



**Università
di Brescia**

Dipartimento di Economia e Management

DOTTORATO DI RICERCA IN MODELLI E METODI PER
L'ECONOMIA E IL MANAGEMENT

MAT/09 - RICERCA OPERATIVA
XXXVIII Cycle

Optimisation Models in Bike Sharing Systems

Nome del Dottorando: Reza Atefi
Supervisore: Prof. M. Grazia Speranza

Academic Year 2025–2026

Abstract

Bike-sharing systems are an important component of sustainable urban mobility, but their reliability is impaired by spatial and temporal imbalances between bicycle supply and user demand. Empty and full stations generate lost demand and unnecessary operating costs, making the Bike Rebalancing Problem (BRP) a central problem at the intersection of vehicle routing and inventory management. This thesis develops Operations Research models and algorithms for the BRP through three studies connected by the use of *demand intervals*.

The first study reviews the BRP literature from an Operations Research perspective. Starting from approximately 948 records, it retains 282 papers for full-text assessment and a final analytical corpus of 104 studies. A unified six-tuple formalisation and a multi-dimensional taxonomy organise the literature by planning horizon, demand representation, fleet structure, operational scope, and rebalancing paradigm. The review finds a mature static deterministic stream but weaker comparability and reproducibility in the dynamic, stochastic, and learning-based streams, and uses these findings to motivate the two methodological studies.

The second study considers a single-vehicle Static Bike Rebalancing Problem with acceptability intervals, repeated visits to stations and the depot, and a freely chosen initial load. One flow-based mixed-integer linear programming formulation represents non-split, without-preemption, and with-preemption policies. A structural theorem proves that the with- and without-preemption policies have the same optimal value under unbounded vehicle capacity, non-negative handling times, an uncharged free initial load, and an unrestricted terminal load. An Adaptive Iterated Local Search (AILS), combining Nested Dynamic Programming with $Q(\lambda)$ -guided perturbation, matches the CPLEX objective on all 60 small random instances, although CPLEX certifies optimality in only 53. On 100 established literature configurations, AILS matches every published best-known solution and 99 of these matches are certified by the published lower bounds. A ten-run component ablation shows the lowest average deviation for the complete AILS (0.64%), while the method also produces feasible solutions for instances of up to 200 stations within seconds. No optimality claim is made where no certificate is available.

The third study develops a Dynamic Bike Rebalancing Problem with Demand Intervals. The CARI procedure corrects boundary-censored station data through a continuous-time birth–death process, propagates bootstrap rate uncertainty, and constructs robust station-period inventory intervals. These intervals are embedded in a rolling-horizon mixed-integer model; at fleet level, vehicle-specific clustering is combined with a VNS–MIP-refinement–VNS workflow and

evaluated by event-driven simulation. The computational study uses 20 instances, each containing 40 stations sampled from 2024 Capital Bikeshare data. In the reported averages, a three-period look-ahead reduces realised loss from 5.0 to 4.1 relative to a one-period policy. The robust interval records realised loss of 4.1, compared with 8.0 for the benchmark interval and 19.2 for the point-target policy; the corresponding no-rebalancing value is 213.9. These are model-based simulation results on the tested instances. Because raw replication-level outputs, confidence intervals, and held-out operational validation are not available, they do not establish universal superiority.

Taken together, the three studies provide a coherent treatment of static and dynamic rebalancing through demand intervals. The evidence comprises synthetic benchmarks, established literature instances, data-derived instances, and simulations parameterised by historical records, with certification status and empirical limitations stated explicitly.

Keywords: bike-sharing systems; bike rebalancing problem; vehicle routing; inventory management; demand intervals; preemption; mixed-integer linear programming; metaheuristics; rolling-horizon optimisation.

Riassunto

I sistemi di bike-sharing costituiscono una componente importante della mobilità urbana sostenibile, ma la loro affidabilità è compromessa dagli squilibri spaziali e temporali tra offerta di biciclette e domanda degli utenti. Stazioni vuote o sature generano domanda persa e costi operativi non necessari, rendendo il *Bike Rebalancing Problem* (BRP) un problema centrale all'intersezione tra instradamento dei veicoli e gestione delle scorte. Questa tesi sviluppa modelli e algoritmi di Ricerca Operativa per il BRP attraverso tre studi collegati dall'impiego degli *intervalli di domanda*.

Il primo studio esamina la letteratura sul BRP da una prospettiva di Ricerca Operativa. A partire da circa 948 record, vengono selezionati 282 articoli per la valutazione del testo completo e un corpus analitico finale di 104 studi. Una formalizzazione unificata in sei elementi e una tassonomia multidimensionale organizzano la letteratura secondo orizzonte di pianificazione, rappresentazione della domanda, struttura della flotta, ambito operativo e paradigma di riequilibrio. La revisione individua un filone statico deterministico maturo, ma una minore comparabilità e riproducibilità nei filoni dinamico, stocastico e basato sull'apprendimento, e utilizza questi risultati per motivare i due studi metodologici successivi.

Il secondo studio considera uno Static Bike Rebalancing Problem con un solo veicolo, intervalli di accettabilità, visite ripetute alle stazioni e al deposito e carico iniziale liberamente scelto. Un'unica formulazione di programmazione lineare intera mista basata sui flussi rappresenta le politiche non-split, senza preemption e con preemption. Un teorema strutturale dimostra che le politiche con e senza preemption hanno lo stesso valore ottimo quando la capacità del veicolo è illimitata, i tempi di movimentazione sono non negativi, il carico iniziale è libero e non contabilizzato e il carico terminale non è vincolato. Un'Adaptive Iterated Local Search (AILS), che combina Nested Dynamic Programming e perturbazione guidata da $Q(\lambda)$, coincide con l'obiettivo CPLEX in tutte le 60 piccole istanze casuali, sebbene CPLEX certifichi l'ottimalità solo in 53 casi. Su 100 configurazioni consolidate della letteratura, AILS coincide con tutte le migliori soluzioni pubblicate e 99 di tali corrispondenze sono certificate dai lower bound pubblicati. Un'ablazione dei componenti basata su dieci esecuzioni mostra la deviazione media più bassa per l'AILS completo (0,64%); il metodo produce inoltre soluzioni ammissibili in pochi secondi per istanze fino a 200 stazioni. Non viene formulata alcuna dichiarazione di ottimalità in assenza di certificazione.

Il terzo studio sviluppa un Dynamic Bike Rebalancing Problem con intervalli di domanda. La

procedura CARI corregge i dati di stazione censurati agli estremi mediante un processo continuo birth–death, propaga l’incertezza delle stime tramite bootstrap e costruisce intervalli robusti di inventario per stazione e periodo. Tali intervalli sono integrati in un modello intero misto a orizzonte mobile; a livello di flotta, il raggruppamento specifico per veicolo è combinato con una procedura VNS–raffinamento MIP–VNS e valutato mediante simulazione a eventi discreti. Lo studio computazionale utilizza 20 istanze, ciascuna con 40 stazioni campionate dai dati Capital Bikeshare del 2024. Nelle medie riportate, un orizzonte di tre periodi riduce la domanda persa realizzata da 5,0 a 4,1 rispetto a una politica con un solo periodo. L’intervallo robusto registra una domanda persa realizzata pari a 4,1, rispetto a 8,0 per l’intervallo benchmark e 19,2 per la politica con target puntuale; il corrispondente valore senza riequilibrio è 213,9. Si tratta di risultati di simulazione basati sul modello per le istanze esaminate. Poiché non sono disponibili gli output delle singole repliche, gli intervalli di confidenza e una validazione operativa fuori campione, essi non dimostrano una superiorità universale.

Nel loro insieme, i tre studi forniscono un trattamento coerente del riequilibrio statico e dinamico tramite gli intervalli di domanda. L’evidenza comprende benchmark sintetici, istanze consolidate della letteratura, istanze derivate da dati e simulazioni parametrizzate mediante dati storici, con lo stato delle certificazioni e i limiti empirici dichiarati esplicitamente.

Parole chiave: sistemi di bike-sharing; problema di bike rebalancing; instradamento dei veicoli; gestione delle scorte; intervalli di domanda; preemption; programmazione lineare intera mista; metaeuristiche; ottimizzazione a orizzonte mobile.

Acknowledgments

Completing this dissertation marks far more than the conclusion of an academic project; it represents the end of one of the most demanding, exhausting, and transformative periods of my life. These years were shaped by uncertainty, persistence, failure, growth, isolation, hope, and continuous learning. Throughout this journey, many individuals contributed, directly and indirectly, to both this work and my personal development. I would therefore like to express my sincere gratitude to all those who accompanied, supported, and inspired me during these years.

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Speranza. Without exaggeration, her role throughout this journey extended well beyond that of a formal academic advisor. Her scientific rigor, intellectual discipline, analytical mindset, and unwavering commitment to high academic standards profoundly shaped both my research approach and my professional development. During the most difficult stages of this PhD, particularly when I encountered scientific dead ends, uncertainty, or mental exhaustion, her guidance and support consistently provided clarity and direction. Beyond her academic responsibilities, she was a source of encouragement and inspiration during many challenging moments of this journey. This dissertation would not have been completed without her continuous trust, guidance, and support.

I would also like to sincerely thank Professor Angelelli and Professor Mor for their valuable guidance, constructive feedback, and continuous support throughout different stages of this research. Their insights and discussions significantly contributed to improving the quality and depth of this work.

My sincere thanks also go to all members of the Operations Research group at the University of Brescia for the stimulating scientific environment, valuable discussions, collaboration, and collegial atmosphere that characterised my years within the group.

In particular, I would like to express my gratitude to Professor Guastaroba for his generosity and support in providing access to part of the facilities and resources required for this research. His assistance played an important role in facilitating several aspects of this work, and many parts of this dissertation would have been considerably more difficult to accomplish without his support.

A significant part of this research was carried out during my visiting research period at TU Delft in the Netherlands, an experience that was undoubtedly among the most valuable periods of my academic life. I would like to sincerely thank Professor Sharif Azadeh for hosting me, for her

scientific support, and for providing an intellectually stimulating and highly professional research environment. My time at TU Delft gave me the opportunity to experience an international academic setting, broaden my scientific perspective, and further develop my professional outlook. I am equally grateful to all colleagues and friends I met during my stay in Delft for creating such a welcoming and supportive environment.

I would also like to thank the administrative staff of the department, the PhD office, and all personnel who, through their professionalism and patience, facilitated many administrative and practical aspects of this journey.

I would particularly like to thank Valentina Morandi, not in the context of academic collaboration, but as a valued friend whose support, encouragement, and kindness meant a great deal to me during difficult periods of this PhD. Some forms of support cannot be adequately described as professional or academic contributions; their value lies in genuine human presence, understanding, and encouragement during moments when the pressures extended far beyond research itself.

I am also deeply grateful to my close friends, Seyed Ali Mir Aghel and Payam Heydarian, whose friendship, patience, and emotional support helped me maintain motivation and perspective throughout many difficult moments of this journey. Some people ease the weight of difficult periods simply through their presence, and they were undoubtedly among those people for me.

My deepest gratitude belongs to my family, especially my parents, who have always been my greatest source of support, strength, and encouragement throughout every stage of my life. No words can adequately express the extent of their sacrifices, patience, love, and unwavering belief in me. Whatever I have achieved is fundamentally rooted in their support and dedication.

I would also like to thank my brothers and sister for their constant encouragement, care, and support. The presence of my family has always been my greatest source of stability and strength.

Finally, I would like to express my sincere gratitude to everyone who contributed to this journey in one way or another, even if their names are not explicitly mentioned here. This dissertation is not solely the result of individual effort, but also the outcome of the support, trust, kindness, and presence of many remarkable people throughout these years.

I hope that this work, however modest, can contribute in some way to the advancement of knowledge in the field.

