** since the regret is unobservable when running the model.

Algorithm 1 UCB-GLM *[39]*

Input: the total rounds T, tuning parameter τ and α.
Initialization: randomly choose $a_t \in [K]$ for $t \in [\tau]$, set $V_{\tau+1} = \sum_{i=1}^{\tau} X_t {X_t}'$
For $t = \tau + 1, \tau + 2, \cdots, T$ do
1. Calculate the maximum-likelihood estimator $\hat{\theta}_t$ by solving the Eq. (9).

$$\sum_{i=1}^{t-1} \big(Y_i - \mu(X_i'\theta)\big) X_i = 0 \tag{9}$$

2. Choose $a_t = \text{argmax}_{a \in [K]} \left(X_{t,a}' \hat{\theta}_t + \alpha \left\| X_{t,a} \right\|_{V_t^{-1}} \right)$
3. Observe Y_t, let $X_t \leftarrow X_{t,a_t}$, $V_{t+1} \leftarrow V_t + X_t {X_t}'$
End For

3.3 Logit model for route choice

The last component is the demand model for a given route design. A multinomial logit model is used for route choice. If the vector of parameters of the utility function is $\boldsymbol{\theta}$, the probability $P_n(i)$ of choosing Option i among J set of routes is shown in Eq. (10), where some may not choose any if $\gamma = 1$, or choose only among the routes if $\gamma = 0$. In the case of a binary logit model, the utility of rejection is set to zero to represent other alternative travel options.

$$P_n(i) = \frac{\exp(\boldsymbol{\theta X_{in}})}{\sum_{j=1}^{J} \exp\big(\boldsymbol{\theta X_{jn}}\big) + \gamma} \tag{10}$$

This approach does have well-known limitations in terms of route overlaps (see *[49-51]*). For this study we focus on the learning aspect and assume that riders perceive route options as sufficiently independent of each other. More sophisticated choice models can also be adopted for implementation but is beyond the scope of this work.

4 ALGORITHM DESIGN

Having introduced the three components of the system, we now propose **Algorithm 2** to solve the sequential route generation problem.

ALGORITHM 2. To solve the sequential route generation problem with UCB-GLM-route planning

1) Formulate the route planning problem under appropriate assumptions.
2) Initialize. $t = 0$, and x_{rs}^{k0}, w_k^0, $K_{rs} = \phi$.
3) Randomly choose Route k_t and $w_{k_t}^t = 1$ ($K_{rs} = \{k_t\}, \forall t \in [1, \tau]$).
4) Observe $z_t = x_{rs}^{kt} \leq d_{rs}^t$ potential passengers and collect information including attributes of Route k_t, X_t, the reward, $Y_{z_t,t}$ and $V_{z_t,t+1} = \sum_{i=1}^{t} z_t X_t {X_t}'$.

5) Repeat Step 3) and 4) until $t = \tau$, tuning parameter.
6) Calculate maximum-likelihood estimator $\hat{\theta}_t$ and choose a route which is expected to achieve the highest arm value.
7) Move the chosen route from the enumerated route set to the chosen route set and observe $z_t = \sum_{k \in K_{rs}} x_{rs}^{kt} \leq \sum_{(r,s)} d_{rs}^t \ \forall (r,s) \in D$ passengers and collect information as the same as Step 4).
8) If stopping criterion is satisfied, stop. Otherwise, go to Step 6) and repeat until $t = T$.

The problem formulated in Step 1 is the same as Eq. (3) – (8). The travel demand between each node pair r and s is fixed for the observation period η, but may change from trial to trial. In the case where demand is elastic, we would change the objective to maximizing total demand served. An example of this is covered later in the experiments.

Before beginning the algorithm, the time step should be 0, and other variables should have no values assigned. For Step 3, during the learning period when $t \leq \tau$, the algorithm chooses a route randomly from an enumerated set of routes which satisfies a prerequisite such as total length. When route k_t is chosen during Step 3, $w_{k_t}^t$ becomes 1 automatically and K_{rs} is a singleton whose element is k_t.

