Operational Innovations for Efficiency and Accessibility of On-demand Mobility in Rural Areas

A Technical Report Submitted to the Rural Safe Efficient Advanced Transportation (R-SEAT) Center and United States Department of Transportation

FINAL REPORT

Principal Investigator:
Yanshuo Sun, Ph.D.
Associate Professor
Department of Industrial & Manufacturing Engineering,
FAMU-FSU College of Engineering
2525 Pottsdamer Street
Tallahassee, FL 32310, USA
Phone: +1 (850) 645-8996
E-mail: y.sun@eng.famu.fsu.edu

Co-Principal Investigator:
Ren Moses, Ph.D., PE
Professor
Department of Civil and Environmental Engineering
Florida A&M University
2525 Pottsdamer Street, Suite 129
Tallahassee, FL 32310, USA
Phone: +1 (850) 410-6191
Email: ren.moses@famu.edu

Research Assistant:
Zhenhao Lan
Doctoral Student
Department of Industrial & Manufacturing Engineering,
FAMU-FSU College of Engineering
2525 Pottsdamer Street
Tallahassee, FL 32310, USA
E-mail: zl23o@fsu.edu

October 2025

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, under the grant 69A3552348321 from the U.S. Department of Transportation's University Transportation Centers Program. The U.S. Government assumes no liability for the contents or use thereof.

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Operational Innovations for Efficiency and Accessibility of On-demand Mobility in Rural Areas		5. Report Date 10/22/2025
		6. Performing Organization Code 59-0977035
7. Author(s) Yanshuo Sun (https://orcid.org/0000-0003-2943-4323) Ren Moses (https://orcid.org/0000-0003-2988-5220) Zhenhao Lan (https://orcid.org/0009-0001-6187-8955)		8. Performing Organization Report No.
9. Performing Organization Name and Address Florida A&M University-Florida State University College of Engineering 2525 Pottsdamer Street Tallahassee, FL 32310		10. Work Unit No. (TRAIS)
		11. Contract or Grant No. 69A3552348321
12. Sponsoring Agency Name and Address Rural Safe Efficient Advanced Transportation (R-SEAT) 2525 Pottsdamer Street Tallahassee, FL 32310		13. Type of Report and Period Covered Final Report Period Covered: 05/01/2024 – 04/30/2025
		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Unlike the fixed-route transit service, microtransit can thrive in rural and suburban areas with low demand density, because its operation is tailored to individual travel plans. Therefore, microtransit has great potential in improving mobility and accessibility for the transportation disadvantaged individuals in rural areas. Although many researchers have investigated how microtransit vehicle schedules and routes can be optimized, it is widely assumed that travel requests submitted by individual riders are accommodated independently, without exploring any coordination among riders. This project introduces a concept of continuous time window expansion (TWE) in on-demand microtransit services. Traditional microtransit systems often employ fixed pickup and drop-off schedules or time windows, which can result in increased passenger rejection rates and operational inefficiencies during periods of high demand. To address these limitations, we develop novel mixed-integer programs that incorporate continuous expansions of rider time windows, regulated through a penalty parameter that balances service quality (measured by schedule disruptions for new riders) with operational efficiency (measured by total route distance). The programs are formulated for both single-vehicle and multi-vehicle fleet scenarios, integrating rider assignment, routing, and time window adjustments. Computational experiments using synthetic data demonstrate that coordinated fleet-wide optimization with TWE can reduce total time window expansions by up to 89.7% and lower operational costs by 19.2% compared to independent single-vehicle optimization. While most instances can be solved optimally by the solver very quickly (e.g., within 25 seconds), the computation time can grow to around 1,000 seconds when there are five vehicles, and more than 30 riders for scheduling, which warrants the design of new solution algorithms through future research. Our research is expected to provide actionable guidance for transit agencies seeking to improve the flexibility and efficiency of microtransit operations through dynamic scheduling policies.

17. Key Words	18. Distribution Statement
Microtransit, on-demand mobility, time window expansion, optimization, performance evaluation	No restrictions

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	51	

ACKNOWLEDGEMENTS

This project was sponsored by the Rural Safe Efficient Advanced Transportation (R-SEAT) Center and United States Department of Transportation. The Principal Investigators would like to thank the representatives of the REAT Center for their valuable feedback throughout the project activities.

EXECUTIVE SUMMARY

Public transit agencies especially those serving rural and small-urban communities are turning to technology-enabled, demand-responsive microtransit to close coverage gaps while protecting service quality for riders with diverse needs. This project develops and evaluates a rider-centric scheduling approach that converts agency promises such as pickup windows, capacity limits, and maximum ride time into explicit optimization constraints and then introduces a disciplined, minute-scale time-window expansion (TWE) mechanism for new requests only. The result is a practical way to enlarge the feasible set at insertion—preserving prior commitments for onboard and scheduled riders—while quantifying any rider inconvenience through transparent penalties and policy caps to help balancing the operational cost with customer service quality.

The research advances two mixed-integer programming (MIP) formulations. The first one is a single-vehicle model which captures the real-time insertion step for one active tour: visited stops remain fixed, the remaining sequence can be reordered, and pickup/drop-off windows for new riders may be symmetrically expanded within an explicit cap. The objective minimizes routing distance plus a priced TWE with certain penalty weight, subject to flow balance, time-propagation, pickup-before-drop-off precedence, capacity, and per-rider maximum ride-time constraints. The second multi-vehicle extension model couples these route-level decisions with fleet-level assignment so new requests can be served by whichever vehicle yields the best network-wide outcome, while maintaining the same protection of previously promised times.

We test our formulations with some synthetic instances which vary spatial dispersion, planning-horizon width, and fleet size. Instances include onboard, scheduled, and new riders with policy-consistent windows and ride-time bounds. We evaluate operational outputs of these two models, for the single vehicle model, TWE can greatly improve the feasibility of new rider insertion with a reasonable cap and penalty. For the multi-vehicle coordination model, it can significantly reduce TWE usage compare to single-vehicle model with the same instance and lower the total travel distance of the same vehicle route also. In summary, the key achievements of this project include the following:

- Formulation of rider-centric scheduling models and introduce continuous TWE for new requests with explicit caps and penalties.
- Development of a fleet-wide coordination system which improves the optimization performance.
- Successful testing the real-time computational performance with different instances
- Providing policy calibration guidance for agency to improve the flexibility and efficiency of microtransit operations.

For transit agencies, the research provides two practical models which can help them improve the performance of their microtranit operations when facing different routing scenarios in rural and small-urban area. This project supported by the U.S. Department of Transportation through the University Transportation Center program tested the models under different instance scales and has proved that proposed models can be solved within reasonable time. The models also align with agency KPIs, integrate with standard solvers, and generate auditable optimization results as total travel distance. Together, these contributions offer a practical blueprint for more reliable, equitable, and productive on-demand microtransit in rural and small-urban America.

Contents

List of Figures

List of Tables

1. INTRODUCTION

1.1 On-Demand Microtransit in Rural and Small-Urban Contexts

Public transit agencies across the United States are increasingly deploying microtransit, which is a form of demand-responsive transit (DRT), to complement or partially replace fixed-route service in low-density settings. In Federal Transit Administration (FTA) Mobility on Demand (MOD) materials, microtransit is described as a multi-rider, dynamically routed service that typically operates within defined zones and often connects to fixed routes for first/last-mile access (Federal Transit Administration, 2023b). The umbrella includes door-to-door operations, virtual-stop (corner-to-corner) designs, hub-to-hub circulators, and commingled ADA (Americans with Disabilities Act) paratransit with general-public service, coordinated through rider apps or call centers, automated scheduling, and real-time vehicle tracking (National Center for Applied Transit Technology, 2023; Shared-Use Mobility Center, 2022). Figure 1 shows how microtransit positions in light of other fixed and flexible transit service types.

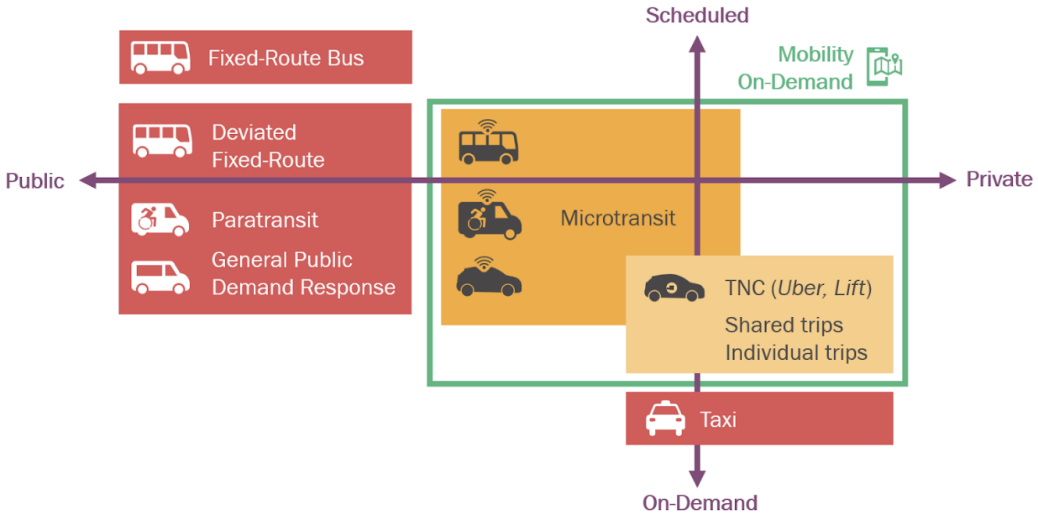


Figure 1: Different types of on-demand transit and microtransit (National Center for Applied Transit Technology, 2023)

With the service definition in place, we next turn to the settings where microtransit's scheduling choices are especially consequential: rural and small-urban systems (National Center for Applied Transit Technology, 2023; Shared-Use Mobility Center, 2025). Rural and small-urban areas face persistent mobility gaps due to long trip distances, sparse demand, and limited fixed-route coverage. National statistics show that rural fleets remain predominantly demand-responsive; recent

Rural Transit Fact Books report tens of thousands of DRT vehicles in service annually, reflecting DRT's centrality to rural mobility portfolios (Mattson, 2024, 2025). Demographic patterns compound the need: rural regions have higher shares of older adults and persons with disabilities than urban areas, who are disproportionately reliant on accessible, flexible services for healthcare, employment, and daily activities (Mattson, 2024). Within this policy landscape, FTA's MOD initiatives encourage agencies to pilot technology-enabled models including microtransit to expand spatial-temporal coverage and improve first/last-mile connectivity (Federal Transit Administration, 2023a).

Industry overviews and advocacy guidance described a typical microtransit travel request as carrying an origin, a destination, and a requested pickup and drop-off time window submitted via app or phone; platforms then batched compatible travel requests subject to pickup/drop-off windows, vehicle capacity, and service-quality limits such as maximum ride time and maximum wait (Disability Rights Education & Defense Fund, 2025). For instance, in the OmniRide Connect microtransit service in Manassas Park and other areas (such as Quantico) in Virginia, their wait time for the service is intended to be no more than 15 minutes from time of reservation to time of pickup from your requested location within the service zone to keep system efficiency (OmniRide (Potomac and Rappahannock Transportation Commission), 2024). A Utah Transit Authority microtransit planning project suggested a different maximum wait time ranging from 15 to 40 minutes for different service zones based on their local demand and vehicle supply (Via Mobility, LLC, 2020).

To compare our modeling with real-world practice, we introduce the metrics most microtransit operations report. Microtransit operations are often evaluated with rider-facing and productivity metrics that scheduling directly controls: riders per vehicle-hour (PVH), average wait/response time, pickup on-time performance (OTP) within the announced window, achieved pooling share, and cost per rider (Federal Transit Administration, 2019; National Center for Applied Transit Technology, 2023). Figure 2 shows customer support metrics often reported to boards and communities (Utah Transit Authority, 2020). In our model, we report total routing distance and total time window expansion which will be introduced in detail in Section 1.2 as evaluation metrics that indicate operating cost per rider.

1.2 Microtransit Scheduling Problem

In this project, we focus on the on-demand microtransit scheduling problem, which is to decide which vehicle serves which pickup–drop-off pair, in what visit order, and at what service start time at each stop, while guaranteeing the service promises communicated to riders with small shuttles or vans. At its core, microtransit scheduling is a dynamic assignment and vehicle routing problem.

