# Dynamic Integration of Heterogeneous Transportation Modes under Disruptive Events

Yukun Yuan, Desheng Zhang†, Fei Miao§, John A. Stankovic⋆, Tian He‡, George Pappas* and Shan Lin

Stony Brook University, †Rutgers University, §University of Connecticut,
*University of Pennsylvania, ⋆University of Virginia, ‡University of Minnesota

***Abstract*—An integrated urban transportation system usually consists of multiple transport modes that have complementary characteristics of capacities, speeds, and costs, facilitating smooth passenger transfers according to planned schedules. However, such an integration is not designed to operate under disruptive events, *e.g.*, a signal failure at a subway station or a breakdown of a bus, which have rippling effects on passenger demand and significantly increase delays. To address these disruptive events, current solutions mainly rely on a substitute service to transport passengers from and to affected areas using ad-hoc schedules and static routes, *e.g.*, sending shuttles to closed subway stations. These solutions are highly inefficient and do not utilize real-time data to estimate dynamic passenger demand. To fully utilize heterogeneous transportation systems under disruptive events, we design a service called eRoute based on a hierarchical receding horizon control framework to automatically reroute, reschedule, and reallocate multi-mode transportation systems based on real-time and predicted demand and supply. Focusing on an integration of subway and bus, we implement and evaluate *eRoute* with large datasets including (i) a bus system with 13,000 buses, (ii) a subway system with 127 subway stations, (iii) an automatic fare collection system with a total of 16,840 readers and 8 million card users from a metropolitan city. The data-driven evaluation results show that our solution improves the ratio of served passengers (RSP) by up to 11.5 times and reduces the average traveling time by up to 82.1% compared with existing solutions.**

## I. Introduction

With rapid population growth and urbanization, people in the metropolitan area increasingly rely on public transportation in their daily lives [14]. In the New York City, the public transportation system serves 52.6% of all residents in the city [14]; whereas in the Chinese city Shenzhen, it serves more than 75% of all urban residents. An urban transportation system consists of multiple transport modes, such as subway and bus. These modes have complementary characteristics to meet the different needs of passengers. For example, the subway serves as the backbone of the urban transportation network, providing high-speed high-capacity transport services across major areas; while the bus is slower and cheaper, spreading all over the entire city areas with a large number of different lines and stations.

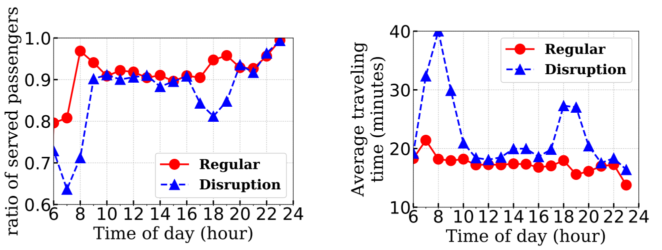
In an integrated transportation system, heterogeneous transport modes are designed to connect with each other by closely-located stations and synchronized schedules to facilitate passenger transfer activities under normal operations. However, under various disruptive events, *e.g.*, a power failure or a signal error, that can cause stations or vehicles to shut down for an unpredictable period, the transportation system becomes disconnected and inefficient, resulting in a surge of stranded passenger and cascading delays in affected areas.

There are very limited solutions to serve stranded passengers in current transportation systems. Existing practices typically provide substitute services using backup vehicles, *e.g.*, dispatching empty shuttles to the closed subway stations [1] [15]. A few recent works have proposed solutions for subway system disruptions, including robust train schedules [8], timetable adjustment [25], taxi recovery services [29], and simple integration between bus and subway systems [12]. However, these works employ localized solutions with static routes or fixed schedules, without dynamic coordination of multiple transport modes.

To achieve sufficient resilience under disruptive events, it is critical to optimally control and coordinate all transport subsystems according to real-time and predicted demand with a global view. In this paper, we design a hierarchical receding horizon control based dynamic integration framework called eRoute for the urban transportation system, which solves the optimization problem periodically and repeatedly during the disruptive event. Focusing on the integration of subway and bus, our control design has the following goal: maximizing satisfied passenger demand under disruptive events while minimizing the cost regarding extra traveling time due to detour and the number of extra vehicles.

The contributions of this work are listed as follows.

- To our knowledge, we conduct the first study on how to address disruptive events in public transportation systems based on real-time multi-source data. Our datasets advance the state-of-the-art in two aspects: (i) one of the most comprehensive datasets, including bus GPS, subway schedules, smart-card payments, from the same city Shenzhen, and (ii) the largest passenger coverage (i.e., 75% of 11 million permanent residents in Shenzhen). Our infrastructures and data are at least one or two orders of magnitude larger than existing academic systems (*e.g.*, [8] [25] [29] [12]).
- Based on these real-world multi-source data, we design a service called eRoute to dynamically reroute, reschedule and reallocate integrated heterogeneous transportation systems under disruptive events to maximally satisfy passenger demand and minimize the corresponding costs. In particular, we formulate a dynamic integration problem




**(a) ratio of served passengers over time** **(b) Average traveling time**

**Fig. 1:** Transportation system performances under disruptions

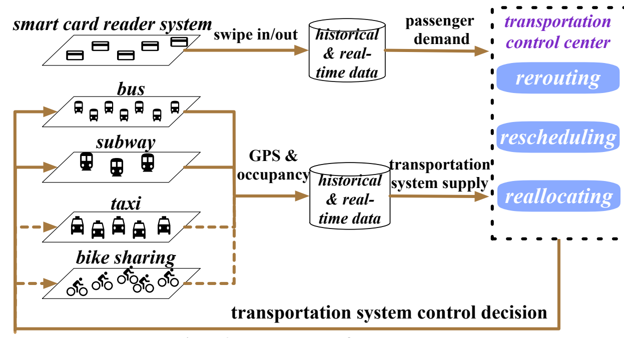



**Fig. 2: eRoute framework**

for the changing transportation network topology under disruptions as an integral multi-commodity max flow problem under dynamic edge capacity. We demonstrate that it is feasible to calculate a solution for this problem within desired control period after applying practical constraints derived from real scenarios, besides proving its NP-hardness.

- To solve this problem, we design a two-level Receding Horizon Control framework to adapt our solutions according to both current and possible future demand. At the higher level, we maximize satisfied passenger demand by obtaining rerouting and reallocation decisions for the overloaded transportation systems, meanwhile, at the lower level, we minimize the cost associated with these rerouting and reallocation decisions while meeting the maximum passenger demand. The interactions between these two levels ensure our framework satisfies the maximum passenger demand with the minimum cost.
- We implement and evaluate eRoute with our datasets that consist of (i) a bus system with 13,000 buses, (ii) a subway system with 127 subway stations. (iii) An automatic fare collection system with a total of 14,270 onboard mobile readers capturing 168,000 bus passengers per hour, and a total of 2,570 static readers capturing 60,000 subway passengers per hour. Compared to various existing approaches with real-time data-driven features, eRoute improves the ratio of served passengers (RSP) by up to 11.5 times and reduces the average traveling time by up to 82.1%.

## II. Motivation

### A. Service Disruption

Service disruptions of public transportation systems have significant impacts on public passengers. They not only introduce travel delays, but also reshape mobility patterns, generating high operation cost due to longer travel distance, local congestion, and the resulting opportunity losses [1].

In this paper, we focus on the service disruptions of public transportation system, *i.e.*, subway and bus system. To fully understand various types of disruptions that occur in urban cities, we provide a taxonomy of disruptions in subway and bus as shown in Table. I. We classify disruptive events into two major categories: small and large. Small disruptive events usually cause delays and cancellations of specific trains, whereas large disruptive events lead to one or multiple lines or stations to be shut down. We aim to address such disruptive events in this work.

Based on this taxonomy, we investigate an incident that occurred in 2013 at Shenzhen, a metropolitan in China with our datasets. A signal problem caused long delays for all trains of a major subway line during the day. Both speeds and frequency of all trains along this line were reduced significantly due to safety concerns. No bus shuttles had been used to address this incident in reality.

We use the smart card reader dataset from the city to analyze the impacts of this disruption on passengers. Here the ratio of served passengers (RSP) represents the ratio between the amount of actually served passengers and the amount of passenger demand, which is obtained by historical data assuming a stable daily passenger demand. In Figure 1a, two curves show the RSP over time on a regular day and on the day with the disruption. We observe that the RSP during peak-hours (6am-9am and 5pm-8pm) under disruption is 35% less than that of the regular day. Meanwhile, as shown in Figure 1b, the average traveling time of passengers under disruption is 31% higher than that of the regular day. The average traveling time increased from 18 minutes to 40 minutes in the morning rush hours. These plots indicate that existing solutions do not effectively handle such disruptions.