In Step 4, as Route k connects Node r and s, at most d_{rs}^t passengers will access the route and decide to take or skip a ride until the capacity of the route is filled or all potential passengers conclude their actions. Although there is only one observation during a time step in the original UCB-GLM, the operator can observe z_t potential passengers. The actual probability of choosing an arm ($P_n(i)$ in Eq. (10)) is not explicitly in the algorithm. We do make use of it in the simulation experiment in Section 5, however, to simulate choice behaviors of users. The users behave according to a binary choice model where the true parameters are known to the simulator but hidden from the algorithm to evaluate the efficiency of learning them.

Step 6 is borrowed from the UCB-GLM algorithm, especially the calculation after time step τ to choose the most popular route. Once a route is chosen, it should be excluded to avoid the duplication of route. Step 7 is also borrowed from UCB-GLM algorithm, especially the calculation after time step τ to choose the most popular route. Several different stopping criteria are applicable to Step 8. For example, the algorithm can terminate when the difference of objective function between previous and current iteration is within a certain tolerance ($|Z_t - Z_{t-1}| \leq \delta$), when the action space w_k^t is stable without any update for several consecutive iterations ($w_k^t = w_k^{t-1} = \cdots = w_k^{t-a}$), or when the number of experiments after τ reaches a budget B ($t - \tau = B$). The last one is the criterion that we considered in the experiments.

5 NUMERICAL EXPERIMENTS

Two example networks are formed to test the proposed algorithm and evaluate the performance. Due to the difference between both networks in terms of complexity and structure, basic assumptions to analyze the system are modified to reflect prevailing conditions.

5.1 Single OD pair with n routes consisted of single link

The first network involves a pair of nodes of origin and destination, and 50 links connecting both, providing 50 different routes to passengers. It is assumed that they only take one-way trips and do not come back to the origin as described in **Figure 3**. Each route has three attributes: travel time, capacity, and attractiveness. While the travel time c_k is given as a continuous value in minutes, the

capacity is an integer variable between 5 and 10. The attractiveness S_k is also a discrete variable with a 3-point scale parameter of 0, 1, and 2. Their attributes are indicated in **Table 1**.

Initially, it is assumed that there are 100 trips that require this service, and the utility function of route k is $U_p^k = 3.5 - 0.3c_k + 0.15S_k$. Each route is ordered to observe passengers with the same amount as the route capacity. The initialization period τ is shorter than the total experiment period $T = 100$ time steps and randomly generated to obtain a relation to the objective function value.

Two different policies are applied. "Risk aversion after random Choice (RC)" only chooses the route set out of explored routes without taking any risk of choosing undiscovered routes, while "Learning Demand (LD)" is the proposed **Algorithm 2**.

For each policy, more than 10,000 simulation are conducted, and **Figure 4a** is the distribution of total passenger costs regardless of their length of τ. During the simulation, the τs are randomly designated to an integer between 13 and 50. The histograms show the significant difference in shapes as well as the mean, 989.49 for RC and 716.60 for LD.

The trend of the average total passenger cost by τ is plotted in **Figure 4b**. As τ increases, total cost obtained in RC decreases and approaches gradually to the curve of LD which indicate almost a simple line parallel to the x-axis. This implies that demand learning is the more efficient process to minimize the total cost with a tighter optimality gap.

5.2 Multiple OD pairs with the single origin on 3×3 grid

The second network is a typical grid consisting of 9 nodes and 24 links. Node 5 is considered as the only origin in the network and the other 8 nodes are destinations. Each link has own capacity and travel time, and those of both directions are the same. This may represent a central business district (CBD) with suburbs surrounding it.