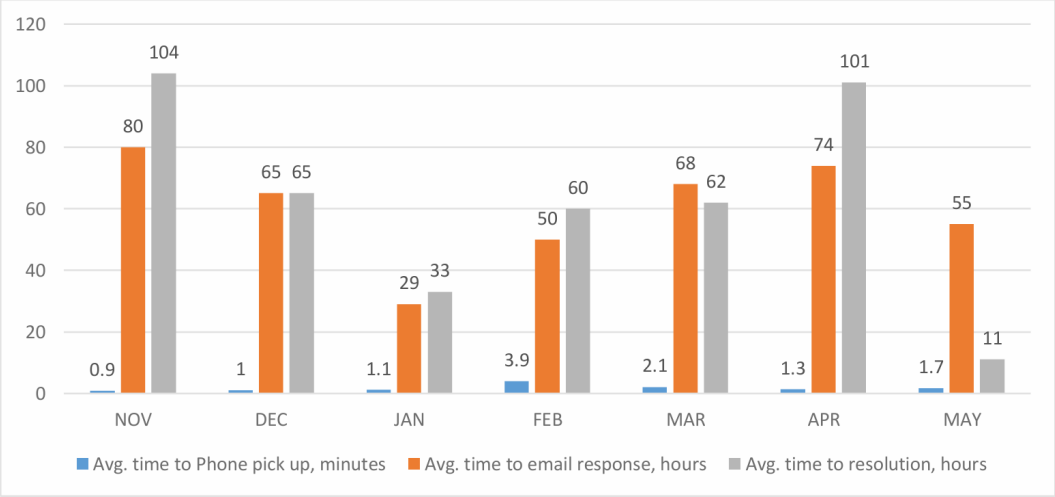


Figure 2: Customer support metrics in a UTA microtransit program (Utah Transit Authority, 2020)

Requests arrive over time; the platform batches them on rolling horizons, checks feasibility against vehicle capacity and time windows, and optimizes a matching between rider trips and vehicles. Putting it together, the microtransit scheduling problem is an online optimization challenge that must translate policy choices (who is served, where, and how) into real-time assignments the fleet can execute.

The commonly adopted operational constraints in microtransit scheduling problem are:

(1) Time windows (pickup and drop-off). Microtransit system typically either define a pickup and drop-off time window for scheduled requests (typically 15–30 minutes of width) or enforce maximum wait time targets around 10–20 minutes (often 15 minutes) (Laredo Metropolitan Planning Organization, 2025; Summit Stage, 2024; Virginia Department of Rail and Public Transportation, 2023).

(2) Vehicle capacity. Feasible routes must respect the maximum vehicle occupancy constraint at all times.

(3) Other special constraints such as maximum wait, detour bounds, zone boundaries, operating hours and transfer rules based on the specific topic and problem background.

These constraints jointly define the feasible set for both scheduled riders (onboard or visited) and newly arriving. In low-supply periods, evaluations reported that agencies additionally applied explicit response-time caps (e.g., 30–45 minutes) so that requests exceeding the cap were rescheduled or denied to contain queues (Utah Transit Authority, 2020).

The objectives of the microtransit scheduling problem are mainly about minimizing denied requests and rider constraint violations and routing cost, maximizing pooling/share rate and operational efficiency and reducing vehicle hours and miles and subject to service-level targets defined by the agency which are determined by the decision variables in the problem such as the acceptance

of each request, the assignment of requests to a vehicle, stop sequence and optional transfer points.

Recent literature has explored various approaches to optimize the rider assignment and vehicle routing decisions in microtransit with minimal total operating cost or maximal operation efficiency. For instance, Li et al. (2024) proposed differentiated scheduling based on traveler flexibility, utilizing piecewise-linear penalties to mitigate the adverse effects of strict windows. Yang et al. (2020) addressed unexpected disruptions through a hybrid scheduling model, allowing early arrivals with penalties while disallowing late arrivals. Additionally, Ma et al. (2023) developed a two-stage stochastic programming approach to dynamically adjust vehicle routes, substantially lowering rejection rates.

However, existing studies on microtransit scheduling problems still face numerous limitations in different aspects. Lots of approaches only implement discrete or scenario-based time window relaxations (Li et al., 2024; Ma et al., 2023; Sarbijan & Behnamian, 2022; Yang et al., 2020), lacking the continuous treatment of time window flexibility. Most current fleet optimization models typically assume fixed time windows or treat time flexibility independently from route optimization. Therefore, the fleet-level optimization integrating time window adjustments with dynamic vehicle routing remains relatively underexplored. There is also a lack of understanding of the explicit trade-offs between time window flexibility and operational efficiency, leaving transit operators without clear guidance on managing these competing demands effectively. To conclude, we identify a significant research gap in the understanding of rider schedule coordination and its role in improving system-level operational performance. Rider schedule coordination here means that the travel requests submitted by new riders in terms of pickup and drop-off time windows can be continuously adjusted by the microtransit operator if such adjustments can lead to a lower routing cost or better operational efficiency in both single-vehicle and multi-vehicle scenarios.

Therefore, this study introduces a novel technique termed Continuous Time Window Expansion (TWE) for microtransit systems, formulated as Mixed-Integer Programming (MIP) models. Unlike discrete or probabilistic methods, the TWE continuously expands pickup and drop-off windows, regulated through an adjustable penalty parameter which can be adjusted by operator to balance service quality with operational efficiency. The time window adjustments and routing decisions are optimized in both single-vehicle and multi-vehicle formulations. The multi-vehicle formulation further considers rider assignments.

1.3 Project Objectives

This project pursues a set of objectives that collectively advance the state of practice for flexible, rider-centric microtransit operations:

1. To better understand the effectiveness of rider behavioral coordination in improving on-

demand microtransit operational efficiency;

2. To design a mathematical optimization model for supporting rider schedule coordination;
3. To test the developed optimization algorithm in both simulated and realistic scenarios;
4. To derive managerial insights that can inform the real-world deployment of the proposed schedule negotiation method.

1.4 Structure of This Report

The remainder of the report proceeds as follows. Section 2 Literature Review summarizes prior work on rider-centric constraints in microtransit and on flexible or soft time window arrangements, positioning the present study. Section 3 Problem Statement introduces the single- and multi-vehicle formulations that integrate continuous TWE with routing and assignment decisions. Section 4 Computational Experiments presents the experimental design, instance generation, and quantitative findings, including sensitivity to policy parameters and the benefits of fleet coordination. Section 5 concludes with key insights, policy implications, and directions for scaling and deployment.

2. LITERATURE REVIEW

This literature review summarizes previous research relevant to rider-centric constraints and flexible scheduling in microtransit route optimization. It is structured into three parts: (1) rider-centric optimization approaches, (2) studies exploring flexible or soft scheduling arrangements, and (3) identification of research gaps.

2.1 Rider Travel Experience in Microtransit Optimization

Microtransit routing problems enhance traditional pickup and delivery vehicle routing models by incorporating rider-centric constraints such as maximum waiting times, detour limits, time window adherence, and ride-duration constraints.

Recent studies have developed exact and heuristic methods to address these rider-experience constraints. Tuncel et al. (2023) encode per-rider service promises through a pickup/response-time cap (a maximum time from request to pickup) and a bound on in-vehicle detour relative to the rider's direct trip. These limits are applied during feasibility checks so that insertions never violate the rider's promised pickup window or their allowed extra ride time. To more accurately reflect rider preferences, Li et al. (2024) developed a flexible scheduling MIP model distinguishing between time-sensitive and flexible travelers. Time-sensitive riders are kept on hard time windows, while flexible riders may accept bounded window adjustments with piecewise-linear penalties to represent rider inconveniences so that any deviation from their preferred pickup or drop-off time is tightly controlled and explicitly priced. The combination of hard windows and small capped adjustments effectively limits individual waiting and ride time, and operators can calibrate these limits so that they match how much extra detour a rider will accept.

Additional evidence that directly ties route–schedule feasibility to rider experience showed that microtransit outcomes depended on how strictly pickup windows and ride time constraints were enforced in the schedule rather than on average targets alone. Markov et al. (2021) demonstrated that enforcing per-rider wait and detour limits in the routing logic materially changed feasible route construction. In particular, the maximum excess ride time is defined in three interchangeable ways: (1) an absolute surplus over direct (shortest-path) time, (2) a ratio of realized ride time over direct time, and (3) a hybrid of both. Tightening these caps reduces insertion opportunities but improves per-rider time reliability; relaxing them expands the feasible set and can change optimal fleet sizing and route construction. At the demand side, He & Ma (2022) estimated a Bayesian-network model for microtransit users' next-ride decisions and found that tolerance to waiting and walking influ-

enced continued usage, reinforcing the need to encode heterogeneous time sensitivities in schedule optimization. Fayed et al. (2024) quantified a trade-off between initial waiting and detour limits in on-demand microtransit: larger batches rarely reduce waiting, but can improve detour per rider due to better pooling. This implied that agencies should set a fixed wait time cap or tight pickup windows and then use detour limits to keep each rider's extra ride time within the allowed limits as pooling increases. From a planning perspective, Rath et al. (2023) scaled field data and scenario simulation across multiple U.S. deployments. This study showed that window and batching settings materially shift realized riders-per-vehicle-hour and rejections. The mechanism was operational: tighter pickup windows and shorter wait targets reduced insertion feasibility, while slightly looser windows increased pooling but must be counterbalanced by per-rider ride time to protect experience.

2.2 Flexible Scheduling Arrangements in Microtransit

Fixed scheduling windows can compromise microtransit performance under uncertain demand and travel conditions, prompting researchers to explore more flexible scheduling frameworks. Yang et al. (2020) addressed disruptions caused by unexpected time window changes through a hybrid time window model permitting early arrivals with penalties while forbidding lateness, significantly outperforming traditional rescheduling methods.

Several studies have adopted soft or flexible scheduling approaches. For example, Ma et al. (2023) introduced a two-stage stochastic programming framework for flexible bus routing, dynamically adjusting routes based on real-time requests and stochastic demand forecasts, significantly reducing request rejection rates and operational costs. Similarly, Sarbijan & Behnamian (2022) explored real-time flexible-window scheduling in feeder transit, achieving notably higher rider coverage without increasing mileage.

For zonal-based flexible bus operations, Lee et al. (2021) jointly optimized stop clustering, routing, and departure times, achieving cost savings compared to fixed schedules. Fu & Chow (2022) enabled synchronized en-route transfers in microtransit by imposing hard time windows at transfer points so that two vehicles meet within a narrow interval and the rider's transfer wait is bounded. From the rider perspective, this design adds time window constraints at intermediate transfer nodes while still respecting pickup and drop-off time window promises.

Complementary work from scheduling and robust operations treated flexibility in a controlled, auditable way. In Figliozzi (2010) and Taş et al. (2014), they modeled soft windows with penalties (and, in the latter, stochastic travel times), showing that bounded deviations can reduce infeasibility without dissolving guarantees. In fixed-route coordination, Gkiotsalitis & Alesiani (2019) and Liu et al. (2017) demonstrated that carefully placed slack and timetable–vehicle co-optimization reduce

early/late violations and protect transfers, a principle directly portable to microtransit when pickup windows are tuned jointly with feasible visit orders. For customized transit design with explicit windows, Li et al. (2018) showed that enforcing fixed pickup windows alongside mixed-load rules improved service regularity; together with the microtransit-focused works above, these studies motivated modeling window flexibility as a bounded decision variable with explicit penalties rather than as an after-the-fact override.

2.3 Literature Summary and Research Gaps

While existing studies increasingly integrated rider-centric constraints within routing optimization models (Chen et al., 2021; Li et al., 2024; Tuncel et al., 2023), most encoded flexibility via coarse batch choices, discrete rule switches, or stepwise penalties; continuous, policy-bounded expansion magnitudes that are priced minute-by-minute and embedded directly in schedule construction remain rare in public microtransit optimization. Moreover, although flexible or probabilistically relaxed windows were explored in flexible-bus and feeder contexts (Lee et al., 2021; Ma et al., 2023; Sarbijan & Behnamian, 2022), very few formulations coupled those bounded adjustments with maximum ride time constraints and multi-vehicle route construction in one optimization.

Additionally, although multi-vehicle coordination models are gaining traction (Fu & Chow, 2022; Tuncel et al., 2023; Veve & Chiabaut, 2022), many either impose fixed time windows or overlook routing optimizations. To address these gaps, this paper introduces a novel, continuous TWE model with penalty parameters to balance service quality and routing efficiency. Our multi-vehicle model integrates rider reassignment among vehicles, routing optimization, and time window adjustments, contributing significant advancements to the state-of-the-art in microtransit optimization.

The literature overview is summarized in Table 1 which is mainly focuses on the research method, literature focus, studied case or used data and the key takeaways from their studies.

Table 1: Overview of analyzed research on Microtransit scheduling.

a/a	Authors (Year)	Method	Focus	Case/Data	Key takeaways
1	Tuncel et al. (2023)	MIP (integrated)	Matching + rebalancing under wait/detour caps	Shared MoD logs	Joint decisions reduced waits/rejections vs. sequential pipelines.
2	Chen et al. (2021)	Bi-objective + heuristic	Customized/zonal routes with time window and excess-travel penalties	Real case applications	Pareto plans respected windows while lowering operator cost.
3	Li et al. (2024)	MIP + preference classes	Time-sensitive vs. flexible riders; piecewise disutility	Synthetic/field params	Modest priced deviations for flexible riders cut system cost.
4	Markov et al. (2021)	Simulation + optimization	Schedule design with rider wait/detour protections	Multi-city scenarios	Enforcing rider bounds altered feasible routes and fleet sizing.
5	He & Ma (2022)	Bayesian networks	Continued-use decision drivers	U.S. microtransit users	Tolerance to wait/walk shaped demand—supports heterogeneous windows.
6	Rath et al. (2023)	Scenario upscaling sim	Deployment portfolio choices	Vendor + city data	Window/batching settings shifted PVH and rejections via scheduling.
7	Alonso-Mora et al. (2017)	ILP on shareability graph	Dynamic assignment with max wait/detour	NYC taxi traces	High pooling achieved while enforcing per-rider time guarantees.
8	Santi et al. (2014)	Analytical + data	Shareability vs. small time/space slack	NYC taxis	Small temporal/spatial slack sharply increased shareability.
9	Stiglic et al. (2015)	Heuristic	Meeting points with bounded walking	Synthetic	Short walks reduced system time with limited user burden.
10	Vazifeh et al. (2018)	Min-cost flow / cover	Minimum fleet size at SLA	NYC taxis	Lower bounds linked fleet to promised response times.
11	Tachet et al. (2017)	Scaling law	Pooling potential vs. city size/density	Multi-city	Superlinear pooling benefits with demand density.
12	Fayed et al. (2024)	Analytical + experiments	Batching frequency vs. initial waits	Simulated demand	Aggressive batching risks longer initial waits—needs safeguards.