### B. eRoute Framework

To address one or multiple simultaneous disruptive events in urban cities, we design a service framework called eRoute, which is one system that can be used by transportation authorities during disruptive events. Figure 2 shows an overview of the eRoute architecture. Our main goal is to balance the supply and demand by dynamically integrating multiple transportation subsystems, *e.g.*, subway, bus, taxi, bike sharing subsystems, etc. In this paper, we focus on two specific transportation systems, *i.e.*, subway and bus, which are usually directly controlled by city government. Given these two systems, disruptive events usually result in dramatically reduced supply at certain locations, and our solution is to automatically compute solutions to increase supply at these locations.

Typically, the supply of bus and subway systems is organized in lines: (i) each bus or subway line has a fixed number of allocated vehicles; (ii) each line has a route where a few bus stops or subway stations are organized in a specific order; (iii) each line has a schedule with which the vehicles leave and arrive at certain bus stops or subway stations. With these three

| | Subway System | | | Bus System | | |
|---|---|---|---|---|---|---|
| Class | Cause | Impact | Solution | Cause | Impact | Solution |
| Small | signal problem | reduced train speed, delay of trains | passengers traffic control, scheduling larger capacity trains | Construction, mechanical failure, accident | service at substitute stations, passengers stuck on the road | substitute service |
| Large | weather disaster, power failure, mechanical train failure | service canceled at affected subway stations | emergency status broadcast, scheduling shuttles | weather disaster (flooding, icy road condition, etc), traffic congestion | service shutdown at a large region, significant delays | Ad-hoc routing |

**TABLE I: Taxonomy of disruptions to subway and bus systems**

features, the key idea of eRoute is to conduct the following three functions to deal with disruptive events:

- **Quantitative Reallocating**. Many disruptive events lead to failures of vehicles and result in a decrease of supply. Given the limited amount of vehicles available for providing transport services, reallocation among lines is critical to increasing supply with small overhead.
- **Spatial Rerouting**. This is an active traffic control strategy that presents alternate routes for buses, trains, and taxis. Rerouting is normally used when the regular route is severely affected by congestion and incidents, here the purpose of rerouting is to re-balance supply with practical constraints across different regions under disruptive events. The alternate route information is disseminated to drivers using control channels in real time.
- **Temporal Rescheduling**. Disruptive events directly affect the schedules of some transportation lines. eRoute reschedules the supply in other lines nearby and other transportation modes, increasing the current supply to the region in order to reinforce the service.

Different from existing works on transportation planning, eRoute is driven by real-time multi-source data, which has rich spatiotemporal information about passenger mobility patterns, and the demand and supply in transportation systems. Existing infrastructure in urban transportation systems has already offered various data to the transportation center over the network in real time. The smart card reader system records the swiping in/out events of every smartcard and then uploads them to the datasets in the database at the transportation center. The format of these datasets is shown in Table II. From these datasets, we extract historical, current, and future passenger demand along every origin-destination (OD) pair. The supply of every subsystem is obtained from GPS and occupancy datasets collected from every vehicle. For example, the GPS device in every bus reports the longitude, latitude, speed, number of the bus and number of the bus line.

Under disruptions, the transportation network topology, and passenger demand change dynamically, so eRoute employs a receding horizon control (RHC) framework to adapt control decisions based on both current and future demand. Our framework allows transportation control center to specify multi-objective optimization goals under the transportation system requirements and constraints. eRoute solves an optimization problem repeatedly at each iteration step of the RHC framework and then updates routing commands, schedules, and vehicle allocation periodically. eRoute can potentially deal with both small and large classes of disruptions. We note that the applicability of our solution is still limited by the scale of disruptive event. eRoute can be applied to deal with disruptive events that affect a single or multiple geographic regions, *e.g.*, a few stations or a city district. eRoute allocates under-utilized transportation system supply nearby to serve affected passengers. eRoute also requires extra buses and trains to serve these affected passengers. Therefore, it is important for cities to anticipate the amount of passengers affected and have sufficient extra buses and trains in stock.

## III. PROBLEM FORMULATION

Both subway and bus networks are main components of the public transportation system, and they are complementary to each other. Either of them has a high potential to offload passengers from the other. Our design explores this potential to interconnect buses and subways dynamically to serve passengers under disruptive events. Specifically, we design one receding horizon control based solution which solves a static optimization problem and sends the control decisions to transportation subsystems once in every iteration. In this section, we formulate the one-iteration optimization problem of handling disruptions by controlling subway-bus integrated network, including rerouting existing bus lines and reallocating extra buses and trains. Our goal is to provide alternative paths for influenced passengers that can meet a) dynamic passenger demand as much as we can with b) minimized cost including detour time due to rerouting and number of extra vehicles needed. We show that our formulation is an integral multi-commodity maximum flow problem under dynamic edge capacity, and it is NP-hard.

### *A. Model Subway-Bus Integrated Network*

We use $N^s$ and $N^b$ to represent the numbers of subway lines and bus lines. We define the subway-bus integrated network as $G = (V, E)$, where $V = V^s \cup V^b$. Every vertex in $V^s$ denotes one subway station and every vertex in $V^b$ represents one bus stop. Here we call either one subway station or bus stop as a

| Smartcard Reader data | |
|---|---|
| Collection Period | 2014/03/03∼2014/03/09<br>2014/02/19∼2014/02/20 |
| Data Size | 10.4 GB |
| Record number | 17,568,574 |
| Format | [Device ID, Smartcard ID, Time,<br>Transaction Type,<br>Metro Station or Bus Line]<br>Bus Boarding &<br>Subway Swiped-In<br>& Subway Swiped-Out; |

**TABLE II: Dataset of Smartcard Reader**

node in the subway-bus integrated transportation network. For any two vertices $v_i, v_j \in V$, if they are visited by the same subway or bus line consecutively, we add one directed edge, $e(v_i, v_j)$ from $v_i$ to $v_j$. There are two attributes of for every directed edge $e_k \in E$: capacity and cost denoted as $w(e_k)$ and $p(e_k)$. Disruptive events can happen at any locations on the road. We incorporate them into the subway-bus integrated network by adding nodes to represent them and edges to connect them with adjacent stops or stations.

Suppose there are $l$ nodes whose services are ceased due to disruptions and let $\mathbf{D} = \{v_{d_1}, v_{d_2}, .., v_{d_l}\}$ be the set of influenced nodes. For example, the services of four subway stations are ceased because of power failure. Then the four corresponding nodes of the ceased subway stations are regarded as the affected nodes. The passengers at these impacted nodes need to find alternative ways to travel to their destinations. We define the origin-destination (OD) pair as $(s_i, t_i)$, representing that there are passengers traveling from $v_{s_i} \in \mathbf{D}$ to $v_{t_i} \in V$. The passengers demand of one OD pair $(s_i, t_i)$ during one time slot, *e.g.*, 20 mins, is represented as $C_i$. The OD pair and corresponding demand indicate mobility patterns of passengers, which can be learned from historical data sets.

### B. Problem Statement

**Definition 1** (Dynamic integration problem (DIP))**.** *In a subway-bus integrated network, suppose there are $l$ nodes influenced due to disruptive events. Given the passenger demand for every OD pair, the number of extra trains $N_s$ and the number of extra buses $N_b$, the problem is to decide how to reroute existing $N^b$ bus lines, reallocate extra buses and trains, and reschedule them such that the supply can meet the passenger demand as much as possible under practical constraints for every OD pair.*

$X^r \in \{0,1\}^{N^b \times l}$ is the decision matrix for rerouting, where $X^r_{k,i} = 1$ if $k$th bus line is rerouted to $i$th influenced transportation node. $X^b \in \{0,1\}^{N_b \times N^b}$ is the decision matrix for reallocating extra $N_b$ buses, where $X^b_{h,k} = 1$ if $h$th extra bus is reallocated to $k$th bus line, otherwise, it is 0. $X^s \in \{0,1\}^{N_s \times N^s}$ is the decision matrix for reallocating extra $N_s$ trains, where $X^s_{h,k} = 1$ if $h$th extra train is reallocated to $k$th subway line.

For every OD pair, we define a passenger flow from a source node $s_i$ to a sink node $t_i$ in the edge-capacitated directed graph $G$: an $s_i - t_i$ flow is a function $f : E \to \mathbb{R}^+$ that assigns a real number to each edge. Intuitively, $f(e) \geq 0$ is the amount of flow carried on the edge $e$, which represents the number of passengers transported along the edge $e$. There are two constraints: (1) capacity constraint: $\forall e \in E, f(e) \leq w(e)$; (2) flow reservation on transit node: for each node $v$ except $s$ and $t$, we have

$$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ leaving } v} f(e) \tag{1}$$

Let $S_i$ be the number of passengers that the integrated network can carry for an OD pair $(s_i, t_i)$, subjected to the link capacity constraint and $S_i$ is called the supply for $(s_i, t_i)$. When the passenger demand of an OD pair $(s_i, t_i)$ is less or equal to $S_i$, the integrated network can fully transport all the passengers without delay. Under disruptive events, there could be multiple OD pairs that need to be addressed simultaneously and the supplies for these OD pairs are usually significantly insufficient, so our goal is to maximize the supplies $J = \sum_i S_i$.