Contents

Abstract	i
Riassunto	iii
Acknowledgments	v
Unified Notation	5
1 The Bike Rebalancing Problem: A review from an operational research perspective	7
1.1 Introduction	7
1.1.1 Motivation for a New Survey	7
1.1.2 Scope and Methodology	8
1.1.3 Contributions of This Survey	10
1.1.4 Paper Organisation	10
1.2 Background and Terminology	10
1.2.1 Terminology Harmonisation	10
1.2.2 Bike-Sharing System Types	11
1.2.3 Historical Evolution of BRP Research	11
1.3 Formal Framework and Taxonomy	12
1.3.1 Formal Definition of the BRP	12
1.3.2 Taxonomy Design and Justification	14
1.3.3 Mapping the Literature onto the Taxonomy	14
1.4 The Static Bike Rebalancing Problem	15
1.4.1 Problem Structure and Complexity	15
1.4.2 Foundational Single-Vehicle Formulations	16
1.4.3 Multi-Vehicle Static BRP	16
1.4.4 Structural Variants of the Static BRP	17
1.4.5 Multi-Type and E-Bike Extensions	17
1.4.6 Maintenance-Integrated Static BRP	17
1.4.7 Green and Sustainability-Oriented Static BRP	18
1.5 The Dynamic Bike Rebalancing Problem	18
1.5.1 Problem Structure and Contrast with Static BRP	18
1.5.2 Foundational Dynamic Formulations	18

1.5.3	Rolling-Horizon and Real-Time Approaches	19
1.5.4	MDP and Stochastic Dynamic Formulations	19
1.5.5	Dynamic BRP with Electric Vehicles	20
1.5.6	Broken Bikes in Dynamic Settings	20
1.6	Stochastic, Robust, and Distributionally Robust BRP	20
1.6.1	Two-Stage Stochastic BRP	20
1.6.2	Robust Optimisation Models	21
1.6.3	Distributionally Robust BRP	21
1.6.4	Stochastic Single-Station Inventory Models	21
1.7	Learning-Based and Prediction-Integrated BRP	22
1.7.1	The Predict-Then-Optimise Pipeline in BRP	22
1.7.2	Demand Forecasting for BRP	22
1.7.3	Reinforcement Learning for Dynamic BRP	22
1.7.4	Integration of Prediction and Optimisation: Design Principles	23
1.8	User-Based and Hybrid Rebalancing	23
1.8.1	Incentive Mechanisms and Pricing	23
1.8.2	Hybrid Operator-User Models	23
1.9	Cross-Cutting Extensions	24
1.9.1	Sustainability and Environmental Objectives	24
1.9.2	Fairness and Service Equity	24
1.9.3	Multi-Modal and Intermodal Integration	24
1.9.4	Broken Bikes and Maintenance Integration	24
1.10	Benchmarks, Datasets, and Reproducibility	25
1.10.1	Benchmark Instance Audit	25
1.10.2	Reproducibility Assessment	25
1.11	Synthesis and Critical Analysis	27
1.11.1	Comparative Analysis of Solution Methods	27
1.11.2	Trade-Off Analysis	27
1.11.3	Maturity Assessment	28
1.12	Research Agenda	28
1.13	Conclusion	29
	Appendices	30
.1	Comprehensive Literature Matrix	30
2	The Static Bike Rebalancing Problem with Preemption	34
2.1	Introduction	34
2.2	Literature Review	36
2.2.1	Split-Demand Pickup-and-Delivery and Preemption	36
2.2.2	Interval-Based Inventory Representations	37

2.2.3	Positioning of This Work	37
2.3	Practical Motivation	38
2.4	Problem Description and Formulation	40
2.4.1	An illustrative example	41
2.4.2	Mathematical formulation	42
2.4.3	An equivalence theorem under unbounded capacity	45
2.5	The Adaptive Iterated Local Search (AILS)	46
2.5.1	Initial Solution Method	46
2.5.2	Nested Dynamic Programming	47
2.5.3	NDP-aware full optimisation	48
2.5.4	$Q(\lambda)$ -guided Perturbation	49
2.6	Computational Results	50
2.6.1	Instance generation	50
2.6.2	Parameter tuning	51
2.6.3	Impact of the Main AILS Components on Algorithmic Performance	53
2.6.4	Impact of the maximum number of visits	55
2.6.5	Comparison of rebalancing policies	56
2.6.6	Validation against the exact method	56
2.6.7	Scalability and robustness	57
2.6.8	Validation on Established Benchmark Instances from the Literature	57
2.6.9	Real-Data Instances	59
2.6.10	Results on Small Real-world Data Instances	60
2.6.11	Results on Medium and Large Real-world data Instances	60
2.7	Conclusions and Future Research	61
	Appendices	66
.1	Nested Dynamic Programming: Derivation	66
.2	Detailed per-instance results	67
3	Dynamic Bike Rebalancing Problem with Demand Intervals	69
3.1	Introduction	69
3.2	Literature Review	71
3.3	Problem Description	74
3.3.1	Demand interval construction	75
3.3.2	Mathematical Formulation and Notation	78
3.4	Rolling-Horizon Fleet Solution Method	81
3.4.1	Temporal decomposition and objective	81
3.4.2	Spatial decomposition by reproducible network partitioning	82
3.4.3	Period-feasible VNS construction and improvement	83
3.4.4	Fixed-route repair, MIP refinement, and feedback VNS	83

3.4.5	State update and event-driven simulation	84
3.5	Computational Results	84
3.5.1	Data instances	85
3.5.2	Experimental design and performance measures	85
3.5.3	Calibration of objective-function weights	85
3.5.4	Effect of rolling-horizon length	86
3.5.5	Sensitivity to the CARI interval-width parameter	86
3.5.6	Comparison of demand representations	87
3.5.7	Value of active rebalancing	87
3.5.8	Summary of findings	88
3.6	Conclusion	88
4	Conclusion	89
	References	91

Introduction

Over the past two decades, cities have faced mounting pressure to reduce congestion, air pollution, and dependence on private motorised transport. Bike-sharing systems (BSS) have become one of the most visible responses to this pressure: by letting users rent a bicycle at one station and return it at another, they provide a low-emission, flexible option for short trips and help close the first- and last-mile gap of public-transport networks (Ricci, 2015). From a handful of experimental schemes in the early 2000s, BSS have grown into a mainstream mobility service operating in hundreds of cities across Asia, Europe, and North America, with documented benefits for emissions, congestion, public health, and integration with mass transit.

This success, however, rests on an operational foundation that is inherently fragile. Because urban travel is spatially and temporally uneven—commuting, leisure, and topography all push bicycles in the same directions at the same times—inventories drift away from balance during the day. Some stations are emptied of bicycles while others fill to capacity, so that prospective renters find no bicycle and returning users find no free dock. Both situations translate directly into lost demand and degraded quality of service, which in turn erode ridership and threaten the long-term viability of the system. Operators counter this drift through *rebalancing*: relocating bicycles among stations, and between stations and a depot, using one or more dedicated vehicles. The resulting optimisation problem—the Bike Rebalancing Problem (BRP)—is the subject of this thesis.

The BRP is computationally hard and conceptually rich because it couples two classical Operations Research problems. On the one hand, deciding which stations to visit and in what order is a vehicle routing problem; on the other, deciding how many bicycles to pick up or deliver at each station is an inventory problem constrained by vehicle and station capacities. Chemla et al. (2013a) formalised the single-vehicle static case as a generalisation of the one-commodity pickup-and-delivery problem and established its combinatorial hardness, and Dell’Amico et al. (2014a) introduced benchmark instances and mathematical formulations that still underpin much of the field. Since then the BRP has expanded into a large family of variants that differ in fleet structure, objective, demand representation, and—most fundamentally—in whether decisions are taken *before* or *during* the operational period.

This last distinction defines the two paradigms that organise the literature and this thesis. The *Static* Bike Rebalancing Problem (SBRP) plans the redistribution in advance, typically overnight when usage is negligible, fixing routes, visit sequences, and pickup/delivery quantities for the

next day (Chemla et al., 2013a; Dell’Amico et al., 2014a). The *Dynamic* Bike Rebalancing Problem (DBRP) instead revises decisions in real time as demand unfolds, usually within rolling- or receding-horizon schemes that repeatedly re-solve a short look-ahead subproblem (Legros, 2019a; Liang et al., 2024a). The static setting offers tractability and a clean basis for exact and heuristic comparison; the dynamic setting offers responsiveness at the price of speed, forecasting, and data-integration requirements. The two are complementary rather than competing, and a coherent treatment of rebalancing benefits from addressing both.

Two modelling threads recur across the BRP literature and are central to this thesis. The first concerns how a station’s requirement is represented. Classical formulations impose a single target inventory at each station, which is operationally convenient but triggers unnecessary transfers whenever the realised inventory deviates from the target. A more flexible alternative, pioneered by Günes Erdoğan et al. (2014), represents the requirement as an acceptability *interval*: any final inventory within the interval is admissible. Demand intervals reduce redundant movements and improve robustness to demand variability, and they have since been used in static, fault-tolerant, and dynamic formulations (Günes Erdoğan et al., 2014; Liang et al., 2024c; Tero J. Ikonen et al., 2024). The second thread concerns how bicycles may be physically moved. Because a station’s demand may exceed the vehicle capacity, or because the network geometry may favour spreading service over several visits, three rebalancing policies have crystallised in the split-demand pickup-and-delivery literature (Hernandez-Perez et al., 2018; Hernández-Pérez and Salazar-González, 2022): the *non-split* policy, in which each station is served in a single visit; the *without-preemption* policy, in which multiple visits are allowed but no station is used as a buffer; and the *with-preemption* policy, in which the vehicle may temporarily buffer bicycles at intermediate stations. These two threads—interval demand and multi-visit policies—provide the common vocabulary that links the studies in this thesis.

Despite a substantial body of work, three gaps motivate the research reported here. First, the BRP literature has grown large and heterogeneous, but it lacks a compact synthesis that links model structure, algorithms, datasets, and research maturity within a single coherent framework, so that the most promising open directions are hard to identify. Second, on the static side, no prior formulation simultaneously combines demand intervals, multiple visits to both stations and the depot, a freely chosen initial vehicle load, and the three rebalancing policies, and methods that scale to realistic instances while jointly optimising routing and loading decisions are scarce. Third, on the dynamic side, the field still lacks a tractable framework that couples a probabilistically grounded characterisation of acceptable station inventories with a rolling-horizon control policy, since targets are often treated as single set points and decoupled from the routing decisions they constrain. The overarching objective of this thesis is therefore to develop Operations Research models and algorithms for the BRP that use demand intervals as a unifying modelling device and span the static and dynamic settings, and to evaluate them on synthetic benchmarks and data-derived model-based experiments.

The thesis pursues this objective through three studies, each forming a self-contained

chapter. The first is a review of the BRP from an Operations Research perspective. Building on a screening of roughly 948 records, reduced to 282 papers for full-text assessment and a final analytical corpus of 104 studies, it proposes a unified six-tuple formalisation of the problem and a multi-dimensional taxonomy that separates planning horizon from demand representation, fleet structure, operational scope, and rebalancing paradigm. The review finds the static deterministic BRP to be methodologically mature, while the dynamic, stochastic, and learning-based streams—though more operationally relevant—remain fragmented by inconsistent benchmarks and limited reproducibility, and it formulates a research agenda that the two subsequent studies take up directly.

The second study addresses the SBRP with demand intervals, multiple visits to stations and the depot, a free initial vehicle load, and three rebalancing policies. It contributes a single flow-based mixed-integer linear programming formulation; a structural equivalence theorem under explicit assumptions of unbounded capacity, non-negative loading and unloading times, an uncharged free initial load, and unrestricted terminal load; and an Adaptive Iterated Local Search combining Nested Dynamic Programming with $Q(\lambda)$ -guided perturbation. On the small random benchmarks, AILS matches the CPLEX objective in all 60 cases, of which CPLEX certifies 53 as optimal. A ten-run ablation study isolates the contributions of NDP-aware evaluation, $Q(\lambda)$, adaptive perturbation, and the specialised local search. External validation on 100 established literature configurations yields 100 matches with published best-known solutions, 99 of which are certified by published lower bounds. The heuristic also returns solutions for instances of up to 200 stations; no optimality claim is made where no certificate is available.

The third study transfers the interval-based perspective to the dynamic setting. CARI estimates latent station-period rental and return rates from boundary-censored observations, propagates bootstrap rate uncertainty through a continuous-time birth–death model, and constructs robust inventory intervals. These intervals are embedded in a rolling-horizon mixed-integer programme, while vehicle-specific clustering and a VNS–MIP-refinement–VNS workflow support fleet-level solution. Event-driven experiments on 20 Capital Bikeshare-derived instances of 40 stations compare objective weights, look-ahead lengths, interval-width settings, three demand representations, and a no-rebalancing baseline. The reported averages favour a three-period look-ahead and the proposed robust interval on realised loss, but the absence of replication-level outputs, confidence intervals, and held-out operational validation limits the scope of inference.

In summary, the contributions of this thesis are: (i) a structured comparative review of the BRP, with a unified formalisation, taxonomy, reproducibility audit, and research agenda; (ii) a unified flow-based MILP, equivalence theorem, and externally benchmarked AILS for the static problem with demand intervals and flexible rebalancing policies; (iii) a censoring-aware dynamic framework linking latent-rate estimation, uncertainty-aware demand intervals, rolling-horizon inventory routing, fleet decomposition, and event-driven evaluation; and (iv) the use of demand intervals as a consistent modelling device connecting the static and dynamic settings. The

remainder of the thesis is organised as follows. The next chapter presents the review of the Bike Rebalancing Problem. The following chapter develops the Static Bike Rebalancing Problem with Flexible Preemption, and the subsequent chapter develops the Dynamic Bike Rebalancing Problem with Demand Intervals. A concluding chapter summarises the findings and outlines directions for future research.

Unified Notation

The notation below is used consistently across the thesis. Symbols shared by the static and dynamic studies retain the same meaning; chapter-specific algorithmic symbols are explicitly scoped to avoid ambiguity. Subscripts identify the relevant station, vehicle, visit, or planning period.

Table 1: Unified notation used throughout the thesis.

Scope	Symbol	Meaning
General	$G = (V, A)$	Directed station network, with location set V and arc set A .
General	$i, j; n$	Location indices; number of stations.
General	u_i	Capacity of station i .
General	Q, Q_k	Capacity of the single vehicle, or of vehicle k in a fleet.
General	τ_{ij}	Travel time from location i to location j .
General	$[b_i, \bar{b}_i], [b_{ip}, \bar{b}_{ip}]$	Acceptable inventory interval at station i , in the static model or in dynamic period p .
Static study	$I; l; m_{\text{vis}}$	Station set; visit-copy index; maximum number of visits to a location.
Static study	$s_i; d_i$	Initial inventory and minimum required number of vacant docks at station i , with $\bar{b}_i = u_i - d_i$.
Static study	t^L, t^U	Per-bicycle loading and unloading times.
Static study	x_{ij}, y_i	Arc-use and visit indicators on the expanded graph.
Static study	v_i^+, v_i^-	Numbers of bicycles loaded onto and unloaded from the vehicle at a visit copy.
Static study	$f_{ij}; g_{i(l)}$	Vehicle load on arc (i, j) ; station inventory after visit copy $i(l)$, with $g_{i(0)} := s_i$.
Static study	z_{ij}	Ordinal position of selected arc (i, j) in the route.
AILS	$Q(s, a)$	$Q(\lambda)$ action-value table; distinct from the Markov generator used in the dynamic study.
AILS	$\alpha_Q, \zeta, \lambda_Q$	Q-learning rate, discount factor, and eligibility-trace coefficient.
AILS	$\phi_0; \chi_{ij}$	Initial sector half-angle; asymmetric travel-time multiplier.
Dynamic study	$N, K, P; p$	Station, vehicle, and period sets; planning-period index.