Table 1 continued on next page

a/a	Authors (Year)	Method	Focus	Case/Data	Key takeaways
13	Veve & Chiabaut (2022)	Demand-driven MIP	Recurring patterns with window/ride-time rules	European microtransit	Stabilized schedules; fewer day-to-day deviations.
14	Fu & Chow (2022)	Optimization with transfers	Synchronized en-route transfers for microtransit	Case study	Coordination reduced travel while protecting transfer times.
15	Lee et al. (2021)	Joint design model	Zonal stops, routing, departures under rider windows	Zonal microtransit	Vehicle-hours fell while meeting rider windows.
16	Sarbijan & Behnamian (2022)	Metaheuristic (real-time)	Collaborative feeder with flexible windows	Synthetic	Higher coverage without more vehicle-hours.
17	Ma et al. (2023)	Two-stage stochastic	Flexible buses with stochastic requests	Urban network	Bounded timing flex cut rejections and mileage.
18	Hansen et al. (2021)	Metrics framework	Microtransit performance/KPIs	U.S. pilots	Window-based OTP and wait metrics tie directly to scheduling.
19	Quadrifoglio et al. (2008)	Simulation study	DRT design: windows vs. productivity	Synthetic networks	Modest relaxations raised productivity in sparse areas.
20	Martínez & Eiró (2012)	Bi-criteria design	Minibus feeder design and stop selection	Lisbon (Sintra line)	Structured design lowered user time at acceptable cost.
21	Narayanan et al. (2020)	Review	Shared autonomous services + on-demand integration	Literature	Highlights scheduling levers and QoS constraints for pooling.
22	Tirachini (2020)	International review	Ride-hailing impacts and policy	Literature	Notes equity/externalities; stresses wait/ride-time protections.
23	Lowalekar et al. (2021)	RL + combinatorial	Zone-path construction for real-time ridesharing	City-scale simulation	Structured feasible paths enabled reliable matching under limits.
24	Iglesias et al. (2018)	Model Predictive Control	Data-driven repositioning/matching	City simulations	Short-term forecasts lowered realized wait without relaxing SLAs.

Table 1 continued on next page

a/a	Authors (Year)	Method	Focus	Case/Data	Key takeaways
25	Braverman et al. (2019)	Queueing/OR	Empty-car routing to support pickups	Analytical network	Asymptotically optimal rebalancing improved pickup reliability.
26	Nourinejad & Ramezani (2020)	System model	Rebalancing with congestion/pricing	Analytical network	Internalizing congestion improved system-wide pickup times.
27	Bimpikis et al. (2019)	Pricing + dispatch	Spatial pricing for balance	Analytical	Pricing with dispatch reduced mismatches and wait.
28	Xue et al. (2021)	MILP + heuristic	Rider scheduling with time windows (O2O)	Industrial data	Window-aware assignment improved on-time performance.
29	Baldacci et al. (2011)	Exact algorithm	PDPTW baselines with hard windows	OR benchmarks	Exact baselines inform feasibility checks for microtransit routing.
30	Kallehauge et al. (2005)	Survey/chapter	VRPTW foundations and column generation	Review	Formal window/precedence tools transfer to microtransit routing.
31	Chen et al. (2024)	Empirical econometrics	Schedule negotiation under window/VOT uncertainty (paratransit)	U.S. program data	Value-of-time heterogeneity motivates priced, bounded flexibility.
32	Arslan et al. (2019)	Dynamic PD model	Ad-hoc driver pickup–delivery with time constraints	Synthetic/operational	Time-feasible matching policies scale to real-time settings.
33	Gkiotsalitis & Alesiani (2019)	Robust optimization	Bus timetable slack under regulatory/resource constraints	Real bus line	Small buffers lowered early/late violations with limited cost.
34	Taş et al. (2014)	Column gen + B&P	Soft time windows with stochastic travel times	Benchmarks	Robust soft-window treatment curbed lateness under variability.
35	(Figliozzi, 2010)	Construct + improve	VRP with soft time window penalties	Benchmarks	Penalizing earliness/tardiness reduced infeasibility and cost.

3. PROBLEM STATEMENT

This section formalizes the real-time decision an operator makes when new requests arrive during operations using an insertion perspective: service that has already been delivered remains fixed, while the remaining nodes in each route may be rescheduled within certain constraints. The setup mirrors control-room practice and balances responsiveness and pooling against reliability commitments. We distinguish three rider states: onboard (pickup completed but not yet dropped off), scheduled (both stops pending), and new (pending acceptance) to reflect different operational obligations. To avoid unnecessary denials in low-density or bursty-demand settings, we allow bounded expansion in new riders' pickup and drop-off time windows when doing so resulting better cost and service quality. The single-vehicle formulation captures the insertion step for one active route; the multi-vehicle extension adds a fleet-assignment system that jointly decides which vehicle serves each new request while re-optimizing affected routes. Together, these models provide a practical basis for real-time microtransit decision support.

3.1 Base Problem

At a certain time, we are given one existing vehicle route, denoted as S, and one set of new rider requests. We seek to optimally accommodate the new requests by revising the existing route S, if feasible at all. The existing route S begins from the start depot o^+ and ends at the return depot o^-, between which a series of pickup and drop-off locations are covered. Without loss of generality, we number those visited and scheduled nodes of route S sequentially, from 1 to n, where $n/2$ gives the number of covered riders as each rider has one pickup location and one drop-off location. Then, the given route is expressed as $S = (o^+, 1, 2, \ldots, n, o^-)$. Based on the current location of the vehicle at the present time $\bar{\tau}_\sigma$, we partition route S into S_v (a tuple of visited nodes that occur before current location σ) and S_f (a tuple of nodes to be visited after location σ). Tuple S_f involves two types of riders, namely onboard riders $\tilde{R}$ and scheduled riders R. Each rider is indexed by a tuple (r^+, r^-), where the superscript "$+$" means pickup and superscript "$-$" means drop-off. For an onboard rider $(r^+, r^-) \in \tilde{R}$, its pickup location has been visited and its drop-off location has not yet been visited, namely r^+ is in tuple S_v and r^- is in tuple S_f. By contrast, for each rider $(r^+, r^-) \in R$, both the pickup and drop-off locations (i.e., r^+ and r^-) are in tuple S_f. Each node that has been visited, namely node i in tuple S_v, has a single timestamp τ_i, which is the actual node visit time. Each node to be visited, namely node i in S_f, is associated with a planned time window $[e_i, l_i]$, regardless of whether it is a pickup or drop-off node. Each rider $(r^+, r^-) \in \tilde{R} \cup R$ is associated with a maximum

ride time, denoted as $L_{(r^+,r^-)}$.

Following the naming convention, each new rider is represented by a tuple of the pickup and drop-off locations, namely $(\bar{r}^+, \bar{r}^-) \in \bar{R}$, where $\bar{R}$ is the set of new requests. The time windows for a new rider are constructed as follows. The pickup time window at location $\bar{r}^+$ is $[e_{\bar{r}+}, l_{\bar{r}+}]$, where $e_{\bar{r}+}$ is the present time plus a fixed buffer η_1 that is sufficient for the rider to walk to the designated pickup location. $l_{\bar{r}+} = e_{\bar{r}+} + \eta_2$, where η_2 is the pickup time window width predetermined by policy, such as 15 minutes. Given the direct travel time from node $\bar{r}^+$ to node $\bar{r}^-$, denoted as $t_{(\bar{r}^+,\bar{r}^-)}$, the earliest arrival time at the drop-off location $\bar{r}^-$ is $e_{\bar{r}-} = e_{\bar{r}+} + t_{(\bar{r}^+,\bar{r}^-)}$. Given the maximum ride time $L_{(\bar{r}^+,\bar{r}^-)}$, the latest drop-off time at $\bar{r}^-$ is $l_{\bar{r}-} = l_{\bar{r}+} + L_{(\bar{r}^+,\bar{r}^-)}$. Therefore, the drop-off time window is $[e_{\bar{r}-}, l_{\bar{r}-}]$.

The so-called *time window expansion* means that when a rider is being accommodated, the associated time windows (pickup and/or drop-off) can be widened. For instance, the pickup time window $[e_{\bar{r}+}, l_{\bar{r}+}]$ is changed to $[e_{\bar{r}+} - \delta_{\bar{r}+}, l_{\bar{r}+} + \delta_{\bar{r}+}]$, where $\delta_{\bar{r}+}$ is the amount of deviation at the pickup location $\bar{r}^+$. Note that the time window is expanded in two directions and the time window width increases by twice of $\delta_{\bar{r}+}$. Similarly, we can decide whether to expand the drop-off time window or not. Time window expansion decisions at the pickup and drop-off locations can be conducted independently. The expansion $\delta_{\bar{r}+}$ (or $\delta_{\bar{r}-}$) also has a penalty that will affect the objective in this problem. The expansion of the time window incurs a rider compensation at a fixed rate λ.

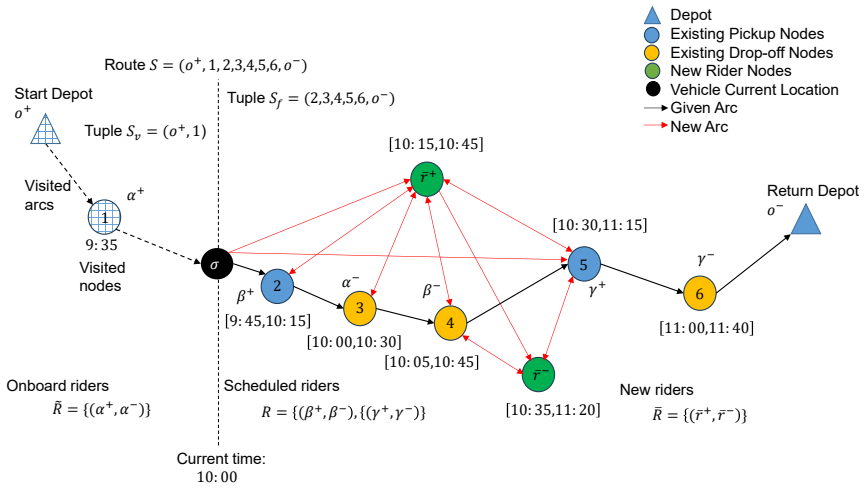


Figure 3: Illustration of the base problem network

Table 2: Notation for the base problem

Indices and Sets	
o^+	Start depot
o^-	End depot
S	A given vehicle route (modeled as a tuple of nodes)
σ	Current vehicle location, defined as a node
S_v	Tuple of visited nodes in route S before the current time
S_f	Tuple of future nodes to be visited in route S after the current time
$\tilde{R}$	Set of onboard riders, each of whom is indexed by (r^+, r^-)
R	Set of scheduled riders, each of whom is indexed by (r^+, r^-)
$\bar{R}$	Set of new riders, each of whom is indexed by $(\bar{r}^+, \bar{r}^-)$
$\bar{P}$	Set of pickup nodes of new riders: $\bar{P} = \{\bar{r}^+ : (\bar{r}^+, \bar{r}^-) \in \bar{R}\}$
$\bar{D}$	Set of drop-off nodes of new riders: $\bar{D} = \{\bar{r}^- : (\bar{r}^+, \bar{r}^-) \in \bar{R}\}$
N	Set of nodes, including σ, o^-, all nodes in tuple S_f, and nodes in $\bar{P} \cup \bar{D}$
A	Set of arcs, indexed by (i, j)
Parameters	
e_i	Earliest possible visit time at node i
l_i	Latest possible visit time at node i
s_i	Service duration at node i
$L_{(r^+,r^-)}$	Maximum ride time for rider $(r^+, r^-) \in \tilde{R} \cup R \cup \bar{R}$
t_{ij}	Travel time from node i to node j
d_{ij}	Travel distance from node i to node j
q_j	Vehicle load change at node j
Q_{max}	Vehicle capacity
M	A sufficiently large positive constant
λ	Weighting factor for time window expansions in the objective function
$\bar{w}_\sigma$	The initial vehicle load at node σ
$\bar{\tau}_\sigma$	Current time
η_1	Fixed buffer time for a rider to walk to the pickup location
η_2	Fixed pickup time window width
δ_{max}	Maximum time window expansion for a new rider
Decision Variables	
x_{ij}	Binary variable, which is 1 if the vehicle travels from node i to node j; 0, otherwise
u_i	A nonnegative variable to represent the service start time at node i
w_i	A nonnegative variable representing the vehicle load after visiting node i
δ_i, δ_j	Magnitudes of time window expansions in one direction at the pickup location i and drop-off location j for rider $(i, j) \in \bar{R}$