Then we have the following realistic constraints for *DIP*:

- **Detour Constraint:** if $X^r_{k,i} = 1$, $f(k,i) \leq \alpha$, where $f(k,i)$ is the function of extra detour time due to rerouting $k$th bus line to $v_i$, and $\alpha$ is the upper bound threshold of such detour time. It ensures that the increase of traveling time for regular bus passengers is not too high. We will discuss how to calculate $f(k,i)$ later.
- **Allocation Constraint:** $X^b \mathbf{1}_{N^b} \preceq \mathbf{1}_{N_b}$ and $X^s \mathbf{1}_{N^s} \preceq \mathbf{1}_{N_s}$, since every extra bus or train should be reallocated to at most one line, where $\mathbf{1}_{N_b}$ is a column vector of all 1s and the length of this vector is $N_b$.
- **Schedule Constraint:** $(X^s)^T \mathbf{1}_{N_s} \preceq \beta$, where $\beta$ is a length $N^s$ column vector and $N^s$ is the number of subway lines considered in our problem. Let $\beta_k$ denote the number of extra trains which can be reallocated to $k$th subway line for $1 \leq k \leq N^s$. We define such constraint, since there exists limitation of the number of trains operated along the same route for safety.
- **Supply Constraint:** To keep high utilization of our limited resource, we constrain that $\forall (s_i, t_i)$, $S_i \leq C_i$, which specifies that the supply of one OD pair is less than or equal to its demand.

### C. Problem Analysis and Transformation

There are some similarities and differences between our problem and the classical max flow problem: we use the source-sink pair to describe every OD pair in $G$. Considering the subway-bus integrated network $G$, simultaneously moving passengers for every OD pair means finding feasible integral flows in $G$. Intuitively, the objective of *DIP* is maximizing the sum of the size of every source-sink pair's flow. However, compared to the classical integral multi-commodity max-flow problem, *DIP* has the following differences:

- *Dynamic graph topology:* in *DIP*, the network topology is dynamic because the rerouting decision $X^r$ affects the edges in the network. For example, when one bus line is rerouted to pass a subway station $d_i$ between two previously consecutive bus stops, two new edges are added to connect the bus line with the subway station, and the edge between these two previously consecutive bus stops is removed.
- *Dynamic edge capacity:* due to the reallocation of extra buses and trains, the capacity of some subway and bus lines would increase dynamically. Therefore, the capacity of corresponding edges in $G$ also dynamically changes based on the reallocation decision, $X^s$ and $X^b$.
- *Constrained capacity of flow:* due to the supply constraint stated previously: the supply of one OD pair is less

than or equal to the demand, the size of one flow of every source-sink pair should be no more than the corresponding demand.

To address dynamic graph topology and constrained capacity, we transform DIP to an Integral Multi-commodity Max-Flow problem (IMCMF) under dynamic edge capacity. The detail of this transform is described in the Appendix. Here we introduce the basic idea briefly.

We first remove the dynamic topology by select all the bus lines candidates near the influenced nodes that meet the detour cost constraint, then tentatively reroute all of them to the influenced nodes. So according to the detour cost constraint, we add the corresponding edges into the integrated subway-bus network and get a new graph $G'$. This allows us to compute passenger flows for the next step, if some of these edges are not used by any flow in the next step, they will be removed, which means that corresponding bus lines do not need to be rerouted. Then we modify $G'$ to satisfy the supply constraint: for every source-sink pair $(s_i, t_i), 1 \le i \le M$, we add one virtual source node and one directed edge from the virtual source node to the source node, and then we also add one virtual sink node and one directed edge from the original sink node to the virtual sink node. The edge capacities of both two added edges are set to be equal to the corresponding source-sink pair demand, $C_i$. The new graph is denoted as $G''$. By this modification, we make sure the supply constraint is satisfied in $G''$. One example is introduced in Figure 15 in the Appendix.

### D. IMCMF under dynamic edge capacity

Let $X \in \mathbb{N}_0^{M\times N}$ denote the decision variable of the IMCMF, where $M$ and $N$ are the numbers of source-sink pairs and edges in $G''$ separately. $X_{i,j}$ represents the size of $i$th source-sink pair's flow along $j$th edge in $G''$. Then the objective is:

$$\max \sum_{i=1}^{M}\sum_{j=1}^{N} X_{i,j}R_{i,j} \tag{2}$$

where $R \in \{0,1\}^{M\times N}$ is the relation matrix for calculating the size of flow of every source-sink pair. $R_{i,j} = 1$ if $j$th edge connects with the source node of $i$th pair. According to Step 2 in the previous section, only one edge is calculated for every OD pair, $\sum_{j=1}^{N} R_{i,j} = 1$ and $S_i = \sum_{j=1}^{N} X_{i,j}R_{i,j}$.

We define $R^{se} \in \{0,1\}^{N^s\times N}$ to represent the relation between subway lines and edges. $R^{se}_{j,k} = 1$, if $k$th edge is created due to $j$th subway line, otherwise, it is 0. $R^{be} \in \{0,1\}^{N^b\times N}$ denotes the relation between bus lines and edges. $R^{be}_{j,k} = 1$, if $k$th edge is created due to $j$th bus line, otherwise, it is 0.

Let $w^s(e_j)$ and $w^b(e_j)$ denote the capacity increase of edge $e_j$ due to reallocating extra trains and buses respectively. We have the following equations:

$$w^s(e_j) = \sum_{i=1}^{N_s}\sum_{k=1}^{N^s} I(X^s_{i,k}) \times C^s \times R^{se}_{k,j}, \quad 1 \le j \le N \tag{3}$$

$$w^b(e_j) = \sum_{i=1}^{N_b}\sum_{k=1}^{N^b} I(X^b_{i,k}) \times C^b \times R^{be}_{k,j}, \quad 1 \le j \le N \tag{4}$$

where the indicator function $I(X^s_{i,k}) = 1$ if and only if $X^s_{i,k} > 0$, otherwise, it is 0. $C^s$ and $C^b$ are the capacities of one train and bus respectively. Considering the edge capacity constraint, we have the constraint:

$$\sum_{i=1}^{M} X_{i,j} \le w(e_j) + w^s(e_j) + w^b(e_j), \qquad 1 \le j \le N \tag{5}$$

where $w(e_j)$ is the original capacity of edge $e_j$. We formulate the flow conservation on transit nodes: the amount of a flow entering an intermediate node is the same that exits the node. Therefore, for $i$th source-sink pair and $k$th node satisfies

$$\sum_{j=1}^{N} X_{i,j}R^{ne}_{k,j} = 0, \quad 1 \le i \le M, \quad 1 \le k \le L \tag{6}$$

where $R^{ne} \in \{-1,0,1\}^{L\times N}$ describes the node-edge relation. $L$ is the number of regular nodes in $G''$. $R^{ne}_{k,j} = 1$ if $j$th edge points to $k$th node and it is -1 if $j$th edge emits from $k$th node. Otherwise, it is 0. The IMCMF problem we consider is:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{M} S_i = \sum_{i=1}^{M}\sum_{j=1}^{N} X_{i,j}R_{i,j} \\
\text{s.t.} \quad & S_i \le C_i \\
& X^b \mathbf{1}_{N^b} \preceq \mathbf{1}_{N_b}, \quad X^s \mathbf{1}_{N^s} \preceq \mathbf{1}_{N_s} \\
& (X^s)^T \mathbf{1}_{N_s} \preceq \beta \\
& (3), \quad (4), \quad (5), \quad (6)
\end{aligned} \tag{7}$$

**Theorem 1.** *The DIP is NP-hard.*

*Proof.* Compared with maximum integral multi-commodity flow, our problem has dynamic edge capacity and the edge capacity is also one decision variable. In *DIP*, let $N_b = N_s = 0$, then maximum integral multi-commodity flow problem is one special case of *DIP*. [10] shows that maximum integral multi-commodity flow is NP-hard. Therefore, *DIP* is also NP-hard. □

Based on the literature [6] [10], integral maximum multi-commodity flow problem is Max SNP-hard even in several particular cases. This result implies that there exists no polynomial time approximation scheme unless P=NP. In fact, [6] proves that IMMF is not only strongly NP-hard but finding an approximate solution within a fixed performance ratio for it is still one NP-hard problem. [11] shows that it is NP-hard to approximate within $m^{\frac{1}{2}-\epsilon}$, where $m$ is the number of edges. Although there exist several works providing one approximation algorithm or linear time algorithm, they require that the graph is one tree [10]. Meanwhile, our problem is still different from the existing dynamic graph problem, where edge capacity changes with time, but it is not one decision variable [9].