Continued on next page

Table 1 continued

Scope	Symbol	Meaning
Dynamic study	$\bar{T}; h_{RH}$	Period duration; maximum rolling look-ahead length.
Dynamic study	$\hat{\lambda}_{ip}, \hat{\mu}_{ip}$	CARI-adjusted latent return and rental rates.
Dynamic study	$\mathbf{G}_i(\lambda, \mu)$	Generator matrix of the station-level birth–death process.
Dynamic study	$\mathcal{U}_{ip}; \Gamma_{ip}$	Bootstrap uncertainty set and service-risk tolerance.
Dynamic study	R_{ip}	Net inventory drift, $\hat{\lambda}_{ip} - \hat{\mu}_{ip}$.
Dynamic study	$I_{ip}, \bar{I}_{ip}$	Inventory immediately before service and at the end of period p .
Dynamic study	$D_{ip}, E_{ip}, \bar{D}_{ip}, \bar{E}_{ip}$	Shortage and excess quantities before service and at the end of a period.
Dynamic study	G_{ip}	Deviation of end-of-period inventory from $[b_{ip}, \bar{b}_{ip}]$.
Dynamic study	$\kappa_p^{\text{olp}}, \kappa_p^{\text{int}}$	Period weights for operational loss and interval deviation.
Dynamic study	$\omega_h^{\text{olp}}, \omega_h^{\text{int}}$	Rolling-horizon discount weights at relative position h .
Dynamic study	ρ_{cl}	Distance-weight coefficient in the fleet-partition score.
Dynamic study	z_{CARI}	Experimental CARI interval-width parameter.
Dynamic study	$\mathcal{J}$	Objective value of the dynamic mixed-integer model.

Chapter 1

The Bike Rebalancing Problem: A review from an operational research perspective

1.1 Introduction

Bike-sharing systems (BSS) have expanded from small experimental programmes to a large-scale urban mobility service with substantial environmental and accessibility benefits (Fishman, 2016a; Laporte et al., 2018a; Soares Machado et al., 2018). Their operational fragility is equally clear: demand is spatially and temporally uneven, so bicycles accumulate in some locations while other stations or zones experience persistent shortage. The resulting stockouts and overflows degrade service quality, reduce effective demand, and impose direct operating costs. The *Bike Rebalancing Problem* (BRP) captures the optimisation problem generated by this imbalance.

In its canonical form, the operator dispatches one or more vehicles to collect bicycles from surplus locations and deliver them to deficit locations while respecting vehicle capacities, station capacities, depot rules, route-duration limits, and service objectives. Since the NP-hardness results of Benchimol et al. (2011), this core routing-and-inventory problem has expanded into a family of formulations that include dynamic control, stochastic and robust planning, prediction-integrated optimisation, user incentives, maintenance, multi-type fleets, equity constraints, and multimodal coupling. The field is now too broad to be organised adequately by a simple static/dynamic split.

1.1.1 Motivation for a New Survey

Existing reviews illuminate parts of this literature but leave an analytical gap. Bibliometric studies identify themes and citation patterns without comparing formulation structure (Si et al., 2019; Zhou et al., 2022; Ntamwiza et al., 2026). Broader bike-sharing reviews discuss rebalancing alongside infrastructure, safety, or adoption (Fishman, 2016a; C. S. Shui and W. Y. Szeto, 2020a). The encyclopedia chapter by Bruck and Subramanian (2023) provides a concise entry point to the principal bike-sharing rebalancing formulations and solution approaches. The closest full methodological survey, Vallez et al. (2021), is narrower in problem scope and predates the

recent expansion of robust, distributionally robust, and learning-based BRP models. A focused review is therefore justified not because the field lacks summaries, but because it lacks a compact synthesis that links model structure, algorithms, datasets, and research maturity within one coherent framework. Table 2 positions the present review against the main prior contributions along the dimensions that define its contribution.

Table 2: Positioning of this review relative to prior BRP-related reviews.

Review (type)	Form. comp.	Multi-dim. tax.	Bench. audit	Reprod. audit	Coded agenda
Bibliometric mapping (Si et al., 2019; Zhou et al., 2022; Ntamwiza et al., 2026)	–	–	–	–	–
Broad BSS reviews (Fishman, 2016a; C. S. Shui and W. Y. Szeto, 2020a)	–	~	–	–	–
Rebalancing-focused (Vallez et al., 2021)	~	~	–	–	–
This review	✓	✓	✓	✓	✓

✓ provided; ~ partial; – not provided. “Form. comp.”: structured comparison of model formulations; “Coded agenda”: research priorities tied to corpus coding.

Three developments make such a survey timely. First, uncertainty handling has moved beyond deterministic target balancing toward two-stage stochastic, robust, and data-driven formulations (Dell’Amico et al., 2018a; Bruck et al., 2025; Q. Chen et al., 2025; Parada et al., 2026; R. Zhang et al., 2025). Second, prediction and control are increasingly coupled through rolling-horizon optimisation, neural forecasting, and reinforcement learning (Yexin Li et al., 2018; Brinkmann et al., 2019a; Chang et al., 2023; Guo et al., 2024b). Third, several extensions—green routing, maintenance integration, user incentives, fairness, and multimodal coordination—have become substantive modelling streams rather than peripheral variants (Yanfeng Li and Yang Liu, 2021; Q. Chen et al., 2023; Pan et al., 2023; Tang et al., 2025; Hu et al., 2025).

1.1.2 Scope and Methodology

The review protocol began with a broad search of the BRP literature and associated backward reference checks, yielding approximately 948 records. After duplicate removal, preliminary screening, and eligibility filtering, 282 papers were retained for detailed full-text assessment. From that set, 104 studies were selected as the final analytical corpus because they addressed the BRP directly at the level of formulation, algorithm design, operational evaluation, benchmarking, or a closely coupled extension, or because they provided indispensable contextual or methodological background. Of these, 92 constitute the core BRP optimisation studies on which the comparative synthesis rests, and 12 are contextual/background references (bibliometric, broad bike-sharing, behavioural, or empirical works) retained for positioning and terminology. All 104 corpus entries are reported in the classification matrix of Appendix .1; the 12 contextual/background references are explicitly flagged there and are excluded from the comparative optimisation

statistics, which throughout this review refer to the 92 core studies. The final corpus spans from Kaltenbrunner et al. (2010) to papers dated 2026 and includes exact and heuristic optimisation papers, stochastic and robust models, forecasting and RL-based studies, and a limited number of contextual review papers used to position the literature. The analytical centre of gravity is operator-based rebalancing, but user-based and hybrid mechanisms are included wherever they interact materially with routing or inventory decisions. Docked, free-floating, and hybrid systems are all covered when they induce distinct optimisation structure.

Search and screening protocol. To make the review auditable, the protocol is specified in Table 3. Bibliographic searches were run on Scopus, Web of Science, and IEEE Xplore, complemented by Google Scholar for one-step backward and forward citation chasing on highly cited and recent works. The search was restricted to English-language items published between January 2010 and the search cut-off, and combined bike-sharing terms with rebalancing and methodological terms. Records were de-duplicated by DOI and title matching in a reference manager, screened at title/abstract level, and then assessed in full text against the inclusion and exclusion criteria in Table 3.

Table 3: Review protocol summary (databases, search terms, and screening criteria).

Protocol element	Specification
Databases	Scopus; Web of Science; IEEE Xplore; Google Scholar (citation chasing)
Search terms	(“bike sharing” OR “bicycle sharing” OR “bike-share” OR “dockless bike”) AND (rebalancing OR repositioning OR relocation OR redistribution OR balancing) AND (routing OR inventory OR optimisation OR “reinforcement learning” OR robust OR stochastic OR forecasting)
Time window	January 2010 – 2026
Search cut-off	Latest update in the first half of 2026
Languages	English only
Document types	Peer-reviewed journal articles; selected high-quality conference papers and preprints
Citation chasing	One-step backward and forward snowballing on seminal and recent works
Duplicate removal	DOI and title matching in a reference manager
Inclusion criteria	Addresses BRP formulation, algorithm, operational evaluation, benchmarking, or a closely coupled extension; or provides essential contextual/methodological background
Exclusion criteria	Infrastructure, planning, safety, or adoption studies without a rebalancing decision; grey literature without peer review; items not retrievable in full text

Because the BRP couples vehicle routing, inventory control, stochastic and data-driven decision-making, and last-mile mobility logistics, it lies squarely within the methodological and applied scope of logistics and transportation operations research, and the synthesis below is framed accordingly.

1.1.3 Contributions of This Survey

The survey makes five main contributions. First, it provides an organising 6-tuple formalisation of the BRP that serves as a common descriptive backbone for the reviewed variants and makes BRP-specific structure—station-capacity coupling, rebalancing targets, censored demand, broken-bike flows, and user-equilibrium response—explicit, rather than a closed formal theory that resolves every modelling distinction. Second, it develops a multi-dimensional taxonomy that distinguishes planning horizon from demand representation, fleet structure, operational scope, and rebalancing paradigm. Third, it synthesises the literature comparatively rather than paper-by-paper, highlighting what is now known about model structure, algorithmic performance, benchmark realism, and reproducibility. Fourth, it closes with a research agenda grounded in the specific bottlenecks revealed by that synthesis rather than in generic calls for “more realistic models.” Fifth, it codes the corpus for data availability, code release, benchmark reuse, parameter reporting, and operational realism, converting previously narrative claims about reproducibility and benchmark fragmentation into auditable evidence.

1.1.4 Paper Organisation

Section 1.2 harmonises terminology and traces the field’s evolution. Section 1.3 presents the formal framework and taxonomy. Sections 1.4–1.9 review the main model families and extensions. Section 1.10 audits data, benchmarks, and reproducibility. Section 1.11 compares solution paradigms and identifies the field’s central trade-offs. Sections 1.12 and 1.13 present the research agenda and concluding synthesis.

1.2 Background and Terminology

1.2.1 Terminology Harmonisation

Terminological inconsistency remains a basic obstacle to cumulative synthesis. Across the reviewed corpus, the same operational act is variously called *rebalancing*, *repositioning*, *relocation*, *redistribution*, or *balancing*. The terms often overlap, but they are not always used with the same level of generality.

In this review, *rebalancing* is the umbrella term for any intervention that corrects spatial supply-demand mismatch. *Repositioning* denotes operator-controlled physical movement of bicycles and is the default term for truck-based formulations. *Relocation* appears most often in free-floating settings, where the target is a zone rather than a docked station (Caggiani et al., 2018; Chang et al., 2021; Du et al., 2020a). *Redistribution* is treated as synonymous with repositioning (Shu et al., 2013; Ma et al., 2021). When discussing the overall problem class, we use *Bike Rebalancing Problem* (BRP) as the harmonised label.

1.2.2 Bike-Sharing System Types

System design determines the structure of the rebalancing problem. In station-based systems, bicycles are stored at fixed capacitated docks, so operators must manage both vehicle capacity and station capacity; inventory targets are naturally station-specific. In free-floating systems, bicycles can be parked anywhere within service areas, so station-capacity constraints disappear but are replaced by zonal coverage requirements and a dispersed-collection problem (Caggiani et al., 2018; Pal and Y. Zhang, 2017; Lu et al., 2023). Hybrid systems combine these structures or allow multiple operating modes in parallel (Fu et al., 2025; Shi et al., 2024).

The literature remains dominated by station-based systems because their graph structure maps directly to capacitated routing formulations. Free-floating models have grown steadily since 2017, especially where unusable bikes, zonal uncertainty, or operational flexibility make station-based abstractions too restrictive. The distribution of reviewed papers across system types is summarised in Table 4.

Table 4: Distribution of reviewed papers by bike-sharing system type. The first three rows constitute the 92 core optimisation studies; the final row lists the 12 contextual/background references.

System Type	Static BRP	Dynamic BRP	Total
Station-based (docked)	38	32	70
Free-floating (dockless)	7	8	15
Hybrid / Mixed	3	4	7
Contextual / background references	—	—	12

1.2.3 Historical Evolution of BRP Research

The literature can be read as four overlapping phases. The first, from 2010 to 2013, established the phenomenon and the foundational optimisation structure. Kaltenbrunner et al. (2010) documented temporal and spatial demand regularities using Barcelona data; Benchimol et al. (2011) proved NP-hardness for the single-vehicle problem; Contardo et al. (2012) introduced the earliest dynamic space-time formulation in the reviewed corpus; and Raviv et al. (2013a) and Chemla et al. (2013a) provided the canonical static penalty-based and arc-indexed formulations.

The second phase, roughly 2014–2017, focused on algorithmic consolidation. Stronger exact formulations, branch-and-cut algorithms, and benchmark instances appeared (Dell’Amico et al., 2014b; Erdogan et al., 2015). The same period introduced interval targets (Erdogan et al., 2014; J. Schuijbroek et al., 2017), competitive metaheuristics (Sin C. Ho and W. Y. Szeto, 2014; Sin C. Ho and W. Y. Szeto, 2017a; Iris A Forma et al., 2015a), and the first user-incentive architecture (Singla et al., 2015). The third phase, 2017–2021, expanded the modelling frontier to stochastic programming, dynamic control, MDP formulations, RL, green routing, and free-floating systems (Dell’Amico et al., 2018a; Yexin Li et al., 2018; Brinkmann et al., 2019a; Yue Wang and W. Y. Szeto, 2018). Since 2021, the field has shifted further toward robust and distributionally

robust optimisation, richer learning pipelines, maintenance-aware rebalancing, fairness, and multimodal integration (Fu et al., 2022; Q. Chen et al., 2023; Guo et al., 2024b; Q. Chen et al., 2025; Hu et al., 2025; Tang et al., 2025).