The maximum number of travelers from Node 5 to others is assumed fixed. As such, travel demands can change for every time step but cannot exceed that value. If a provided route set is attractive enough, the system can attract more demand. Therefore, the objective function of the route planning problem for this experiment is the maximization of total induced demand. Link and node properties and route information are indicated in **Figure 5** and **Table 2**.

All available routes that start from Node 5 are enumerated, and each route never passes a node already visited. 60 routes are constructed as arms which the algorithm can choose. The budget for routes, the maximum number of routes, is assumed and fixed to 10.

We make the following assumptions to minimize interventions from other system elements to concentrate more on the route planning performance of the methodology. Firstly, potential users perceive routes independent from one another despite the correlation among them that exists due to overlapping sections. They are also not allowed to transfer between routes, i.e. transfer costs are high. Moreover, travel time between ODs is the only feature of concern while other travel disutilities like expected wait time, access/egress cost, and operational delays are negligible.

One of the largest differences from Numerical Experiment 1 is the objective function of the problem, Eq. (11), maximizing total ridership instead of minimizing the passenger cost. Eq. (13) – (18) correspond to Eq. (4) – (8). Except for Eq. (15), the only difference between two constraint groups is the existence of time step indicators. In addition, two constraints, Eqs. (12) and (13), are related to the demand level between nodes r and s at time step t, d_{rs}^t. It changes for every time step as the provided route set differs, but there is an upper limit, the maximum demand, $\bar{d}_{rs}$. These come from the assumption that the system has no obligation to transport all demand. Namely, Eq.

(12) is newly introduced to set up maximum demand for the OD pairs. Actual demand conservation is defined in Eq. (13).

$$\max \sum_{(r,s)\in D} \sum_{k\in K_{rs}} x_{rs}^{kt} \tag{11}$$

s.t.

$$d_{rs}^{t} \le \bar{d}_{rs}, \ \ \forall t, \ \ (r,s) \in D \tag{12}$$

$$\sum_{k\in K_{rs}} x_{rs}^{kt} = d_{rs}^{t}, \ \ \forall t, \ \ (r,s) \in D \tag{13}$$

$$\sum_{k\in K} \sum_{r\in P_l^k, s\in Q_l^k} x_{rs}^{kt} \le u_l, \ \ \forall t, \ \ l \in A \tag{14}$$

$$c_{rs}^{k} = \sum_{l\in L_{rs}^{k}} c_l \,, \ \ \forall k, \ \ (r,s) \in D \tag{15}$$

$$x_{rs}^{kt} \le M w_k^t, \ \ \forall k,t, \ \ (r,s) \in D \tag{16}$$

$$x_{rs}^{kt} \ge 0, \ \ \forall k,t, \ \ (r,s) \in D \tag{17}$$

$$w_k^t \in \{0,1\}, \ \ \forall k,t \tag{18}$$

where,
x_{rs}^{kt}: passenger flow between origin r and destination s using route k at time t
D: OD pair set
K_{rs}: set of routes providing service between origin r and destination s,
where $K = \bigcup_{(r,s)} K_{rs}$ is the set of all routes
$\bar{d}_{rs}$: maximum demand between origin r to destination s
d_{rs}^{t}: demand between origin r to destination s at time t
P_l^k: set of nodes on route k located before link l
Q_l^k: set of nodes on route k located after link l
u_l: capacity of link l
A: set of all links
c_l: travel cost of link l
L_{rs}^{k}: set of links included in route k between origin r and destination s
w_k^t: 1 if route k is used, otherwise 0 at time t

Although the routes are not capacitated, routes cannot accommodate more passengers than capacities of links. It is assumed that potential travelers to different nodes do not impose any congestion effects on each other. They might accept different routes as independent ones. For example, if both Route 5-2-1 and 5-2-3 are provided, people who depart to Node 2 consider both

routes as 2 distinguishable alternatives. It raises the probability of an accepting system by increasing the total utility of the provided route set.

