Case Studies on Transport Policy 20 (2025) 101448

Contents lists available at ScienceDirect

Case Studies on Transport Policy

journal homepage: www.elsevier.com/locate/cstp

Design and implementation of a network-aware automated bus scheduling system for optimizing operational efficiency and financial performance

Pradeep Borade [a], Gajendra Nagar [a], Hemant K. Suman [b], Lakshay Lakshay [c,*], Ravi Gadepalli [d], Nomesh B. Bolia [a]

[a] Department of Mechanical Engineering, IIT Delhi, India
[b] Centre for Transportation Systems (CTRANS), IIT Roorkee, India
[c] Department of Mechanical Engineering, IIT (BHU) Varanasi, India
[d] Transit Intelligence LLP, Bengaluru 560 041, India

ARTICLE INFO

Keywords:
Bus schedule optimization
Automation
Overlaps
Rescheduling
Redeployment
Decision support system

ABSTRACT

This work develops decision support models for bus schedule optimization to improve the existing public transport (PT) systems schedule. It particularly addresses the issue of *overlapping trips,* where multiple buses are scheduled to travel the same route simultaneously. This occurs due to the absence of network-aware scheduling, a significant problem of operational inefficiency in PT systems. To remove such overlapping trips, our approach presents mixed-integer linear programming (MILP) models and innovative solution algorithms for efficient rescheduling and redeployment. To prove the efficacy of the developed model, we use data from Bengaluru Metropolitan Transport Corporation (BMTC), India's largest urban bus fleet operator, and optimize the schedule with no additional capital expenditure and minimal changes to the existing schedule. The work is carried out in close consultation with BMTC and led to the development of a computer-based decision support bus-schedule optimization toolkit (B-SOT). The B-SOT automates the schedule development process using simple CSV files as input and output, making it easy to use for officials at all levels. The toolkit is sufficiently comprehensive to be applied to other cities with contextual adjustments. Overall, the decision support models and the computer-based toolkit developed are novel, modular, and quite useful for PT decision-making to optimize bus schedules while freeing up some buses for deployment elsewhere.

1. Introduction

Transportation systems in developing countries are characterized by low vehicle ownership rates and high dependence on public transport (PT) systems for daily commutes to fulfil various activities (Suman et al., 2017; Thynell et al., 2010). The city governments typically provide formal bus or rail-based PT services like city buses, Bus Rapid Transit (BRT), and sub-urban rail or metro rail systems. In most cases, the available capacity and pace of development of these formal PT systems are inadequate to handle the increasing travel demand (Badami and Haider, 2007; Cervero, 2000; Cervero and Golub, 2007; Goel and Tiwari, 2016; Phun and Yai, 2016). Therefore, optimizing the service delivery of operators to maximize services provided within the available resources is crucial to meet people's mobility needs in a cost-efficient manner (Suman and Bolia, 2019; Suman et al., 2021). To achieve this, data-driven operational planning practices are required to ensure effective resource utilization.

1.1. Background

City bus operators in developing countries still follow manual scheduling systems, which are updated only periodically, often needing to incorporate network awareness, which results in suboptimal utilization of available resources. In cities with manual scheduling systems, these buses are scheduled from several depots in silos without accounting for other depots in the network, which leads to inadvertent overlaps in trips and schedules of some buses. Overlaps represent the condition when multiple buses are scheduled to operate on a particular route at the same departure time in the same direction. According to the literature, such overlaps may be defined as a subset of bus bunching (Andres and Nair, 2017; Bartholdi and Eisenstein, 2012). The bus bunching problem negatively impacts both operational and financial

* Corresponding author.
E-mail address: lakshay.mec@iitbhu.ac.in (L. Lakshay).

https://doi.org/10.1016/j.cstp.2025.101448
Received 15 October 2024; Received in revised form 7 April 2025; Accepted 7 April 2025
Available online 9 April 2025

efficiency. This problem is mostly studied in terms of the random variability in the travel times of buses and headway deviation at the bus stops due to traffic congestion and the random number of passengers boarding/alighting. However, the concept of overlapping trips in bus bunching is not thoroughly addressed in the existing literature.

Due to the presence of overlapping trips, resulting in inefficient scheduling and lack of communication between depots may result in uneven headway, an uneven load distribution, and inefficient capacity utilization at the stops (Cats et al., 2016). Further, this vulnerability can lead to the degradation of the public transport system, which significantly reduces its attractiveness (Lysgaard and Wøhlk, 2014). Also, due to these overlaps of buses, there is an uneven distribution of passengers at the stops; one bus may get more passengers that will increase overcrowding, denied boarding, and discomfort for the passengers, while other buses get the lesser passengers that may reduce the utilization of the buses; this may result in long-term negative change in the passenger demands. Further, this resulted in a decline in bus productivity, an increase in operational costs, and an increase in the cost per passenger on underutilized buses (Nesheli and Ceder, 2015; Daganzo, 2009). Furthermore, this not only prolongs the waiting time for passengers, but also leads to confusion and dissatisfaction among them (Petit et al., 2018). So, analytically resolving overlapping trips in schedules is a crucial concern for bus operators. Even though cities are aware of these problems, the analytical complexity of updating schedules manually constrains them from adopting the necessary changes.

To address these gaps, bus systems need to transition to data-driven, optimized and automated approaches while meeting their unique operational planning needs (Gwilliam, 2003; Patel et al., 2019; Sohail et al., 2004). Accordingly, this work presents a data-driven approach for solving the Multi-Depot Vehicle scheduling problem (MDVSP), which is commonly observed in cities with several bus routes. The existing solutions to MDVSP are adapted to the real-world context of Bengaluru to develop a practical schedule optimization methodology for meeting the operational planning needs of the transit agency by optimally minimizing overlapping trips. The scope of this work is to address the issue of *overlapping trips* and thus improve operational efficiency.

The structure of the remaining paper is as follows: Section 2 describes the literature review on various strategies to enhance operational efficiencies and financial performance. Section 3 provides a detailed description of the problem and the specific challenges posed by overlapping. Further, this section also provides details on the implications of overlapping trips on operational efficiency. Section 4 describes the solution methodology, including data cleansing techniques, rescheduling algorithms, and a heuristic based on the mathematical model solution results. Section 5 presents a case study of one of the largest PT agencies in India, BMTC, and how the optimal schedule can be formulated to accommodate all trips with the fewest number of buses. Finally, a discussion of the case study is provided, followed by concluding remarks and possible future extensions of the work.

2. Literature review

An efficient bus planning process is important for any transit agency, as it affects operational and financial efficiencies. These efficiencies can be achieved by utilizing the resources effectively (Andrade-Michel et al., 2021). Multiple works in the literature focus on improving operational performance by efficiently utilizing and allocating resources to minimize total operational costs, including the fixed cost of procuring buses and the operational cost (Forbes et al., 1994; Zhoucong et al., 2013; Wei et al., 2013; Ceder, 2011; Hassold and Ceder, 2014; Aziz et al., 2022). The planning process of traditional bus and rail-based PT systems is commonly divided into a set of strategic, tactical, and operational planning activities. Strategic and tactical planning is carried out with the primary objective of designing the route networks and service frequencies to meet user needs optimally, while operational planning focuses majorly on vehicle scheduling, crew scheduling and rostering while other smaller problems include maintenance scheduling and parking management (Patel et al., 2019; Bunte and Kliewer, 2009).

In cities with established bus-based PT systems, the route networks evolve over several years and have limited flexibility for change (Heyken Soares et al., 2019). Therefore, tactical and operations planning offers a higher scope for optimization. This involves allocating fleet and crew to meet the service needs while minimizing the cost of operations for the transit operator (Ceder and Wilson, 1986; Desaulniers and Hickman, 2007; Ibarra-Rojas et al., 2015). Forbes et al. (Forbes et al., 1994) is one of the earliest works that solve vehicle scheduling problems (VSPs) by minimizing the total fixed and operational costs of buses. Followed by several works emphasizing different objective functions and considerations. Zhoucong et al. (Zhoucong et al., 2013) solve vehicle scheduling with even headway and minimize the total bus fleet and operational cost. Wei et al. (Wei et al., 2013) propose a vehicle scheduling solution that minimizes the total fleet size. Hassold & Ceder (Hassold and Ceder, 2014) solve VSP for multiple vehicle types and minimize the total fixed cost of buses and operational costs of serving the given number of bus trips.

Further, optimization of the bus schedule may play an important role in improving operational efficiency from both the passenger's and the operator's point of view (Shang et al., 2023). Different strategies are used to achieve this, such as allowing changes to parameters like speed, start time, and headway on the route. Sun et al. (Sun et al., 2021) present a flexible bus route optimization model to optimize operator and passenger costs simultaneously by changing the shutter travel time. He et al. (He et al., 2019) present a strategy to improve operational stability by varying the bus speed. Ruiz et al. (Ruiz et al., 2017) has developed a frequency optimization model to improve service quality and social equity. Naumann et al. (Naumann et al., 2011) solve the problem of optimizing the schedule by minimizing the sum of planned and deviation costs for public transport. Yan et al. (Yan et al., 2012) optimize the schedule by minimizing the cost of schedule deviation from the actual schedule by taking into account the uncertainty in the travel time of the bus. Desfontaines & Desaulniers (Desfontaines and Desaulniers, 2018) solve vehicle scheduling problems by minimizing the total operational cost. The work uses a new strategy and allows the modification in the trip's start time while minimizing the deviation of trips from the actual schedule. Furthermore, even after optimizing the schedule and allocating resources, maintaining headway between consecutive trips on high-frequency routes might be difficult, resulting in overlapping bus schedules.

In the current paper, we focus on the problem of *overlapping* bus schedules, a special case of bus bunching problems that negatively impact operational and financial efficiency. Bus bunching is a phenomenon where multiple buses serving a bus route arrive at a bus stop simultaneously (He et al., 2019). There are various control strategies used for addressing the bus bunching problems that arise due to uneven and high travel demand on a given route or corridor (Andres and Nair, 2017; Bartholdi and Eisenstein, 2012; Daganzo, 2009; Daganzo and Pilachowski, 2011; Delgado et al., 2012; Hernández et al., 2015; Lizana et al., 2014; Schmöcker et al., 2016). Different strategies are used in the literature to mitigate the bus bunching problem. Daganzo & Pilachowski (Daganzo and Pilachowski, 2011) developed an adaptive control system that works by changing the cruising speed of buses based on the time–space between the front and rear buses. For better performance, the space between buses is monitored frequently. Daganzo (Daganzo, 2009) suggested a bus-holding technique to deal with the issue of bus bunching. Based on real-time data on route headway, the author suggested a system that dynamically determines bus holding time at route stops. Bartholdi & Eisenstein (Bartholdi and Eisenstein, 2012) proposed a coordinated approach to reduce the bus bunching issue and equalise the headway on the route through a self-equalising scheme. Following any disruption on the route, buses autonomously adjust their spacing without management intervention or with driver awareness. Andres & Nair (Andres and Nair, 2017) address the problem of bus bunching using dynamic bus holding and headway prediction. In order to decrease the variance in headway, this study presents a methodology that allows buses to modify dwell time at bus stops. Petit et al. (Petit et al., 2018) proposed a bus substitution approach to mitigate the problem of bus bunching. If there is any service delay, an alternative bus will be provided to serve the remaining bus trips. To make the optimal decision for substitution, a modelling framework was developed that minimizes the total costs for passengers and the agency.

Furthermore, public transport businesses concentrate on the effective use of resources, particularly vehicles. Therefore, they have been required to act market-oriented instead of the conventional monopolistic strategy. So, operations research supported by decision support systems and advanced computational techniques can play an important role here (Kliewer et al., 2006). In the past, numerous optimization toolkits have been developed within the mobility context. For example, Aziz et al. (Aziz et al., 2022) present a planning toolkit based on optimization for on-demand mobility services. Garikapati (Garikapati, 2019) presents an optimization toolkit for autonomous shuttles. Further, Ocalir-Akunal (Ocalir-Akunal, 2016) presents a review of various decision support systems for transport planning but mostly focuses on airline crew scheduling, road traffic control, etc. Further, a summary of key aspects considered in the existing literature is presented in Table 1.