Given the above problem statements, we define the network underlying the base problem, which is denoted as $G = \{N, A\}$ and illustrated in Figure 3. The set of nodes N consists of the current vehicle location σ, the return depot o^-, all nodes in tuple S_f, and nodes associated with new riders, described as follows. For the new riders in set $\bar{R}$, we define set $\bar{P}$ that contains all new riders' pickup nodes $\bar{r}^+$ and set $\bar{D}$ that contains new riders' drop-off nodes $\bar{r}^-$. Then, all nodes in $\bar{P} \cup \bar{D}$ are included in set N. Each pair of distinct nodes in N is connected with a directed arc (i, j) unless i is the drop-off location and j is the pickup location for the same rider or i is the return depot and j is the current vehicle location. Note that not all new arcs are shown and only a single new rider is shown for illustration purposes in Figure 3. Each arc $(i, j) \in A$ is associated with a travel time t_{ij}. The base optimization problem is defined on graph G with the objective of minimizing the total travel distance and the impact of time window expansion. The complete formulation for the base problem using notation in Table 2 is thus presented as follows:

$$\underset{\{x_{ij},u_i,w_i,\delta_i,\delta_j\}}{\text{Minimize}} \quad \sum_{(i,j)\in A} d_{ij}x_{ij} + \lambda \sum_{(i,j)\in \bar{R}} (\delta_i + \delta_j) \tag{3.1}$$

$$\text{s.t.} \quad \sum_{j\in N} x_{ij} = 1, \quad \forall i \in N \setminus \{o^-\}, \tag{3.2}$$

$$\sum_{i\in N} x_{ij} = 1, \quad \forall j \in N \setminus \{\sigma\}, \tag{3.3}$$

$$u_i + s_i + t_{ij} - (1 - x_{ij})M \leq u_j, \quad \forall (i, j) \in A \tag{3.4}$$

$$e_i \leq u_i \leq l_i, \quad \forall i \in N \setminus \{\sigma\} \setminus \bar{P} \setminus \bar{D} \tag{3.5}$$

$$e_i - \delta_i \leq u_i \leq l_i + \delta_i, \quad i \in \bar{P} \tag{3.6}$$

$$0 \leq \delta_i \leq \delta_{max}, \quad \forall i \in \bar{P} \tag{3.7}$$

$$0 \leq \delta_j \leq \delta_{max}, \quad \forall j \in \bar{D} \tag{3.8}$$

$$u_\sigma = \bar{\tau}_\sigma, \tag{3.9}$$

$$u_i + s_i + t_{ij} \leq u_j, \quad \forall (i, j) \in R \cup \bar{R} \tag{3.10}$$

$$u_j - (u_i + s_i) \leq L_{(r^+,r^-)}, \quad \forall (i, j) \in R \cup \bar{R} \tag{3.11}$$

$$w_\sigma = \bar{w}_\sigma, \tag{3.12}$$

$$w_j \geqslant w_i + q_j - M(1 - x_{ij}), \quad \forall (i, j) \in A \tag{3.13}$$

$$w_i \leq Q_{\text{max}}, \quad \forall i \in N \tag{3.14}$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A \tag{3.15}$$

$$u_i \geq 0, \quad \forall i \in N \tag{3.16}$$

The objective Eq. (3.1) minimizes the weighted sum of total routing distance $\sum_{(i,j)\in A} d_{ij}x_{ij}$ and

time window expansion cost $\sum_{(i,j)\in\bar{R}}(\delta_i+\delta_j)$. The weighting factor λ in the objective function determines the trade-off between routing efficiency (measured by routing cost) and service level for new riders (measured with the time window expansion and the associated trip disruption).

Constraint (3.2) ensures that for all nodes except for the return depot o^-, the outflow should be one unit. Similarly, Constraint (3.3) ensures that all nodes except for the current location σ, the inflow should be one unit. Constraint (3.4) states that when arc (i,j) is visited, $u_i+s_i+t_{ij}\leq u_j$; the positive increase in node visit time thus prevents subtours. Constraint (3.5) ensures that the vehicle must visit node i between e_i and l_i for an unvisited node in the given route, namely nodes in tuple S_f. Constraint (3.6) ensures that the time window $[e_i,l_i]$ at node i can be expanded to $[e_i-\delta_i,l_i+\delta_i]$, which will incur a penalty in the objective in the amount of $\lambda\delta_i$. For practical reasons, the magnitude of the expansion is bounded above by δ_{max}, as enforced by Constraints (3.7) and (3.8). δ_{max} represents the maximum time window expansion from its initial scheduled value which is predetermined by policy, such as 5 minutes or 10 minutes. Constraint (3.9) specifies the node visit time at location σ as the current time $\bar{\tau}_\sigma$. Constraint (3.10) enforces that for each scheduled rider in R and the new rider, the drop-off location can be visited only after the pickup location. Constraint (3.11) imposes the maximum ride time for each rider in R and new riders. Note that for each onboard rider in $\tilde{R}$, we do not need to impose the maximum ride time limit because enforcing the delivery time window for an onboard rider sufficiently ensures the compliance with the maximum ride time limit. Constraint (3.12) initializes the number of onboard riders at the current time. Constraints (3.13) and (3.14) ensure that the vehicle's load cannot exceed its capacity at any time. Constraints (3.15) and (3.16) define decision variables x_{ij} and u_i to be binary and continuous, respectively.

3.2 Extension to Multiple Vehicles

In practice, the operator rarely optimizes one vehicle in isolation. When several routes are concurrently en route, multi-vehicle assignment of new requests can (i) eliminate long detours on an already busy vehicle, (ii) avoid denials by tapping spare capacity elsewhere, and (iii) reduce total deadhead through better geographic matching. A fleet coordination is therefore needed to jointly decide which vehicle accepts each new request and how each chosen route is locally re-optimized. The multi-vehicle model preserves the same routing which each vehicle has a fixed visited route and assigned future route and TWE remains available only for new riders under the certain cap and penalty. Additionally, coupling constraints ensure that each accepted new rider is served by exactly one vehicle and that pickup and drop-off occur on the same route.

We may allow idle vehicles at the depot to enter service if they offer a cheaper or more reliable way to absorb new demand. This is frequent in rural/small-urban settings where spatial dispersion

makes cross-coverage efficient only for certain OD patterns. Conversely, if all active vehicles can feasibly absorb the requests with small TWE, the optimizer may choose not to dispatch an idle unit, preserving operator-hours.

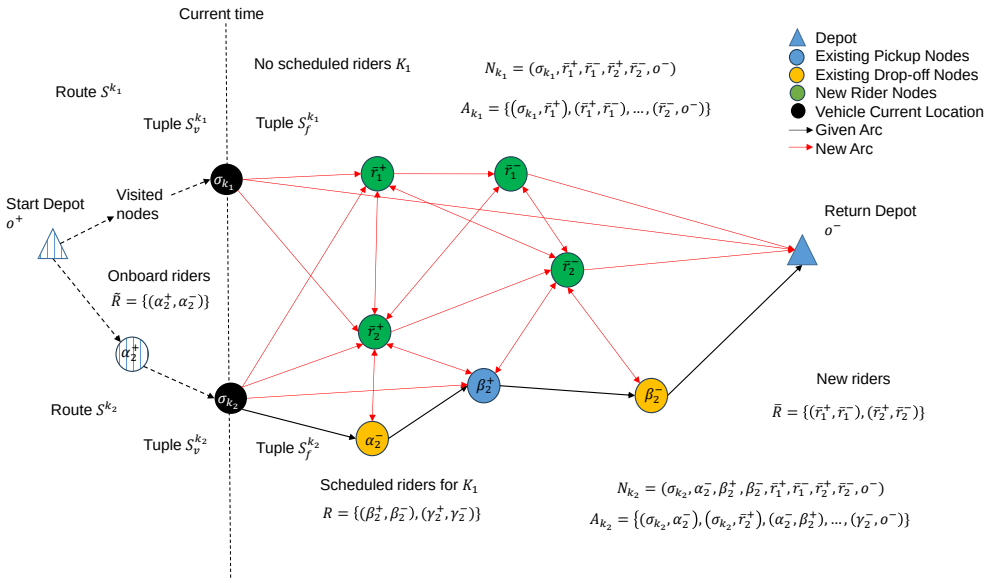


Figure 4: Illustration of the extended problem network

We denote a set of vehicles as K, each of which $k \in K$ is associated with an existing route, denoted as S^k. Note that set K may include an idle vehicle at the depot, which can be dispatched in case that none of the existing vehicles can feasibly or economically cover one or several requests. Vehicle k has a set of nodes N^k, consisting of the current vehicle location σ_k, return depot o^-, all to-be-visited nodes in tuple S_f^k, and nodes associated with new riders. Given N^k, the set of arcs for vehicle k, denoted as A^k, is constructed as follows: create a directed arc (i, j) to connect nodes i and j in N^k, except when i and j are the drop-off and pickup nodes for the same rider or i is the return depot and j is the current vehicle location.

Figure 4 illustrates the underlying network for each vehicle. Vehicle k_1 is currently idle and can go to the pickup location of each new rider or directly to the return depot. By contrast, vehicle k_2 has one onboard and one to-be-visited rider. Either vehicle can serve zero, one, or two new riders.

As shown in the Figure 4, the multi-vehicle network has a set of vehicle K indexed by k. For each vehicle $k \in K$, it has a set of all possible nodes N^k consists of the current vehicle location σ_k, the return depot o^-, all nodes in tuple S_f^k, and nodes associated with new riders and a set of arcs A^k consists all possible feasible arcs connecting those nodes. Note that not all possible arcs in A^k are shown Figure 4. In the extension problem, in order to insert all the new riders into all current routes S_f^k, we want the solver to select the best possible arcs in A^k. Each vehicle will be reassigned and have a new route in order to serve all new riders' requests, subject to capacity, time window,

and maximum ride-time constraints.

Following the above definition and with notation in Table 3, the multi-vehicle formulation is written as follows:

Table 3: Additional notation for the multi-vehicle formulation

Indices and Sets	
K	Set of vehicles, indexed by k
σ_k	Current location of vehicle k, at current time τ_σ
S^k	Given route of vehicle k
S_v^k	Tuple of visited nodes by vehicle k before current time
S_f^k	Tuple of future nodes to be visited by vehicle k after current time
$\tilde{R}^k$	Set of onboard riders for vehicle k, each of whom is indexed by (r_k^+, r_k^-)
R^k	Set of scheduled riders for vehicle k, each of whom is indexed by (r_k^+, r_k^-)
$\bar{R}^k$	Set of new riders for vehicle k, each of whom is indexed by $(\bar{r}_k^+, \bar{r}_k^-)$
N^k	Set of relevant nodes for vehicle k
A^k	Set of feasible arcs for vehicle k
Parameters	
$\bar{w}_\sigma^k$	Current load of vehicle k
Decision variables	
x_{ij}^k	Binary variable to be 1 if vehicle k travels from node i to node j; 0, otherwise
u_i^k	Service start time at node i by vehicle k
w_i^k	Load of vehicle k after completing service at node i

$$\underset{\{x_{ij}^k, u_i^k, w_i^k, \delta_i, \delta_j\}}{\text{Minimize}} \quad \sum_{k\in K}\sum_{(i,j)\in A^k} d_{ij}\, x_{ij}^k + \lambda \sum_{(i,j)\in \bar{R}} (\delta_i + \delta_j) \tag{3.17}$$

$$\text{s.t.} \quad \sum_{j\in N^k} x_{ij}^k = 1, \quad \forall k \in K, i \in N^k \setminus \{o_k^-\} \setminus \bar{P} \setminus \bar{D}, \tag{3.18}$$

$$\sum_{i\in N^k} x_{ij}^k = 1, \quad \forall k \in K, j \in N^k \setminus \{\sigma_k\} \setminus \bar{P} \setminus \bar{D}, \tag{3.19}$$

$$\sum_{k\in K}\sum_{j\in N^k} x_{\bar{r}^+,j}^k = 1, \quad \forall (\bar{r}^+, \bar{r}^-) \in \bar{R}^k, \tag{3.20}$$

$$\sum_{k\in K}\sum_{i\in N^k} x_{i,\bar{r}^-}^k = 1, \quad \forall (\bar{r}^+, \bar{r}^-) \in \bar{R}^k, \tag{3.21}$$

$$\sum_{j\in N^k} x_{\bar{r}^+,j}^k = \sum_{i\in N^k} x_{i,\bar{r}^-}^k \quad \forall k \in K, (\bar{r}^+, \bar{r}^-) \in \bar{R}^k, \tag{3.22}$$

$$u_i^k + s_i + t_{ij} - \left(1 - x_{ij}^k\right) M \le u_j^k, \quad \forall k \in K, (i,j) \in A^k, \tag{3.23}$$

$$e_i \le u_i \le l_i, \quad \forall i \in N^K \setminus \{\sigma\} \setminus \bar{P} \setminus \bar{D}, \tag{3.24}$$

$$e_i - \delta_i \le u_i \le l_i + \delta_i, \quad i \in \bar{P} \cup \bar{D}, \tag{3.25}$$

$$0 \le \delta_i \le \delta_{max}, \quad \forall (i,j) \in \bar{R}, \tag{3.26}$$

$$0 \le \delta_j \le \delta_{max}, \quad \forall (i,j) \in \bar{R}, \tag{3.27}$$

$$u_{\sigma_k}^k = \tau_\sigma, \quad \forall k \in K, \tag{3.28}$$

$$u_i^k + s_i + t_{ij} \le u_j^k, \quad \forall (i,j) \in R^k \cup \bar{R}^k \tag{3.29}$$