Ten trials are given to form a proposed route set after the initial learning. For each trial, the route with the maximum demand is chosen. **Table 3** describes the example results of a route proposal. As the algorithm is set to maximize the demand, routes are stretched as far as possible. For reality, route length constraints may be useful to avoid suggesting excessively long routes. Among 197 passengers, 93.3725 (standard deviation of 0.5073) are predicted to take the system according to the mean of 1,000 simulations.

We consider an oracle reference scenario with perfect information. We solve the problem with a genetic algorithm under default settings in MATLAB which ends up serving 96.8879 demand (standard deviation of 0.4889), the mean of 1,000 runs. This means our proposed algorithm provided the objective value within a 3.6% gap of the heuristic solution to the oracle policy.

Table 4 explains how much demand is served by 10 routes. The most well-served node is Node 2, being served 74.0% of demand from the origin, Node 5. It is followed by Node 8 (72.7%), Node 6 (60.1%), and Node 4 (57.3%). Because vehicles can reach all of the nodes adjacent to Node 5 within a relatively short time, potential passengers may feel it is convenient to use the system. On the other hand, people heading to Node 1, 3, 7, and 9 find that the transit system based on this route set is inconvenient to get their destinations, resulting in less than half the demand served. This imbalance may be one of the factors that make the transit system less competitive to other modes.

6 CONCLUSION

The costly acquisition of demand information prohibits transit operators from providing the best service to the public under uncertainty. We propose an alternative planning strategy involving sequential route design as well as a solution algorithm based on reinforcement learning to solve it.

The methodology is a combination of the route planning problem, MAB, and logit model. The suggested algorithm exploits the structure of transit route design and estimates parameters of the link functions to learn the distributions for route choice and acceptance. Results from numerical experiments show that the algorithm can achieve route sets within a reasonable range from the optimal value.

We conclude that the proposed algorithm can assist operators' decision-making under a sequential planning process. Among candidate routes, the algorithm can recommend the route set that satisfies the objective function maximizing the ridership or minimizing the average wait time based on learning processes.

The primary computational bottlenecks to the proposed method pertain to those that already exist in route generation. A larger candidate route set requires more computing resources not only for the set generation but also for the evaluation of alternatives in each time step.

Although the numerical experiments represent stylistic scenarios, they can be used to analyze last-mile services of regional transit users and transportation hubs. Nonetheless, we make assumptions like independent routes, transfer availability, and exclusion of expected waiting time or other travel disutilities. This study attempted to focus on acquiring the information of prevailing demand and thus left those other components out to minimize the noise. However, they should be considered in future research.

ACKNOWLEDGEMENT
This study is supported by C2SMART, a USDOT Tier 1 University Transportation Center, and NSF CMMI-1652735.

AUTHOR CONTRIBUTIONS

The authors confirm contribution to the paper as follows: study conception and design: G. Yoon, J.Y.J. Chow; data preparation: G. Yoon; analysis and interpretation of results: G. Yoon, J.Y.J. Chow; draft manuscript preparation: G. Yoon, J.Y.J. Chow. All authors reviewed the results and approved the final version of the manuscript.