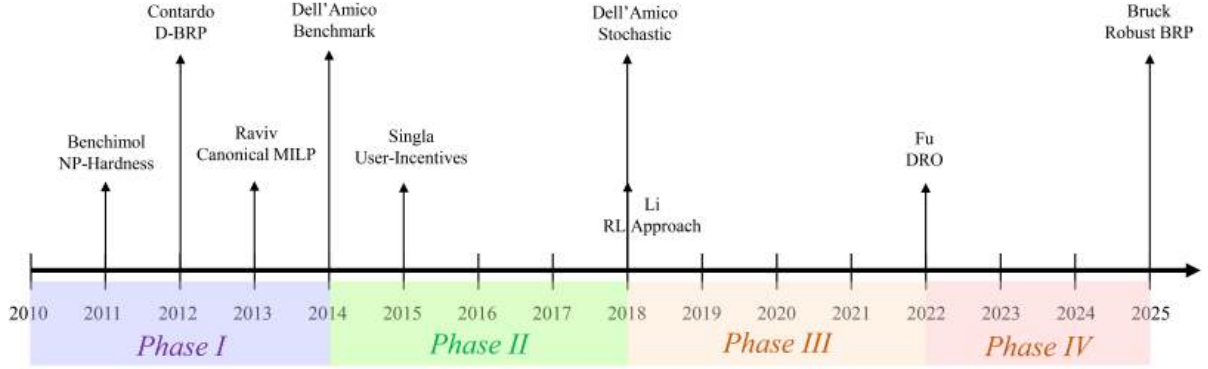


Figure 1: Historical evolution of the BRP literature, with key milestones identified in the reviewed corpus. Phase I: observational and foundational; Phase II: algorithmic consolidation; Phase III: Modelling expansion; Phase IV: Robust and intelligent integration.

1.3 Formal Framework and Taxonomy

1.3.1 Formal Definition of the BRP

We represent the BRP by the 6-tuple

$$\text{BRP} = \langle \mathcal{G}, \mathcal{V}, \mathcal{D}, \mathcal{O}, \mathcal{C}, \mathcal{H} \rangle, \quad (1)$$

where $\mathcal{G}$ is the network representation, $\mathcal{V}$ the vehicle fleet, $\mathcal{D}$ the demand model, $\mathcal{O}$ the objective, $\mathcal{C}$ the operational constraints, and $\mathcal{H}$ the planning horizon. The value of this representation is not merely classificatory. Most meaningful advances in the literature correspond to enriching one or more components: moving from complete graphs to space-time or zone graphs, from homogeneous to heterogeneous fleets, from deterministic to stochastic or robust demand, or from single-period to multi-period planning.

The same abstraction also clarifies which modelling choices are tightly coupled. Static arc-indexed formulations are feasible only on relatively compact network representations, whereas time-expanded networks grow with both the number of nodes and the number of periods. Likewise, moving from a single homogeneous truck to heterogeneous fleets or electric vehicles fundamentally changes both route feasibility structure and solution-method design.

A common backbone across most formulations is the station inventory balance equation. Let p_i be initial inventory at station i , and let y_i^+ and y_i^- denote delivered and collected bicycles. Then

$$s_i = p_i + y_i^+ - y_i^-, \quad 0 \leq s_i \leq C_i, \quad (2)$$

Table 5: Mapping of major BRP variants onto the 6-tuple. $\hat{D}$: deterministic demand; $\tilde{D}$: stochastic demand; $\bar{D}$: robust uncertainty set; D^{ML} : data-driven/learning-based demand.

Variant	$\mathcal{G}$	$\mathcal{V}$	$\mathcal{D}$	$\mathcal{O}$	$\mathcal{C}$	$\mathcal{H}$
1-S-BRP	single depot	$ \mathcal{V} = 1$	$\hat{D}$	cost/penalty	capacity	single
m-S-BRP	single depot	$ \mathcal{V} > 1$	$\hat{D}$	cost/penalty	capacity, time	single
MD-S-BRP	multi-depot	hetero. $\mathcal{V}$	$\hat{D}$	cost+inventory	capacity	single
D-BRP	space-time	$ \mathcal{V} \geq 1$	$\hat{D}$	unmet demand	TW, capacity	multi
SDIRP	station graph	$ \mathcal{V} \geq 1$	$\tilde{D}$	exp. unmet	capacity	multi
Robust-BRP	station graph	$ \mathcal{V} \geq 1$	$\bar{D}$	worst-case cost	capacity	single
BRPSD	complete graph	$ \mathcal{V} \geq 1$	scenarios	exp. recourse	capacity	single
Green-BRP	station graph	fuel vehicles	$\hat{D}$	cost+emission	TW, budget	single
Broken-BRP	station graph	$ \mathcal{V} \geq 1$	$\hat{D}$	cost+penalty	capacity	single
Incentive-BRP	user network	users	$\tilde{D}$	unmet demand	budget	multi
FF-BRP	zone graph	$ \mathcal{V} \geq 1$	$\hat{D}$	cost+penalty	zone capacity	single/multi

where C_i is station capacity. In deterministic settings, s_i is driven toward a target or interval through a penalty function; under uncertain demand, the same structure becomes an expected-penalty or worst-case-penalty problem,

$$\min \mathbb{E}_{\xi} \left[\sum_i f_i(s_i, \xi_i) \right], \quad (3)$$

and in dynamic settings it becomes a state-transition equation,

$$s_i^{\tau+1} = s_i^{\tau} + y_i^{+, \tau} - y_i^{-, \tau} + r_i^{\tau} - d_i^{\tau}, \quad (4)$$

with returns r_i^{τ} and rentals d_i^{τ} evolving over time (Contardo et al., 2012; Gleditsch et al., 2024a; Liang et al., 2024b).

The value of the 6-tuple $\langle G, V, D, O, C, H \rangle$ is not that it supersedes the routing- and inventory-routing taxonomies from which it descends, but that it makes the BRP-specific structure these generic templates leave implicit explicit and comparable across variants. The capacity and constraint components (C, O) encode the two-sided station-capacity coupling and the rebalancing targets or fill-level intervals that distinguish the BRP from the standard pickup-and-delivery and inventory-routing settings, in which depots are uncapacitated and targets are exogenous. The demand component D separates observed trip counts from the latent, censored demand that station saturation and stock-outs hide, a distinction that ordinary vehicle-routing descriptions do not require. The graph and capacity components (G, C) jointly express docked, free-floating,

and hybrid feasibility, while the horizon component H records whether recourse is static, rolling-horizon, or fully sequential. Finally, D and O together accommodate user-equilibrium response under incentives and broken-bike flows that remove vehicles from circulation. The tuple is therefore best read as an organising device that aligns heterogeneous formulations onto a common descriptive backbone; it does not resolve every deeper modelling distinction, such as the precise treatment of stochastic recourse or the boundary between operator- and user-based control, which the subsequent sections address individually.

1.3.2 Taxonomy Design and Justification

A binary static/dynamic taxonomy is no longer sufficient. A static model can be deterministic, stochastic, or robust; a dynamic model can be a rolling-horizon MILP, an MDP, an approximate dynamic program, or an RL policy. Cross-cutting features such as maintenance, sustainability, fairness, or user incentives cut across both temporal classes. We therefore organise the literature along five axes grounded in the 6-tuple: planning horizon, demand representation, fleet structure, operational scope, and rebalancing paradigm.

This taxonomy separates what determines mathematical structure from what modifies it. The primary split remains planning horizon because it changes the state space and the nature of recourse. Demand representation is the next most consequential dimension because it determines whether the model is deterministic, probabilistic, or robust. Fleet, scope, and paradigm then capture the major practical enrichments observed in the literature. The resulting classification is compact enough to be usable and rich enough to avoid misleading category assignments.

Within the demand-representation axis we treat the observed-versus-censored distinction as a first-class sub-dimension rather than an incidental modelling detail. Because bikes and docks are both capacity-constrained, recorded trips understate true demand whenever a station is empty at pickup or full at return, so a model calibrated on observed flows optimises against a systematically biased signal. We therefore classify each study by whether it (i) uses observed demand directly, (ii) acknowledges censoring without correcting for it, or (iii) explicitly de-censors or models latent demand. This coding feeds directly into the benchmark and reproducibility audit of Section 1.10, where the prevalence of uncorrected censoring is quantified, and it motivates several of the research priorities in Section 1.12.

1.3.3 Mapping the Literature onto the Taxonomy

Appendix .1 provides the full paper-by-paper classification. At a high level, the corpus is still led by static operator-based formulations, but the centre of innovation has shifted toward dynamic, uncertain, and data-driven models. Static contributions remain the dominant source of exact formulations and benchmark development. Dynamic contributions dominate the literature on operational responsiveness, MDP modelling, and RL. Uncertainty-focused papers remain fewer in number but are methodologically important because they challenge the deterministic

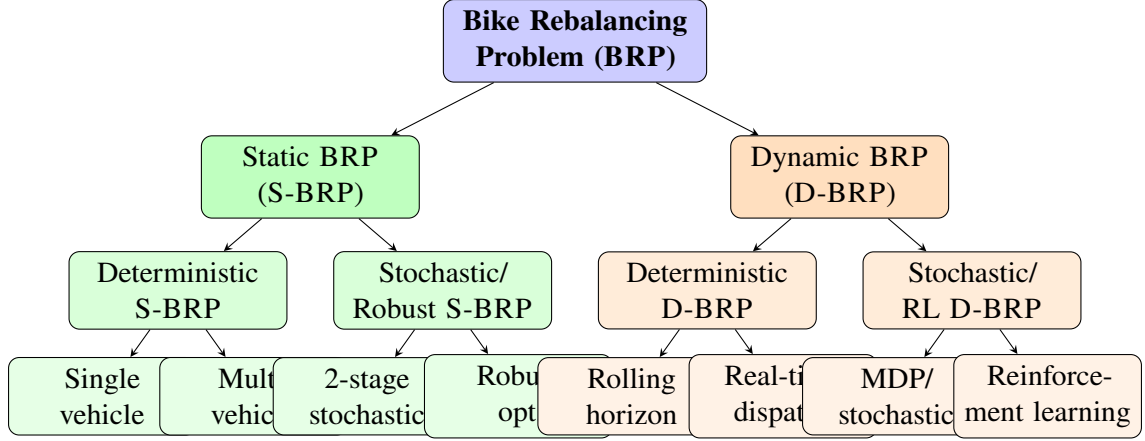


Figure 2: Multi-dimensional BRP taxonomy. The primary split is by planning horizon (A1). Cross-cutting secondary dimensions A2–A5 (demand model, fleet structure, operational scope, rebalancing paradigm) modify formulations within each branch.

assumptions built into early benchmark culture. Extension streams—maintenance, green routing, user incentives, fairness, and multimodal integration—remain thinner but are increasingly central to deployment-oriented modelling.

In the classification matrix the 12 contextual/background references are labelled as such and are excluded from all quantitative statements about formulation type, algorithm class, and benchmark practice; every count, share, and audit figure reported below refers to the 92 core BRP optimisation studies. The contextual references are retained only to harmonise terminology and to position the field, so that the comparative analysis is not diluted by works that do not contain a rebalancing optimisation model.

1.4 The Static Bike Rebalancing Problem

1.4.1 Problem Structure and Complexity

The canonical S-BRP is a single-period pickup-and-delivery routing problem on a complete graph with deterministic imbalance, capacitated vehicles, and station-capacity constraints. Decisions couple route design and loading quantities, so even simplified versions inherit the combinatorial difficulty of the VRP and TSP.

Complexity results are settled rather than tentative. Benchimol et al. (2011) established NP-hardness for the single-vehicle problem on several graph classes, and Chemla et al. (2013a) showed that although optimal loading is polynomial for a fixed visit sequence, the joint routing-and-loading problem is NP-hard. Later work confirms that exact solution remains practical only on modest instances and that feasibility becomes especially difficult when station imbalance approaches or exceeds vehicle capacity (Kadri et al., 2016a; Fabio Cruz et al., 2017; Bruck et al., 2019a). In practice, exact methods remain concentrated below roughly 60–80 stations, while larger instances require decomposition, matheuristics, or metaheuristics.

1.4.2 Foundational Single-Vehicle Formulations

Foundational single-vehicle models established the mathematical grammar of the field. Chemla et al. (2013a) formalised the static problem with arc variables and load propagation, using a polynomial loading subproblem inside an NP-hard routing shell. The key load relation,

$$l_j = l_i + d_j x_{ij} \quad \forall (i, j) \in \mathcal{A}, \quad (5)$$

made explicit how pickup and delivery decisions are embedded in route feasibility. Raviv et al. (2013a) shifted the focus from exact target satisfaction to convex shortage penalties, which better align the model with service quality and have since influenced much of the subsequent literature.

A second foundational line replaces fixed targets with acceptable intervals. Erdogan et al. (2014) modelled interval feasibility directly, J. Schuijbroek et al. (2017) derived intervals analytically from queueing models, and Teemu J. Ikonen et al. (2025) tightened the interval formulation through generalised disjunctive programming. These studies are important because they move the S-BRP closer to how operators actually define service quality: not as exact inventories, but as ranges that limit stockout and overflow risk.