Although the *DIP* is hard to solve in nature, we argue that the disruptions in urban transportation systems usually only affect small numbers of stations and the subway lines and bus lines around them. Therefore, the input size of DIP is not large. Based on our linear integer programming (LIP) formulation of the problem, existing solvers in Matlab can solve them relatively quickly. In our evaluation with large-scale datasets, the optimal solution of the problem can be obtained in $1 \sim 2$

minutes, which is fast enough for transportation system control in reality.

## IV. RHC Algorithm Design

The transportation control center receives real-time streaming data including smart card records, vehicles' GPS locations, and occupancy status with timestamps periodically. These real-time data streams are then processed to predict the spatiotemporal patterns of passenger demand. Based on the prediction, the control center can utilize a receding horizon control (RHC) algorithm to calculate a control solution periodically and repeatedly in real-time, to match predicted passenger demands.

To obtain optimal bus/train line rerouting and extra bus/train reallocation decisions, we consider two main objectives: 1) maximize the passenger transport for the overloaded transportation systems under disruptive events; 2) minimize the rerouting and reallocation cost while achieving the maximum passenger transport. When disruptive events happen, the transportation network topology and passenger demand change dynamically, and the system should address these challenges and adapt the solutions according to both current and possible future demand. Hence, we design a hierarchical RHC algorithm. The higher level problem is based on our problem formulation in Section III, which suggests the passenger flows and their paths (rerouting decisions), and the reallocation of buses and trains. The lower level problem is to minimize the rerouting cost and reallocation cost regarding extra detour time and the number of additional buses or trains reallocated.

### A. Variables, constraints and objective functions

We assume that the optimization time horizon is $T$, indexed by $t = 1, ..., T$. We first reformulate the variables in the optimization time horizon. Let $w(e_k, t)$ be the capacity of $e_k$ during time slot $t$. $C_i(t)$ represents the passengers demand of $i$th source-sink pair $(s_i, t_i)$ during time slot $t$, which can be predicted based on the historical dataset and real-time sensor information. We define $X_{i,j}(t) \in \mathbb{N}_0^{M\times N}$ as the decision variable of the integral multi-commodity max-flow during time slot $t$. Meanwhile, $X^b(t) \in \{0,1\}^{N_b \times N^b}$ and $X^s(t) \in \{0,1\}^{N_s \times N^s}$ represent the reallocation decision of extra buses and trains during time slot $t$.

**Modeling Circulating Supply:** The routing and allocation in subway-bus network have to meet spatiotemporal constraints, due to operating schedules and road conditions. For example, a bus may become available for reallocation after transporting all passengers at its final stop in a finite optimization horizon. We define $W^b \in \mathbb{N}_+^{N^b}$, one column vector to denote the number of time slots needed to complete one end-to-end trip of all bus lines. For instance, $W_i^b$ is the number of time slot needed to finish one trip of $i$th bus line. Let $U^b(t) \in \mathbb{N}_+^{N_b}$ be one column vector to represent the number of time slot needed to finish current bus line trip at the beginning of time slot $t$. We note that $U^b(t)$ may change over time due to congestion and road conditions. More importantly, it is directly affected by the rerouting decision. For instance, it costs 4 time slots to finish one trip of the first bus line, and the first extra bus reallocated to the first bus line at time slot 1. So at the time slot 2, we have the following values: $U_1^b(2) = 3$. Then the relation between $U_i^b(t)$ and $U_i^b(t-1)$ is:

$$U_i^b(t) = \max\{0, \max\{\sum_{j=1}^{N^b} X_{i,j}^b(t-1) W_i^b, U_i^b(t-1)\} - 1\} \quad (8)$$

where $t \geq 2$ and $U_i^b(1) = 0$ for $1 \leq i \leq N_b$. Based on $U^b(t)$, $\gamma^b(t) \in \{0,1\}^{N_b}$ is one vector column to describe whether every extra bus can be reallocated during time slot $t$. It is clear that if $U_i^b(t) > 0$, $i$th extra bus is still operating for one existing bus line and it cannot be reallocated, otherwise, it can be reallocated. We have the following equation:

$$\gamma_i^b(t) = I_1(U_i^b(t)) \quad (9)$$

where $I_1(U_i^b(t))$ is an indicator function, and it is equal to 1 if $U_i^b(t) = 0$, otherwise, it is 0. Then for $i$th bus, during time slot $t$, it cannot be reallocated to more than $\gamma_i^b(t)$ bus lines:

$$X^b(t)\mathbf{1}_{N^b} \preceq \gamma^b(t) \quad (10)$$

We remark that it's possible that $\sum_{j=1}^{N^b} X_{i,j}^b(t) = 0$, however, $i$th bus also contributes to one existing bus line, because of operating for one existing bus line. Hence, we define $O^b(t) \in \{0,1\}^{N_b \times N^b}$ to denote which bus line that every extra bus contributes to during time slot $t$. $O_{i,j}^b(t) = 1$ if $i$th bus is operated for $j$th bus line during time slot $t$, otherwise, it is 0. Then, we have the following relation:

$$O_{i\cdot}^b(t) = O_{i\cdot}^b(t-1) I_2(U_i^b(t)) + X_{i\cdot}^b(t) \quad (11)$$

where $O_{i\cdot}^b$ is the $i$th row of $O^b(t)$ and $I_2(U_i^b(t))$ is also one indicator function. $I_2(U_i^b(t)) = 1$ if $U_i^b(t) > 0$, otherwise, it is 0. Finally, we describe the capacity increase of $e_j$ during time slot $t$ due to $N_b$ extra buses:

$$w^b(e_j, t) = \sum_{i=1}^{N_b} \sum_{k=1}^{N^b} I(O_{i,k}^b(t)) \times C^b \times R_{k,j}^{be} \quad (12)$$

where $C^b$ is the capacity that one extra can provide. The circulating supply model and constraint of subway trains are similar to that of buses. We also define the notations: $U^s(t)$, $W^s$, $\gamma^s(t)$, $O^s(t)$ and $w^s(e_j, t)$. Due to the space limitation, we skip the details of definitions. The circulating supply model and constraints of trains is described in the Appendix B. The equations for extra trains' reallocation is:

$$U_i^s(t) = \max\{0, \max\{\sum_{j=1}^{N^s} X_{i,j}^s(t-1) W_i^s, U_i^s(t-1)\} - 1\}$$

$$\gamma_i^s(t) = I_1(U_i^s(t)) \quad (13)$$

$$X^s(t)\mathbf{1}_{N^s} \preceq \gamma^s(t) \quad (14)$$

$$O_{i\cdot}^s(t) = O_{i\cdot}^s(t-1) I_2(U_i^s(t)) + X_{i\cdot}^s(t) \quad (15)$$

$$w^s(e_j, t) = \sum_{i=1}^{N_s} \sum_{k=1}^{N^s} I(O_{i,k}^s(t)) \times C^s \times R_{k,j}^{se} \quad (16)$$

The edge capacity constraint of flow during every time slot $1 \leq t \leq T$ is defined as:

$$\sum_{i}^{M} X_{i,j}(t) \leq w(e_j, t) + w^s(e_j, t) + w^b(e_j, t), 1 \leq j \leq N. \quad (17)$$

Considering the flow conservation on transit nodes, we define:

**Algorithm 1:** ***RHC algorithm for real-time transportation system control***

**Input:** Time horizon $T$ minutes, period of updating control solution $t_1$ minutes; number of time slots to finish one bus or subway line trip $W^b, W^s$; train and bus capacity $C^s, C^b$; geometrical information of transportation nodes; parameter $\theta$

**Output:** Control decision: $X^r$, $X^s$, $X^b$

```
while At the beginning of every update period t_1 minutes do
    Update the number of available buses, N_b and trains, N_s; update the passengers demand of every OD pair based on the historical dataset and real-time sensor information; update prediction of C_i(t) for the time horizon T; update the edge capacity during the time horizon T, w(e_j, t); update the parameter β
    Solve the max-flow problem (19) and get the optimal solution set {X̂(t), X̂^s(t), X̂^b(t)} of problem (19)
    Solve the min-cost problem (22) to get the control decision, which achieves the minimum cost.
    Send the control decision of according to solution: X, X^s, X^b.
end while
return Control decision
```

$$\sum_{j=1}^{N} X_{i,j}(t) R^{ne}_{k,j} = 0,\ 1 \le i \le M, 1 \le k \le L. \tag{18}$$

## B. A Hierarchical RHC Algorithm

Our goal of the higher level RHC problem formulation is to seek the dynamic rerouting and reallocation decision based on predicted passenger demand. This formulation is based on the problem transformation in the previous section:

$$\begin{aligned} \max_{X(t), X^b(t), X^s(t)} \quad & \sum_{t=1}^{T}\sum_{i=1}^{M}\sum_{j=1}^{N} X_{i,j}(t) R_{i,j} \\ \text{s.t.} \quad & \sum_{i=1}^{M}\sum_{j=1}^{N} X_{i,j}(t) R_{i,j} \le C_i(t) \\ & (X^s)^T \mathbf{1}_{N_s} \preceq \beta, \quad (8) \sim (18) \end{aligned} \tag{19}$$

After solving the above problem, we obtain the maximum demand that the system can support, which is equal to the amount of the supply that the system needs to provide.