REFERENCE

1. Desaulniers, G., and Hickman, M. D., 2007. Public transit. *Handbooks in Operations Research and Management Science*, *14*, 69-127.
2. Guihaire, V., and Hao, J. K., 2008. Transit network design and scheduling: A global review. *Transportation Research Part A 42*(10), 1251-1273.
3. Schöbel, A., 2012. Line planning in public transportation: models and methods. *OR Spectrum* 34(3), 491-510.
4. Canca, D., De-Los-Santos, A., Laporte, G., and Mesa, J. A., 2017. An adaptive neighborhood search metaheuristic for the integrated railway rapid transit network design and line planning problem. *Computers & Operations Research*, *78*, 1-14.
5. Lampkin, W., and Saalmans, P. D., 1967. The design of routes, service frequencies, and schedules for a municipal bus undertaking: A case study. *Journal of the Operational Research Society*, *18*(4), 375-397.
6. Silman, L. A., Barzily, Z., and Passy, U., 1974. Planning the route system for urban buses. *Computers & Operations Research*, *1*(2), 201-211.
7. Mauttone, A., and Urquhart, M.E., 2009. A route set construction algorithm for the transit network design problem. *Computers & Operations Research*, *36*(8), 2440-2449.
8. Ngamchai, S., and Lovell, D. J., 2003. Optimal time transfer in bus transit route network design using a genetic algorithm. *Journal of Transportation Engineering*, *129*(5), 510-521.
9. Borndörfer, R., Grötschel, M., and Pfetsch, M. E., 2007. A column-generation approach to line planning in public transport. *Transportation Science* 41(1), 123-132.
10. Szeto, W. Y., and Jiang, Y., 2014. Transit route and frequency design: Bi-level modeling and hybrid artificial bee colony algorithm approach. *Transportation Research Part B: Methodological*, *67*, 235-263.
11. Chow, J. Y. J., and Regan, A.C., 2011a. Real option pricing of network design investments. *Transportation Science*, *45*(1), 50-63.
12. Flyvbjerg, B., 2007. Cost overruns and demand shortfalls in urban rail and other infrastructure. *Transportation Planning and Technology*, *30*(1), 9-30.
13. Wisniewski, M. Chicago Tribune, 2018. "Column: Want $50? Planning agency will pay to hear about your commute." *Chicago Tribune.* http://www.chicagotribune.com/g00/news/columnists/wisniewski/ct-biz-cmap-commute-survey-50-getting-around-20180830-story.html, Accessed on Oct 14, 2018.
14. Chicago Metropolitan Agency for Planning, 2018. Community Data Snapshots Raw Data, June 2017 Release. https://datahub.cmap.illinois.gov/dataset/community-data-snapshots-raw-data, Accessed on Oct 14, 2018.
15. Guo, Q.W., Chow, J. Y. J., and Schonfeld, P., 2018. Stochastic dynamic switching in fixed and flexible transit services as market entry-exit real options. *Transportation Research Part C: Emerging Technologies*, *94*, 288-306.
16. Allahviranloo, M., and Chow, J. Y. J., 2019. A fractionally owned autonomous vehicle fleet sizing problem with time slot demand substitution effects. *Transportation Research Part C: Emerging Technologies*, *98*, 37-53.