However, to the best of our knowledge, none of the studies focus on solving and developing tools to address overlapping issues which are causing operational efficiencies in the existing system. Moreover, to improve urban PT services, data is the most important requirement for such an optimization toolkit, and also the data processing must be automated. Further, with the development of communication

Table 1

Comparative literature review for financial and operational efficiency improvement studies.

Author(s)	Objective	Constraints	Financial efficiency	Operational efficiency	Overlapping / bunching?	Tool developed	Large realistic network
Kliewer et al. (Kliewer et al., 2006)	Minimize user & operator cost	Vehicle flow	✔	✔	×	✔	✔
Daganzo (Daganzo, 2009)	Minimize bus bunching	Bus delay, deviation from actual headway	×	✔	✔	×	×
Naumann et al. (Naumann et al., 2011)	Optimize cost of planned & disruption	Vehicle flow	✔	✔	×	×	×
Daganzo & Pilachowski (Daganzo and Pilachowski, 2011)	Minimize bus bunching	Headway calculation, traffic disturbances, passenger arrivals	×	✔	✔	×	×
Yan et al. (Yan et al., 2012)	Minimize schedule deviation cost	Schedule deviation, slack time limitation	×	✔	×	×	×
Bartholdi & Eisenstein (Bartholdi and Eisenstein, 2012)	Minimize bus bunching	Headway calculation, velocity calculation	×	✔	✔	×	×
Wei et al. (Wei et al., 2013)	Minimize bus procurement & operating cost	Vehicle flow	✔	×	×	×	×
Zhoucong et al. (Zhoucong et al., 2013)	Minimize bus procurement & operating cost	Vehicle flow	✔	×	×	×	×
Hassold & Ceder (Hassold and Ceder, 2014)	Minimize bus procurement & operating cost	Vehicle flow	✔	×	×	×	×
Nesheli & Ceder (Nesheli and Ceder, 2015)	Minimize total passenger travel time & waiting time	Minimum headway limitation, vehicle flow	×	✔	×	×	×
Cats et al. (Cats et al., 2016)	Cost-benefit analysis of congestion effects	Passenger flow, vehicle riding, dwell time	×	✔	×	×	✔
Andres & Nair (Andres and Nair, 2017)	Minimize overtaking	Vehicle flow, arrival time	×	✔	✔	×	×
Ruiz et al. (Ruiz et al., 2017)	Optimize headway, service level & equity	Bus service level, public transport need index	×	✔	×	×	✔
Desfontaines & Desaulniers (Desfontaines and Desaulniers, 2018)	Minimize operating & penalties cost for changes in departure time	Vehicle flow	×	✔	×	×	×
Petit et al. (Petit et al., 2018)	Minimize agency & passenger cost	Bus available, bus deviation from actual schedule	×	✔	✔	×	×
He et al. (He et al., 2019)	Improve stability of bus line	Stability index, dynamic circle headway	×	✔	✔	×	×
Sun et al. (Sun et al., 2021)	Minimize operating & waiting time cost	Vehicle flow, vehicle capacity	✔	✔	×	×	×
Aziz et al. (Aziz et al., 2022)	Minimize bus procurement cost	Vehicle flow, vehicle capacity, subtour elimination	✔	×	×	✔	✔
Shang et al. (Shang et al., 2023)	Minimize bus procurement, operating & passenger cost	Vehicle flow, vehicle capacity	✔	✔	×	×	✔
Current work	Optimize the operational & financial performance by minimizing overlapping trips & number of buses required	Vehicle flow, overlapping	✔	✔	✔	✔	✔ (45 depots and 2,203 routes)

technologies, from the Internet of Things (IoTs) to the Internet of Wearables (IoWs), a large amount of data is automatically collected from multiple sources, including mobile phones, smartwatches, and smart cards. In the past, PT services have utilized data from these sources along with other data such as automatic vehicle location (AVL) and automatic passenger count (APC) (Dao, 2022; Ometov et al., 2016; Pi et al., 2018; Zhong et al., 2014). Cao & Wachowicz (Cao and Wachowicz, 2019) propose a stand-alone platform for automating the data-analysis process to enhance service quality in mobility. In line with this context, this work develops a bus schedule optimization toolkit (B-SOT), which analyzes the existing schedule and eliminates overlapping trips in an automated way.

To sum up, the existing literature employs various strategies to enhance operational efficiencies and financial performance. Some studies focus on improving financial performance by minimizing resources, such as the number of buses, or reducing the cost of meeting demand. Additionally, some studies have focused on enhancing operational efficiency by implementing strategies that allow adjustments to parameters such as speed, start time, and headway on the route. Then, the work related to bus bunching addresses the impact of variability in travel times and headways at the bus stops due to traffic congestion and passengers boarding/alighting. However, improving the operational efficiencies and financial performance that our work addresses are caused by overlapping trips generated due to depot-level planning and a lack of communication between schedule planners at different depots, which is not addressed in the literature. Also, in the past, numerous optimization toolkits have been developed within the mobility context; however, none of the studies focused on solving and developing tools to address overlapping issues that are causing operational inefficiencies and affecting financial performance in the existing system.

3. Problem Description

This work considers a transit network with multiple depots. Each depot is managed by different authorities, and the daily bus schedule is independently generated to serve the demands of the large transit network. Each bus starts its operation from a particular depot and returns to the same depot after completing its scheduled trips. The buses on each route originate from one or more depots depending on the minimum dead-mileage and ensure services are provided from both ends of the route. Each depot develops its timetable and bus schedule under its jurisdiction. When buses on a given route are scheduled from different depots independently, there is a possibility of *overlapping*.

Overlapping of buses is defined as when multiple buses are scheduled at the same time to serve the same route, i.e., from the same origin to the same destination. Such trips are referred to as overlapping trips. Fig. 1 depicts the illustration of overlapping trips used in this paper. Uneven distribution of bus allocations in the network and an unreliable timetable in certain areas of the city are some of the evident issues of such overlapping trips. Further, such overlaps cause some buses to run beyond their capacity while some run empty, resulting in trips that do not generate any additional revenue. Consequently, it is essential to eliminate these overlaps from the existing schedule so that the saved buses can be deployed to other routes or areas of the city where more service is required, thereby expanding the network and connecting an increasing number of individuals to PT. Due to the enhanced efficiency of the redeployment of the freed buses, the PT agency can further generate more revenue.

Mathematical constraint programming models and solution algorithms are developed in this paper to eliminate overlapping trips from the daily schedules. As already indicated, overlapping trips cause redundancies and lead to underutilization of existing resources (buses). A complete overhaul of the schedule is not possible for a bus network of large size currently in use by millions of commuters. The sheer disruption, inconvenience to citizens, and possible backlash require the development of an approach that is acceptable to decision-makers.

Thus, we present a novel technique that first identifies the overlapping trips in the timetables and then presents methodology to determine the optimal bus schedule to minimize such overlapping trips. The purpose of this study is to analyze the existing schedule and eliminate overlapping trips while causing as little disruption as possible to the remaining schedule. To do so, we present a data-driven optimization toolkit to manage such overlapping services and reschedule trips to ensure adequate headway between services.

4. Methodology to address overlapping issues

An optimization-based Desktop *Toolkit* is developed in this work to address overlapping issues in the existing bus schedule. The toolkit ensures minimum deviation in the revised schedule from the existing one and provides the improved schedule with zero overlaps. Fig. 2 presents the *Toolkit* framework that provides the optimized bus schedule in a reasonable amount of computational time.

The existing scheduling data of buses after cleaning is fed as input to the *Toolkit*. Then the *Toolkit* adopts a two-step approach – the first is to reschedule overlapping trips to the maximum possible extent, and the second is to redeploy the non-overlapping trips of the excluded buses optimally; both steps are detailed in Sections 4.1 and 4.2, respectively. The output of the *Toolkit* is an improved schedule with zero overlaps. The detailed design, software packages and GUI of this toolkit is described in Section 6. Efforts have been made to make it platform-independent.

4.1. Rescheduling overlapping trips

Rescheduling is one of the two major steps in the *Toolkit* to reduce overlaps from the existing bus schedule. The *Toolkit* first reschedule overlapping trips by a maximum amount of time (δ_{max}) before or after

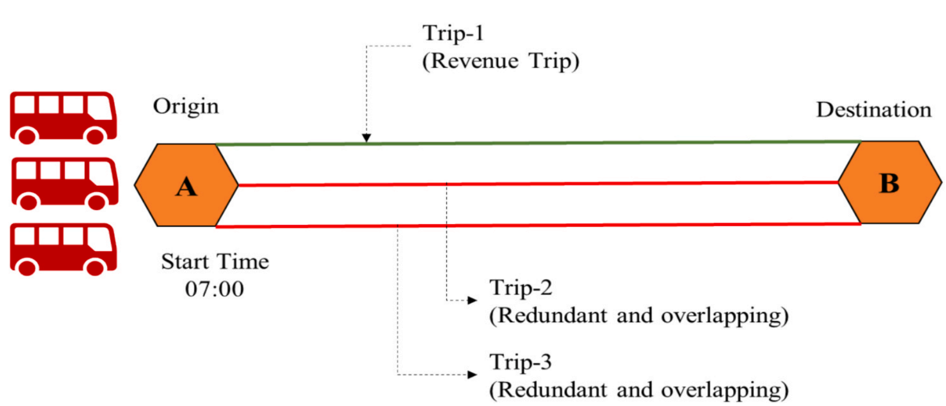

Fig. 1. Representation of overlapping trips.

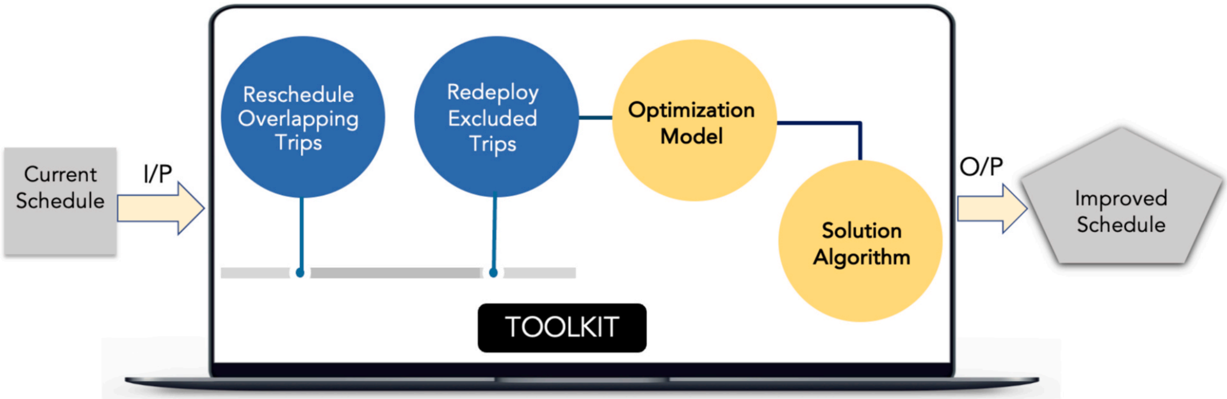

Fig. 2. Framework of desktop-based *toolkit*.

their existing dispatch times. Where, δ_{max} is the minimum value of headway required at origin, o, of any given route to maintain the minimum headway from operational perspective. Its value is either provided by the transit agencies or estimated based on the safe headway requirements between the vehicles. Table 2 presents the list of notations used in models and algorithms developed in this work.