In General, the problem (19) has multiple optimal solutions, i.e., different flow and reallocation assignments can achieve the same supply in the integrated network. Assume the optimal solutions of (19) is a set $\{\hat{X}(t), \hat{X}^s(t), \hat{X}^b(t)\}$. They provide the maximum value of supply. Therefore, we introduce the lower level which is formulated to choose the optimal flow and reallocation assignments with the minimum cost. In this level, the goal is rerouting existing bus lines and reallocating the extra vehicle supplies along different lines with the minimal cost. We minimize the cost to satisfy the supply achieved in (19), which consists of the rerouting cost and the number of extra buses or trains.

The rerouting cost is defined as:

$$J_r = \sum_{j=1}^{N} I(\sum_{i=1}^{M} X_{i,j}(t)) p(e_j) \tag{20}$$

where $p(e_j)$ is the cost of traveling along edge $e_j$. The number of extra buses and trains used is modeled as:

$$J_n = \sum_{k=1}^{N_b}\sum_{j=1}^{N^b} X^b_{i,j}(t) + \sum_{k=1}^{N_s}\sum_{j=1}^{N^s} X^s_{i,j}(t) \tag{21}$$

We define a weight parameter $\theta$ when summing up the costs related to both objectives. The formulation of minimizing cost is shown as follows:

$$\min_{X(t), X^b(t), X^s(t) \in \{\hat{X}(t), \hat{X}^b(t), \hat{X}^s(t)\}} \quad J = \sum_{t=1}^{T} (J_r + \theta J_n) \tag{22}$$

## C. RHC Framework Implementation

We adopt basic linear regression technique to predict passenger demand of different OD pairs based on historical and real-time datasets. We define the time horizon is $T$ minutes and the length of every time slot is $t_1$ minutes. The previous proposed RHC based problem formulation is embedded in one iteration of our RHC algorithm, and we update the control decision every time slot, $t_1$ minutes. The pseudo-code of RHC algorithm is shown as Algorithm. 1. For simplicity, we assume that the two-level RHCs have the same timescale.

This RHC algorithm is triggered when one or multiple disruptive events occur in the transportation systems, which cause subway stations or bus stops to close. This algorithm periodically makes control decisions every $t_1$ minutes until the transportation system recovers from the disruption. At the beginning of every $t_1$ minutes, it updates the locations and occupancy status of all the available extra buses and trains, and predicted passengers demand of every OD pairs till the future $T$ time horizon. Then it solves the problem (19) to the optimal solution set to transport the maximum number of passengers, solves the problem (22) to obtain the rerouting decision of existing bus lines, extra bus and train assignments that minimizes the control cost.

# V. Evaluation

## A. Methodology

To evaluate eRoute in a real-world scenario, we use the dataset described in Table II to conduct a data-driven analysis. We can see that a smartcard record contains the location, time, and transport mode that one passenger swipes the smart card. Based on this dataset, we can extract the origin and

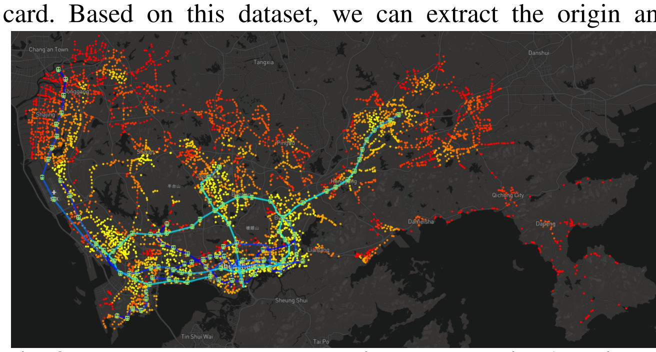


**Fig. 3: Passengers demand density over the city (the lighter the icon, the higher the demand density)**

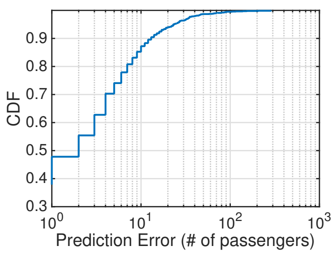



**Fig. 4: CDF of Prediction Error**

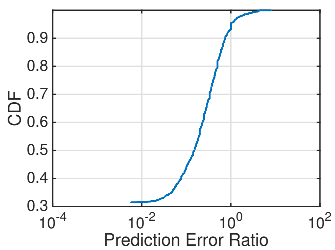



**Fig. 5: CDF of Prediction Error Ratio**

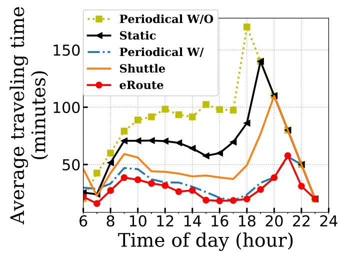



**Fig. 6: Average Traveling Time**

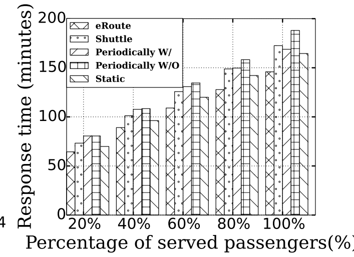



**Fig. 7: Response Time vs. % of Served Passengers**

destination of a trip for each passenger. Figure 3 shows the passenger demand density of the subway and bus system over one city. Then we can estimate the demand for each OD pair, and predict the future demand for each OD pair using linear regression.

We also use a dataset of GPS traces of all buses along 800 different bus lines in the same city. Every bus has networked GPS that can upload real-time location information every 30 seconds. One record in this dataset contains a plate number, a bus line number, a time stamp in seconds, GPS Coordinates, and a real-time speed. Based on this dataset, we can estimate the schedules of every bus line and the trip time at the different time of the day. We can also estimate the real-time passenger demand and available capacity of one bus by combining smartcard reader data and bus GPS data, as all buses use the smartcard system. The locations of subway stations and bus stops are obtained from online digital map service provider. The typical capacity of a city bus is 60 passengers. We assume 25 extra buses can be reallocated.

This dataset contains one disruptive event: a signal problem starting from 7 am to the end of the day that causes significant delays on all trains of a subway line. Both the train speed and the frequency of trains were reduced due to safety concerns. No bus shuttles had been used to address this incident in reality based on an analysis of our data.

To show the effectiveness of eRoute, we compare it with the following existing solutions to handle disruptions: (i) Periodical W/: transportation center makes control decision of rerouting and reallocating available vehicles once every time slot without looking forward to the future several time slots; (ii) Periodical W/O: control center only reroutes existing bus lines once every time slot; (iii) Static: transportation center makes control decision only once at the beginning of the disruptions; (iv) Shuttle: control center utilizes dedicated shuttles running along the influenced subway line to provide substitute services; (v) Nearest: for every influenced subway stations, only a fixed number (10) of geographically close bus lines are considered, such as rerouting and reallocating extra vehicles to these bus lines.

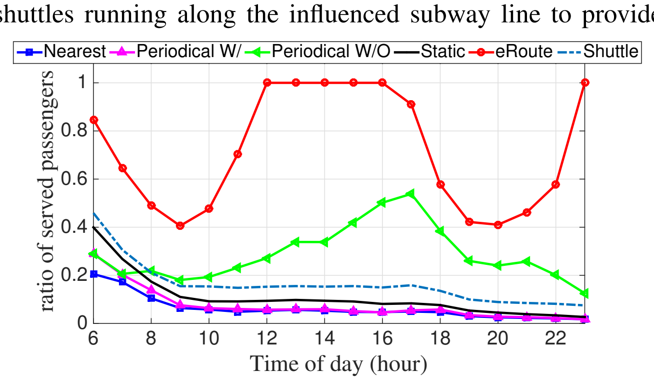



**Fig. 8: RSP over time**

The performance metrics considered include: (i) the ratio of served passengers (RSP): the ratio between the number of served passengers and the number of passenger demand; (ii) average traveling time; (iii) response time: time to serve fixed percentage of passengers. In the experiment, for our eRoute, the length of every time slot is 20 minutes and then the time horizon is 6 time slots.