17. Ryzhov, I. O., and Powell, W. B., 2011. Information collection on a graph. *Operations Research*, *59*(1), 188-201.
18. Powell, W. B., and Ryzhov, I. O., 2012. *Optimal learning* (Vol. 841). John Wiley & Sons.
19. Sutton, R. S., and Barto, A. G., 2018. *Reinforcement learning: An introduction*. MIT press.
20. Gittins, J., Glazebrook, K., and Weber, R., 2011. *Multi-armed bandit allocation indices*. John Wiley & Sons.
21. Ceder, A. and Wilson, N. H. M., 1986. Bus network design. *Transportation Research: Part B* 20(4), 331-344.
22. Chien, S., Yang, Z., and Hou, E., 2001. Genetic algorithm approach for transit route planning and design. *Journal of Transportation Engineering* 127(3), 200-207.
23. Yu, B., Yang, Z., Jin, P., Wu, S., and Yao, B., 2012. Transit route network design-maximizing direct and transfer demand density. *Transportation Research: Part C* 22, 58-75.
24. An, K., and Lo, H. K., 2015. Robust transit network design with stochastic demand considering development density. *Transportation Research Part B: Methodological*, *3*(81), 737-754.
25. An, K., and Lo, H. K., 2016. Two-phase stochastic program for transit network design under demand uncertainty. *Transportation Research Part B: Methodological*, *84*, 157-181.
26. Pinelli, F., Nair, R., Calabrese, F., Berlingerio, M., Di Lorenzo, G. and Sbodio, M. L., 2016. Data-driven transit network design from mobile phone trajectories. *IEEE Transactions on Intelligent Transportation Systems*, 17(6), 1724-1733.
27. Zhong, S., Zhou, L., Ma, S., Jia, N., Zhang, L., and Yao, B., 2018. The optimization of bus rapid transit route based on an improved particle swarm optimization. *Transportation Letters* 10(5), 257-268.
28. Liang, J., Wu, J., Gao, Z., Sun, H., Yang, X., and Lo, H. K., 2019. Bus transit network design with uncertainties on the basis of a metro network: A two-step model framework. *Transportation Research Part B 126*, 115-138.
29. Chow, J. Y. J., and Regan, A. C., 2011b. Network-based real option models. *Transportation Research Part B: Methodological*, *45*(4), 682-695.
30. Mohammed, A., Shalaby, A., and Miller, E. J., 2006. Empirical analysis of transit network evolution: case study of Mississauga, Ontario, Canada, bus network. *Transportation research record*, *1971*(1), 51-58.
31. Li, Z. C., Guo, Q. W., Lam, W. H., and Wong, S. C., 2015. Transit technology investment and selection under urban population volatility: A real option perspective. *Transportation Research Part B: Methodological*, *78*, 318-340.
32. Sun, Y., Guo, Q., Schonfeld, P., and Li, Z., 2017. Evolution of public transit modes in a commuter corridor. *Transportation Research Part C: Emerging Technologies*, *75*, 84-102.
33. Yu, W., Chen, J., and Yan, X., 2019. Space–Time Evolution Analysis of the Nanjing Metro Network Based on a Complex Network. *Sustainability*, *11*(2), 523.
34. Chow, J. Y. J., and Sayarshad, H. R., 2016. Reference policies for non-myopic sequential network design and timing problems. *Networks and Spatial Economics*, *16*(4), 1183-1209.
35. Powell, W. B., 2007. *Approximate Dynamic Programming: Solving the curses of dimensionality* (Vol. 703). John Wiley & Sons.
36. Ayyadevara, V.K., 2018. *Pro Machine Learning Algorithms: A Hands-On Approach to Implementing Algorithms in Python and R*. Apress.
37. Liu, K. and Zhao, Q., 2010. Distributed learning in cognitive radio networks: Multi-armed bandit with distributed multiple players. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on* (pp. 3010-3013). IEEE.
38. Villar, S.S., Bowden, J. and Wason, J., 2015. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics* 30(2), 199-215.
39. Li, L., Lu, Y., and Zhou, D., 2017. Provably optimal algorithms for generalized linear contextual bandits. In *Proc. 34th International Conference on Machine Learning-Volume 70,* 2071-2080.

40. Zhou, J., Lai, X., and Chow, J. Y. J., 2019. Multi-armed bandit on-time arrival algorithms for sequential reliable route selection under uncertainty. *Transportation Research Record*, doi:10.1177/0361198119850457.

41. Bubeck, S. and Cesa-Bianchi, N., 2012. *Foundations and Trends in Machine Learning*, 5(1), 1-122.

42. Vermorel, J. and Mohri, M., 2005. Multi-armed bandit algorithms and empirical evaluation. Proc. *European conference on machine learning*, 437-448.

43. Römer, C., Hiry, J., Kittl, C., Liebig, T., and Rehtanz, C., 2019. Charging control of electric vehicles using contextual bandits considering the electricaldistribution grid. *arXiv preprint*, arXiv:1905.01163.

44. Huang, Y., Zhao, L., Powell, W.B., Tong, Y. and Ryzhov, I. O., 2019. Optimal Learning for Urban Delivery Fleet Allocation. *Transportation Science*, 53(3), 623-641.

45. Zolfpour-Arokhlo, M., Selamat, A., Hashim, S. Z. M. and Afkhami, H., 2014. Modeling of route planning system based on Q value-based dynamic programming with multi-agent reinforcement learning algorithms. *Engineering Applications of Artificial Intelligence*, 29, 163-177.