As explained in the introduction Section, overlapping trips are those trips where headway ($H^{ro}_{t,t+1}$) between consecutive trips at the origin of a given route is zero (i.e., $H^{ro}_{t,t+1} = \hat{\tau}^{ro}_{t+1} - \hat{\tau}^{ro}_{t} = 0$, where, $\hat{\tau}^{ro}_{t}$ is the start time of t^{th} trip on route, r, at origin, o). The number of overlaps with t^{th} trip is given by ω^{ro}_{t}. Out of ω^{ro}_{t}, the number of overlapping trips that can be eliminated by rescheduling (M^{ro}_{t}) is given by expression (1).

$$M^{ro}_{t} = min\left\{ \omega^{ro}_{t}, \frac{\left[min\left(\delta_{max}, [H^{ro}_{t-1,t}] - \alpha\right) + min\left(\delta_{max}, [Hn^{ro}_{t,t+\omega}] - \alpha\right) \right]}{\delta_{max}} \right\}$$
$$\forall r \in R, o \in O(r), t \in T(o) \tag{1}$$

where, δ_{max} is the maximum allowable change in the dispatch time of a given trip from its existing schedule so that none of the trips of a given bus deviate from the current schedule beyond a permissible limit. $O(.)$ is a function that returns the origin of a given route, and $T(.)$ and $\overline{T}(.)$ are sets of all the trips and non-overlapping trips, respectively, from a given origin.

The *Toolkit* reschedules trips if the following two conditions are met:

(a) Overlapping trips are present in the existing schedule (i.e., $\omega^{ro}_{t} > 0$), where $\forall r \in R$, $o \in O(r), t \in T(o)$.
(b) Headway between non-overlapping consecutive trips is sufficient to reschedule at least one trip between them (i.e., $max\left(Hn^{ro}_{i-1,i}, Hn^{ro}_{i,i+1}\right) \geq 2\delta_{max}$), where $\forall r \in R$, $o \in O(r), i \in \overline{T}(o)$.

If the first condition is not satisfied, then there is no need to reschedule any trip because overlaps are already zero. Moreover, if the second condition is not met, there is no scope for rescheduling. Even if the *Toolkit* reschedules only one overlapping trip before or after its existing schedule, it will be unable to maintain the minimum headway of δ_{max}. For example, let's consider five consecutive trips from a given origin of a route as t_1, t_2, t_3, t_4, and t_5 with dispatching times as 6:00, 6:05, 6:05, 6:10, and 6:15, respectively. Here, t_3 is overlapping with t_2, thus the headway between these two consecutive trips (i.e., t_2 and t_3) is zero. If t_3 is removed, then headways between the remaining non-overlapping consecutive trips (t_1 and t_4) with t_2 will be 5 min for both the cases. If $\delta_{max} = 5min$, then 2 δ_{max} will be 10 min and headways between the remaining non-overlapping consecutive trips will be less than 2 δ_{max} (i.e., $[max\left(Hn^{ro}_{i-1,i}, Hn^{ro}_{i,i+1}\right) = 5] < 2\delta_{max}$). Thus, t_3 cannot be rescheduled either before or after the dispatch time of t_2 (i.e., 6:05).

Next, the Algorithm 1 presents the detailed steps involved in rescheduling overlapping trips before and after the actual schedules (or dispatch times). The algorithm 1 is only applicable if the aforementioned two conditions are satisfied. Lines 1 – 5 clean the input data and arrange the cleaned data in the required format and input appropriate values of the given parameters. Line 6 estimates headway between consecutive trips. Lines 7 – 14 count the number of overlaps with each trip. Lines 15 – 18 checks, estimate, define, and gather relevant information needed for rescheduling. Lines 19 – 27 present the scheduling algorithm for the scenarios where preceding headway is more than or equal to the succeeding headway. However, Lines 28 – 35 present the scheduling algorithm for the instances where preceding headway is less than the succeeding headway.

Algorithm 1: Rescheduling of overlapping trips

1 • **Get** the cleaned bus schedule data (input to the *Toolkit*).
2 • **Sort** the data according to 'route_number', 'start_point', 'end_point', 'start_time', 'end_time'.
3 • **Group** trips for a given route.
4 • **Define** maximum deviation allowed from the current schedule, $\pm\delta_{max}$
5 • **Define** minimum headway required to avoid overlapping, i.e.,α
6 • **Estimate** headway of consecutive trips $\left(t^{th} \& t+1^{th}\right)$ for each route in the existing schedule, i.e., $H^{ro}_{t,t+1} = \hat{\tau}^{ro}_{t+1} - \hat{\tau}^{ro}_{t}$.
7 **Initialize** $r = 1, t = 1$
8 **Set** Number of overlaps,$\omega = 0, \gamma = t$
9 **If** $H^{ro}_{t,t+1} \geq \alpha$, where α is the minimum headway required to avoid overlapping.
10 **Set** $\omega^{ro}_{t} = 0$
11 **Do** $t \leftarrow t+1$
12 **Else**
13 **Do** $\omega \leftarrow \omega + 1$
14 **Do** $\omega^{ro}_{\gamma} = \omega$
15 • **Check** headway of the preceding and succeeding trips, i.e., $H^{ro}_{t-1,t}$ and $H^{ro}_{t,t+\omega}$
16 • **Estimate** M^{ro}_{t} using expression (1)
17 • **Reschedule** n^{th} trips;$n \in \left(1, 2, 3, 4, \cdots\cdots, M^{ro}_{t}\right)$
18 **Initialize** $k = 1$, $l = 1$; indices to reschedule trips
19 **If** $H^{ro}_{t-1,t} \geq H^{ro}_{t,t+\omega}$ (preceding headway is more than or equal to succeeding headway).
20 **while** $k \leq \lceil M^{ro}_{t}/2 \rceil$, reschedule trip by dispatching ahead of the existing schedule.
21 $\hat{\tau}^{ro}_{k} \leftarrow \hat{\tau}^{ro}_{t} - k\alpha; k \leftarrow k+1$
22 **End while**
23 **If** $l \leq \lceil M^{ro}_{t}/2 \rceil$, reschedule trip by dispatching after the existing schedule.
24 $\hat{\tau}^{ro}_{l} \leftarrow \hat{\tau}^{ro}_{t} + l\alpha;\ l \leftarrow l+1;\ n = k + l - 2$
25 **Elseif** $n + 1 \leq M^{ro}_{t}$, reschedule remaining trips by dispatching ahead of the existing schedule.
26 $\hat{\tau}^{ro}_{n+1} \leftarrow \hat{\tau}^{ro}_{t} + [n + 1 - (l-1)]\alpha;\ n \leftarrow n+1$
27 **Else end if**
28 **Else if** $H^{ro}_{t-1,t} < H^{ro}_{t,t+\omega}$ (preceding headway is less than succeeding headway).
29 **while** $l \leq \lceil M^{ro}_{t}/2 \rceil$, reschedule trip by dispatching after the existing schedule.

(*continued on next page*)

(*continued*)

Algorithm 1: Rescheduling of overlapping trips
$\widehat{\tau}_l^{ro} \leftarrow \widehat{\tau}_t^{ro} + l\alpha$; $l \leftarrow l+1$
If $k \leq \lceil M_t^{ro}/2 \rceil$, reschedule trip by dispatching ahead of the existing schedule.
$\widehat{\tau}_k^{ro} \leftarrow \widehat{\tau}_t^{ro} - k\alpha$; $k \leftarrow k+1$; $n = k + l - 2$
Elseif $n + 1 \leq M_t^{ro}$, reschedule the remaining trips by dispatching after their existing schedule.
$\widehat{\tau}_{n+1}^{ro} \leftarrow \widehat{\tau}_t^{ro} + [n + 1 - (k-1)]\alpha$; $n \leftarrow n+1$
Else end if
End if
End

Table 2
List of notations.

Category	Symbol	Description
Sets	D	Set of existing depots in the PT network
	B	Set of buses available in the network
	V	Set of buses with overlapping trips after rescheduling,$V \subseteq B$
	I	Set of total scheduled trips in the network
	$\aleph$	Set of total trips (non-overlapping trips of the removed buses) to redeploy
	Γ_ϑ	Set of total trips (non-overlapping trips) of ϑ^{th} bus to redeploy,$\Gamma_\vartheta \subseteq \aleph$
	$\beth$	Set of first trip of each removed bus,$\beth \subseteq \aleph$
	C	Set of clusters to redeploy (only those clusters where $x_{c\ell} \neq 0$)
	J_ϑ	Set of possible clusters ϑ^{th} vehicle can cover in a day
	$N[c]$	Set of clusters which are spatially not connected to cluster c
	$T(o)$	Set of overlapping trips at origin o
	$\overline{T}(o)$	Set of non-overlapping trips at origin o
	R	Set of Routes in the network
	$O(r)$	Set of origins at route r
Parameters	φ	Maximum running time of ϑ^{th} bus in a day
	θ	Maximum travel time between two clusters i.e., inter-cluster travel
	$\widehat{\tau}_\ell^\vartheta$	Start time of ℓ^{th} trip of ϑ^{th} bus
	$\check{\tau}_\ell^\vartheta$	End time of ℓ^{th} trip of ϑ^{th} bus
	$\widetilde{\tau}_\ell^\vartheta$	Break time after ℓ^{th} trip of ϑ^{th} bus
	$\widehat{\xi}_c$	Start time of c^{th} cluster
	$\check{\xi}_c$	End time of c^{th} cluster
	$\widetilde{\xi}_c$	Break time after c^{th} cluster
	$\alpha_{c\overline{c}}$	Inter-cluster time from cluster c to cluster $\overline{c} \neq c$
	$H_{t,t+1}^{ro}$	Headway between consecutive trips t and $t+1$ at the origin o of route r
	$Hn_{i-1,i}^{ro}$	Headway between non-overlapping consecutive trips $i-1$ and i at origin o of route r
	$Hn_{t,t+\omega}^{ro}$	Headway between non-overlapping trips t and $t+\omega$ at the origin o of route r
	$\widehat{\tau}_t^{ro}$	Start time of t^{th} trip on route r, at origin,o
	δ_{max}	Maximum deviation allowed in the start time of a scheduled trip
	α	Minimum safety headway required to avoid overlap
	ω_t^{ro}	Number of overlaps with t^{th} trip at origin o of route r.
	M_t^{ro}	Number of overlapping trips removed with t^{th} trip at origin o of route r after rescheduling
	$O(.)$	Function that represents the origin of route r
	$T(.)$	Function that represents the total trip at origin o of route r
	$\overline{T}(.)$	Function that represents the total non-overlapping trip at origin o of route r
Variables	$x_{c\ell}$	Binary decision variable, equals 1 if t^{th} trip is assigned to c^{th} cluster, otherwise 0.
	y_c	Binary decision variable, equals 1 if c^{th} cluster used, otherwise 0
	$z_{\vartheta jc}$	Binary decision variable, equals 1 if c^{th} cluster is assigned to ϑ^{th} bus in its j^{th} trip-leg, otherwise 0.
	$Y_{\vartheta c}$	Binary decision variable, equals 1 if ϑ^{th} buses assigned to c^{th} cluster, otherwise 0
	$X_{\vartheta j}$	Binary decision variable, equals 1 if ϑ^{th} buses assigned to their j^{th} trip-leg, otherwise 0.

The rescheduling step has two major benefits from the perspective of PT agencies: it ensures minimum scheduling disruptions and significantly reduces the computational burden in the redeployment step (detailed in Section 4.2). However, as explained in the previous example, this step may not guarantee the 100 % elimination of overlaps in all the scenarios. The efficiency of the rescheduling algorithm depends on how much leeway we have before and after overlapping trips. There are two scenarios from expression (1): first, $M_t^{ro} = \omega$, and second, $M_t^{ro} < \omega$. If the first scenario is true, then rescheduling alone is sufficient to address all overlapping issues. However, that cannot be ensured in every instance, and overlaps are present even after rescheduling. For such cases ($M_t^{ro} < \omega_t^{ro}$), this work develops mathematical formulations and efficient solution algorithms in the next Section. The next step only applies to a limited set of buses which are still involved in overlaps.

4.2. *Redeploying excluded trips*

If there are still overlapping trips after rescheduling (i.e., $M_t^{ro} < \omega_t^{ro}$), the *Toolkit* will first remove all such overlapping trips. However, during the removal of the overlapping trips, non-overlapping trips will also be removed since all the trips of a bus which is involved in overlap(s) are eliminated. Thus, we need to re-deploy buses to the non-overlapping trips which must be present in the system. The *Toolkit* does this process efficiently using the following steps.