Exact single-vehicle algorithms have since improved the frontier without changing the basic conclusion. Branch-and-cut, branch-and-bound, and disaggregation-based methods provide strong optimality machinery (Erdogan et al., 2015; Kadri et al., 2016a; Bruck et al., 2019a), but they do not eliminate the structural scale limit. Single-vehicle exact methods are now theoretically mature, not operationally sufficient.

1.4.3 Multi-Vehicle Static BRP

The multi-vehicle S-BRP adds assignment and coordination effects to an already hard routing-and-loading problem. Dell’Amico et al. (2014b) provided the decisive exact benchmark contribution by comparing four formulations and showing the superiority of an arc-flow model strengthened with infeasible-path cuts. Dell’Amico et al. (2016) then demonstrated that destroy-and-repair heuristics built on the same structural insights can deliver near-exact performance on small instances and strong performance on larger ones.

Subsequent progress has mostly come from decomposition and neighbourhood design rather than from radically new formulations. Iris A Forma et al. (2015a) used clustering and reduced-network optimisation to extend solvability; Sin C. Ho and W. Y. Szeto (2017a) showed that problem-aware LNS with station-type information scales well to instances above 500 stations; and W. Y. Szeto and C. S. Shui (2018) exploited the fact that loading can be solved exactly on fixed routes, making route-first/load-second architectures particularly effective. The shared lesson is clear: large-scale performance depends less on generic metaheuristic complexity than on whether the algorithm exploits the separable structure of routing, loading, and station typology.

1.4.4 Structural Variants of the Static BRP

Three static variants matter structurally. The first allows multiple visits or split demand, which becomes necessary when vehicle capacity is small relative to station imbalance (Chemla et al., 2013a; Bulhoes et al., 2018; Fabio Cruz et al., 2017; J. Liu et al., 2025). This increases flexibility but complicates both feasibility checking and exact separation. The second is the multi-depot setting, where vehicle staging locations become endogenous to routing efficiency and fleet utilisation (Lv et al., 2020; Ying Liu et al., 2018; Lv et al., 2022). The third is heterogeneous fleets, where capacity classes, vehicle technology, or hub-and-spoke structures change both route design and marginal service cost (Ying Liu et al., 2018; D. Huang et al., 2020; Ma et al., 2021; M. Chen and W. Y. Szeto, 2023a).

These variants are not cosmetic extensions. Each weakens one of the simplifying assumptions that made the original static benchmark culture possible: single service, single depot, and homogeneous trucks. Their practical relevance is high, but benchmark comparability declines sharply once these assumptions are relaxed.

1.4.5 Multi-Type and E-Bike Extensions

Multi-type systems convert the S-BRP from a one-commodity to a multi-commodity problem. Yanfeng Li et al. (2016) modelled multiple bicycle types with substitution and compartment-sharing strategies, showing that operational flexibility can materially reduce cost. Cui et al. (2025) extended this idea to mixed fleets of conventional bikes and e-bikes, with charging and non-charging station types and hybrid genetic search. The modelling implication is obvious: once bike types are differentiated, station balance is no longer a scalar target and loading feasibility becomes commodity-specific. That substantially raises algorithmic difficulty even before uncertainty or dynamic control are introduced.

1.4.6 Maintenance-Integrated Static BRP

Maintenance-integrated static models jointly rebalance usable bicycles and collect or repair unusable ones. Yue Wang and W. Y. Szeto (2018) first combined broken-bike collection with green objectives; Du et al. (2020a), Wu et al. (2024), Bruno A. Neumann-Saavedra and Cavagnini (2025), and Hu et al. (2025) expanded the idea through free-floating, recycling, column-generation, and on-site-repair formulations. The core structural change is that the problem becomes at least two-commodity: usable bicycles must be delivered where demand exists, while broken bicycles must be collected or repaired according to a different spatial logic.

This extension is operationally significant because unusable-bike accumulation directly erodes effective capacity, especially in free-floating systems. It also exposes a methodological weakness of the literature: maintenance-aware formulations are realistic, but benchmark and reproducibility infrastructure remain thin.

1.4.7 Green and Sustainability-Oriented Static BRP

Green static formulations augment service and routing objectives with emission costs or carbon-policy constraints. Early models used load-dependent emission functions (Yue Wang and W. Y. Szeto, 2018; Yue Wang and W. Y. Szeto, 2021a); later work adopted richer fuel and speed models and explicit multi-objective optimisation (M. Chen and W. Y. Szeto, 2023a; Lv et al., 2024). Yiming Li and Sun (2023) showed that different policy instruments—cap, tax, trading, and offset—generate materially different routing incentives, while H. Luo et al. (2020) demonstrated that life-cycle emission minimisation can conflict with service-optimal fleet sizing.

The main conclusion from this stream is not simply that greener routing is possible. It is that sustainability changes the efficient operating point of the system. Lower-emission solutions often use fewer or lighter trips and can therefore reduce service robustness unless policy design and service objectives are coordinated explicitly.

1.5 The Dynamic Bike Rebalancing Problem

1.5.1 Problem Structure and Contrast with Static BRP

The defining feature of the D-BRP is sequential recourse. Inventories, vehicle locations, and demand evolve during the planning horizon, so decisions at time τ alter both the state and the feasible action set at time $\tau + 1$. The field’s central computational problem is therefore not only route construction but the control of myopia: how much future information should be represented, and at what computational cost.

Dynamic formulations differ mainly in how they approximate this lookahead problem. Some embed time directly in a space-time network; some solve repeated deterministic subproblems in rolling horizon; some formulate the system as an MDP or stochastic program; and some learn a policy offline and execute it online. Table 6 summarises the main families.

1.5.2 Foundational Dynamic Formulations

The foundational dynamic papers established time-expanded inventory control as the natural extension of static routing. Contardo et al. (2012) formulated peak-hour rebalancing on a space-time network and highlighted a subtle but critical issue: fractional relaxations can imply artificial bicycle creation unless inventory-flow consistency is handled carefully. Dong Zhang et al. (2017a) generalised the approach to richer fleets and time-indexed inventory evolution. These papers clarified the dynamic state structure, but they also exposed the main limitation of time-expanded exact models: size grows quickly with both network and temporal resolution.

Table 6: Dynamic BRP modelling and solution approaches.

Approach	Formulation	Representative papers
Space-time network	Arc-flow LP/MILP	Contardo et al. (2012), Dong Zhang et al. (2017a)
Deterministic rolling horizon	Iterative MILP	C. S. Shui and W. Y. Szeto (2018), Gleditsch et al. (2024a), Liang et al. (2024b)
MDP / stochastic DP	Bellman equation, ADP	Legros (2019a), Brinkmann et al. (2019a), X. Luo et al. (2022b)
Two-stage stochastic	Scenario-based MILP	Warrington and Ruchti (2019), Xiang Li et al. (2024a)
Reinforcement learning	Q-learning, DQN, actor-critic	Yexin Li et al. (2018), Guo et al. (2024b), Shi et al. (2024)
Reactive real-time	Rule-based dispatch	Jimenez-Merono and Soriguera (2024), Legros (2019a)

1.5.3 Rolling-Horizon and Real-Time Approaches

Rolling-horizon methods dominate deterministic dynamic rebalancing because they trade global optimality for real-time solvability. A forecast is used to solve a short-horizon subproblem; the first action is executed; the state is updated; and the process repeats. C. S. Shui and W. Y. Szeto (2018) provided an early green rolling-horizon implementation, while Gleditsch et al. (2024a) offered the most operationally rigorous recent study, showing on Oslo data that subproblem quality and system-level performance do not always coincide. In particular, better route columns in an isolated subproblem need not yield fewer violations in closed-loop simulation.

Recent work has also clarified that modelling assumptions inside the rolling-horizon solver matter at least as much as the solver itself. Liang et al. (2024d) compared time discretisation, trip representation, and event-sequence assumptions and found that trip-distribution and event-order choices have the strongest performance effects. Liang et al. (2024d) showed that folding-horizon strategies can reduce myopia relative to standard rolling horizon, and Jimenez-Merono and Soriguera (2024) showed that reactive dispatch can be surprisingly competitive when forecasts are weak. The practical conclusion is blunt: dynamic sophistication is useless if the model of near-term demand evolution is wrong.

1.5.4 MDP and Stochastic Dynamic Formulations

Theoretically, the MDP is the cleanest description of dynamic rebalancing under uncertainty. The value function

$$V^r(\mathbf{s}^r) = \min_{\mathbf{a}^r \in \mathcal{A}(\mathbf{s}^r)} \left[c(\mathbf{s}^r, \mathbf{a}^r) + \mathbb{E}_{\xi^r} \left(V^{r+1}(T(\mathbf{s}^r, \mathbf{a}^r, \xi^r)) \right) \right] \quad (6)$$

shows exactly why the problem is hard: the state must encode inventories, vehicle positions, and loads, and the action space is combinatorial. Exact dynamic programming is therefore infeasible except in stylised settings.

The literature responds through approximation. Legros (2019a) derived threshold-type structural policies for a stylised cyclic network; Brinkmann et al. (2019a) combined MILP lookahead with simulation-based value approximation; X. Luo et al. (2022b) developed approximate dynamic programming for free-floating repositioning; and Warrington and Ruchti (2019) and Xiang Li et al. (2024a) connected rolling horizon with scenario-based stochastic approximation. These papers collectively show that stochastic dynamic modelling is conceptually mature, but not yet standardised enough for fair method comparison.

1.5.5 Dynamic BRP with Electric Vehicles

Electric-vehicle integration adds battery state and charging decisions to dynamic routing. Yue Wang and Szeto (2021) is the key reference in the reviewed corpus, showing that battery management changes route feasibility and can require anticipatory positioning near charging infrastructure. This is not a minor engineering detail. Once charging enters the state space, the D-BRP becomes a joint inventory-routing-energy problem.

1.5.6 Broken Bikes in Dynamic Settings

Dynamic maintenance-aware formulations address an operational reality that static models only approximate: broken bicycles are revealed over time and may require urgent collection or repair. Cai et al. (2022a) introduced stochastic broken-bike arrivals in a dynamic routing framework; Chang et al. (2021) forecast both usable-bike imbalance and damaged-bike locations jointly; and Fan et al. (2025) analysed dockless repair-removal policies through a queueing model. The common thread is that maintenance demand is neither stationary nor identical to customer demand, so it cannot be handled well by naive add-on penalties.

1.6 Stochastic, Robust, and Distributionally Robust BRP

1.6.1 Two-Stage Stochastic BRP

Two-stage stochastic programming treats route design and fleet commitment as first-stage decisions and demand-dependent service penalties as recourse. Dell’Amico et al. (2018a) established the benchmark formulation for the stochastic static BRP, using an L-shaped branch-and-cut on scenarios derived from the Dell’Amico benchmark family. The formulation is important because it formalised uncertainty without abandoning the routing structure of the deterministic S-BRP.

Subsequent work improved both benchmark quality and risk representation. Parada et al. (2026) showed that the original stochastic benchmark is too easy and proposed a harder 100-instance suite together with a disaggregated integer L-shaped method. Cavagnini et al. (2024) and Walker and Kwon (2024) incorporated CVaR to hedge service failures, and Maggioni et al. (2019a) quantified the value of stochastic modelling through transshipment formulations. The main substantive result across these studies is that uncertainty matters, but benchmark design matters almost as much: weak scenario libraries can make sophisticated methods look stronger than they are.

1.6.2 Robust Optimisation Models

Robust formulations seek solutions that remain effective across uncertainty sets rather than under estimated distributions. Bruck et al. (2025) introduced the main explicit robust static BRP in the reviewed corpus, using cardinality-constrained demand uncertainty and showing that robust solutions typically deploy more slack—more vehicles or shorter routes—than deterministic optima. Fu et al. (2022) incorporated robustness into location-and-service design, and R. Zhang et al. (2025) proposed a sample-based robust framework for multi-period free-floating rebalancing with local uncertainty neighbourhoods and linear decision rules.

Robust models are attractive when demand distributions are unstable or hard to estimate, but they remain computationally demanding and benchmark-fragmented. The empirical question is not whether robustness improves worst-case performance; it does. The real question is when its conservatism is worth the operational cost, and the literature is still thin on that trade-off at scale.

1.6.3 Distributionally Robust BRP

Distributionally robust optimisation (DRO) occupies the middle ground between scenario-based stochastic programming and set-based robustness. Q. Chen et al. (2023) introduced a target-based SDRO formulation with a dissatisfaction risk measure that protects poorly served zones rather than only minimising aggregate expected shortage. Q. Chen et al. (2025) extended that framework to heterogeneous user classes, and Q. Chen et al. (2024) applied target-level DRO to free-floating dynamic rebalancing.

These models are methodologically significant because they tie uncertainty handling to service distribution, not only to average cost. They are among the strongest examples in the corpus of uncertainty modelling aligned with policy questions such as fairness and differentiated service quality.

1.6.4 Stochastic Single-Station Inventory Models

Before routing is optimised, the operator must decide what inventory levels are worth protecting. Raviv and Kolka (2013a) gave the foundational single-station stochastic model, linking stockout

and overflow penalties to the time between repositioning events. Datner et al. (2019a) extended the problem to network-level interactions through user roaming, and Gammelli et al. (2022) studied the prescriptive value of forecast-driven inventory policies. These papers matter because they show that the BRP is partly an inventory-setting problem and not only a routing problem.