46. Khadilkar, H., 2018. A scalable reinforcement learning algorithm for scheduling railway lines. *IEEE Transactions on Intelligent Transportation Systems*, *20*(2), 727-736.

47. Tsai, C. W., Hsia, C. H., Yang, S. J., Liu, S. J., and Fang, Z. Y., 2020. Optimizing hyperparameters of deep learning in predicting bus passengers based on simulated annealing. *Applied Soft Computing*, 106068.

48. Cats, O., and West, J., 2020. Learning and Adaptation in Dynamic Transit Assignment Models for Congested Networks. *Transportation Research Record*, 0361198119900138.

49. Ben-Akiva, M. and Bierlaire, M., 1999. Discrete choice methods and their applications to short term travel decisions. *Handbook of transportation science*, 23, 5-33.

50. Cascetta, E., Nuzzolo, A., Russo, F. and Vitetta, A., 1996. A modified logit route choice model overcoming path overlapping problems. Specification and some calibration results for interurban networks. In *Proc. 13th ISTTT, Lyon, France, 24-26 July.*

51. Prashker, J. and Bekhor, S., 1999. Stochastic user-equilibrium formulations for extended-logit assignment models. *Transportation Research Record* 1676, 145-152.

Table of Figures

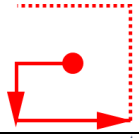
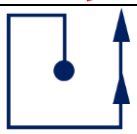
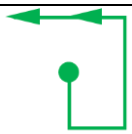
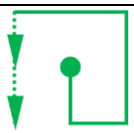
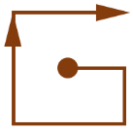
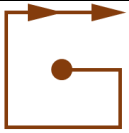
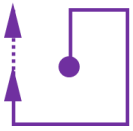
Table 1. Network properties of Case Study 1

Route index	Capacity	Travel time	Attractive-ness	Route index	Capacity	Travel time	Attractive-ness
1	7	12.78402	2	26	5	6.563081	0
2	10	19.64027	0	27	10	15.49458	2
3	9	12.08471	0	28	8	6.294078	0
4	9	13.17178	1	29	5	9.127546	0
5	9	8.981365	1	30	8	6.436196	1
6	9	12.96086	1	31	5	13.08447	2
7	5	13.47545	0	32	10	17.04014	1
8	5	16.62467	0	33	10	16.72975	2
9	7	15.43484	1	34	9	13.88374	0
10	9	19.20215	0	35	7	13.10808	2
11	10	19.11368	2	36	9	10.94936	1
12	7	17.07971	0	37	9	10.57389	1
13	7	7.931309	0	38	6	8.209589	2
14	7	14.83193	1	39	5	19.67188	1
15	8	16.34944	0	40	9	18.93461	1
16	8	8.71545	1	41	7	9.513882	0
17	9	7.466427	1	42	8	5.621179	1
18	5	19.84065	1	43	10	6.426785	0
19	8	15.00345	1	44	5	10.57066	0
20	8	17.31445	2	45	7	13.89486	2
21	5	14.54217	0	46	8	12.00166	2
22	6	9.616747	2	47	9	7.50225	0
23	7	14.31186	2	48	5	11.15243	2
24	10	15.48292	1	49	9	14.98024	1
25	8	16.52937	0	50	10	5.216627	2

Table 2. Link information of network in Case study 2

Index	*i*	*j*	Capacity	Travel cost (min)	Index	*i*	*j*	Capacity	Travel cost (min)
1	1	2	23	6.496	13	2	1	19	8.696
2	2	3	15	9.781	14	3	2	23	10.474
3	4	5	22	10.441	15	5	4	40	5.930
4	5	6	40	11.951	16	6	5	22	7.320
5	7	8	30	8.668	17	8	7	18	6.284
6	8	9	28	8.705	18	9	8	20	7.799
7	1	4	30	6.509	19	4	1	25	11.843
8	4	7	25	5.324	20	7	4	26	5.085
9	2	5	28	9.582	21	5	2	40	8.514
10	5	8	40	7.530	22	8	5	15	5.369
11	3	6	30	8.035	23	6	3	20	10.450
12	6	9	17	5.442	24	9	6	17	6.212