For removing the overlaps, the *Toolkit* keeps only the bus with a maximum number of trips per day and removes all other buses from the existing schedule. After the removal of trips, the efficient redeployment of non-overlapping removed trips is done in two sub-steps: a) formation of clusters and b) efficient assignment of clusters to buses. For this, the work develops two mathematical Models M1 and M2. The following assumptions are made for formulation:

- It is assumed that scheduled service trips are given and have a predetermined and fixed start time, end time, and running time.
- The capacitated depot and the capacitated bus are assumed, and demand does not exceed the fixed capacity of bus and depot during the assignment of service trips.
- A homogenous diesel bus fleet is assumed in this work.
- Each bus starts from a given depot and returns to the same depot.
- The model uses all predetermined parameters that remain constant throughout formulation.
- Given traffic situation and demand are supplied as input to the model.
- The analysis of optimizing operational efficiency and financial performance limits the scope of the problem to tactical planning. Stochastic and dynamic fluctuations in service are not considered in the current formulation.

4.2.1. *Formation of clusters*

We need to form the cluster(s) for the trips of each bus since, otherwise, dealing with each trip separately will significantly increase the computational complexity of the problem. A cluster is defined as a continuous sequence of a set of trips of a bus (ϑ), where the start time of the next trip ($\widehat{\tau}_{\ell+1}^\vartheta$) is equal to the sum of the end time of the previous trip ($\check{\tau}_\ell^\vartheta$) and the break time of the previous trip ($\widetilde{\tau}_\ell^\vartheta$), i.e., $\widehat{\tau}_{\ell+1}^\vartheta = \check{\tau}_\ell^\vartheta + \widetilde{\tau}_\ell^\vartheta$. If there is any discontinuity (i.e., $\widehat{\tau}_{\ell+1}^\vartheta > \check{\tau}_\ell^\vartheta + \widetilde{\tau}_\ell^\vartheta$) in the trip sequence of a given bus due to removal of the overlapping trips, then there will be more than one clusters for the trips of a particular bus. Next, we present the mathematical model to form the minimum number of clusters corresponding to the non-overlapping trips of the removed buses.

Model M1: Cluster Formation Model

$$Min\sum_{c=1}^{\aleph} y_c \tag{2}$$

Subject to

$$y_c \geq x_{c\ell} \quad \forall c \in \aleph, \ell \in \aleph \tag{3}$$

$$\widehat{\tau}^{ro}_{\ell+1} \bullet x_{c\ell+1} + M(1 - x_{c\ell+1}) \geq \left(\check{\tau}^{rd}_{\ell} + \bar{\tau}^{\bar{r}d}_{\ell}\right) \bullet x_{c\ell}$$
$$\forall c \in \aleph, \ell \in \aleph, r \in R, \bar{r} \in R, o \in O(r), d \in D(\bar{r}), o(\ell+1) = d(t) \tag{4}$$

$$\sum\nolimits_{c=1}^{\aleph} x_{c\ell} = 1 \quad \forall \ell \in \aleph \tag{5}$$

$$\widehat{\xi}_c = \min_{\ell \in \aleph}\left(\widehat{\tau}^{ro}_{\ell} \bullet x_{c\ell}\right) \quad \forall c \in C r \in R, \bar{r} \in R, o \in O(r) \tag{6}$$

$$\check{\xi}_c = \max_{\ell \in \aleph}\left(\check{\tau}^{rd}_{\ell} \bullet x_{c\ell}\right) \quad \forall c \in C r \in R, \bar{r} \in R, o \in O(r), d \in D(\bar{r}) \tag{7}$$

The objective of the model is presented in expression (2), i.e., to minimize the number of clusters for the trips to be redeployed. Constraint (3) ensures that even if one trip is assigned to a cluster then also that cluster must be created. This tries to add more and more trips to a given cluster so that the number of clusters can be reduced. Constraint (4) ensures that a new cluster must be formed if the vehicle serving the ℓ^{th} trip cannot serve the $\ell + 1^{th}$ trip due to time and space connectivity. Either the vehicle is unable to reach on time at the start point of the $\ell + 1^{th}$ trip or the start point itself is different from the end point of the ℓ^{th} trip. Constraint (5) ensures that each trip is assigned to one cluster only. Finally, the constraints (6) & (7) store the start and end time of a cluster, respectively.

4.2.2. Optimum assignment of clusters to buses

Once the clusters are formed, we need to assign clusters to buses optimally so that all clusters can be served by the minimum number of buses. Thus, the objective function of the model is to minimize the total number of buses required to complete all the clusters (expression (8).

Model M2: Assignment Model of clusters

$$Min \sum_{\vartheta \in V} X_{\vartheta 1} \tag{8}$$

Subject to

$$\sum_{\vartheta \in V} \sum\nolimits_{j=1}^{J_\vartheta} Z_{\vartheta jc} = 1 \quad \forall c \in C \tag{9}$$

$$\sum_{c \in C} Z_{\vartheta jc} \leq 1 \quad \forall \vartheta \in V, j = \{1, 2, .., J_\vartheta\} \tag{10}$$

$$\sum_{c \in C} \check{\xi}_c \bullet Z_{\vartheta j+1c} - \sum_{c \in C} \widehat{\xi}_c \bullet Z_{\vartheta 1c} \leq \varphi \quad \forall \vartheta \in V, j = \{1, 2, .., J_\vartheta\} \tag{11}$$

$$\check{\xi}_c \bullet Z_{\vartheta jc} \leq \left(\widehat{\xi}_{\bar{c}} - \alpha_{c\bar{c}}\right) \bullet Z_{\vartheta j+1\bar{c}} + M \bullet \left(1 - Z_{\vartheta j+1\bar{c}}\right) \quad \forall \vartheta \in V, j = \{1, 2, .., J_\vartheta\}, c \& \bar{c} \in C, \bar{c} \neq c \tag{12}$$

$$X_{\vartheta j} = \sum_{c \in C} Z_{\vartheta jc} \quad \forall \vartheta \in V, j = \{1, 2, .., J_\vartheta\} \tag{13}$$

$$Y_{\vartheta c} = \sum\nolimits_{j=1}^{J_\vartheta} Z_{\vartheta jc} \quad \forall \vartheta \in V, c \in C \tag{14}$$

$$X_{\vartheta j} \geq X_{\vartheta j+1} \quad \forall \vartheta \in V, j = \{1, 2, .., J_\vartheta - 1\} \tag{15}$$

$$Z_{\vartheta jc} \leq 1 - \sum\nolimits_{c' \in N[c]} Z_{\vartheta j+1c'} \quad \forall \vartheta \in V, c \in C, j = \{1, 2, .., J_\vartheta\} \tag{16}$$

Expression (8) minimizes the number of buses assigned to their first trip-leg. Trip-leg can be defined as sequence of clusters assigned bus has to complete after redeployment. Constraint (9) ensures that every cluster must be assigned only to a single bus. Constraint (10) ensures that only one cluster is served by a bus on its j^{th} leg. Constraint (11) ensures that each bus can travel for a maximum of φ hours from its starting time. Constraint (12) ensures that a bus, ϑ, can only serve a set of clusters until time connectivity is ensured. M in equation is used for larger positive number. Constraints (13) & (14) are definitional constraints for vehicle to trip-leg and vehicle to cluster assignments, respectively. The last constraint (15) makes sure that the bus will always make its first trip before making its second or subsequent trips. Constraint (16) is used to make sure that two clusters are only assigned to the same bus if they are spatially connected. This means that if the destination of the last cluster $c \in C$ is not the same as the source of the next cluster $c' \in C$, there should be a time gap between the start time of cluster $c' \in C$ and the end time of cluster $c \in C$ equal to the deadhead time required to cover the distance between the clusters.

5. Case study

The city of Bengaluru (earlier known as Bangalore) presents a representative case for urban bus systems in developing countries. Bengaluru has a population of more than 12 million inhabitants spread across the municipal area of 741 square kilometers and the urban agglomeration around it. Bengaluru Metropolitan Transport Corporation (BMTC) is used as a case study to develop the solution into an optimization toolkit that can be deployed in other cities as well. BMTC is India's largest urban public bus operator with a fleet of 6,700 buses, including 859 Air-Conditioned (AC) and 5,833 Non-AC buses. These buses operate from 45 depots and are spread across 2,203 routes covering the Bengaluru Metropolitan area providing 1.1 million daily service-km catering to a daily ridership of 3.5 million passengers. In addition to the details of ridership and service kilometers which is mentioned in Section 1, Table 3 summarizes the other service-related parameters from BMTC schedule.

Despite having such a large operation, BMTC's planning and scheduling practices are predominantly manual in nature. BMTC centrally determines route-specific frequency based on travel demand patterns and policy headways. Each depot independently schedules its buses within its jurisdiction, resulting in overlapping trips. It is necessary to co-ordinate timetables across several depots to ensure consistency in passenger service delivery is necessary to optimize service provision and maximize ridership. The issues faced by BMTC are symptomatic of other bus systems in developing countries with similar analytical constraints. The following sub-Sections first define the schedule format of BMTC, and then highlight the issue of overlapping trips in the existing schedule.

5.1. Existing bus schedule (Form-IV)

For identifying the extent of overlapping issues in the current scheduling practices, we use existing scheduling information of BMTC. Throughout the entire manuscript, we refer to their bus schedule as "Form-IV" (as designated by BMTC). Form IV contains data regarding routes, stops, trips, and services. Table 4 summarizes scheduling and network related information from BMTC Form-IV.

A typical Form-IV contains 20 columns, each representing the different attributes, and each row represents one trip of a bus. Some of the key attributes that we use in this work are given in Table 5 and their details in Appendix A1.

Table 3
Details of BMTC schedule (BMTC, 2023).

Description	Details
Number of buses	~6700
Number of daily trips	~50,000
Number of geographic routes	2203
Number of depots	45
Road network coverage	2522 km
Service km per day	1.1 million-km
Traffic revenue per day	35 million rupees

Table 4
Summary of scheduling and BMTC network details.

Sr No	Description	Count
1	Total number of trips	99,099
2	Total route numbers	9,993
3	Total schedule ids	6,247
4	Number of depots	45

As shown in Table 4, the dataset contained 9,993 number of routes. However, it is already known that the city contains 2,203 geographic routes. Therefore, it indicates that data cleansing is necessary to conduct a meaningful analysis. Further, it is identified that the dataset also contains numerous other data anomalies, such as manual typing errors, route matching, and dead trips (i.e., non-revenue generating trips), which must be addressed. The details of all such issues are explained next. Additionally, the raw dataset contained numerous manual typing errors, such as incorrect end-time calculations and misspelled route numbers. We eliminated all these inaccuracies and ensured consistency in the end time for each trip using the running time and the start time. Similarly, in some instances, the same schedule ID is assigned to two buses that are physically distinct. For this identified case, we created new separate schedule IDs for different buses. Further, in Form-IV, the majority of daily bus trips involve multiple routes. In order to switch between routes, buses make additional short trips. These trips are identified as having a duration of less than 30 min (for AC) and 15 min (for Non-AC). This trip is represented as "Dead Trip" in the schedule as shown in Table 5. If it is a non-revenue-generated trip, then there will be a "1" in the "Dead Trip" column, and if it is a revenue-generated trip, then there will be a "0" in the "Dead Trip" column. The information for dead trips is provided by the agency, so it is important to remove this trip from the analysis to avoid any resource assignment to this trip, as it does not generate any revenue. The rational for choosing different dead trip duration for AC and Non-AC is as follows: AC buses predominantly serve longer routes, including suburban routes, airport routes, and routes connecting employment centres, which have 50–75 km route lengths and an average travel time of around 02:30 h. While non-AC buses serve relatively shorter routes within city boundaries with 10–30 km route lengths and an average travel time of 01:15 h. In consultation with BMTC, it is decided that any trip less than 20–25 % of the route length is represented as "Dead Trip" in the schedule and the actual trip duration (revenue) trip time is different than this dead-trip time. Based on this recommendation, 15 min for non-AC and 30 min for AC bus criteria is considered. However, the developed models are generic enough to take other time as well based on the chosen scenario.