### B. Results

*1) Prediction error:* In eRoute, we use one linear model to predict the passengers demand of different OD pairs during each time slot (20 minutes), then use them to run our RHC algorithm. We evaluate the accuracy of our prediction method by using one-day data as the testing set and five-days data as the training set. Figure 4 and 5 show the CDF of prediction error and prediction error ratio respectively. Here, we have 72 OD pairs. 80.0% passenger demand of one OD pair during one time slot is no more than 30 and 50.0% of that is fewer than 10 passengers. The prediction error of nearly 90% of OD pairs is less than ten passengers, which demonstrates that simple linear model can predict passengers demand fairly accurately. In Figure 5, prediction error ratio of some OD pairs is large since the actual passengers demand is very small (<5) resulting large prediction error ratio.

*2) Comparison of five solutions:* Figure 8 plots the RSP of five solutions over the day. The performance of eRoute decreases from 6 am to 9 am and then it increases with reduced passengers demand, but it still significantly outperforms the other solutions. Compared to the widely used Static solution, eRoute achieves up to 2.82 times higher RSP during rush hours. This is because that eRoute benefits from our RHC by considering the passengers demand in the future several time slots. Even compared to the second best solution periodically control with extra buses, eRoute achieves 64.1% higher RSP during 19:00-19:59. In the off-peak hours, *e.g.*, 12:00-15:59, with the decrease of passengers demand, eRoute can serve all the passengers.

There are also several observations: the first one is updating the control decision dynamically can improve the performance. Comparing static solution to eRoute and periodical with extra buses, although all of the three solutions reallocate the same number of extra buses, static solution has a lower RSP than

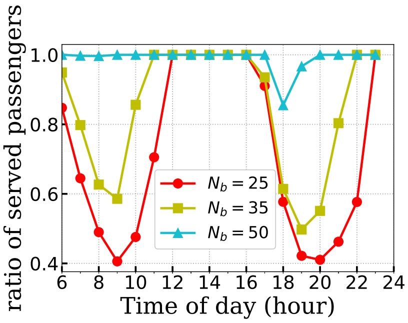



**Fig. 9: Impact of Extra Vehicles**

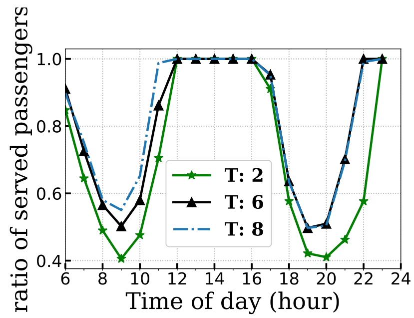



**Fig. 10: Impact of Different Horizon Length**

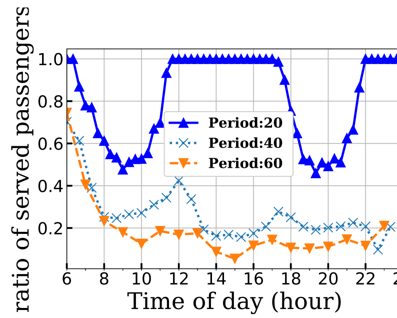



**Fig. 11: Impact of Different Update Period**

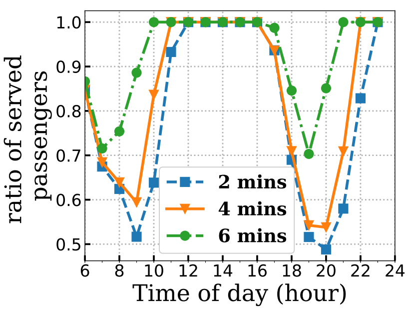



**Fig. 12: Impact of detour constraint**

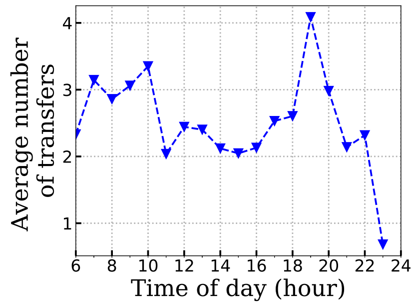



**Fig. 13: Overhead of eRoute**

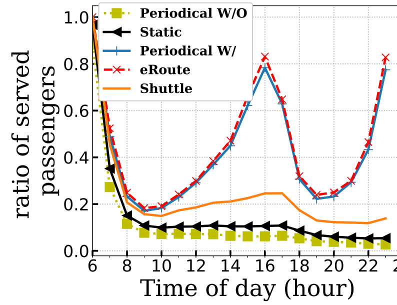



**Fig. 14: eRoute under a Bus Disruption**

the other two solutions do during most time of the day, such as only 50.0% of periodical with extra buses and 20.1% of eRoute during 10:00-10:59. The second observation is rerouting the existing bus lines can provide part of substitute supply, but more extra buses are needed to move most of the passengers. The last one is when comparing the performance of nearest bus lines solution and eRoute, more bus lines which are considered in the subway-bus integrated network would provide more alternative paths and higher supply. However, sometimes, more bus lines would increase the traveling time of passengers due to multi-hop transfer. Therefore, in eRoute, we only consider the bus lines that are around the influenced subway line.

Figure 6 plots the average passenger traveling time of five solutions over time. The traveling time of a passenger includes the waiting time for the transport service and the actual traveling time. From this Figure 6, we can see that the average traveling time of eRoute increases from 7:00 to 9:00 because the surge of passenger demand results in the growth of waiting time at the influenced stations. eRoute still outperforms all the other solutions. For example, compared to periodical control with extra buses, the passenger traveling time with eRoute is still 20.8% less at 9 am. We see the traveling time of all solutions except eRoute decrease at the end of the day, and this is because many passengers arrived at the influenced station after 19:00 do not receive any transport service due to the limited supply, so it only counts their waiting time. We see Here we did not plot the curve of the Nearest solution since its average traveling time is much higher than the others.

We also evaluate the performances of eRoute with a different number of total available buses. As shown in Figure 9, the more vehicles we use, the higher RSP we can achieve. When the number of total available extra buses is 25, the RSP can reach 100% during off-peak hours. When 50 buses are used, the RSP is almost 100% in the whole day.

*3) Response Time:* Once disruptions occur, eRoute reroutes the nearby buses to pick up passengers strained at the influenced subway stations, which has a very short response time. Other solutions like Shuttle need to dispatch extra buses from the distant terminals, which usually takes a long time to reach the influenced stations. Figure 7 shows how long each solution takes to pick up a certain percentage of passengers during the first two-hours after disruptions. We can see that it takes eRoute 88 minutes and Shuttle 101 minutes to move 40% of the passengers, which suggests eRoute has 12.9% faster response time to move 40% of all the passengers.

*4) Time Horizon:* Figure 10 plots the performance of eRoute with different prediction time horizon: 2, 6 and 8 time slots. The observation is that during off-peak hours, the percentage of served passengers of 8 time slots horizon outperforms that of 2 and 6 time slots horizon with average gains of 36.2% and 12.3% respectively. The reason for this observation is that a shorter time horizon means that only the demand in the very recent future is considered, which misses opportunities to achieve better control. During the other hours of the day, the percentage of served passengers with time horizon 6 and 8 is similar.

*5) Control Update Period:* Figure 11 plots the performance of eRoute with different control update periods: 20, 40 and 60 minutes. The prediction time horizon is set to be 120 mins. We can see that shorter control update period can increase the performance of eRoute, as it allows more frequent control decisions for passenger demand changes: when the time slot length is 20 minutes, it improves the performance by up to 2.4 and 2.7 times compared with time slot lengths of 40 and

60 minutes respectively at 8:00.

*6) Detour constraint:* Figure 12 plots the ratio of served passengers of eRoute with different detour constraints: 2, 4, and 6 minutes. The observation is that higher detour constraint can increase the ratio of served passengers, since it allows more bus lines are detoured to influenced nodes, and then provides higher supply for delivering passengers.

*7) Overhead of eRoute:* The overhead of eRoute is measured by the average number of transfers. Since passengers are delivered by rerouted bus lines, which cannot connect origin and destination directly and increases the number of transfers they make. Figure 13 plots the average number of transfers during the day. We can see that passengers need to transfer more than twice to reach their destinations. Specifically, during rush hours, three or more transfers are needed due to higher passenger demand making passengers find the longer alternative path to their destinations.

*8) Disruption to bus system:* To demonstrate the flexibility of our design, we conduct another simulation to evaluate the performance of eRoute under a disruptive failure of the bus system. Here we simulate the transportation network when the bus service is shut down at a few stops in one region of the city due to accidents, and then we apply eRoute to transport strained bus passengers. The simulated disruption to bus system influences six stations, 48 bus lines and 4456 affected passengers from 6:00 to 23:00. Under the real-world scenario, the disruption may not last for one day. To evaluate eRoute for a long period, we assume it lasts for one day. The nearby subway lines are also utilized in our solution to provide transfers. Figure 14 shows the RSP of five solutions under such a scenario. From Figure 14, we can see that the RSP of eRoute decreases during rush hour, *e.g.*, from 7:00 to 9:00 and from 16:00 to 19:00 due to the higher passengers demand. eRoute outperforms all the other solutions. For instance, compared with periodical control with buses, the RSP with eRoute is still 6.8% higher during off-peak hours.