Table 3 Proposed route set diffence between algorithms

Best solution from Genetic algorithm (MATLAB)		Best example from proposed algorithm	
	[Red route: 5-4-7] ▪ 5-4-7-8-9-6-3-2-1 ▪ 5-4-7-8-9-6-3-2 ▪ 5-4-7-8-9-6-3		[Red route: 5-4-7] ▪ 5-4-7-8-9-6-3-2-1 (shortened) ▪ 5-4-7-8-9-6-3-2 (shortened) ▪ 5-4-7-8-9-6-3 (excluded)
	[Dark blue route: 5-2-1] ▪ 5-2-1-4-7-8-9-6-3 ▪ 5-2-1-4-7-8-9		[Dark blue route: 5-2-1] ▪ 5-2-1-4-7-8-9-6-3 ▪ 5-2-1-4-7-8-9-6 (extended)
	[Green route: 5-8-9] ▪ 5-8-9-6-3-2-1 ▪ 5-8-9-6-3-2		[Green route: 5-8-9] ▪ 5-8-9-6-3-2-1-4-7 (extended) ▪ 5-8-9-6-3-2-1-4 (extended)
	[Brown route: 5-6-9] ▪ 5-6-9-8-7-4-1-2-3 ▪ 5-6-9-8-7-4-1		[Brown route: 5-6-9] ▪ 5-6-9-8-7-4-1-2-3 ▪ 5-6-9-8-7-4-1-2 (extended)
	[Purple route:5-2-3] ▪ 5-2-3-6-9-8-7-4		[Purple route:5-2-3] ▪ 5-2-3-6-9-8-7-4-1 (new) ▪ 5-2-3-6-9-8-7-4
Served demand	97.2989 / 197 = 49.4%		95.1816 / 197 = 48.3%

Note: Circles are starting points of routes and arrowheads are ends.

Table 4 Served demand by routes

j	Served demand $/d_{ij}$	Served demand by route									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	2.5982 / 20 = 13.0%	0.0000	0.0001	1.2988	0.0000	0.0002	1.2988	0.0001	0.0002	0.0000	0.0002
2	19.9840 / 27 = 74.0%	0.0000	0.0003	6.6611	0.0000	0.0000	6.6611	0.0003	0.0000	6.6611	0.0000
3	1.1930 / 21 = 5.7%	0.0005	0.0177	0.0000	0.0005	0.0000	0.0000	0.0177	0.0000	1.1565	0.0000
4	17.2026 / 30 = 57.3%	8.5464	0.0000	0.0526	8.5464	0.0013	0.0526	0.0000	0.0013	0.0000	0.0019
6	14.4280 / 24 = 60.1%	0.0070	0.2428	0.0000	0.0070	4.8922	0.0000	0.2428	4.8922	0.0800	4.0639
7	12.2200 / 26 = 47.0%	6.0448	0.0000	0.0372	6.0448	0.0186	0.0372	0.0000	0.0186	0.0003	0.0185
8	17.4469 / 24 = 72.7%	0.2424	8.3988	0.0015	0.2424	0.0536	0.0015	8.3988	0.0536	0.0009	0.0535
9	8.0504 / 25 = 32.3%	0.0555	1.9233	0.0003	0.0555	1.3806	0.0003	1.9233	1.3806	0.0226	1.3084
Σ	93.1231 / 197 = 47.3%	14.8966	10.5830	8.0516	14.8966	6.3465	8.0516	10.5830	6.3465	7.9214	5.4464