To ensure that the bus departs from and returns to the same depot, dead trips are listed as the first and last on the schedule. We excluded these trips from our analysis because they did not generate revenue and there is no involvement of overlapping in such dead trips. These trips are added to the improved schedule based on the requirement. Moreover, Form-IV includes variously scheduled weekday, weekend, and holiday bus trips. In this classification, some buses have the same daily schedule. Consequently, the itinerary for only all the weekly trips is optimized. After cleaning the dataset, it is then necessary to divide the dataset into trips with and without AC buses. This division is required to ensure that after the redeployment step, all buses AC trips must be completed by AC buses only and vice versa. In the Form-IV, AC and Non-AC trips are distinguished by a V or KIAS code in the *schedule-id* field, while it is left blank for trips by Non-AC buses.

5.2. *Issue of overlapping trips*

From the cleaned Form-IV, it is observed that overlapping is indeed a critical issue causing operational inefficiencies. In the Form-IV, there are 725 AC buses servicing 5,223 trips per day. Out of these trips, around 9 % of trips (451, to be exact) are overlapping. Similarly, in the case of Non-AC trips, there are a total of 5,446 buses, servicing 60,858 trips per day and around 6 % (3,604 to be precise) are found to be overlapping trips. It is also important to determine the distribution of these overlaps over the day. Fig. 3 illustrates time-based overlaps for AC and Non-AC buses. The Fig. 3 reveals that the greatest number of overlaps occur between 11 AM and 4 PM, which is not even a peak hour band (where we would not expect the most commuters). It clearly shows the need for interventions. During the analysis it is further observed that these overlaps are the results of poor planning rather than high demand for the service. The route- and depot-specific overlapping analysis is also performed on the cleaned Form-IV to find overlapping trip hotspots. The top ten routes and the number of overlaps present on each route are listed in Table 6. Further from Table 6, it is observed that around 57 % of the trips on route 500-D are overlapping.

Through the depot-wise analysis, the functional characteristics of each depot are depicted in Fig. 4. It is observed that each depot has an average bus capacity of 130. Hence operating on an average of 1500 trips per day. It should also be noted that each depot runs an average of 270 different routes throughout the city on a given weekday. In overlapping analysis of depots, Fig. 5 shows the number.

of overlapping trips for each depot. The Fig. 5 reveals that in some depots, number of overlapping trips are even more than 200, hence causing numerous system inefficiencies.

It is evident from the case study that a significant proportion of the trips are overlapping, resulting in system inefficiencies. Technological intervention is necessary to determine a data-driven and automated optimized approach to meet the daily operations planning is needed.

6. Results and discussion

The results of BMTC's case study are presented in this Section. The two-step approach of the toolkit is applied effectively to manage the overlapping trips. While applying the toolkit, it is preferred to

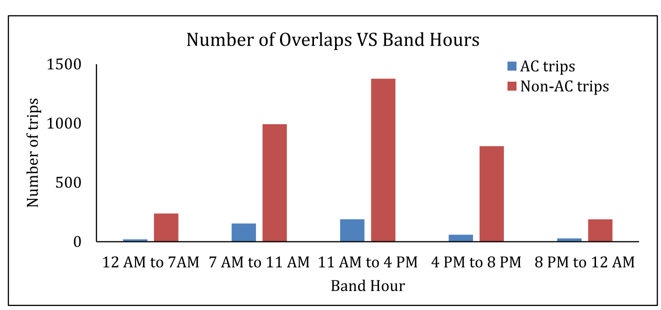

Fig. 3. Hourly band wise number of overlaps in existing schedule.

Table 5
Form-IV snippet with trip details.

Schedule id	Day type	Route number	Start point	End point	Start time	End time	Running time	Dead trip
10 J/5	All Days	10-J	160	1373	05:30:00	06:00:00	00:30:00	0
10 J/3	All Days	10-J	160	1373	08:15:00	09:00:00	00:45:00	0
10 J/2	All Days	10-J	160	1373	08:45:00	09:30:00	00:45:00	0
10 J/2	All Days	10-J	1373	145	09:45:00	09:50:00	00:05:00	1
10 J/3	All Days	10-J	160	1373	09:55:00	10:40:00	00:50:00	0

Table 6
Top 10 routes with their overlaps.

Route number	Number of overlaps	% Overlaps
500-D	345	56.4 %
600-F	238	51.9 %
401-M	222	39.8 %
V-500CA	211	44.7 %
V-500D	176	46.1 %
356-CW-FLY	155	44.7 %
V-335E	154	30.2 %
290-E	153	40.5 %
378	152	38.7 %
252	143	44.7 %

reschedule the overlaps as much as possible so that the revised schedule will be as close as possible to the existing one.

As discussed in Section 4, we develop a desktop-based Toolkit to eliminate the overlapping trips from the existing schedule. This PT Toolkit serves the purpose of both the dashboard and as a decision support system. The Toolkit is developed using python language. Different python packages, including NumPy, Pandas, Datetime, Bisect, and Time are used, and the graphical user interface of the Toolkit is coded using the streamlit module. The Toolkit has three different tabs: the 'Home' tab, the 'Results' tab and the 'Analysis' Tab.

All the required input file and parameter-related information are provided in the 'Home' tab (i.e., a landing page). As shown in Fig. 6, the 'Home' tab has four primary inputs: (i) selecting the existing schedule (ii) specifying the rescheduling parameter of δ_{max}, (iii) specifying the maximum running time of a bus in a day φ and lastly, (iv) specifying the inter-cluster travel time θ. After inputting these parameters, simply clicking on the *solve* button on the screen will run the developed models and algorithms. As an output, a revised schedule without any overlapping trips is generated. The console also shows the steps in progress, i.e., Rescheduling, Removing overlaps, and Redeployment of non-overlapping trips. In the 'Results' tab, the critical results, including the total number of overlaps removed; the number of buses saved; and the number of additional trips that can be done using saved buses, are displayed. Also, in this tab, we can download the output schedule, i.e., the optimized schedule zero overlaps, and the excel file containing the list of all the saved buses. Lastly, in the analysis tab**,** depot-wise analysis of the number of buses before and after optimization, distance, and time

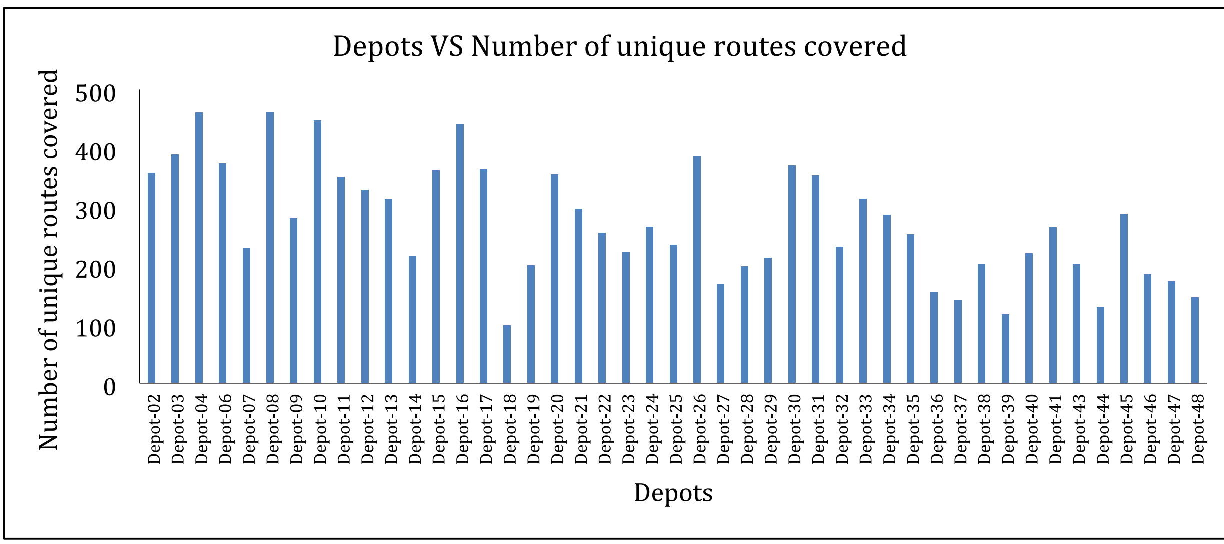

Fig. 4. Depot-wise details routes covered in existing BMTC schedule.

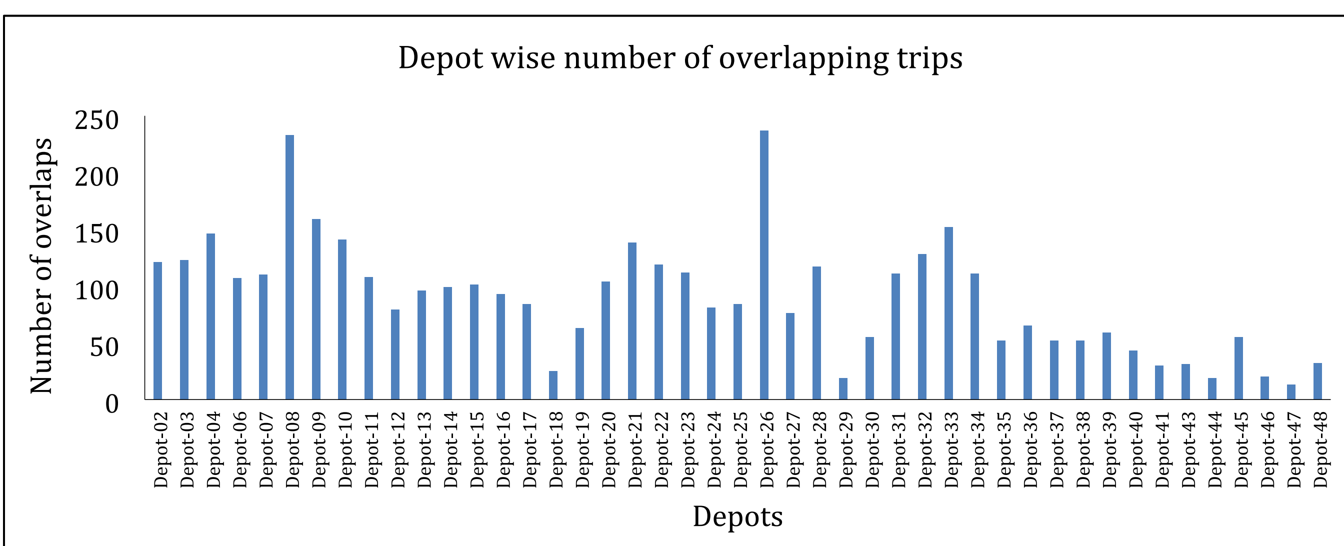

Fig. 5. Depot-wise details overlapping trips in existing BMTC schedule.

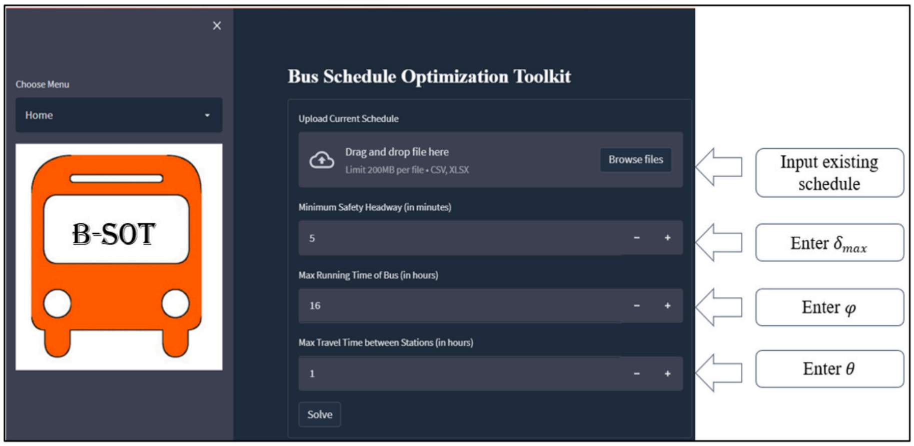

Fig. 6. The graphic user interface of the desktop-based toolkit.

comparison is done. The results of individual tabs will be explained in sub-Section 6.2 and 6.3.