## VI. Discussion

Our work only considers the maximum number of passengers that can be delivered by the integrated subway-bus system simultaneously, but our framework can be extended with the model of passengers' behaviors, *e.g.*, tolerance in traveling time and number of transfers, willingness to switch to other means of transportation, and choices of traveling paths. There are multiple models describing passengers' behaviors, such as multinomial logit model [4], $0-1$ integer linear programming model [26], and dynamic Markov models [24]. We can incorporate the passenger behaviors model in our framework by using it to calculate the number of passengers that can be served of the different OD pairs and the cost of decisions when given the subway-bus integrated network.

Meanwhile, because of the dynamic change of the integrated transportation system, searching for an alternative path also incurs the overhead of passengers. Under the disruptions, transportation authorities can broadcast the paths information at the stations or stops or provide one mobile app for transportation map so that passengers can search their paths easily.

## VII. Related Work

Two types of works are related to our eRoute: (i) urban data-driven applications and analysis and (ii) solutions to handle disruptive events on urban train system.

**Urban data-driven applications and analysis.** There exist some prior works, which either propose data-driven applications or formulate generic models to capture urban phenomena by data analysis. The increasing of availability of urban transportation sensors has encouraged a surge of work focusing on design data-driven applications. Many novel applications are proposed to improve the efficiency of urban transportation system, *e.g.*, providing last-mile transit service to deliver passengers [34], designing a win-win taxi carpool services for both passengers and drivers [32] [21], helping taxi drivers find next passengers efficiently [30] [27] [28]. Based on the collected large-scale data, some works focus on data-driven analysis to formulate generic models to understand urban features, *e.g.*, inferring real-time traffic speeds [33], inferring human mobility patterns across the city [31], investigating spatiotemporal segmentation information of trips inside a metro system [35], predicting traffic in a bike sharing system [18], calculating traffic volume on road segments [22], and inferring traffic cascading patterns [19].

**Solution to handle disruptive events:** They are classified into two directions [12]: pre-disruption preparedness and post-disruption response. While the former focuses on improving robustness of the transportation network, including schedules, frequencies and routes, and the latter seeks to respond to occurred disruptions.

Pre-disruption preparedness is to prepare certain measures before disruption happens. An alternative direction is designing a robust schedule to enhance potential recovery actions. [37] [36] [5] [20] and [8] consider robust train scheduling. Specifically, [5] computes robust routing schedules to improve robustness by either increasing the delay absorption capacity or explicitly providing potential recovery possibilities. [12] studies that metro network resilience to disruptions can be enhanced by localized integration between bus services and subway stations to achieve the desired resilience to potential disruptions. However, their design relies on manual and local incremental adjustments on bus routes, and it generates static and fixed routes. Differently, our solution dynamically adjusts bus routes and based on the passenger demand.

Post-disruption response focuses on coming up with responsive measures for subway system disruptions to alleviate consequences. [7] and [17] state that in the case of a disruption, the first task is keeping subway system running, including timetable adjustment [25], and re-scheduling rolling stock and crew [3] [23] [2] [4]. [13] and [15] introduce shuttle bus services in the disrupted area intelligently which requires extra shuttle buses rather than detouring existing bus lines. [29] has taxis instead of buses as the recovery service for on-board passengers in a public tram system. [16] responds to serious disruptions by redesigning the lines in a particular region around the disruption. Thereby, it removes part of existing lines and establishes new lines. However, these work does not consider the dynamic integration between heterogeneous transportation systems. In contrast, our solution utilizes data-driven insights to perform this integration in real time under disruptive events.

## VIII. Conclusions

In this work, we design, implement and evaluate eRoute for dynamic public transportation integration under disruptive events based on real-world multi-source data from the Chinese city Shenzhen including a 13 thousand bus system, a subway system with 127 subway stations, and an automatic fare collection system with a total of 16,840 card readers and 8 million card users. Our endeavors offer a few valuable insights for fellow researchers to conduct the similar investigations: (i) under disruptive events, the existing effort for transfers within public transportation provides an opportunity to dynamically integrate them without requiring ad-hoc efforts, *e.g.*, extra bus lines; (ii) given spatial temporal partitions of public transportation systems and natures of disruptive events, we can deliver strained passengers with a hierarchical receding horizon control framework to reduce their affected travel time with minimal overheads; (iii) our work only focused on the technical frontier on the modeling and resource allocation framework, and it is more challenging to establish right policies that would make large-scale deployment feasible to reduce impacts of disruptive events and increase transportation resilience.

## Acknowledgment

This research is supported in part by the NSF grants CNS 1553273, CNS 1463722, CPS 1646470 and CNS 1446640.

## References

[1] F. T. Administration. Transit asset management practices a national and international review. 2010.

[2] G. Budai, G. Maróti, R. Dekker, D. Huisman, and L. Kroon. Rescheduling in passenger railways: the rolling stock rebalancing problem. *Journal of scheduling*, 2010.

[3] V. Cacchiani, A. Caprara, L. Galli, L. Kroon, G. Maróti, and P. Toth. Railway rolling stock planning: Robustness against large disruptions. *Transportation Science*, 2012.

[4] L. Cadarso, Á. Marín, and G. Maróti. Recovery of disruptions in rapid transit networks. *Transportation Research Part E: Logistics and Transportation Review*, 2013.

[5] A. Caprara, L. Galli, L. Kroon, G. Maróti, and P. Toth. Robust train routing and online re-scheduling. In *OASIcs-OpenAccess Series in Informatics*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2010.

[6] M.-C. Costa, L. Létocart, and F. Roupin. Minimal multicut and maximal integer multiflow: A survey. *European Journal of Operational Research*, 2005.

[7] D. De Almeida, F. Trouve, and B. Hebben. A decision aid module for resources rescheduling when managing disturbances. In *Poster presented at the World Congress of Railway Research (WCRR'03)*, 2003.

[8] M. Fischetti, D. Salvagnin, and A. Zanette. Fast approaches to improve the robustness of a railway timetable. *Transportation Science*, 2009.

[9] M. A. Fonoberova and D. D. Lozovanu. The minimum cost multi-commodity flow problem in dynamic networks and an algorithm for its solving. *Computer Science*, 2005.

[10] N. Garg, V. V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 1997.

[11] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, and M. Yannakakis. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. *Journal of Computer and System Sciences*, 2003.

[12] J. G. Jin, L. C. Tang, L. Sun, and D.-H. Lee. Enhancing metro network resilience via localized integration with bus services. *Transportation Research Part E: Logistics and Transportation Review*, 2014.

[13] J. G. Jin, K. M. Teo, and L. Sun. Disruption response planning for an urban mass rapid transit network. In *transportation research board 92nd annual meeting, Washington DC*, 2013.

[14] R. L. Kemp and J. C. Stephani. Urban transportation innovations worldwide: A handbook of best practices. 2015.

[15] K. Kepaptsoglou and M. G. Karlaftis. The bus bridging problem in metro operations: conceptual framework, models and algorithms. *Public Transport*, 2009.

[16] A. Kiefer, S. Kritzinger, and K. F. Doerner. Disruption management for the viennese public transport provider. *Public Transport*, 2016.

[17] L. Kroon and D. Huisman. Algorithmic support for railway disruption management. In *Transitions Towards Sustainable Mobility*. Springer, 2011.

[18] Y. Li, Y. Zheng, H. Zhang, and L. Chen. Traffic prediction in a bike-sharing system. ACM SIGSPATIAL, 2015.

[19] Y. Lian, Z. Jiang, and Y. Zheng. Inferring traffic cascading patterns. ACM SIGSPATIAL, 2017.

[20] Y. Liu, C. Liu, N. J. Yuan, L. Duan, Y. Fu, H. Xiong, S. Xu, and J. Wu. Intelligent bus routing with heterogeneous human mobility patterns. *Knowledge and Information Systems*, 2017.

[21] S. Ma, Y. Zheng, and O. Wolfson. Real-time city-scale taxi ridesharing. *IEEE Transactions on Knowledge and Data Engineering*, 2015.

[22] C. Meng, X. Yi, L. Su, J. Gao, and Y. Zheng. City-wide traffic volume inference with loop detector data and taxi trajectories. ACM SIGSPATIAL 2017.

[23] L. K. Nielsen. *Rolling Stock Rescheduling in Passenger Railways: Applications in short-term planning and in disruption management*. Number EPS-2011-224-LIS. 2011.

[24] A. Pentland and A. Liu. Modeling and prediction of human behavior. *Neural computation*, 1999.

[25] C. G. Walker, J. N. Snowdon, and D. M. Ryan. Simultaneous disruption recovery of a train timetable and crew roster in real time. *Computers & Operations Research*, 2005.