1.7 Learning-Based and Prediction-Integrated BRP

1.7.1 The Predict-Then-Optimise Pipeline in BRP

Prediction-integrated BRP models attempt to reduce the gap between demand forecasting and operational planning. Regue and Recker (2014) already contained the essential architecture: predict station states, translate uncertainty into service targets, then optimise redistribution. More recent work strengthens both layers. Chang et al. (2023) combined a multi-task spatio-temporal neural network with rolling-horizon green relocation in a free-floating system, while J. Liu et al. (2025) coupled demand prediction, simulation-based target setting, and large-scale overnight routing.

The value of this line of work is not simply higher forecast accuracy. It is the recognition that forecast design should be judged by prescriptive value: whether it yields better routing and inventory decisions, not only lower prediction error.

1.7.2 Demand Forecasting for BRP

Forecasting papers in the corpus show two consistent points. First, bike-sharing demand is strongly structured by time, location, and substitution effects among nearby stations or zones (Kaltenbrunner et al., 2010; Caggiani et al., 2018; X. Gao and Lee, 2019; Xinyu Li et al., 2023). Second, observed demand is censored when stations are empty or full, so raw trip data systematically understate true demand (X. Gao and Lee, 2019; D. Huang et al., 2020; Tang et al., 2025). Models that ignore censoring or substitution often optimise against the wrong target.

The best forecasting papers therefore contribute operationally only when they address these distortions or evaluate downstream decisions directly, as in Gammelli et al. (2022). This is a stricter standard than conventional forecasting accuracy and it is the right one for BRP research.

1.7.3 Reinforcement Learning for Dynamic BRP

RL methods replace repeated explicit optimisation by learned policies. Yexin Li et al. (2018) established feasibility in a clustered city-scale setting. Guo et al. (2024b) developed a more deployment-oriented architecture combining offline RL, online prediction updates, and fairness-aware dispatch. Shi et al. (2024) pushed the frontier further by embedding rebalancing and pricing in a competitive duopoly solved with deep Q-learning.

The attraction of RL is amortised decision-making: once trained, policies can act almost instantly. The downside is dependence on simulators, training data, and hidden modelling assumptions. In this literature, algorithmic sophistication has advanced faster than validation discipline.

1.7.4 Integration of Prediction and Optimisation: Design Principles

Three design principles recur across the prediction-integrated literature. Jointly forecasting correlated demand streams is usually better than treating pickups and returns independently (Chang et al., 2023; Xinyu Li et al., 2023). Richer predictors do not guarantee better decisions when irreducible uncertainty dominates, which is why robust downstream optimisation can still matter even after strong forecasting (Gammelli et al., 2022; Q. Chen et al., 2025). Finally, almost all reviewed learning-based systems are evaluated offline on fixed datasets; adaptation to concept drift and live operational deployment remains largely unresolved.

1.8 User-Based and Hybrid Rebalancing

1.8.1 Incentive Mechanisms and Pricing

User-based rebalancing changes the control lever from truck routes to incentives, prices, or bids. Singla et al. (2015) introduced the main benchmark architecture in the corpus, combining station selection with adaptive pricing for redirected users. Subsequent work formalised the user response more explicitly through bilevel pricing, variational inequalities, and auction or equilibrium models (Haider et al., 2018; J. Zhang et al., 2019; Cheng et al., 2021; J. Wang and Yan Wang, 2021; Jiang and Ouyang, 2022).

This stream matters for two reasons. First, it can reduce operator cost by shifting some balancing burden to users. Second, it introduces incentive-compatibility and equilibrium issues absent from operator-only models. The difficulty is that user response is uncertain, strategic, and context-dependent, so the apparent simplicity of incentive-based balancing is deceptive.

1.8.2 Hybrid Operator-User Models

Hybrid models jointly optimise truck operations and user incentives. Yanfeng Li and Yang Liu (2021) remains the clearest formal reference in the reviewed corpus: truck routes, load quantities, and incentive offers are optimised together under logit-based user response. The main insight is structural rather than numerical. Once incentives are available, the best truck routes change because users can absorb low-density tasks that are inefficient for vehicles. This is one of the strongest arguments in the literature for treating operator and user actions as complements rather than substitutes.

1.9 Cross-Cutting Extensions

1.9.1 Sustainability and Environmental Objectives

Green BRP research broadens the objective function beyond service and distance. Static and dynamic studies show that emission-aware routing can materially reduce fuel use and carbon cost, but usually by changing fleet utilisation and service patterns (C. S. Shui and W. Y. Szeto, 2018; Chang et al., 2023; M. Chen and W. Y. Szeto, 2023a; Lv et al., 2024). H. Luo et al. (2020) adds an important system-level point: life-cycle emissions may favour a smaller fleet with less frequent redistribution, which directly conflicts with service-oriented logic. Sustainability therefore cannot be treated as a minor penalty term; it changes what counts as an efficient operating policy.

1.9.2 Fairness and Service Equity

Equity-oriented papers reject purely aggregate performance criteria. Q. Chen et al. (2023) introduced a dissatisfaction risk measure that protects the worst-served zones, while Q. Chen et al. (2025) accounted for heterogeneous user classes and Guo et al. (2024b) embedded fairness priority into RL dispatch. The collective message is that efficiency-focused rebalancing systematically favours dense or profitable areas unless equity is made explicit in the objective or constraints. In other words, fairness does not emerge automatically from better optimisation.

1.9.3 Multi-Modal and Intermodal Integration

Multimodal studies link bike rebalancing to transit access and scheduling. Pan et al. (2023) and Tang et al. (2025) show that rail-access demand can change where and when bike availability matters most, while Yang Liu et al. (2022) provides behavioural evidence on integrated metro-bike route choice. F. Gao et al. (2021) adds a methodological warning for free-floating systems: zonal aggregation can materially distort both demand estimates and policy conclusions. This stream is still early, but it is strategically important because the value of rebalancing depends heavily on first/last-mile context.

1.9.4 Broken Bikes and Maintenance Integration

Maintenance integration now spans detection, collection, repair, and joint routing. Kaspi et al. (2016) supplied the statistical starting point through transaction-based broken-bike detection; later work combined detection with relocation and operational response (Chang et al., 2021; Cai et al., 2022a; Hu et al., 2025; Fan et al., 2025). The key lesson is that maintenance is not a side constraint. It modifies effective inventory, shifts spatial demand for vehicle service, and often requires different resource types from standard rebalancing.

1.10 Benchmarks, Datasets, and Reproducibility

1.10.1 Benchmark Instance Audit

Empirical evaluation is one of the field’s weakest points. Roughly two-thirds of the reviewed papers use real-world data, most commonly from Vélib, Citi Bike, Capital Bikeshare, BIXI, and a smaller set of dockless systems. But the use of real data does not guarantee comparability. Preprocessing choices, target construction, temporal aggregation, and objective definitions differ widely across studies.

The main benchmark facts are straightforward. Exact static methods rely heavily on the Dell’Amico 2014 instances; larger heuristic studies use heterogeneous city-scale datasets; and dynamic, stochastic, maintenance-aware, and learning-based models typically rely on paper-specific data pipelines. Network realism is therefore uneven, temporal realism is often limited to short windows, and operational realism remains partial because loading times, crew constraints, demand censoring, and heterogeneous travel speeds are often simplified or ignored. The literature has many datasets but very few reusable benchmarks.

Table 7: Selected real-world datasets used in the reviewed BRP literature.

System	Stations	Papers	Notes
Vélib (Paris)	1,200+	6+	Industry benchmark; Forma, Raviv, Lv2024
Citi Bike (NYC)	615+	8+	Ho2017, Liu2025, Walker2024
Capital Bikeshare (DC)	329–600	5	Chen2023A2, Chen2025A, Gao2019
Oslo Bysykkel	158	1	Gleditsch2024
Bixi (Montréal)	varies	2	Liang2024
Helsinki CityBike	150+	1	Ikonen2025
Shenzhen FFBS	500-grid	2	Chang2023, Gao2021
Beijing FFBS	29 zones	1	Ma2021
Nanjing (docked)	1,089	1	Tian2020
Houston BCycle	varies	2	Cavagnini2024, Walker2024

1.10.2 Reproducibility Assessment

Reproducibility is below the standard expected in mature OR fields. Public benchmark instances exist mainly for the classical static case. Code release is rare; fully runnable repositories are essentially absent in the reviewed corpus. Parameter reporting is acceptable in many heuristic papers but inconsistent in exact, stochastic, and learning-based studies, and computational environment details are often incomplete.

The practical consequence is severe: cross-paper comparison is costly and often impossible. Without shared instances, shared simulation environments, and shared evaluation protocols,

each paper becomes a self-contained experiment. That slows cumulative progress even when individual contributions are technically strong.

To make these observations auditable rather than impressionistic, each of the 92 core optimisation studies was coded along two groups of attributes: reproducibility (data type, public-benchmark reuse, code release, availability of a runnable repository, completeness of parameter reporting, disclosure of the computational environment, and release of a shared simulator) and operational realism (explicit modelling of loading/handling times, crew or shift constraints, demand censoring, and heterogeneous travel speeds). A study was credited with an attribute only when it was reported explicitly; ambiguous cases were coded conservatively as absent. The aggregate results are summarised in Table 8. The figures are the present review’s coding and are necessarily approximate at the margins, but they are stable enough to support the qualitative conclusions drawn above and below.

Reproducibility and operational-realism audit (92 core studies)

Attribute	Studies	Share
<i>Reproducibility</i>		
Uses real-world data	62	67%
Synthetic / literature instances only	30	33%
Reuses a public benchmark instance set	17	18%
Releases source code	6	7%
Provides a runnable repository	1	1%
Reports full parameter settings	58	63%
Discloses computational environment	22	24%
Shares a simulation environment	2	2%
<i>Operational realism</i>		
Models loading / handling times	24	26%
Models crew or shift constraints	9	10%
Accounts for demand censoring	14	15%
Models heterogeneous travel speeds	11	12%

Table 8: Coding of reproducibility and operational-realism practice across the 92 core BRP optimisation studies. Shares are rounded; an attribute is credited only when explicitly reported.

These figures make the field’s two structural weaknesses concrete. On the reproducibility side, real data are common but reusable artefacts are not: public-benchmark reuse and full parameter reporting are the only practices adopted by a majority or near-majority, whereas code release, runnable repositories, environment disclosure, and shared simulators remain rare. On the realism side, the operational features most relevant to deployment—handling times, crew constraints, demand censoring, and heterogeneous speeds—are each modelled in only a minority of studies, confirming that operational realism is partial and unevenly distributed rather than systematically addressed.

1.11 Synthesis and Critical Analysis

1.11.1 Comparative Analysis of Solution Methods

Table 9 summarises the solution-method landscape. Exact methods remain the standard for understanding formulation strength and generating ground truth on small instances. Matheuristics and metaheuristics dominate large-scale static applications because they exploit route/load decomposition and station structure. Rolling-horizon methods are the default for deterministic dynamic control because they fit operational decision windows. MDP/approximate dynamic programming and RL become attractive when the objective is long-run control rather than repeated short-horizon optimisation. Incentive-based models rely more heavily on bilevel, equilibrium, and pricing machinery because user response is central.

Table 9: Solution method landscape across BRP variant classes. • = primary contribution; ◦ = secondary or supporting role.

Solution Method	1-S-BRP	m-S-BRP	D-BRP	Stoch.	RL	Incentive
Exact B&C / B&B	•	•	—	•	—	—
L-shaped / Benders	—	◦	—	•	—	—
Math heuristic	•	•	◦	—	—	—
LNS / ALNS	◦	•	•	—	—	—
GA / evolutionary	•	•	◦	—	—	—
ILS / tabu search	•	•	◦	—	—	—
Column generation	•	•	•	—	—	—
Rolling horizon	—	◦	•	•	—	—
MDP / ADP	—	—	•	•	•	—
Deep RL (DQN, A3C)	—	—	•	—	•	—
Stochastic approx.	—	—	•	•	•	—
Pricing / VI	—	—	◦	—	—	•
Bilevel / KKT	—	—	—	—	—	•

No single method family dominates all variants. Exact approaches are indispensable for classical static and stochastic formulations but do not scale to the largest operational systems. Metaheuristics scale, but usually without optimality guarantees on realistic instances. Rolling horizon is operationally plausible but sensitive to demand modelling and simulator quality. RL offers near-instant online decisions after training, but its apparent speed is purchased with modelling opacity and high validation requirements. Method choice should therefore follow the structure of the problem class, not current fashion.

1.11.2 Trade-Off Analysis

Seven trade-offs organise the field. First, realism versus tractability: every extension that matters operationally—uncertainty, heterogeneity, maintenance, sustainability, fairness—also makes exact solution harder. Second, exactness versus scalability: the gap between proven optimality and city-scale deployment remains wide. Third, static planning versus dynamic responsiveness: overnight optimisation is easier, but it cannot absorb intra-day shocks. Fourth, modelling richness versus benchmark comparability: richer

models produce narrower and less reusable empirical evidence. Fifth, efficiency versus equity: aggregate service objectives systematically bias resources toward dense areas unless fairness is imposed explicitly. Sixth, efficiency versus sustainability: lower-emission solutions can conflict with service-optimal fleet use. Seventh, predictive sophistication versus implementability: the most elaborate learning pipelines may not dominate simpler reactive or rule-based policies once deployment cost is considered.

These trade-offs are not peripheral observations; they explain why the field has progressed unevenly. Classical static BRP is mature because its trade-offs are comparatively manageable. Dynamic, stochastic, and deployment-oriented variants are less mature because they sit where several trade-offs collide at once.