6.1. Rescheduling

Recall from Section 4 that the first step of the Toolkit is to perform rescheduling to reduce the overlapping to the possible extent. Following the steps of Algorithm 1, based on consultation with the BMTC, appropriate values are assigned to the parameters. Value of minimum safety headway δ_{max} is kept at 5 min. Further, only the last overlapping trip is rescheduled out of all the overlapping trips. Upon application of Algorithm 1, it is observed that the number of overlapping trips decreased significantly for both AC and Non-AC buses. As depicted in Fig. 7, the number of overlaps decreases from 451 to 283, representing a 38 % decrease in total AC trip overlaps. The value of $|V|_{AC}$, i.e., number of buses with overlapping trips after rescheduling is 194. The same procedure is applied to Non-AC dataset as well. Fig. 8 reveals that the number of overlapping Non-AC trips is reduced from 3,604 to 1,593, which is 56 percent of the total number of Non-AC overlaps and $|V|_{non-AC}$ is 853. The remaining overlapping trips are then removed from the existing schedule and all those trips given as input to Model M1 for the formation of clusters.

6.2. Redeployment of excluded trips

To further remove the overlapping trips, Model M1 (for formation of clusters) and M2 (for optimal assignment of clusters to buses) are applied on modified Form-IV after rescheduling. The results of these models are presented next.

Model M1: the strategy is already defined in Section 4 to keep only the bus with the maximum number of trips in a day. The number of excluded AC and Non-AC bus are 194 and 853, respectively. Following the first step, clusters are formed using Model M1, the value of $\sum_{c=1}^{\aleph} y_c$, i.e., minimum number of clusters formed is 455, and the number of redeployment trips $|\aleph|_{AC}$ is 1,018 for AC buses. Similarly, for Non-AC, the number of clusters formed using Model M1 is 2,149, and $|\aleph|_{non-AC}$ is 7720. Fig. 9 shows cluster-wise frequency distribution of $\sum_{\prime} x_{c\prime}$ (i.e., total number of trips). It is observed that around 500 clusters have only two trips, while around 50 clusters have nine trips in them.

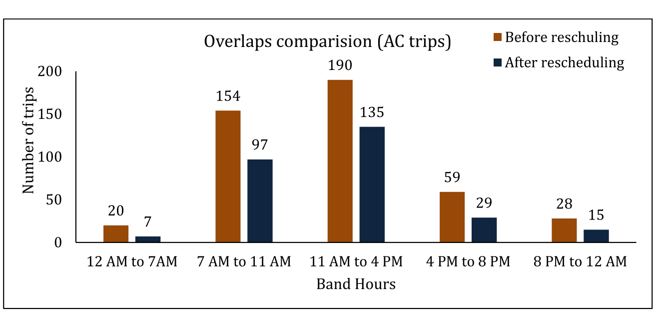

Fig. 7. Hourly band-wise overlap distribution after rescheduling (AC trips).

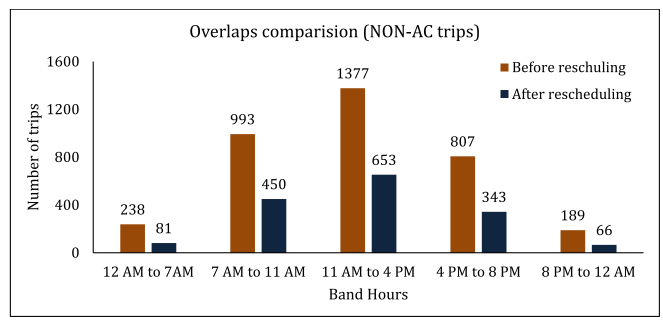

Fig. 8. Hourly band wise overlap distribution after rescheduling (Non-AC trips).

After clusters formation, Model M2 determines the optimal assignment of buses to each cluster. Fig. 10 presents the solution approach to solve model M2. Each step of this solution methodology is explained next. The first step in solving model M2 is to determine values of all the parameters. Proper consultation with BMTC has been done to set the values of all the parameters. For instance, φ and θ are set to 960 mins (16 h) and 60 mins (1 h), respectively. Recall that these parameters are needed to ensure that the start time of the next cluster must be greater than the end time of the previous cluster and inter-cluster travel time (θ).

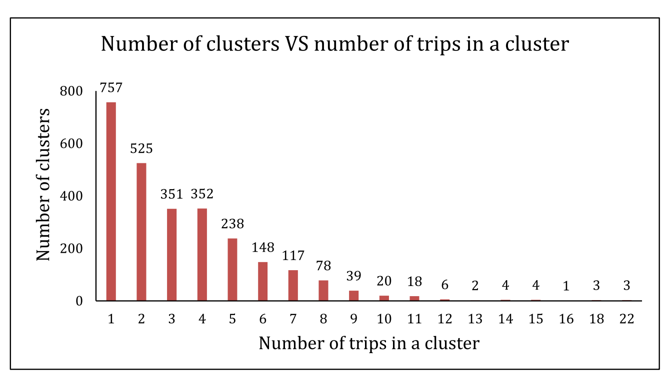

Fig. 9. Distribution of cluster-wise number of trips (AC + Non-AC).

The rational for setting $\varphi = 960\,\text{mins}\,(16\,\text{hours})$ is that the total running time of a bus must be less than 16 h (as provided by BMTC). Similarly, the rationale for setting $\theta = 60\,\text{mins}\,(1\,\text{hour})$ is to keep the complexity in calculation of inter-cluster times under check. As BMTC has a wide route network, calculating inter-cluster time is computationally complex. Due to the unavailability of the exact inter-cluster time (required to go from one stop to any other stop) with BMTC, we take the average running time of the complete network. However, we highlight here that our models are capable of taking any value of θ provided by the agency.

In order to make model M2 computationally tractable, it is also required to determine the range of variable j. If j is not calculated, the model will take the value of j equal to the total number of clusters formed i.e., $J_\vartheta = 455$ for each AC vehicle ϑ. This leads to the creation of additional decision variables as well as an increase in the number of constraints, which will further increase the complexity of the problem. Therefore, it is important to find the upper bound of j. We have used a greedy approach to find the upper bound of j, that intelligently reduces the search space, without leaving any feasible solution space. In this approach we find a good upper bound for j. To do so, we need to monitor the progression of a given bus in a day. The following procedure is adopted to determine the upper bound: A bus starts with the cluster where the running time is minimum (say $\propto$ hours) among all other clusters. After completing all the trips in that cluster, it will move to the next cluster within a maximum of θ hours. This bus will next keep on going to the same cluster throughout its total shift time of φ hours. The upper bound of j will be then the ratio of total shift time to the time of completing the cluster plus inter-cluster time: $j \leq \frac{\varphi}{(\theta+\propto)}$. In case of the AC trips, the value of j comes out to be 11, as $\propto = 30mins$.

After setting these values, it turns out that the total number of variables generated for AC trips are 2,277,275. To solve the model M2, we used IBM CPLEX on a machine with 32 GB of RAM and an Intel Xeon E3 processor. The M2 model is applied to specific instances of the problem. We iteratively increased the number of clusters in order to obtain Model M2′s results. The reason for considering such instances is because the problem is not tractable using exact methods for large instances. The results obtained from Model M2 show that 47 buses are required to complete all non-overlapping trips for 50 clusters, and 97 buses for 100 clusters. Table 7 contains the outcomes of other instances.

However, as expected, Model M2 is incapable of providing a solution for the entire dataset using the exact solution approach. The computer ran out of memory while attempting to store all the generated variables and solve all the constraints. Therefore, we devise a heuristic (explained in Fig. 10) to determine the minimum number of buses required to cover all non-overlapping trips.

During the design of this heuristic, it is ensured that all constraints of model M2 are satisfied, and the results are found in acceptable time. The procedure in this heuristic involves sorting all clusters by their start and run times. Then, a bus is assigned to the first cluster selected. Then, it checks for the subsequent cluster that can be assigned to the same bus while meeting all constraints of the Model M2. For instance, if a bus is assigned to cluster c, then it is needed to algorithmically ensure that it can be assigned to any other $\bar{c}$, if and only if the start time of $\bar{c}$ is greater than the sum of end time of c and inter cluster time θ. While doing these assignments, it is also ensured that buses are running for φ hrs or lesser.

To demonstrate the efficacy of this heuristic, we test it on the same scenarios with varying numbers of clusters from the AC dataset. Table 7 presents a comparison analysis between model M2 and heuristic algorithm. It is evident from Table 6 that for a small number of clusters, model M2 and algorithm produce identical results. However, after 207 clusters and more than 48 h of waiting, the model M2 is unable to

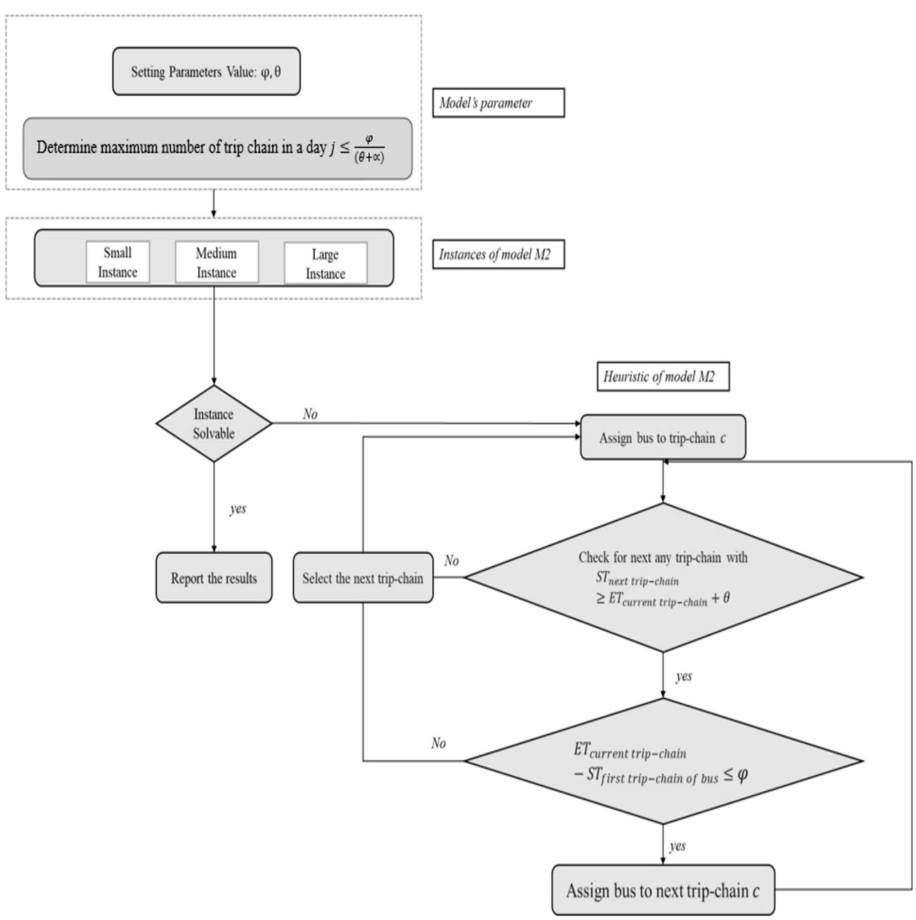

Fig. 10. Flowchart for assigning buses to clusters after redeployment.

Table 7
Comparison between results of model M2 and the heuristic algorithm.