$$u_j^k - (u_i^k + s_i^k) \le L_{(r^+,r^-)} + \left(1 - x_{ij}^k\right) M, \quad \forall k \in K, \left(r^+, r^-\right) \in \tilde{R}^k \cup R^k \cup \bar{R}^k, (i,j) \in A^k \tag{3.30}$$

$$w_{\sigma_k}^k = \bar{w}_\sigma^k, \quad \forall k \in K, \tag{3.31}$$

$$w_j^k \geqslant w_i^k + q_j - M(1 - x_{ij}^k), \quad \forall k \in K, (i,j) \in A^k, \tag{3.32}$$

$$w_i^k \le Q_{\max}, \quad \forall k \in K, i \in N^k, \tag{3.33}$$

$$x_{ij}^k \in \{0,1\}, \quad \forall k \in K, (i,j) \in A^k, \tag{3.34}$$

$$u_i^k \ge 0, \quad \forall k \in K, i \in N^k, \tag{3.35}$$

$$w_i^k \ge 0, \quad \forall k \in K, i \in N^k, \tag{3.36}$$

The optimization objective Eq. (3.17) minimizes the total cost over all vehicles in K and time window expansion costs. Constraints (3.18) and (3.19) together ensure that each vehicle k leaves from the current location once, arrives at the return depot once, and visits (i.e., arrives at and leaves from) other nodes associated with riders in $\tilde{R}^k \cup R^k$ once. New riders' pickup or drop-off nodes are excluded from constraints (3.18) and (3.19), as vehicle k is not required to visit them. Con-

straint (3.20) ensures that exactly one vehicle in K leaves from the pickup nodes of new rider $(\bar{r}^+, \bar{r}^-) \in \bar{R}$. Constraint (3.21) ensures exactly one vehicle arrives at the drop-off node of a new rider. Constraint (3.22) guarantees that the same vehicle leaves from $\bar{r}^+$ and arrives at $\bar{r}^-$, thus serving new rider $(\bar{r}^+, \bar{r}^-) \in \bar{R}$. Constraint (3.23) enforces an increase in node visit time, namely, $u_j^k \geq u_i^k + s_i + t_{ij}$, if an arc is visited by vehicle k. Constraint (3.24) enforces time window constraints for onboard and to-be-visited riders. Constraint (3.25) models the time window expansion at each new rider's pickup and drop-off location. Constraints (3.26) and (3.27) limit the time window expansion for each new pickup node $\bar{r}^+$ and drop-off node $\bar{r}^-$. Constraint (3.28) specifies the visit time $u_{\sigma_k}^k$ to the known current time τ_σ for vehicle k. Constraint (3.29) ensures that each scheduled rider in R^k and every new rider in $\bar{R}$ has their pickup location visited before their drop-off location. Constraint (3.30) imposes a maximum ride time for each new rider and M is to make sure only the chosen arcs will have this constraint. Constraint (3.31) initializes the current vehicle load. Constraints (3.32) and (3.33) ensure that vehicle k's load cannot exceed capacity Q_{max} at any time. Constraint (3.34) defines decision variable x_{ij}^k as binary. Constraints (3.35) and(3.36) define decision variables u_i^k and w_i^k to be continuous.

4. COMPUTATIONAL EXPERIMENTS

In this section, we describe a series of numerical experiments conducted to evaluate the performance of both the single-vehicle and multi-vehicle models introduced earlier. We begin by outlining the computational environment, including the hardware and software settings. Subsequently, we discuss how problem instances of varying sizes and complexity are generated. The experiments are divided into two main parts: single-vehicle and multi-vehicle tests. Finally, we provide a brief discussion highlighting the principal insights drawn from the results. All experiments are performed on a personal computer equipped with an Intel Core i9-12900H processor (2.50 GHz) and 32 GB RAM. The operating system is Windows 11, and mixed integer linear problems are directly solved using CPLEX.

4.1 Evaluation Metrics and Baselines

Performance metrics. Consistent with the modeling objectives, we report:

- the optimized objective value z (weighted sum of route distance and TWE penalties);
- total routing distance $\sum_{(i,j)} d_{ij}x_{ij}$;
- total time window expansion (TWE) $\sum_{(i,j)\in\bar{R}}(\delta_i + \delta_j)$ and per-new-rider averages;
- computation time (wall clock seconds) and solver termination status;
- instance descriptors that drive complexity: number of requests, $|N|$ and $|A|$.

Note that in our formulations (Sections 3.1–3.2) all new riders must be served; acceptance is not a decision variable. Thus, TWE functions as the safety valve that restores feasibility when strict windows would otherwise preclude service.

Comparative settings. We contrast:

1. No flexibility (TWE-off): $\delta_{\text{max}} = 0$, quantifying how often strict windows force costly detours or infeasibility;
2. Bounded flexibility (TWE-on): the policy setting used in our main tests with $\delta_{\text{max}} > 0$ and penalty weight λ;
3. Independent vs. Coordinated fleets: the Independent Scenario (each vehicle handles only its own new requests) vs. the Coordinated Scenario (new requests are pooled and assigned network-wide).

4.2 Synthetic Data Generation

We first describe how travel requests are generated. Uniformly distributed nodes are sampled from a 100×100 coordinate plane as riders' pickup or drop-off locations. The distance d_{ij} between any two locations, e.g., i and j, is estimated as Euclidean distance; the average travel speed is 4, which yields the travel time between two locations, namely t_{ij}. Each pickup node i is associated with a time window (e_i, l_i). As the vehicle operating period is $(600, 1440)$, the left time window e_i is sampled from $(700, 1320)$ and $l_i = e_i + \eta_2$, with $\eta_2 = 20$. The corresponding drop-off time window (e_j, l_j) is computed as $e_j = e_i + t_{ij}$ and $l_j = l_i + 2 \cdot t_{ij}$. The above time window setup ensures either the earliest pickup time or the latest drop-off time of a rider can be accommodated. After generating H rider requests following the above procedure, where H is the total number of requests to be specified later, we next select those requests relevant to an instance, namely determining sets $\tilde{R}$, R, and $\bar{R}$.

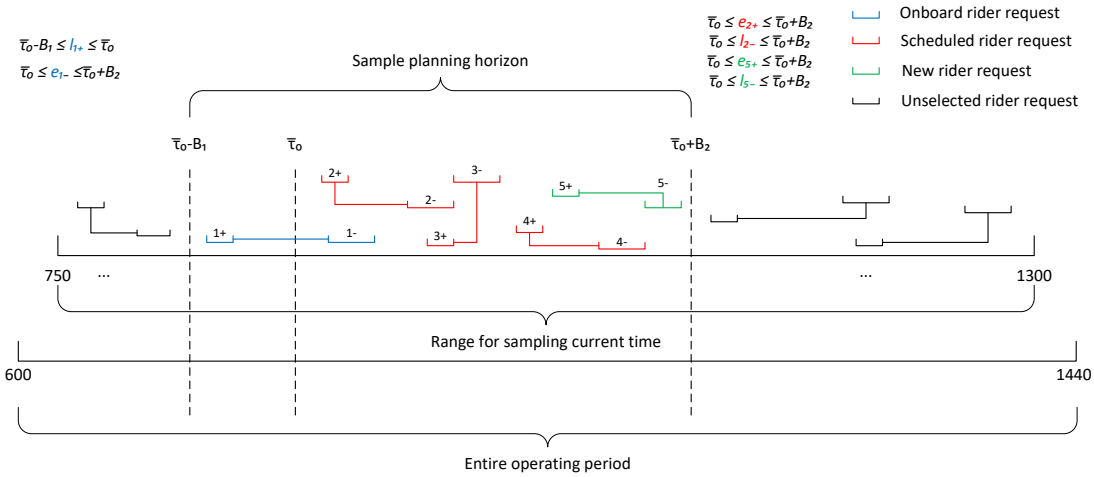


Figure 5: Illustration of time scheme

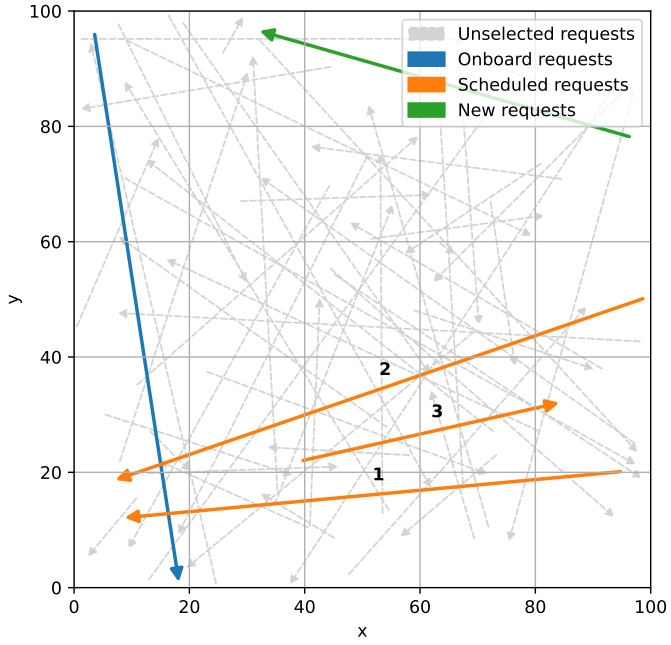
As shown in Figure 5, we build a planning horizon $[\bar{\tau}_\sigma - B_1, \bar{\tau}_\sigma + B_2]$ after introducing two known buffers namely B_1 and B_2. Here, the current time $\bar{\tau}_\sigma$ is randomly selected from range $(750, 1300)$. For each of H sampled requests, denoted as (r^+, r^-), we determine its type, namely classifying it into one of sets $\tilde{R}$, R, and $\bar{R}$. When the latest pickup time l_{r^+} is earlier than the current time, i.e., $\bar{\tau}_\sigma - B_1 < l_{r^+} < \bar{\tau}_\sigma$, and the earliest drop-off time is between $\bar{\tau}_\sigma$ and $\bar{\tau}_\sigma + B_2$, namely, $\bar{\tau}_\sigma < e_{r^-} < \bar{\tau}_\sigma + B_2$, the request (r^+, r^-) is classified as onboard and thus part of set $\tilde{R}$. When the earliest pickup time and the latest drop-off time are both within $[\bar{\tau}_\sigma, \bar{\tau}_\sigma + B_2]$, namely $\bar{\tau}_\sigma < e_{r^+} < \bar{\tau}_\sigma + B_2$ and $\bar{\tau}_\sigma < l_{r^-} < \bar{\tau}_\sigma + B_2$, we randomly designate the request with probability p as a new request (thus in set $\bar{R}$); otherwise, the request is classified as scheduled (thus in set R). Figure 6 illustrates five sampled requests relevant to a planning horizon from a total of 60 requests

over the entire operating period, and detailed schedule information (mainly time windows) for each sampled request.

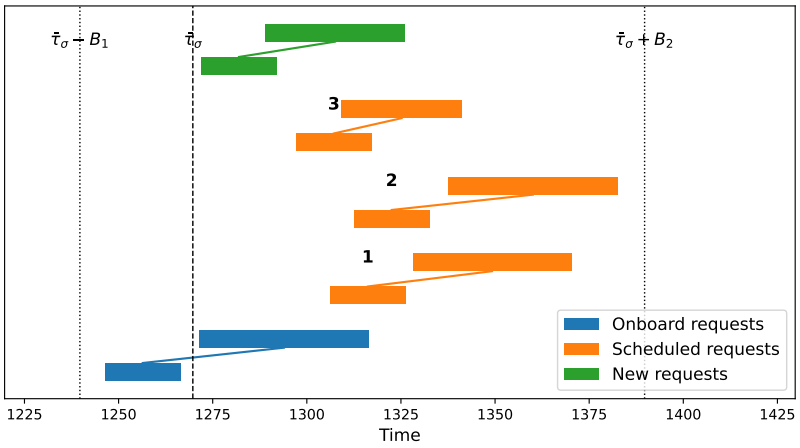
After selecting and classifying requests relevant to a planning horizon, we build the set of network nodes N by (1) considering all relevant pickup and drop-off nodes, (2) setting the current vehicle location σ at the middle point of one randomly selected onboard rider request or at location $(40, 40)$ in case there are no onboard riders, and (3) fixing return depot o^- at location $(50, 50)$. The procedure in the Base Problem section is used to construct the set of arcs A.

To ensure there is a feasible initial route S covering riders in set $\tilde{R} \cup R$, we conduct time window compatibility checking as follows. All nodes in N excluding new rider nodes in $\bar{P} \cup \bar{D}$ and the return depot o^- are first chronologically sequenced by the left time window. Starting from the current location σ, we sequentially append nodes to route S if the vehicle can reach the node within its allowable time window (before the right time window) and still satisfy all precedence and capacity constraints. If the time window is violated for any node, the current instance is discarded. The above checking, if passed, ensures that all selected requests in $\tilde{R}$ and R can be served in sequence without violating time windows or capacity limits.

To systematically evaluate the performance of the model under different operational settings, we construct a series of synthetic instance groups by varying two key parameters: the number of rider requests H and buffer B_2, which determines the total number of scheduled and new requests for any given H. Specifically, we consider three levels for H (i.e., $H \in \{30, 60, 90\}$), two levels for B_2 (i.e., $B_2 \in \{90, 120\}$ resulting in a total of six instance groups. For each group, we generate five random instances, yielding a total of 30 single-vehicle instances. Each instance is labeled as SV-H-B_2-m, where "SV" is short for Single-Vehicle and m is a serial number in each instance group. Table 4 further provides baseline values for other parameters.