1.11.3 Maturity Assessment

Table 10: Maturity assessment of major BRP research streams. Scale: ★ = nascent, ★★ = developing, ★★ ★ = established, ★★ ★★ = mature.

Stream	Model.	Algo.	Bench.	Reprod.	Deploy
1-vehicle S-BRP	★ ★ ★★	★ ★ ★★	★ ★ ★	★★	★★
Multi-veh. S-BRP	★ ★ ★★	★ ★ ★	★ ★ ★	★★	★★
Deterministic D-BRP	★ ★ ★	★ ★ ★	★★	★★	★ ★ ★
Stochastic/2SP-BRP	★ ★ ★	★ ★ ★	★★	★★	★
Robust BRP	★★	★★	★	★	★
Learning-based D-BRP	★★	★ ★ ★	★	★	★★
Green BRP	★ ★ ★	★★	★★	★	★
Maintenance-BRP	★ ★ ★	★★	★★	★	★
Incentive / user-based	★ ★ ★	★★	★	★	★★
Equity-oriented BRP	★★	★★	★	★	★
Multi-modal BRP	★★	★★	★	★	★
E-bike / multi-type BRP	★★	★★	★	★	★

The maturity pattern is uneven but coherent. Single-vehicle and multi-vehicle static S-BRP are structurally mature and algorithmically well developed. Deterministic dynamic BRP is methodologically credible but benchmark-fragmented. Stochastic and robust streams are conceptually mature but computationally immature at realistic scale. Learning-based and extension streams are promising but reproducibility-poor. In blunt terms, the field knows how to solve the classical problem well; it does not yet know how to compare or validate the richer ones consistently.

1.12 Research Agenda

The priorities below are anchored in the corpus coding of Sections 1.3 and 1.10 rather than in generic appeals for more realism, and the three most urgent items map directly onto the weakest measured dimensions in Table 8.

The review points to nine priorities. **R1: benchmark infrastructure.** The field needs a standard multi-scale public library spanning static, dynamic, stochastic, maintenance-aware, multi-type, and city-scale instances. This priority follows directly from the audit: only about one in five core studies reuses a public benchmark set and shared simulators are essentially absent, so the absence of reusable instances—not a shortage of datasets—is the binding constraint on comparability. **R2: tighter exact methods for medium-scale static instances.** The most credible route is decomposition-based exact optimisation that pushes the frontier from laboratory scale toward operationally relevant sizes. **R3: integrated stochastic-dynamic control.** Static uncertainty models and dynamic routing models have largely evolved in parallel and need to be unified. The coding reinforces this: demand censoring is accounted for in only about 15% of core studies, so most dynamic models still optimise against biased demand signals. **R4: learning with operational validation.** RL and prediction-integrated methods require stronger simulators, online adaptation protocols, and deployment-style evaluation. With code released in roughly 7% of studies and runnable repositories all but absent, learning-based claims are at present difficult to validate or reproduce, which makes operational validation infrastructure a precondition for cumulative progress here. **R5: scalable maintenance-aware and multi-type algorithms.** These extensions are realistic but currently under-solved at scale. **R6: joint operator-user optimisation.** Hybrid balancing remains too thinly developed relative to its practical promise. **R7: equity-by-design at station resolution.** Fairness is still mostly handled at aggregated zone level. **R8: multimodal optimisation.** Bike rebalancing should be embedded more explicitly in transit-access systems. **R9: empirically calibrated green models.** Emission-aware formulations need better linkage to measured vehicle behaviour and actual operating data.

These priorities are not equally urgent. Benchmark standardisation, stochastic-dynamic integration, and operational validation are the three bottlenecks most likely to unlock cumulative progress across the rest of the field.

1.13 Conclusion

This review has condensed 104 publications on the BRP into a coherent analytical structure built on the 6-tuple formalisation and a multi-dimensional taxonomy, with the comparative analysis resting on the 92 core optimisation studies and supported by an explicit coding of their reproducibility and operational-realism practice. The main conclusion is straightforward. The classical static deterministic BRP is a mature OR problem with strong formulations, clear complexity results, and competitive exact and heuristic methods. Everything beyond that core—dynamic control, uncertainty, learning, maintenance, fairness, sustainability, and multimodal coupling—is both more relevant to practice and less mature methodologically.

The field's next phase should not be driven by adding one more extension to one more bespoke model. It should be driven by integration and validation: shared benchmarks, reproducible evaluation, uncertainty-aware dynamic control, and deployment-oriented comparison of optimisation and learning methods. Without that shift, the literature will continue to grow in volume faster than it grows in cumulative scientific value.

.1 Comprehensive Literature Matrix

Table 11 reports the complete classification of the 104 papers in the final analytical corpus. To reduce space while preserving scientific readability, the appendix table is retained in landscape format with tighter typography rather than being split into a denser two-paper-per-row layout.

Entries flagged as contextual/background in the **Type** column are the 12 references retained for positioning and terminology; the remaining 92 rows are the core BRP optimisation studies that underpin every quantitative statement in the main text. The contextual entries are listed here for completeness but are excluded from the comparative counts and from the reproducibility audit of Table 8.

Table 11: Comprehensive BRP Literature Matrix (104 papers in the final analytical corpus; 92 core optimisation studies and 12 contextual/background references, the latter flagged in the Type column).

Paper	Year	Type	Hor.	Demand	Fleet	Form.	Solution	Scope	Par.
Kaltenbrunner et al. (2010)	2010	Empirical	—	DD	—	Other	Data mining	Std	—
Benchimol et al. (2011)	2011	S-BRP	S	Det	1	VRP	Exact B&B	Std	Op
Contardo et al. (2012)	2012	D-BRP	M	Det	m	NF	LP/exact	Std	Op
Chemla et al. (2013a)	2013	S-BRP	S	Det	1	VRP	Exact B&C	Std	Op
Raviv and Kolka (2013a)	2013	S-BRP	S	Stoch	1	Other	DP/queue	Std	Op
Raviv et al. (2013a)	2013	S-BRP	S	Det	m	VRP	Exact MILP	Std	Op
Shu et al. (2013)	2013	S-BRP	S	Det	m	NF	LP+heur.	Std	Op
Dell’Amico et al. (2014b)	2014	S-BRP	S	Det	m	VRP	Exact B&C	Std	Op
Erdogan et al. (2014)	2014	S-BRP	S	Det	1	VRP	Exact B&C	Std	Op
Sin C. Ho and W. Y. Szeto (2014)	2014	S-BRP	S	Det	1	VRP	ITS	Std	Op
Regue and Recker (2014)	2014	D-BRP	M	DD	m	Other	ML+CCP	Std	Op
Iris A Forma et al. (2015a)	2015	S-BRP	S	Det	m	VRP	Matheur.	Std	Op
Erdogan et al. (2015)	2015	S-BRP	S	Det	1	VRP	Exact B&C	Std	Op
Singla et al. (2015)	2015	D-BRP	M	Stoch	Usr	Other	UCB pricing	Std	User
Dell’Amico et al. (2016)	2016	S-BRP	S	Det	m	VRP	D&R heur.	Std	Op
Kadri et al. (2016a)	2016	S-BRP	S	Det	1	VRP	Exact B&B	Std	Op
Kaspi et al. (2016)	2016	D-BRP	M	DD	—	Other	Bayesian	Maint	Op
Yanfeng Li et al. (2016)	2016	S-BRP	S	Det	1	VRP	HGA	Multi-type	Op
W. Y. Szeto et al. (2016)	2016	S-BRP	S	Det	1	VRP	CRO	Std	Op
Fishman (2016a)	2016	Survey	—	—	—	—	Literature	—	—
Fabio Cruz et al. (2017)	2017	S-BRP	S	Det	1	VRP	ILS	Std	Op
Sin C. Ho and W. Y. Szeto (2017a)	2017	S-BRP	S	Det	m	VRP	LNS+TS	Std	Op
Pal and Y. Zhang (2017)	2017	S-BRP	S	Det	m	VRP	NLNS+VND	Std	Op
J. Schuijbroek et al. (2017)	2017	S-BRP	S	Stoch	m	VRP	Queue+MSST	Std	Op
Dong Zhang et al. (2017a)	2017	D-BRP	M	Det	m	NF	LP	Std	Op
Caggiani et al. (2018)	2018	D-BRP	M	DD	m	Other	NARX+VRP	Std	Op
Dell’Amico et al. (2018a)	2018	S-BRP	S	Stoch	m	VRP	L-shaped	Std	Op
Haider et al. (2018)	2018	D-BRP	M	Det	Usr	Other	Bilevel	Std	User
Ying Liu et al. (2018)	2018	S-BRP	S	Det	H	VRP	CRO	Std	Op
W. Y. Szeto and C. S. Shui (2018)	2018	S-BRP	S	Det	m	VRP	Exact load	Std	Op
Bulhoes et al. (2018)	2018	S-BRP	S	Det	m	VRP	B&C	Std	Op
Yue Wang and W. Y. Szeto (2018)	2018	S-BRP	S	Det	m	VRP	MILP+heur	Green+Maint	Op
Laporte et al. (2018a)	2018	Survey	—	—	—	—	Literature	—	—
Soares Machado et al. (2018)	2018	Survey	—	—	—	—	Literature	—	—

Continued on next page

Table 11 continued

Paper	Year	Type	Hor.	Demand	Fleet	Form.	Solution	Scope	Par.
Yexin Li et al. (2018)	2018	D-BRP	M	DD	m	MDP	RL (PG)	Std	Op
C. S. Shui and W. Y. Szeto (2018)	2018	D-BRP	M	Det	1	VRP	ABC+RH	Green	Op
Brinkmann et al. (2019a)	2019	D-BRP	M	Stoch	1	MDP	Lookahead	Std	Op
Datner et al. (2019a)	2019	S-BRP	S	Stoch	—	Other	MILP+search	Std	Op
Legros (2019a)	2019	D-BRP	M	Stoch	1	MDP	Analytic	Std	Op
Maggioni et al. (2019a)	2019	S-BRP	S	Stoch	—	NF	2SP/MS-SP	Std	Op
Si et al. (2019)	2019	Survey	—	—	—	—	Bibliometric	—	—
X. Gao and Lee (2019)	2019	D-BRP	M	DD	m	Other	ML+routing	Std	Op
Bruck et al. (2019a)	2019	S-BRP	S	Det	1	VRP	Exact B&C	Std	Op
Warrington and Ruchti (2019)	2019	D-BRP	M	Stoch	m	NF	2-stage SA	Std	Op
J. Zhang et al. (2019)	2019	D-BRP	M	Det	Usr	Other	VI	Std	User
Lv et al. (2020)	2020	S-BRP	S	Det	m	VRP	D&R+VNS	Std	Op
Du et al. (2020a)	2020	S-BRP	S	Det	H	VRP	GA	Maint	Op
D. Huang et al. (2020)	2020	S-BRP	S	DD	H	VRP	ABC	Std	Op
H. Luo et al. (2020)	2020	S-BRP	S	Det	m	VRP	MIP+routing	Green	Op
C. S. Shui and W. Y. Szeto (2020a)	2020	Survey	—	—	—	—	Literature	—	—
Dezhi Zhang et al. (2020)	2020	S-BRP	S	Det	m	VRP	ATS+ILS	Maint	Op
Tian et al. (2020)	2020	D-BRP	M	DD	m	Other	Rule-based	Std	Op
Cheng et al. (2021)	2021	D-BRP	M	Stoch	Usr	Other	Bidding	Std	User
Yanfeng Li and Yang Liu (2021)	2021	S-BRP	S	Det	m	VRP	MILP	Std	Hyb
Ma et al. (2021)	2021	S-BRP	S	Stoch	H	Other	MILP+decomp	Std	Op
Chang et al. (2021)	2021	D-BRP	M	DD	m	Other	Attn-NN+routing	Maint	Op
Vallez et al. (2021)	2021	Survey	—	—	—	—	Sys. review	—	—
J. Wang and Yan Wang (2021)	2021	D-BRP	M	Stoch	Usr	Other	Bilevel	Std	User
Yue Wang and W. Y. Szeto (2021a)	2021	S-BRP	S	Det	m	VRP	ABC	Green+Maint	Op
Yue Wang and Szeto (2021)	2021	D-BRP	M	Det	m	VRP	MIP+RH	Std	Op
F. Gao et al. (2021)	2021	Survey	—	DD	—	Other	GeoStat	—	—
Fu et al. (2022)	2022	S-BRP	S	Rob	m	Other	Robust+B&C	Std	Op
Lv et al. (2022)	2022	S-BRP	S	Det	m	VRP	Fuzzy+heur	Std	Op
Cai et al. (2022a)	2022	D-BRP	M	Stoch	m	VRP	SetPart	Maint	Op
X. Luo et al. (2022b)	2022	D-BRP	M	Stoch	1	MDP	ADP	Std	Op
Gammelli et al. (2022)	2022	S-BRP	S	DD	—	Other	VPRNN	Std	Op
Jiang and Ouyang (2022)	2022	D-BRP	M	Stoch	m	Other	GNE	Std	User
Yang Liu et al. (2022)	2022	Survey	—	DD	—	Other	MNL	—	—
Zhou et al. (2022)	2022	Survey	—	—	—	—	Bibliometric	—	—
Chang et al. (2023)	2023	D-BRP	M	DD	m	VRP	STNN+ALNS	Green+Maint	Op