Model/ Heuristics	Model M2				Heuristic algorithm	
Number of clusters	Number of buses from model M2	Number of constraints	Number of variables	Calculation time w/o heuristic	Calculation time with heuristic algorithm	Number of buses from algorithm
50	47	27,500	26,600	4.49 sec	391 milli sec	47
100	97	110,000	103,200	41.2 sec	1.45 sec	97
150	141	247,500	229,800	2 min 46 sec	2.87 sec	141
200	188	440,000	406,400	7 min 51 sec	5.01 sec	188
205	191	462,275	426,810	9 min 43 sec	5.23 sec	191
207	193	471,339	435,114	9 min 45 sec	5.48 sec	193
≥210	–	≥485,100	≥447,720	No results even after 2 days	5.55 sec	193

provide any solutions. As stated earlier, the Model M2 itself is incapable of providing the exact solution for the entire dataset. The heuristic is, however, applied to the entire set of AC and Non-AC trips. To complete the re-deployable trips, the heuristic provides that only 183 buses (out of 194 excluded buses) will be required to satisfy all of the model M2′s specified constraints. This saves eleven AC buses. Similarly, this heuristic when applied to Non-AC buses utilizes 759 Non-AC buses, resulting in the saving 94 Non-AC buses.

6.3. *Analysis tab of toolkit*

After redeploying the AC and Non-AC trips, comparison of improved with existing schedule is done based on key performance indicators (KPIs) such as distance travelled per day, number of trips per day, and servicing time per day. The median distance travelled by buses decreases up to 8 % after redeployment (Fig. 11). Similarly, there is also up to 11 % decrease in average distance travel by buses after redeployment. If we talk about the other KPIs then there is decrease of 10 % in number of trips per day (Fig. 12) and average running time of buses by more than 11 % (Fig. 13). After redeployment, no bus is servicing more than 22 trips and 16 h per day. This results in saving of additional operational hours or running time. Whereas, in the existing schedule 12 buses were running beyond the recommended operational time (i.e., 16 h).

Further, the work highlights that the maximum number of buses saved per depot is 6 and in total there is a saving of more than 60 buses for 13 depots together. The detailed statistics is presented in Table 8. The number of buses saved per depot reflects how inefficiently the particular depot was operating its buses.

The format of the final output of the toolkit is indeed similar to that of the initial schedule (FROM-IV). An example of this is shown in Table 9 below, where two trips from start-point 5848 to end-point 8456 start exactly at the same time, i.e., 06:50 and another two trips start at the same time, i.e., 06:55 AM. Based on the approach mentioned in the methodology, schedule number 252F/24 is excluded from the existing schedule and schedule number 252F/8 is rescheduled at 07:00 AM. The updated start time and end time are added to the final schedule. Similarly, the other trips of bus number 252F/24 are also excluded from the schedule.

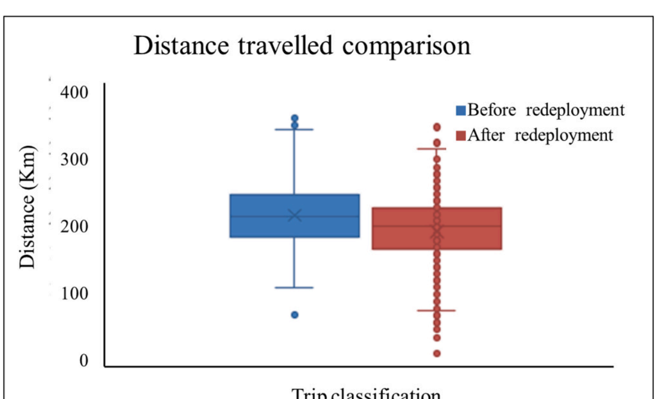

Fig. 11. Distance travelled comparison.

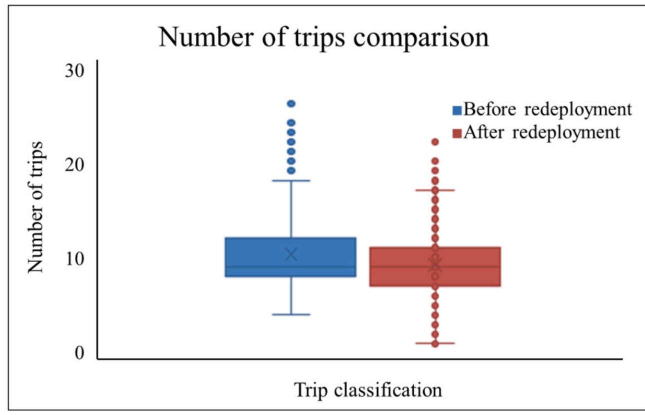

Fig. 12. Number of trips comparison.

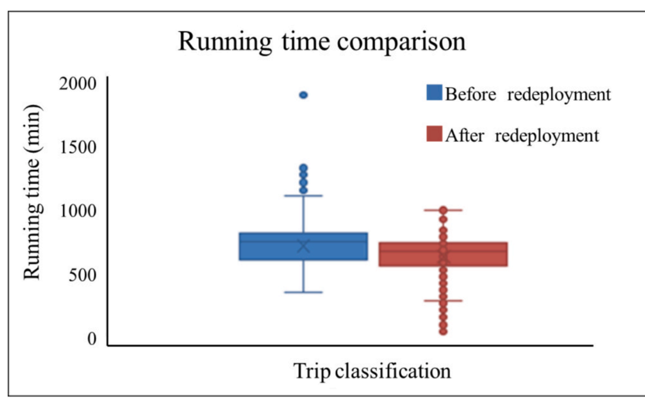

Fig. 13. Running time comparison.

Table 8
Depot wise details of saved buses.

Number of saved buses	Number of depots	Depot number
0	11	All remaining
1	6	02, 15, 17, 18, 35, 36
2	7	13, 21, 22, 31, 34, 39, 40
3	8	04, 06, 11, 19, 32, 33, 37, 45
4	7	07, 08, 23, 24, 27, 28, 38
5	3	10, 14, 25
6	3	09, 12, 26

Moreover, it is also ensured that rescheduling and redeployment will not increase the headway and thus will improve the customer experience and level of service. Infact, the overall methodology is designed to either achieve reduction or keep the headway the same. In particular, during rescheduling the overlapping trips, these are either shifted in a forward or backward direction with a rescheduling condition (e.g. $H^{ro}_{t-1} or H^{ro}_{t} \geq 2\delta_{max}, where H^{ro}_{t} = \hat{\tau}^{ro}_{t+1} - \hat{\tau}^{ro}_{t}$). If rescheduling condition is met, then trips are rescheduled with a minimum δ_{max} amount of time. Further this will reduce the maximum headway either in the forward direction or in the backward direction by δ_{max}, that help in reducing the passenger's waiting time at the bus stops. Also, if this condition is not met, then redeployment steps are followed. During this step, one of the overlapping buses with the maximum number of trips per day remains in schedule, and the rest of the overlapping buses are removed with their schedule and redeployed. So, while redeploying the non-overlapping trips, the timetable of trips remains the same and does not affect the existing headway between two consecutive trips. The Fig. 14 illustrates this by comparing the existing schedules and output schedule generated by developed methodology in this work.

From Fig. 14, it can be seen that overlapping trips (zero headways) from the initial schedule are removed in the output by either rescheduling or redeploying, and also it can be observed that higher headways decreased in the output by shifting the overlapping trips either in the forward or backward direction. The diagram above indicates that the developed methodology won't have any negative impact on waiting times, duration, or passenger experience. Overall, the results of the case study reveal that removing overlaps not only adds monetary benefits to operators but can also be beneficial for users if saved buses are used on other routes. Thus, frequency on other routes will increase and consequently the less waiting times for users.

In terms of monetary benefits to BMTC, there can be an additional potential revenue of 4.2 million USD per year after implementation of the toolkit. Following parameters are assumed to estimate the additional revenue: i) 1 USD = 70 Indian Rupees, ii) daily average distance travel by buses is 200 km, iii) operational cost per bus per km is rupees 50, iv) total working days in a year is 300, and v) total number of buses saved is 105. The PT Toolkit can optimize a large and complex network with all the trips of the PT agency, as there is no restriction on the number of trips to handle. It cleans the data and converts it into a standardized format. The Toolkit can be customized based on the PT agency's requirements. As an output, it gives an optimized timetable in the exact same format as of the input file. It also gives the file containing the list of saved buses and a schedule and route-wise analysis containing before and after comparison of trips. Overall, the Toolkit can be used by any public transportation agency after contextual modification to the existing schedule, which can vary from agency to agency.

7. Conclusions and operational implications

The paper presents a data-driven bus schedule optimization toolkit that analyses the existing bus services in cities and recommends a revised schedule that improves the service level with the least operational cost. We analyze the existing bus services in a large-scale bus

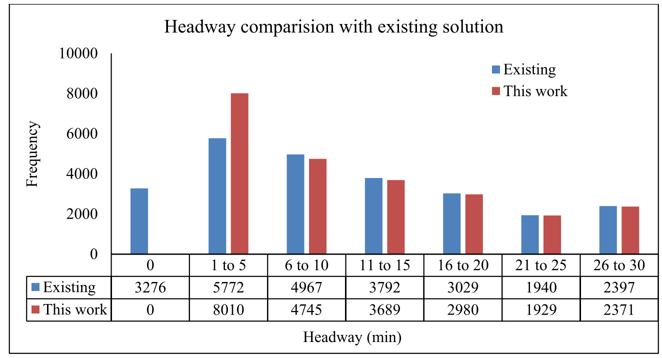

Fig. 14. Headway comparison with the existing solution.

network like the BMTC services in Bengaluru to identify that overlapping trips is a key concern which was found to be a potential issue even in other cities and developed a novel methodology to address the issue. An optimized schedule is developed by eliminating the overlaps, i.e., trips between an origin and destination pair starting at the same time. We propose an innovative approach to first adjust the start time of overlapping trips by a few minutes where feasible and redeploy the remaining overlapping trips and their buses.

We present two MILP models, one to minimize the number of clusters of non-overlapping trips, and the second model to minimize the number of buses required to satisfy the cluster demand. The proposed models and algorithms offer a modular approach that is easily adaptable to the needs of decision-makers. The models are unique in the fact that they intelligently find the solution to such a complex problem in a realistic time. Moreover, the models are data-driven optimization models that seek to determine the minimum number of buses, which can in turn provide both operational and financial benefits. Regarding applicability, the models are suitable to both developing and developed countries. Additionally, the models are applicable to battery-powered electric buses with minimal contextual modifications. Further, to demonstrate the effectiveness of these models and algorithms, they are applied to the 6,700-fleet BMTC network, which provides over 1.1 million bus-km of service every day. The overlaps in the existing schedule are completely eliminated and thus there are zero overlaps in the improved schedule. For the case of the AC buses, the number of overlapping trips is reduced from 451 to 0. This demonstrated the effectiveness of the Toolkit while ensuring that the resulting scheduling doesn't disrupt the existing schedule too much. Similarly, the number of overlaps for Non-AC buses was reduced from 3,950 overlapping trips to 0, resulting in the saving of 94 Non-AC buses. Due to elimination of overlapping trips, operational expenses will decrease. Further, more revenue can be generated with the saved buses as there is a saving of both AC as well as Non-AC buses. These saved buses can be used on other routes where service demand is high but currently there aren't enough buses running on them.

The results demonstrate that the Toolkit can be useful for reviewing the city's timetable, eliminating redundant and overlapping trips, and getting a new optimized timetable. BMTC is currently implementing the

Table 9
Existing v/s updated schedule of route number 252-F.