(a) Selected requests among all requests in one operating day



(b) Instance time scheme

Figure 6: Single-vehicle instance illustration

Table 4: Summary of experimental parameters

Parameter	Description	Value
q_j	Vehicle load change at a node	{-1, 0, 1}
B_1	Pre-buffer time	15
B_2	Post-buffer time	{90,120}
H	Number of requests in entire operating period	{30,60,90}
p	Classification probability	0.25
s	Service time per rider node	1
η_2	Pickup time window width	20
δ_{max}	Max time window expansion	20
Q_{max}	Vehicle capacity	6
λ	Penalty weight	{0.1,0.5,1}
M	A sufficiently large constant	1,000

We next construct multi-vehicle instances by adapting the above single-vehicle instance generation procedure.

For each vehicle in a K-vehicle fleet, we independently generate a single-vehicle instance from a pool of H requests. All K single-vehicle instances have the same current time and planning horizon $[\bar{\tau}_\sigma - B_1, \bar{\tau}_\sigma + B_2]$. Nonetheless, each vehicle has its unique set of onboard riders $\tilde{R}^k$ and a unique set of scheduled riders R^k; none of onboard or scheduled riders could appear in two single-vehicle instances. The set of new requests is defined as $\bar{R} = \cup_k \bar{R}^k$, which means all vehicles share the same pool of new requests. We build the set of network nodes N^k for each vehicle k using the same way as in the single-vehicle instance which determine its own current location σ_k and the fixed return depot o_k^- located at $(50, 50)$. Then, the set of arcs A^k is built using the procedure in the Extension to Multiple Vehicles section considering all possible nodes in N^k. Other parameter values remain the same as in the single-vehicle instance.

We eventually build six instance groups by fixing H at 60, and varying two key parameters: post-buffer B_2 (i.e., $B_2 \in \{60, 90\}$) and the fleet size K (i.e., $k \in \{3, 4, 5\}$). Similar to single-vehicle instances, each group has five random instances, each of which has a serial number m. Therefore, multi-vehicle instances are labeled as MV-B_2-K-m, where "MV" is short for Multi-Vehicle.

4.3 Single-Vehicle Experiments

4.3.1 Benchmark Instance Analysis We begin our analysis by choosing a simple benchmark instance SV-60-90-2, which has only one onboard rider (labeled as $\tilde{1}$) and two scheduled riders (labeled as 2 and 3). Two new riders ($\bar{4}$ and $\bar{5}$) need to be accommodated. Figure 7 shows the initial and optimized routes under different values of λ. When λ is relatively small, such as 0.1, substantial time window expansions are used to keep the total distance of the new route small (269.1). Notably, the latest drop-off time must be postponed by the maximum extent, namely 20, to enable efficient routing. When $\lambda = 0.5$, the same route remains optimal. By contrast, when λ further increases to 1.0, time window expansions are eliminated. In this case, a very different route is generated to avoid adjusting the drop-off time window for new rider $\bar{4}$. As the route distance increases dramatically, the optimization objective (a weighted sum of route distance and time window expansion penalty, as defined in Eq. (3.1)) increases, due to an increasing λ value.

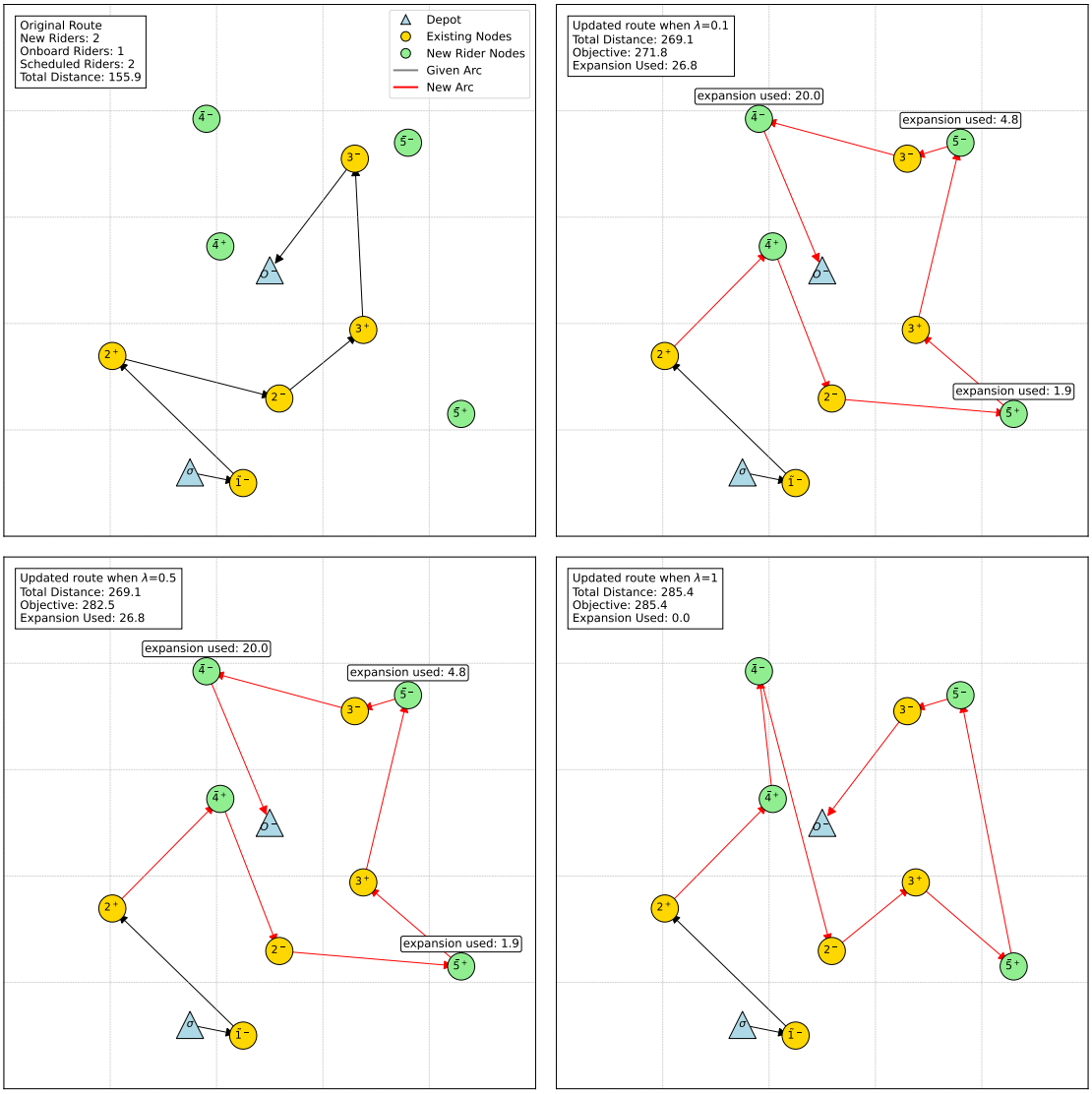


Figure 7: Optimization results from instance SV-60-90-2

4.3.2 Second Instance Analysis While in the above instance SV-60-90-2, a high λ value eventually drives time window expansions to zero, this is not always possible when new riders must be served, as shown in the next example. Instance SV-60-90-4 has three scheduled riders (labeled as 1, 2 and 3) but no onboard riders. Two new riders ($\bar{4}$ and $\bar{5}$) need to be accommodated. Figure 8 shows how the optimization results vary with λ. When λ is very small, substantial time window expansions are used, in order to pick up and drop off new rider $\bar{5}$ before picking up scheduled rider 1. When λ increases to 0.5, the total time window expansion reduces, and the route is revised to pick up scheduled rider 1 before serving new rider $\bar{5}$. When λ is at or above 1.0, a very different route emerges as the optimum one, while time window expansions remain necessary. In other words, in Instance SV-60-90-4, new rider $\bar{5}$ cannot be served, unless time window expansions are allowed.

The benefit of time window expansion is to make an infeasible route feasible, thus avoiding rider service rejection, albeit at additional expansion costs.

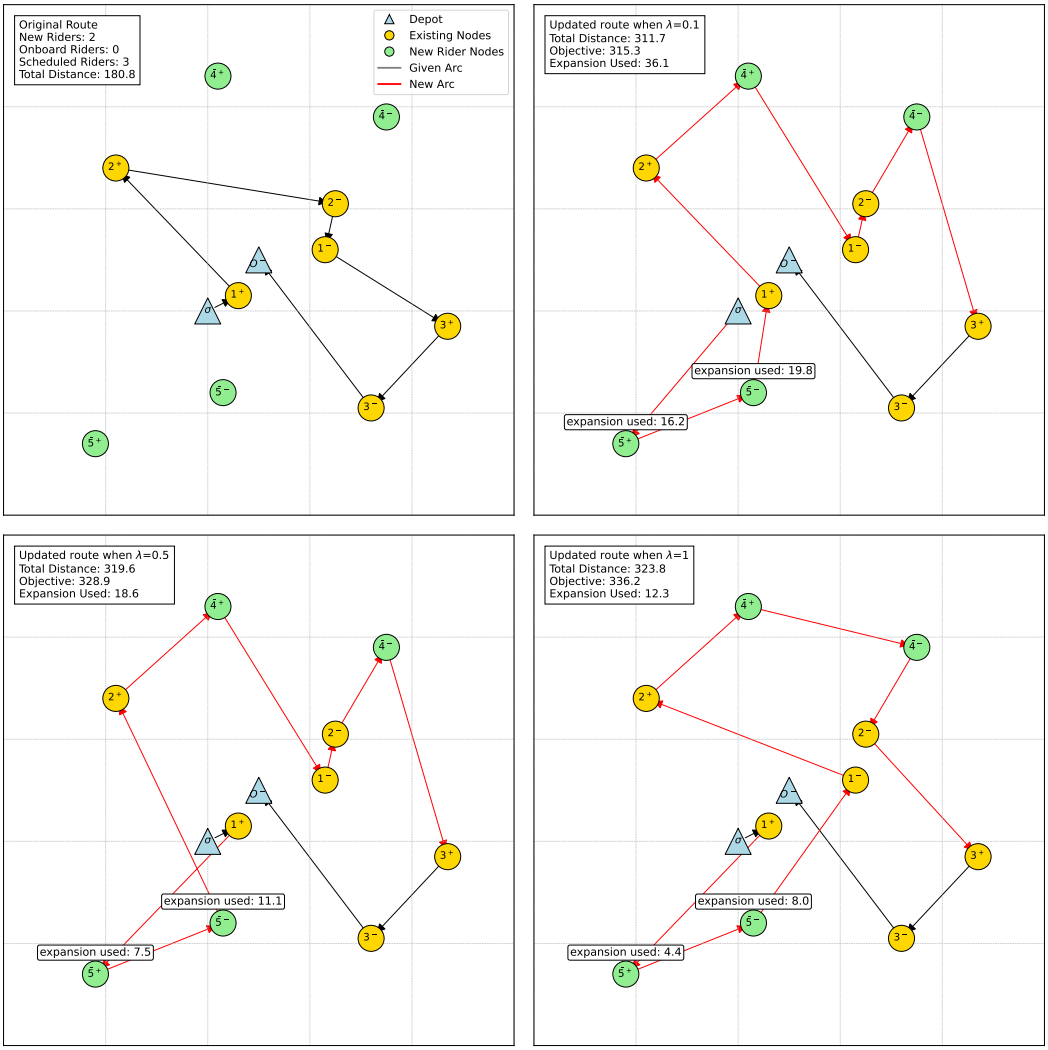


Figure 8: Optimization results from instance SV-60-90-4

4.3.3 Computational Performance of Single-Vehicle Model After understanding the impact of λ on the optimization results, we next fix λ at 0.5 and solve all instances in six instance groups, described earlier. Table 5 show that the computation time increases with both the total number of requests H and post-buffer time B_2. Small instances, such as those consisting of fewer than ten requests when $H = 60$ and $B_2 = 90$, can be solved in less than five seconds; by contrast, larger instances with around 20 requests can take more than 1,000 seconds. Clearly, the computation time grows nonlinearly with the problem size. We further note that when compared to the number of requests, increases in the number of nodes $|N|$ and number of arcs $|A|$ tend to have a more pronounced impact on the computation time as the Pearson's correlation coefficient between the total

number of requests and computation time is 0.66; by contrast, the correlation coefficient between the number of arcs $|A|$ and the computation time is 0.82, which is also higher than the correlation coefficient (0.71) between the number of nodes $|N|$ and the computation time.