[26] H. Yin, B. Han, D. Li, J. Wu, and H. Sun. Modeling and simulating passenger behavior for a station closure in a rail transit network. *PLoS one*, 2016.

[27] J. Yuan, Y. Zheng, X. Xie, and G. Sun. T-drive: Enhancing driving directions with taxi drivers' intelligence. *IEEE Transactions on Knowledge and Data Engineering*, 2013.

[28] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie. T-finder: A recommender system for finding passengers and vacant taxis. *IEEE Transactions on Knowledge and Data Engineering*, 2013.

[29] A. Z. Zeng, C. F. Durach, and Y. Fang. Collaboration decisions on disruption recovery service in urban public tram systems. *Transportation research part E: logistics and transportation review*, 2012.

[30] D. Zhang and T. He. pcruise: Reducing cruising miles for taxicab networks. In *Real-Time Systems Symposium (RTSS)*. IEEE, 2012.

[31] D. Zhang, J. Huang, Y. Li, F. Zhang, C. Xu, and T. He. Exploring human mobility with multi-source data at extremely large metropolitan scales. In *Mobicom*. ACM, 2014.

[32] D. Zhang, Y. Li, F. Zhang, M. Lu, Y. Liu, and T. He. coride: carpool service with a win-win fare model for large-scale taxicab networks. In *Sensys*. ACM, 2013.

[33] D. Zhang, J. Zhao, F. Zhang, and T. He. Urbancps: a cyber-physical system based on multi-source big infrastructure data for heterogeneous model integration. In *ICCPS*. ACM, 2015.

[34] D. Zhang, J. Zhao, F. Zhang, R. Jiang, and T. He. Feeder: supporting last-mile transit with extreme-scale urban infrastructure data. In *IPSN*. ACM, 2015.

[35] F. Zhang, J. Zhao, C. Tian, C. Xu, X. Liu, and L. Rao. Spatiotemporal segmentation of metro trips using smart card data. *IEEE Transactions on Vehicular Technology*, 2016.

[36] P. J. Zwaneveld, L. G. Kroon, H. E. Romeijn, M. Salomon, S. Dauzere-Peres, S. P. Van Hoesel, and H. W. Ambergen. Routing trains through railway stations: Model formulation and algorithms. *Transportation science*, 1996.

[37] P. J. Zwaneveld, L. G. Kroon, and S. P. Van Hoesel. Routing trains through a railway station based on a node packing model. *European Journal of Operational Research*, 2001.

## Appendix

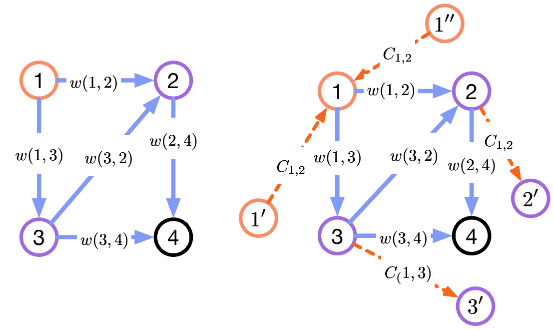
### A. Transformation

**Step 1:** removing dynamic graph topology by rerouting every bus line to all the influenced nodes without conflict with the detour cost constraint. Firstly, we introduce how to calculate the detour cost function $f(k,i)$. Suppose $v_{k_1}$ is the nearest node to $v_{d_i}$ that the $k$th bus line visits, and $v_{k_2}$ is the next node which the $k$th bus line visits after $v_{k_1}$. Then $f(k,i) = travelling\_time(v_{k_1}, v_{d_i}) + travelling\_time(v_{d_i}, v_{k_2}) - travelling\_time(v_{k_1}, v_{k_2})$, where $travelling\_time(v_{k_1}, v_{d_i})$ is a function computing the travelling time from $v_{k_1}$ to $v_{d_i}$. Without losing generality, we use the euclidean distance between two locations to estimate the traveling time for this function in this paper. When road congestion information is available, this function can be generalized to include real-time traveling time. If $f(k,i) \le \alpha$, we add two virtual directed edges $e(v_{k_1}, v_{d_i})$ and $e(v_{d_i}, v_{k_2})$ and remove the existing edge $e(v_{k_1}, v_{k_2})$. We call all the edges added based on the existing transportation network as regular edges and all the edges added on the basis of rerouting as virtual edges. It is noted that the cost of every regular edge is 0, as they don't introduce any extra detour. The cost of every virtual edge is equal to the corresponding distance between two transportation nodes. After the above procedure, we have a topology with all possible reroutes. We may not reroute every bus line and therefore need every virtual edge, so some of the virtual edges will be eliminated after rerouting if there is no passenger flow using it. The graph generated after adding virtual edges is denoted as $G'$.

**Step 2:** was modifying $G'$ to satisfy the supply constraint. For every source-sink pair, we add one virtual source and one virtual sink node, and two directed edges to connect them: one edge is from the virtual source node to the node of origin, the other is from the sink node to the virtual sink node. The edge capacity of both two added edges is set to be equal to the corresponding source-sink pair demand, and its cost is set to be 0. Figure. 15 provides one example to illustrate the basic idea to modify $G'$ into $G''$. After this modification, we only need to find the flow from the virtual source node to virtual sink node in $G''$ for every source-sink pair, and it is clear that such flow satisfies the supply constraint. After the above two steps, *DIP* is changed to find the IMCMF with dynamic edge capacity in $G''$.

### B. Circulating supply of trains

We define $W^s \in \mathbb{N}_+^{N^s}$, one column vector to denote the number of time slots needed to complete one end-to-end trip of all subway lines. For instance, $W_i^s$ is the number of time slot needed to finish one trip of $i$th subway line. Let $U^s(t) \in \mathbb{N}_+^{N_s}$ be one column vector to represent the number of time slot needed to finish current subway line trip at the beginning of time slot $t$. For instance, it costs 4 time slots to finish one trip of the first subway line, and the first extra train reallocated to the first subway line at time slot 1. So at the time slot 2, we




(a) $G'$ with four transportation nodes where two source-sink pair (1,2) and (1,3).

(b) $G''$ after adding four virtual nodes and the corresponding virtual edges, whose capacity is the passengers demand.

**Fig. 15: An example of Problem Transformation Step 2.**

have the follow values: $U_1^s(2) = 3$. Then the relation between $U_i^s(t)$ and $U_i^s(t-1)$ is:

$$U_i^s(t) = \max\{0, \max\{\sum_{j=1}^{N^s} X_{i,j}^s(t-1)W_i^s, U_i^s(t-1)\} - 1\} \tag{23}$$

where $t \ge 2$ and $U_i^s(1) = 0$ for $1 \le i \le N_s$. Based on $U^s(t)$, $\gamma^s(t) \in \{0,1\}^{N_s}$ is one vector column to describe whether every extra train can be reallocated during time slot $t$. It is clear that if $U_i^s(t) > 0$, $i$th extra train is still operating for one existing subway line and it cannot be reallocated, otherwise, it can be reallocated. We have the following equation:

$$\gamma_i^s(t) = I_1(U_i^s(t)) \tag{24}$$

where $I_1(U_i^s(t))$ is an indicator function, and it is equal to 1 if $U_i^s(t) = 0$, otherwise, it is 0. Then for $i$th train, during time slot $t$, it cannot be reallocated to more than $\gamma_i^s(t)$ subway lines:

$$X^s(t)\mathbf{1}_{N^s} \preceq \gamma^s(t) \tag{25}$$

We remark that it's possible that $\sum_{j=1}^{N^s} X_{i,j}^s(t) = 0$, however, $i$th train also contributes to one existing subway line, because of operating for one existing subway line. Hence, we define $O^s(t) \in \{0,1\}^{N_s \times N^s}$ to denote which subway line that every extra train contributes to during time slot $t$. $O_{i,j}^s(t) = 1$ if $i$th train is operated for $j$th subway line during time slot $t$, otherwise, it is 0. Then, we have the following relation:

$$O_{i\cdot}^s(t) = O_{i\cdot}^s(t-1)I_2(U_i^s(t)) + X_{i\cdot}^s(t) \tag{26}$$

where $O_{i\cdot}^s$ is the $i$th row of $O^s(t)$ and $I_2(U_i^s(t))$ is also one indicator function. $I_2(U_i^s(t)) = 1$ if $U_i^s(t) > 0$, otherwise, it is 0. Finally, we describe the capacity increase of $e_j$ during time slot $t$ due to $N_s$ extra trains:

$$w^s(e_j, t) = \sum_{i=1}^{N_s}\sum_{k=1}^{N^s} I(O_{i,k}^s(t)) \times C^s \times R_{k,j}^{se} \tag{27}$$

where $C^s$ is the capacity that one extra train can provide.