Schedule id	Route number	Start point	End point	Start time	End time	Updated start time	Updated end time
251D/1	252-F	5848	8456	06:50	07:35	06:50	07:35
252F/24	252-F	5848	8456	06:50	07:35	Excluded	Excluded
252F/6	252-F	5848	8456	06:55	07:40	06:55	07:40
252F/8	**252-F**	**5848**	**8456**	**06:55**	**07:40**	**07:00**	**07:45**
252F/22	252-F	5848	8456	07:05	07:50	07:05	07:50
252F/11	252-F	5848	8456	07:10	07:55	07:10	07:55
252F/24	252-F	8456	5848	07:40	08:25	Excluded	Excluded
252F/8	252-F	8456	5848	07:45	08:30	07:45	08:30

case study's findings on actual city routes (see attached letter of support from BMTC officials). Once implemented, we are confident that these results will assist the operational manager in determining the optimized schedule of the bus without any overlapping trips, with minimum possible modifications to the current schedule. The revised schedule can retain existing bus users and attract users from other modes of transport as well. Additionally, some other benefits such as better service level can also be accrued. Ridership will increase as a result of more frequent services, which will reduce congestion, emissions, and lead to environmental benefits. The proposed Toolkit is scalable, flexible, and replicable, and after making contextual modifications any PT official from a different city can use it with ease.

The work in this paper has several possible extensions. Subject to availability of data, the model varying inter-cluster times can be considered for more accurate parameters, and hence more buses saved. Dynamic rescheduling of buses on account of breakdown or route congestion is another major direction for future work. Lastly, models that integrate such bus rescheduling with crew scheduling, another major area of concern for city operators, can be developed. The above extensions refer to Internal Combustion Engine (ICE) based buses. The problem gets more interesting as Electric Vehicles (EV) enter the picture. While rescheduling EVs, constraints on their range, charging times and charger locations need to be incorporated in the model. The models developed in this work can be a good starting point for such models. Thus, overall, this is quite an exciting problem, with myriad problems still to be solved, particularly complicated due to the network and technology complexities.

CRediT authorship contribution statement

Pradeep Borade: Writing – review & editing, Writing – original draft, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Gajendra Nagar:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Hemant K. Suman:** Writing – original draft, Supervision, Methodology, Funding acquisition, Conceptualization. **Lakshay Lakshay:** Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Methodology, Conceptualization. **Ravi Gadepalli:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Conceptualization. **Nomesh B. Bolia:** Supervision, Resources, Methodology, Conceptualization.

Funding

This research was supported by the Science and Engineering Research Board (SERB), Government of India, under Grant No. SRG/2022/00711.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

Table A1
Details of each column attributes for Form-IV.

Attribute	Description
Schedule id	Bus number in numeric form
Schedule number	Bus number in alphanumeric form (to identify AC and Non-AC bus)
Form four name	Weekly schedule details (Days on which bus operates)
Route number	Route number in alphanumeric form
Start point	Source of the trip
End point	Destination of the trip
Start time	Dispatch time of the trip
End time	End time of trip
Running time	Running duration of trip
Break time	Break time of bus after each trip
Dead trip	Dead trip indicator in binary form (from depot to source and destination to depot)
Depot name	Depot from which bus operates

References

Andrade-Michel, A., Ríos-Solís, Y.A., Boyer, V., 2021. Vehicle and reliable driver scheduling for public bus transportation systems. Transp. Res. B Methodol. 145, 290–301.
Andres, M., Nair, R., 2017. A predictive-control framework to address bus bunching. Transp. Res. B Methodol. 104, 123–148.
Aziz, H.A., Garikapati, V., Rodriguez, T.K., Zhu, L., Sun, B., Young, S.E., Chen, Y., 2022. An optimization-based planning tool for on-demand mobility service operations. Int. J. Sustain. Transp. 16 (1), 45–56.
Badami, M.G., Haider, M., 2007. An analysis of public bus transit performance in Indian cities. Transp. Res. A Policy Pract. 41 (10), 961–981.
Bartholdi, J.J., Eisenstein, D.D., 2012. A self-coördinating bus route to resist bus bunching. Transp. Res. B Methodol. 46 (4), 481–491.
BMTC (2023). Bengaluru Metropolitan Transport Corporation. Retrieved from https://mybmtc.karnataka.gov.in/info-1/BMTC+Glance/en (accessed on 07 January 2023).
Bunte, S., Kliewer, N., 2009. An overview on vehicle scheduling models. Public Transp. 1 (4), 299–317.
Cao, H., Wachowicz, M., 2019. The design of an IoT-GIS platform for performing automated analytical tasks. Comput. Environ. Urban Syst. 74, 23–40.
Cats, O., West, J., Eliasson, J., 2016. A dynamic stochastic model for evaluating congestion and crowding effects in transit systems. Transp. Res. B Methodol. 89, 43–57.
Ceder, A.A., 2011. Public-transport vehicle scheduling with multi vehicle type. Transp. Res. Part C Emerging Technol. 19 (3), 485–497.
Ceder, A., Wilson, N.H., 1986. Bus network design. Transp. Res. B Methodol. 20 (4), 331–344.
Cervero, R., 2000. Informal transport in the developing world. United Nations Centre for Human Settlements (Habitat), Nairobi. ISBN 92-1-131453-4.
Cervero, R., Golub, A., 2007. Informal transport: A global perspective. Transp. Policy 14 (6), 445–457.
Daganzo, C.F., 2009. A headway-based approach to eliminate bus bunching: Systematic analysis and comparisons. Transp. Res. B Methodol. 43 (10), 913–921.
Daganzo, C.F., Pilachowski, J., 2011. Reducing bunching with bus-to-bus cooperation. Transp. Res. B Methodol. 45 (1), 267–277.
Dao, N.N., 2022. Internet of wearable things: Advancements and benefits from 6G technologies. Futur. Gener. Comput. Syst.
Delgado, F., Munoz, J.C., Giesen, R., 2012. How much can holding and/or limiting boarding improve transit performance? Transp. Res. B Methodol. 46 (9), 1202–1217.
Desaulniers, G., Hickman, M.D., 2007. Public transit. Handbooks Oper. Res. Management Sci. 14, 69–127.

Desfontaines, L., Desaulniers, G., 2018. Multiple depot vehicle scheduling with controlled trip shifting. Transp. Res. B Methodol. 113, 34–53.
Forbes, M.A., Holt, J.N., Watts, A.M., 1994. Theory and Methodology An exact algorithm for multiple depot bus scheduling. Eur. J. Oper. Res. 72.
Garikapati, V., 2019. Optimizing Fleet Operations in Automated Mobility Districts: Serving On-Demand Mobility with Automated Electric Shuttles (No. NREL/PR-5400-73631). National Renewable Energy Lab.(NREL), Golden, CO (United States).
Goel, R., Tiwari, G., 2016. Access–egress and other travel characteristics of metro users in Delhi and its satellite cities. IATSS Res. 39 (2), 164–172.
Gwilliam, K., 2003. Urban transport in developing countries. Transp. Rev. 23 (2), 197–216.
Hassold, S., Ceder, A.A., 2014. Public transport vehicle scheduling featuring multiple vehicle types. Transp. Res. B Methodol. 67, 129–143.
He, S.X., Dong, J., Liang, S.D., Yuan, P.C., 2019. An approach to improve the operational stability of a bus line by adjusting bus speeds on the dedicated bus lanes. Transp. Res. Part C Emerging Technol. 107, 54–69.
Hernández, D., Muñoz, J.C., Giesen, R., Delgado, F., 2015. Analysis of real-time control strategies in a corridor with multiple bus services. Transp. Res. B Methodol. 78, 83–105.
Heyken Soares, P., Mumford, C.L., Amponsah, K., Mao, Y., 2019. An adaptive scaled network for public transport route optimisation. Public Transp. 11 (2), 379–412.
Ibarra-Rojas, O.J., Delgado, F., Giesen, R., Muñoz, J.C., 2015. Planning, operation, and control of bus transport systems: A literature review. Transp. Res. B Methodol. 77, 38–75.
Kliewer, N., Mellouli, T., Suhl, L., 2006. A time-space network based exact optimization model for multi-depot bus scheduling. Eur. J. Oper. Res. 175 (3), 1616–1627.
Lizana, P., Muñoz, J.C., Giesen, R., Delgado, F., 2014. Bus control strategy application: case study of Santiago transit system. Procedia Comput. Sci. 32, 397–404.
Lysgaard, J., Wøhlk, S., 2014. A branch-and-cut-and-price algorithm for the cumulative capacitated vehicle routing problem. Eur. J. Oper. Res. 236 (3), 800–810.
Naumann, M., Suhl, L., Kramkowski, S., 2011. A stochastic programming approach for robust vehicle scheduling in public bus transport. Procedia Social. Behav. Sci. 20, 826–835.
Nesheli, M.M., Ceder, A., 2015. Improved reliability of public transportation using real-time transfer synchronization. Transp. Res. Part C Emerging Technol. 60, 525–539.
Ocalir-Akunal, E.V., 2016. Decision support systems in transport planning. Procedia Eng. 161, 1119–1126.
Ometov, A., Bezzateev, S.V., Kannisto, J., Harju, J., Andreev, S., Koucheryavy, Y., 2016. Facilitating the delegation of use for private devices in the era of the internet of wearable things. IEEE Internet Things J. 4 (4), 843–854.
Patel, D., Narmawala, Z., Tanwar, S., Singh, P.K., 2019. A systematic review on scheduling public transport using IoT as tool. Smart Innov. Commun. Comput. Sci. 39–48.
Petit, A., Ouyang, Y., Lei, C., 2018. Dynamic bus substitution strategy for bunching intervention. Transp. Res. B Methodol. 115, 1–16.
Phun, V.K., Yai, T., 2016. State of the art of paratransit literatures in Asian developing countries. Asian Transport Stud. 4 (1), 57–77.
Pi, X., Egge, M., Whitmore, J., Qian, Z.S., Silbermann, A., 2018. Understanding transit system performance using avl-apc data: An analytics platform with case studies for the pittsburgh region. J. Public Transp. 21 (2), 19–40.
Ruiz, M., Segui-Pons, J.M., Mateu-LLadó, J., 2017. Improving Bus Service Levels and social equity through bus frequency modelling. J. Transp. Geogr. 58, 220–233.
Schmöcker, J.D., Sun, W., Fonzone, A., Liu, R., 2016. Bus bunching along a corridor served by two lines. Transp. Res. B Methodol. 93, 300–317.
Shang, H., Liu, Y., Wu, W., Zhao, F., 2023. Multi-depot vehicle scheduling with multiple vehicle types on overlapped bus routes. Expert Syst. Appl. 228.
Sohail, M., Maunder, D.A.C., Miles, D.W.J., 2004. Managing public transport in developing countries: Stakeholder perspectives in Dar es Salaam and Faisalabad. Int. J. Transp. Manag. 2 (3–4), 149–160.
Suman, H.K., Bolia, N.B., 2019. Mitigation of overcrowding in buses through bus planning. Public Transp. 11 (1), 159–187.
Suman, H.K., Bolia, N.B., Tiwari, G., 2017. Comparing public bus transport service attributes in Delhi and Mumbai: Policy implications for improving bus services in Delhi. Transp. Policy 56, 63–74.
Suman, H., Larrain, H., Muñoz, J.C., 2021. The impact of using a naïve approach in the limited-stop bus service design problem. Transp. Res. A Policy Pract. 149, 45–61.
Sun, J.Y., Chen, Y.Y., Huang, J.L., Wei, P.Y., Song, C.C., 2021. Flexible Bus Route Optimization Scheduling Model. Adv. Civ. Eng.
Thynell, M., Mohan, D., Tiwari, G., 2010. Sustainable transport and the modernisation of urban transport in Delhi and Stockholm. Cities 27 (6), 421–429.
Wei, M., Sun, B., Jin, W., 2013. A bi-level programming model for uncertain regional bus scheduling problems. J. Transp. Syst. Eng. Inf. Technol. 13, 106–112.
Yan, Y., Meng, Q., Wang, S., Guo, X., 2012. Robust optimization model of schedule design for a fixed bus route. Transp. Res. Part C Emerging Technol. 25, 113–121.
Zhong, C., Huang, X., Arisona, S.M., Schmitt, G., Batty, M., 2014. Inferring building functions from a probabilistic model using public transportation data. Comput. Environ. Urban Syst. 48, 124–137.
Zhoucong, X., Peijia, H., Jing, T., Liping, L., 2013. Transit Vehicles Intelligent Scheduling Optimization Based on the Division of Characteristic Periods. Procedia. Soc. Behav. Sci. 96, 1502–1512.