Table 5: Comparison of instance statistics and computation time (single-vehicle)

Instance	Total Requests	$\lvert\tilde{R}\rvert$	R	$\lvert\bar{R}\rvert$	$\lvert N\rvert$	$\lvert A\rvert$	Objective	Comp. Time
SV-30-90-1	3	0	2	1	8	52	164.2	1
SV-30-90-2	3	1	1	1	7	39	161.3	1
SV-30-90-3	3	0	1	2	8	52	171.7	1
SV-30-90-4	3	0	2	1	8	52	186.5	1
SV-30-90-5	3	0	2	1	8	52	141.2	2
SV-30-120-1	4	1	2	1	9	68	201.1	1
SV-30-120-2	4	0	3	1	10	85	253.3	2
SV-30-120-3	3	0	2	1	8	52	193.4	2
SV-30-120-4	3	0	1	2	8	52	214.2	1
SV-30-120-5	4	1	2	1	9	68	213.8	2
SV-60-90-1	5	1	3	1	11	105	274.4	3
SV-60-90-2	5	1	2	2	11	105	282.5	4
SV-60-90-3	7	2	3	2	14	176	325.1	3
SV-60-90-4	5	0	3	2	12	126	328.9	3
SV-60-90-5	6	1	4	1	13	150	260.0	2
SV-60-120-1	9	2	5	2	18	298	456.1	11
SV-60-120-2	10	3	4	3	19	334	557.8	12
SV-60-120-3	9	3	4	2	17	265	508.5	13
SV-60-120-4	9	2	5	2	18	298	498.6	15
SV-60-120-5	9	2	5	2	18	298	430.9	8
SV-90-90-1	10	3	4	3	19	334	557.8	14
SV-90-90-2	13	4	5	4	24	542	535.2	22
SV-90-90-3	13	5	6	2	23	497	509.5	18
SV-90-90-4	13	4	7	2	24	542	621.0	19
SV-90-90-5	12	3	6	3	23	496	649.2	21
SV-90-120-1	20	6	10	4	34	1107	1203.5	189
SV-90-120-2	23	4	14	5	44	1872	1403.8	1455
SV-90-120-3	22	5	12	5	41	1622	1350.7	1234
SV-90-120-4	20	7	11	2	35	1176	1024.6	125
SV-90-120-5	18	5	10	3	33	1042	1198.2	98

4.3.4 Sensitivity to time window Policy Parameters (Single Vehicle) We summarize qualitative responses observed across instance groups:

- **Penalty weight λ.** Larger λ discourages TWE, leading the solver to prefer longer detours and, in some cases, different visit orders. When λ is small, modest expansions are frequently used to unlock pooling opportunities and shorten distance.

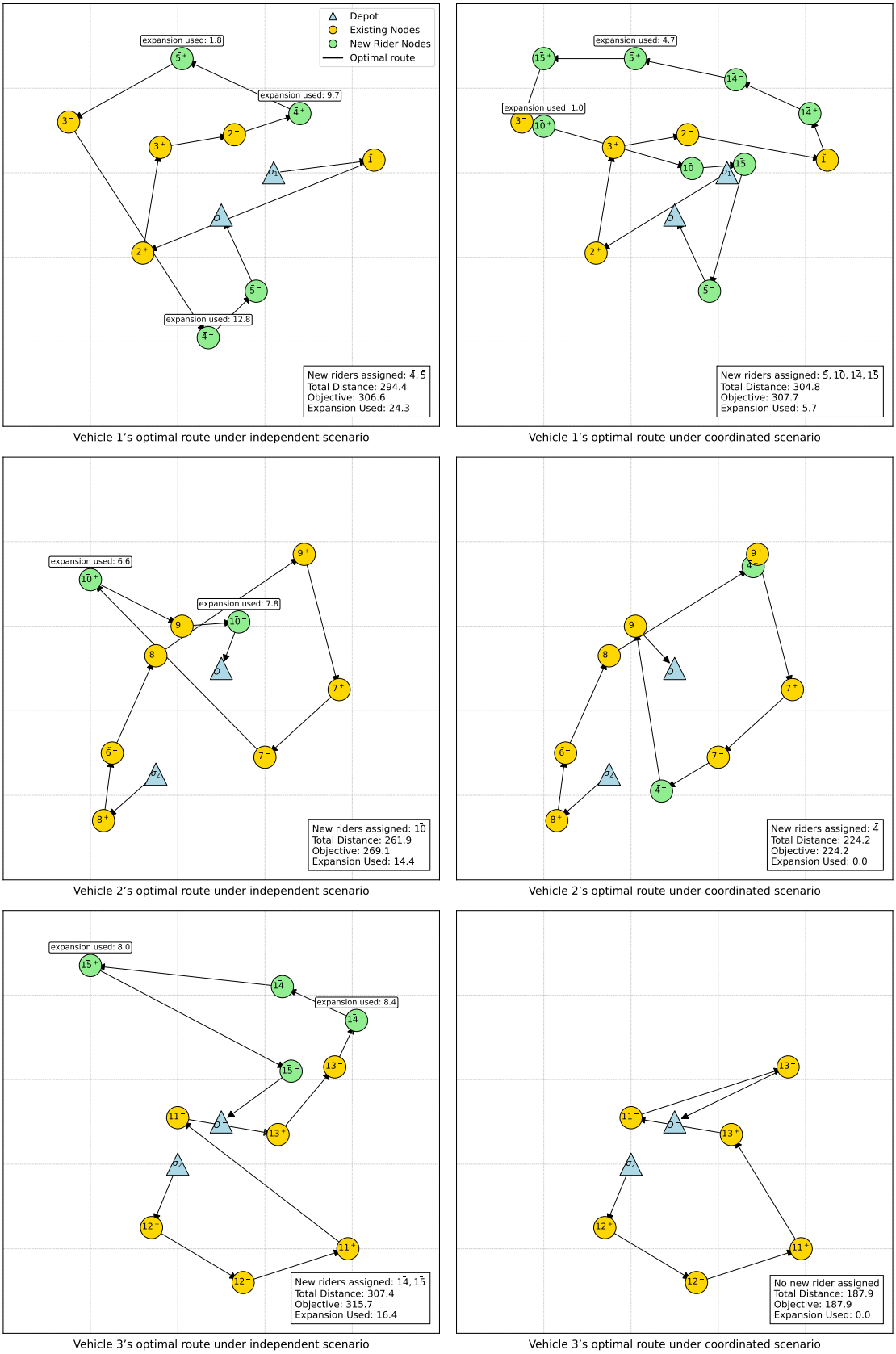
- **Expansion cap $\delta_{\max}$.** Tight caps limit feasibility restoration; when demand is bursty, small

increases in $\delta_{\max}$ can transform an infeasible schedule into a feasible one with negligible expansion per rider.

4.4 Multi-Vehicle Experiments

We illustrate the advantage of multi-vehicle optimization with an example instance MV-90-3-1, consisting of three vehicles and five new rider requests ($\bar{4}$, $\bar{5}$, $1\bar{0}$, $1\bar{4}$, and $1\bar{5}$). As a benchmark, we solve three separate single-vehicle optimization problems (in the so-called *Independent Scenario*), where each vehicle has its own new riders. Then, we solve a single multi-vehicle optimization problem (in the so-called *Coordinated Scenario*), where a common set of new rider requests is available to each of three vehicles. Figure 9 shows the detailed optimization results, including route and schedule adjustments, under two different scenarios. Table 6 further presents key statistics about problem size and cost metrics. Clearly, sharing new requests among vehicles in the Coordinated Scenario significantly reduces the total cost (by 19.2%), total time window expansions (by 89.7%), and total route distance (by 17.0%), with the Independent Scenario as the benchmark.

In Table 7, we seek to highlight the major difference in request-to-vehicle assignment between two scenarios. In the Independent Scenario, five new riders are distributed across all three vehicles for service, and each vehicle is allowed to cover its received requests only. For instance, new rider $1\bar{0}$ is accessible by vehicle 2 only, not by vehicle 1. In the Coordinated Scenario, new rider requests can be assigned to any suitable vehicle, which leads to very different request assignment results, shown in Table 7. Here, new rider $1\bar{0}$ is to be served by vehicle 1. Those cost reductions observed in Table 6 are largely attributed to the new request-to-vehicle assignment, given the more suitable match between new request characteristics and existing vehicle route attributes. The improved match between vehicles and rider requests is also evidenced by greatly reduced TWE in the Coordinated Scenario (see Figure 9).



Vehicle 1's optimal route under independent scenario

Vehicle 1's optimal route under coordinated scenario

Vehicle 2's optimal route under independent scenario

Vehicle 2's optimal route under coordinated scenario

Vehicle 3's optimal route under independent scenario

Vehicle 3's optimal route under coordinated scenario

Figure 9: Comparison of optimization results under two scenarios

Table 6: Comparison of single-vehicle and multi-vehicle optimization ($\lambda = 0.5$)

Scenario	Vehicle Number	Total req.	$\lvert\tilde{R}\rvert$	$\lvert R\rvert$	$\lvert\bar{R}\rvert$	Objective z	TW expansion	Total distance
Independent	1	5	1	2	2	306.6	24.3	294.4
	2	5	1	3	1	269.1	14.4	261.9
	3	5	0	3	2	315.7	16.4	307.4
	Total	15	2	8	5	891.4	55.1	863.7
Coordinated	1	7	1	2	4	307.7	5.7	304.8
	2	5	1	3	1	224.2	0.0	224.3
	3	3	0	3	0	187.9	0.0	187.9
	Total	15	2	8	5	719.8	5.7	717.0
Difference (%)	-	–	–	–	–	-19.2	-89.7	-17.0

Table 7: New rider assignment under different scenarios

Scenario	Vehicle ID	Assigned New Riders ($\bar{r}$)
Independent	1	$\bar{4}$, $\bar{5}$
	2	$\bar{10}$
	3	$\bar{14}$, $\bar{15}$
Coordinated	1	$\bar{5}$, $\bar{10}$, $\bar{14}$, $\bar{15}$
	2	$\bar{4}$
	3	–

We next explore the impact of post-buffer B_2 and fleet size K on the total computation time. Table 8 presents some key statistics (e.g., number of scheduled requests) for each multi-vehicle instance and the corresponding solution time. $|N|$ and $|A|$ here represent the total number of nodes and the total number of arcs, respectively, over all vehicles. Clearly, most instances can be solved within 25 seconds, although the last instance group requires substantial computation time. Notably, instance MV-90-5-5 cannot be solved optimally within the time limit of 2,700 seconds. We next seek to explain why multi-vehicle instances consisting of around 20 requests in total, such as instances MV-90-4-m, can be solved in less than 25 seconds, while single-vehicle instances involving a similar number of requests, such as instances SV-90-120-m, take much longer to solve. The primary reason lies in the number of requests (both scheduled and new) that must be assigned and

routed per vehicle. When a single vehicle is responsible for serving many requests with overlapping time windows (e.g., for pickups or drop-offs), the number of feasible routing plans and vehicle schedules increases substantially. In the single-vehicle instances SV-90-120-m, the average number of such requests per vehicle is 20.6. By contrast, in the multi-vehicle instances MV-90-4-m, the average is only 6.7 requests per vehicle. This substantial difference implies that the routing optimization for one vehicle in a multi-vehicle setting is much less complex and requires significantly less computation time. Even though optimization must be performed for three vehicles, the total computation time remains low—primarily because routing a single vehicle with a limited number of riders is not computationally demanding.

Based on this key observation, the high computational efficiency of solving a single-vehicle routing problem for limited rider requests should be leveraged to the largest possible extent in a promising custom algorithm, to be discussed in the Conclusion section.

Table 8: Comparison of instance statistics and computation time (multi-vehicle)

Instance	Total requests	$\lvert\tilde{R}\rvert$	$\lvert R\rvert$	$\lvert\bar{R}\rvert$	$\lvert N\rvert$	$\lvert A\rvert$	Objective value	Comp. time (s)
MV-60-3-1	12	2	7	3	40	477	612.4	2
MV-60-3-2	11	2	5	4	42	530	753.2	3
MV-60-3-3	10	3	5	3	37	407	683.1	2
MV-60-3-4	9	2	4	3	34	338	570.8	1
MV-60-3-5	9	1	5	3	35	359	524.9	1
MV-60-4-1	15	3	8	4	59	787	874.7	5
MV-60-4-2	14	4	6	4	56	709	759.0	5
MV-60-4-3	14	3	6	5	63	905	748.2	9
MV-60-4-4	13	3	6	4	55	681	694.3	6
MV-60-4-5	15	4	7	4	58	762	702.6	7
MV-60-5-1	17	4	9	4	72	938	914.6	9
MV-60-5-2	16	4	7	5	78	1109	1024.7	13
MV-60-5-3	17	5	7	5	79	1141	1152.4	16
MV-60-5-4	17	4	8	4	70	885	851.2	10
MV-60-5-5	16	4	7	5	78	1127	858.0	14
MV-90-3-1	15	2	8	5	54	896	719.8	7
MV-90-3-2	15	2	9	4	50	762	625.9	6
MV-90-3-3	16	4	8	4	50	763	845.4	6
MV-90-3-4	17	3	9	5	57	1001	814.1	9
MV-90-3-5	16	3	8	5	55	932	784.1	9
MV-90-4-1	21	3	13	5	77	1372	971.2	24
MV-90-4-2	23	4	14	5	80	1489	959.2	23
MV-90-4-3	20	4	11	5	74	1266	867.3	23
MV-90-4-4	19	3	12	4	67	1027	935.4	19
MV-90-4-5	18	3	11	4	65	970	798.8	18
MV-90-5-1	32	4	21	7	126	2993	1458.4	1624
MV-90-5-2	29	5	19	5	103	1977	1244.7	245
MV-90-5-3	30	3	21	6	115	2480	1315.0	733
MV-90-5-4	27	6	16	5	98	1784	1084.3	145
MV-90-5-5	34	7	18	9	143	3884	-	>2700

4.5 Managerial Interpretation

Setting policy caps. $\delta_{\max}$ should reflect what the agency is willing to advertise as "governed flexibility." Typical practice confines new riders to small caps (e.g., single-digit minutes) while keeping existing commitments (onboard/scheduled) strictly within published windows.

Choosing the penalty weight λ. Calibrate λ so that the objective is interpretable in a common cost unit. Two pragmatic pathways are:

1. Dollarization: convert distance to $ via a cost-per-mile (or vehicle-hour) and set λ to the service credit per minute of expansion that the agency is willing to compensate.

2. Service-level targeting: sweep λ and select the smallest value that attains target objective with acceptable cost.

Operational advice. (1) Run the independent baseline to reveal local bottlenecks; (2) turn on coordination to share new requests across routes; (3) tighten $\delta_{\max}$ once acceptance stabilizes; (4) monitor the share and magnitude of expansions and publish rider-facing credits consistent with λ.

5. CONCLUSION

5.1 Summary

Despite extensive research on demand-responsive microtransit, existing systems face operational inefficiencies due to fixed time windows that increase rider rejection rates and route detours during peak demand periods. Therefore, we propose two MIP formulations with continuous Time Window Expansions (TWE) for single-vehicle and multi-vehicle scenarios, respectively. The multi-vehicle formulation enables dynamic, fleet-level optimization of pickup and drop-off time windows, vehicle routing, and rider assignments. A penalty parameter is designed to balance service quality and operational efficiency, namely, schedule disruption and driving distance.

Through our computational experiments, we highlight the following findings.

- Coordinated multi-vehicle optimization reduced cumulative time window expansions by up to 89.7% versus uncoordinated single-vehicle optimizations.
- Operational costs decreased by 19.2% with fleet-wide TWE compared to single-vehicle optimizations.
- Total driving distances were lowered by approximately 17.0% through fleet coordination.
- Optimal solutions were typically obtained within 25 seconds for most instances, affirming computational viability for real-time decision-making.

These results enable transit operators to dynamically adjust schedules. Stakeholders, including transit agencies and riders, benefit from enhanced service responsiveness and resource utilization.

We next discuss the limitations of this study. In this study, a commercial solver is used to directly solve both single-vehicle and multi-vehicle formulations. However, for large-scale instances, such as those involving more than 10 vehicles and 20 new rider requests, a more scalable and efficient solution strategy is to adopt a phased framework. One potential framework is to decompose the overall decision-making into two stages: (i) rider-to-vehicle assignment and (ii) single-vehicle routing. By doing so, we avoid the computational challenge in solving a large multi-vehicle routing problem. The motivation behind this approach is to exploit the high computational efficiency of solving single-vehicle routing problems. The main challenge lies in capturing the interaction between the assignment and routing stages, which can be addressed using a learning-based optimization technique that predicts routing costs based on rider-to-vehicle assignment decisions.

5.2 Future Research

Some future research directions are summarized as follows.

- We can integrate real-time traffic and demand variability into the TWE model using simulation-based frameworks to enhance model realism.
- We can evaluate rider acceptance of dynamically adjusted time windows through stated-preference surveys to improve rider-centric outcomes.
- We can extend the TWE approach to mixed-fleet scenarios, including autonomous and conventional vehicles, to enable comprehensive service optimization.

REFERENCES

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., & Rus, D. (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 114(3), 462–467.

Arslan, A. M., Agatz, N., Kroon, L., & Zuidwijk, R. (2019). Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers. *Transportation Science*, 53(1), 222–235.

Baldacci, R., Bartolini, E., & Mingozzi, A. (2011). An exact algorithm for the pickup and delivery problem with time windows. *Operations Research*, 59(2), 414–426.

Bimpikis, K., Candogan, O., & Saban, D. (2019). Spatial pricing in ride-sharing networks. *Operations Research*, 67(3), 744–769.

Braverman, A., Dai, J. G., Liu, X., & Ying, L. (2019). Empty-car routing in ridesharing systems. *Operations Research*, 67(5), 1437–1452.

Chen, S., Rahman, M. H., Marković, N., Siddiqui, M. I. Y., Mohebbi, M., & Sun, Y. (2024). Schedule negotiation with ada paratransit riders under value of time uncertainty. *Transportation Research Part B: Methodological*, 184, 102962.

Chen, X., Wang, Y., Wang, Y., Qu, X., & Ma, X. (2021). Customized bus route design with pickup and delivery and time windows: Model, case study and comparative analysis. *Expert Systems with Applications*, 168, 114242.

Disability Rights Education & Defense Fund (2025). Topic guide on on-time performance (ADA paratransit scheduling practices). https://dredf.org/ADAtg/OTP.shtml.

Fayed, L., Nilsson, G., & Geroliminis, N. (2024). On the effect of batching for on-demand high-capacity micro-transit services. In *Proceedings of the 24th Swiss Transport Research Conference (STRC)* Monte Verità, Ascona, Switzerland.

Federal Transit Administration (2019). *Mobility on Demand: Performance Measures and Evaluation Framework (Report No. FTA-MPM-0152)*. Technical report, U.S. Department of Transportation. https://www.transit.dot.gov/research-innovation.

Federal Transit Administration (2023a). *Mobility on Demand (MOD) Sandbox Program — Synthesis of Project Evaluations*. Technical report, U.S. Department of Transportation. https://www.transit.dot.gov/research-innovation/mobility-demand-mod-sandbox-program.

Federal Transit Administration (2023b). Shared mobility definitions. https://www.transit.dot.gov/regulations-and-guidance/shared-mobility-definitions.

Figliozzi, M. A. (2010). An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. *Transportation Research Part C: Emerging Technologies*, 18(5), 668–679.

Fu, Z. & Chow, J. Y. (2022). The pickup and delivery problem with synchronized en-route transfers for microtransit planning. *Transportation Research Part E: Logistics and Transportation Review*, 157, 102562.

Gkiotsalitis, K. & Alesiani, F. (2019). Robust timetable optimization for bus lines subject to resource and regulatory constraints. *Transportation Research Part E: Logistics and Transportation Review*, 128, 30–51.

Hansen, T., Walk, M., Tan, S., & Mahmoudzadeh, A. (2021). Performance measurement and evaluation framework of public microtransit service. *Transportation Research Record*, 2675(12), 201–213.

He, J. & Ma, T.-Y. (2022). Examining the factors influencing microtransit users' next ride decisions using bayesian networks. *European Transport Research Review*, 14(1), 47.

Iglesias, R., Rossi, F., Wang, K., Hallac, D., Leskovec, J., & Pavone, M. (2018). Data-driven model predictive control of autonomous mobility-on-demand systems. In *2018 IEEE international conference on robotics and automation (ICRA)* (pp. 6019–6025).: IEEE.

Kallehauge, B., Larsen, J., Madsen, O. B., & Solomon, M. M. (2005). Vehicle routing problem with time windows. In *Column generation* (pp. 67–98). Springer.

Laredo Metropolitan Planning Organization (2025). *Microtransit Feasibility Study Report*. Technical report. https://www.laredompo.org/wp-content/uploads/2025/05/Microtransit_FR_Web_Version.pdf.

Lee, E., Cen, X., & Lo, H. K. (2021). Zonal-based flexible bus service under elastic stochastic demand. *Transportation Research Part E: Logistics and Transportation Review*, 152, 102367.

Li, H., Lei, Z., & Ukkusuri, S. V. (2024). Flexible microtransit scheduling with time sensitive travelers. In *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 2832–2837).: IEEE.

Li, Z., Song, R., He, S., & Bi, M. (2018). Methodology of mixed load customized bus lines and adjustment based on time windows. *Plos one*, 13(1), e0189763.

Liu, T., Ceder, A., & Chowdhury, S. (2017). Integrated public transport timetable synchronization with vehicle scheduling. *Transportmetrica A: Transport Science*, 13(10), 932–954.

Lowalekar, M., Varakantham, P., & Jaillet, P. (2021). Zone path construction (zac) based approaches for effective real-time ridesharing. *Journal of Artificial Intelligence Research*, 70, 119–167.

Ma, W., Zeng, L., & An, K. (2023). Dynamic vehicle routing problem for flexible buses considering stochastic requests. *Transportation Research Part C: Emerging Technologies*, 148, 104030.

Markov, I., Guglielmetti, R., Laumanns, M., Fernández-Antolín, A., & de Souza, R. (2021). Simulation-based design and analysis of on-demand mobility services. *Transportation Research Part A: Policy and Practice*, 149, 170–205.

Martínez, L. M. & Eiró, T. (2012). An optimization procedure to design a minibus feeder service: an application to the sintra rail line. *Procedia-Social and Behavioral Sciences*, 54, 525–536.

Mattson, J. (2024). *Rural Transit Fact Book 2024*. Technical report, Small Urban and Rural Transit Center (SURTC), UGPTI. https://www.ugpti.org/resources/reports/downloads/dp-325.pdf.

Mattson, J. (2025). *Rural Transit Fact Book 2025*. Technical report, Small Urban and Rural Transit Center (SURTC), UGPTI. https://www.ugpti.org/resources/reports/downloads/dp-330.pdf.

Narayanan, S., Chaniotakis, E., & Antoniou, C. (2020). Shared autonomous vehicle services: A comprehensive review. *Transportation Research Part C: Emerging Technologies*, 111, 255–293.

National Center for Applied Transit Technology (2023). What are on-demand transit and microtransit? https://n-catt.org/guidebooks/on-demand-transit-and-microtransit-where-and-why/what-are-on-demand-transit-and-microtransit/.

Nourinejad, M. & Ramezani, M. (2020). Ride-sourcing modeling and pricing in non-equilibrium two-sided markets. *Transportation Research Part B: Methodological*, 132, 340–357.

OmniRide (Potomac and Rappahannock Transportation Commission) (2024). Omniride connect microtransit rider's guide. PDF brochure. https://omniride.com/service/connect/riders-guide/.

Quadrifoglio, L., Dessouky, M. M., & Ordóñez, F. (2008). A simulation study of demand responsive transit system design. *Transportation Research Part A: Policy and Practice*, 42(4), 718–737.

Rath, S., Liu, B., Yoon, G., & Chow, J. Y. (2023). Microtransit deployment portfolio management using simulation-based scenario data upscaling. *Transportation Research Part A: Policy and Practice*, 169, 103584.

Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S. H., & Ratti, C. (2014). Quantifying the benefits of vehicle pooling with shareability networks. *Proceedings of the National Academy of Sciences*, 111(37), 13290–13294.

Sarbijan, M. S. & Behnamian, J. (2022). Real-time collaborative feeder vehicle routing problem with flexible time windows. *Swarm and Evolutionary Computation*, 75, 101201.

Shared-Use Mobility Center (2022). Microtransit: Module in the sumc mobility learning center. https://learn.sharedusemobilitycenter.org/learning_module/microtransit/.

Shared-Use Mobility Center (2025). Microtransit – sumc mobility learning center. https://learn.sharedusemobilitycenter.org/learning_module/microtransit/.

Stiglic, M., Agatz, N., Savelsbergh, M., & Gradisar, M. (2015). The benefits of meeting points in ride-sharing systems. *Transportation Research Part B: Methodological*, 82, 36–53.

Summit Stage (2024). *Microtransit Feasibility Analysis*. Technical report, Summit County, CO. https://cms3.revize.com/revize/summitcoco/Documents/Services/Transit%20Summit%20Stage/Summit%20County%20Transit%20Board/Microtransit%20Feasibility%20Study%20Final%20Report%20051024.pdf.

Tachet, R., Sagarra, O., Santi, P., Resta, G., Szell, M., Strogatz, S. H., & Ratti, C. (2017). Scaling law of urban ride sharing. *Scientific reports*, 7(1), 42868.

Taş, D., Gendreau, M., Dellaert, N., Van Woensel, T., & De Kok, A. (2014). Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach. *European Journal of Operational Research*, 236(3), 789–799.

Tirachini, A. (2020). Ride-hailing, travel behaviour and sustainable mobility: an international review. *Transportation*, 47(4), 2011–2047.

Tuncel, K., Koutsopoulos, H. N., & Ma, Z. (2023). An integrated ride-matching and vehicle-rebalancing model for shared mobility on-demand services. *Computers & Operations Research*, 159, 106317.

Utah Transit Authority (2020). *Microtransit Pilot Evaluation Report*. Technical report. https://rideuta.com/.

Vazifeh, M. M., Santi, P., Resta, G., Strogatz, S. H., & Ratti, C. (2018). Addressing the minimum fleet problem in on-demand urban mobility. *Nature*, 557(7706), 534–538.

Veve, C. & Chiabaut, N. (2022). Demand-driven optimization method for microtransit services. *Transportation Research Record*, 2676(3), 58–70.

Via Mobility, LLC (2020). *Utah Transit Authority Microtransit Planning Project*. Technical report, Utah Transit Authority, Salt Lake City, UT. https://www.rideuta.com/-/media/Files/About-UTA/Reports/2021/UTA_Microtransit_Consulting_Report_Final.pdf.

Virginia Department of Rail and Public Transportation (2023). *Rural Microtransit Case Study and Report*. Technical report, DRPT. https://drpt.virginia.gov/wp-content/uploads/2023/05/drpt-rural-microtransit-case-study-and-report-final.pdf.

Xue, G., Wang, Z., & Wang, G. (2021). Optimization of rider scheduling for a food delivery service in o2o business. *Journal of Advanced Transportation*, 2021(1), 5515909.

Yang, H., Zhao, L., Ye, D., & Ma, J. (2020). Disturbance management for vehicle routing with time window changes. *Operational Research*, 20, 1093–1112.