Continued on next page

Table 11 continued

Paper	Year	Type	Hor.	Demand	Fleet	Form.	Solution	Scope	Par.
M. Chen and W. Y. Szeto (2023a)	2023	S-BRP	S	Det	H	VRP	MILP+CPLEX	Green	Op
Q. Chen et al. (2023)	2023	D-BRP	M	DRO	m	NF	TSDRO	Std	Op
Lu et al. (2023)	2023	S-BRP	S	DD	m	VRP	Community+heur	Std	Op
Qiao et al. (2023)	2023	S-BRP	S	Det	m	VRP	Analysis	Std	Op
Xinyu Li et al. (2023)	2023	D-BRP	M	DD	—	Other	IrrCNN+LSTM	Std	—
Pan et al. (2023)	2023	S-BRP	S	DD	m	VRP	GWR+MILP	Multi-modal	Op
Yiming Li and Sun (2023)	2023	S-BRP	S	Det	H	VRP	MILP	Green	Op
Cavagnini et al. (2024)	2024	S-BRP	S	Stoch	1	VRP	Benders	Std	Op
Gleditsch et al. (2024a)	2024	D-BRP	M	Det	H	VRP	CG+RH	Std	Op
Liang et al. (2024b)	2024	D-BRP	M	DD	m	VRP	MILP+RH	Std	Op
Liang et al. (2024d)	2024	D-BRP	M	Det	m	VRP	MILP	Std	Op
Xiang Li et al. (2024a)	2024	D-BRP	M	Stoch	m	VRP	RH+stoch	Std	Op
Lv et al. (2024)	2024	S-BRP	S	Det	m	VRP	MOEA/D	Green	Op
Silva et al. (2024)	2024	D-BRP	M	DD	m	Other	Priority rules	Std	Op
Guo et al. (2024b)	2024	D-BRP	M	DD	H	VRP	RL+GBM	Std	Op
Jimenez-Merono and Soriguera (2024)	2024	D-BRP	M	Stoch	m	Other	Reactive	Std	Op
Q. Chen et al. (2024)	2024	D-BRP	M	DRO	m	Other	DRO	Std	Op
Walker and Kwon (2024)	2024	S-BRP	S	Stoch	1	VRP	Benders	Std	Op
Wu et al. (2024)	2024	S-BRP	S	Stoch	m	VRP	2-stage iter.	Maint	Op
Shi et al. (2024)	2024	D-BRP	M	Stoch	m	MDP	D3QN	Std	Hyb
Bruck et al. (2025)	2025	S-BRP	S	Rob	m	VRP	B&C	Std	Op
Q. Chen et al. (2025)	2025	D-BRP	M	DRO	m	NF	RS+LDR	Std	Op
Cui et al. (2025)	2025	S-BRP	S	Det	m	VRP	HGS	Multi-type	Op
Fan et al. (2025)	2025	D-BRP	M	Stoch	m	Other	Queueing	Maint	Op
Hu et al. (2025)	2025	S-BRP	S	Det	H	VRP	MILP+heur	Maint	Op
Teemu J. Ikonen et al. (2025)	2025	S-BRP	S	Det	m	VRP	GDP+B&C	Std	Op
J. Liu et al. (2025)	2025	S-BRP	S	DD	m	VRP	NARX+MILP	Std	Op
Bruno A. Neumann-Saavedra and Cavagnini (2025)	2025	S-BRP	S	Det	m	VRP	CG heur.	Maint	Op
Tang et al. (2025)	2025	D-BRP	M	DD	m	Other	Sched.+ML	Multi-modal	Op
R. Zhang et al. (2025)	2025	D-BRP	M	Rob	m	NF	SRO+LDR	Std	Op
Zhao et al. (2025)	2025	S-BRP	S	Rob	m	Other	2-stage Rob	Std	Op
Fu et al. (2025)	2025	Survey	—	Det	m	VRP	MILP mode	Std	—
Parada et al. (2026)	2026	S-BRP	S	Stoch	m	VRP	L-shaped	Std	Op
Ntamwiza et al. (2026)	2026	Survey	—	—	—	—	Bibliometric	—	—

Chapter 2

The Static Bike Rebalancing Problem with Preemption

2.1 Introduction

In recent decades, urban transportation systems have faced growing challenges, such as traffic congestion, air pollution, limited space, and the urgent need to reduce fossil fuel use. Against this backdrop, bike-sharing systems (BSS) have emerged as a key tool for promoting sustainable urban mobility. These systems mitigate traffic congestion and urban pollution by enabling users to rent bicycles from designated stations and return them to any station within a predefined network (Ricci (2015)) and help bridge the first- and last-mile gap in transportation networks (Delle Donne et al., 2025, G. Huang and Xu, 2023, Banisharif-Dehkordi et al., 2025).

BSSs now operate in hundreds of cities across Europe, North America, and Asia. Despite this expansion, their central operational challenge remains the imbalance between bicycle availability at stations and actual user demand. This mismatch, often caused by uneven urban travel patterns, can leave stations either empty or full, resulting in lost demand, where users abandon trips or choose less efficient alternatives. Such inefficiencies reduce both system effectiveness and sustainability.

Researchers have proposed two main approaches to tackle this issue: the Static Bike Rebalancing Problem (SBRP) and the Dynamic Bike Rebalancing Problem (DBRP). SBRP involves planning rebalancing actions before the operational period, usually during low-demand hours such as overnight. This process includes deciding how many bicycles to pick up or deliver at each station, planning vehicle routes, and scheduling visits. These pre-planned decisions are crucial, as they shape daily system performance and influence service quality and operational costs (Chemla et al., 2013a; Dell’Amico et al., 2014a). In contrast, DBRP addresses real-time inventory management, allowing continuous adjustments as demand fluctuates. While DBRP offers greater flexibility, it also introduces higher computational demands, requiring fast algorithms, accurate demand forecasts, and real-time data integration (Legros, 2019a; Liang et al., 2024a).

Effective station inventory management is essential for smooth operations. Traditional methods often rely on fixed target inventory levels for each station, but real-world experience shows that this can lead to unnecessary transfers and increased costs. To overcome this, recent studies (Günes Erdoğan et al., 2014; Tero J. Ikonen et al., 2024; Liang et al., 2024c) advocate for managing stations with an

ideal inventory interval rather than a fixed number. This approach, suitable for both static and dynamic settings, accommodates natural demand variations, reduces redundant movements, saves time and costs, and enhances overall efficiency.

Because the demand of a station may exceed the vehicle capacity, or because the geographical configuration of the network may favour distributing the service across several visits, three policies have emerged in the pickup-and-delivery literature (Hernandez-Perez et al., 2018; Hernández-Pérez and Salazar-González, 2022):

- the *non-split* policy, where each station is visited at most once and its demand is fully served in that single visit;
- the *without-preemption* policy, where multiple visits are allowed but pickups can only be performed at source stations and deliveries only at sink stations, so the vehicle cannot use a station as a temporary buffer;
- the *with-preemption* policy (or, simply, *preemption*), where multiple visits are allowed and the vehicle may temporarily collect bicycles from a sink station, or deliver bicycles to a source station, provided that the final inventory complies with the demand interval.

These policies have been introduced and analysed by Hernandez-Perez et al. (2018) and Hernández-Pérez and Salazar-González (2022) in the context of the Split-Demand One-Commodity Pickup-and-Delivery Travelling Salesman Problem (SD1PDTSP), and are the natural building blocks of any BSS rebalancing model. The contribution of the present paper is *not* to introduce these policies but to investigate, model, and solve a bike-sharing variant of the SD1PDTSP that simultaneously features (i) multiple visits, (ii) temporary buffering via preemption, (iii) demand intervals at the station level, and (iv) a free initial load at the depot, which can also be visited several times.

The contributions of this paper are the following.

- (C1) We introduce a flow-based MILP formulation** for the single-vehicle Static Bike Rebalancing Problem with demand intervals, multiple visits to stations and the depot, a free initial vehicle load, and the three rebalancing policies described above. The formulation closes a modelling gap left open by Hernández-Pérez and Salazar-González (2022), who consider an SD1PDTSP without demand intervals.
- (C2) We prove a structural equivalence theorem** (Theorem 2.4.3) showing that, when vehicle capacity is unbounded, handling times are non-negative, the uncharged initial load is freely chosen, and terminal load is unrestricted, every with-preemption solution can be transformed into a without-preemption solution of no greater cost. Hence, any cost advantage of preemption in this setting can arise only from binding capacity constraints.
- (C3) We design an Adaptive Iterated Local Search (AILS)** that integrates a Nested Dynamic Programming (NDP) procedure to determine the optimal loading and unloading plan for each candidate route. NDP-aware neighbourhood operators evaluate moves using the complete routing-and-handling objective rather than travel cost alone.
- (C4) We develop a $Q(\lambda)$ -guided adaptive perturbation** that selects the destruction operator and removal intensity according to the search state. The reward combines local and global improvement, while eligibility traces assign credit to recently selected actions.
- (C5) We provide a multi-part computational assessment.** The study covers 180 random instances

and 120 BICIMIA-derived instances, a ten-run component ablation, policy and visit-limit analyses, and 100 established literature configurations. AILS matches all 60 CPLEX objectives on the small random set, with 53 certified optima; on the literature set it matches all published best-known solutions, 99 of which equal published lower bounds. Results without an exact certificate are reported as incumbent, feasibility, runtime, or dispersion comparisons rather than as proofs of optimality.

The remainder of the paper is structured as follows. Section 2.2 reviews the literature on bike-sharing rebalancing and on the split-demand pickup-and-delivery problems that motivate our approach. Section 2.3 describes the real-world operational setting that motivates our problem. Section 2.4 defines the problem formally, introduces the rebalancing policies, presents the MILP formulation, and states the equivalence theorem. Section 2.5 describes the AILS metaheuristic. Section 2.6 reports the computational experiments. Section 2.7 concludes and discusses directions for future research.

2.2 Literature Review

The literature relevant to our problem spans three threads: (i) static and dynamic bike-rebalancing problems, (ii) split-demand pickup-and-delivery routing, and (iii) interval-based inventory representations. We review each in turn and conclude with a positioning of our contribution.

Chemla et al. (2013a) were among the first to formalise the single-vehicle SBRP, proposing a branch-and-cut algorithm and a tabu search that scales to 100 stations. Dell’Amico et al. (2014a) introduced standard benchmark instances and developed compact and arc-flow MILP formulations for the multi-vehicle SBRP. Exact methods continued to improve in the following years: Günes Erdoğan et al. (2014) cast the problem as a one-commodity pickup-and-delivery TSP with demand intervals; Güneş Erdoğan et al. (2015) provided an exact branch-and-cut for the static rebalancing problem with deterministic demand; and Bulhões et al. (2018a) addressed the multi-vehicle case with multiple visits per station. Heuristic methods evolved in parallel, including the iterated local search of Fábio Cruz et al. (2017), the destroy-and-repair scheme of Dell’Amico et al. (2016), and the hybrid large neighbourhood search of Sin C. Ho and W.Y. Szeto (2017c) and Sin C Ho and Wai Yuen Szeto (2017). Metaheuristic approaches based on artificial bee colony, particle swarm optimisation, and chemical reaction optimisation have also been applied (Y. Wang and W.Y. Szeto, 2018; Bruck et al., 2019b; Yue Wang and W.Y. Szeto, 2021b).

Dynamic rebalancing has received increasing attention as data availability has improved. Time-space network formulations (Contardo et al., 2012; Dong Zhang et al., 2017b), rolling-horizon frameworks (Legros, 2019a), and online heuristics combined with short-term demand prediction (X. Luo et al., 2022b; Gleditsch et al., 2022; Liang et al., 2024a) are now the dominant approaches. Reinforcement learning and approximate dynamic programming have also been explored (Guo et al., 2024a; Cai et al., 2022b). While our work is on the static case, the algorithmic ideas developed below—in particular the NDP-aware local search—can serve as a building block in rolling-horizon dynamic frameworks.

2.2.1 Split-Demand Pickup-and-Delivery and Preemption

The closest stream of literature to our problem deals with the One-Commodity Pickup-and-Delivery Travelling Salesman Problem (1-PDTSP) introduced by Hernández-Pérez and Salazar-González (2004),

in which a single capacitated vehicle moves a commodity between pickup and delivery locations to satisfy known demands. The Split-Demand variant (SD1PDTSP), formalised by Hernandez-Perez et al. (2018) and refined by Hernández-Pérez and Salazar-González (2022), allows a customer to be visited several times and forms the natural template for the problem studied here. Hernandez-Perez et al. (2018) explicitly define the three rebalancing policies that we adopt and analyse: non-split, without preemption, and with preemption. They develop a matheuristic that decomposes customer demands into partial demands, solves a 1-PDTSP through a variable neighbourhood search, and refines the resulting solution via a MILP-based branch-and-cut. Their experiments on instances with up to 500 customers show that the split-demand option helps reduce cost, while the preemption option yields more limited gains. Subsequent extensions in this family include the precedence-constrained version of Salazar-González and Santos-Hernández (2017) and the worst-case analysis of Nowak et al. (2008) for split-load pickup-and-delivery.

In the BSS literature, the three policies have appeared, sometimes implicitly, in a number of works. The non-split policy is the simplest and is adopted by W.Y. Szeto and C. Shui (2018), Liang et al. (2024a), and Bruno A Neumann-Saavedra and Cavagnini (2024), among others. The without-preemption policy underlies the algorithms of Raviv et al. (2013b), Gaspero et al. (2016), Bulhões et al. (2018a), and Bruck et al. (2019b). The with-preemption policy is considered, sometimes under different names, by Chemla et al. (2013a), J. Schuijbroek et al. (2017), Fábio Cruz et al. (2017), and Sin C. Ho and W.Y. Szeto (2017c). Despite this variety, no prior work has provided a unified MILP formulation that simultaneously handles demand intervals, multiple visits to stations and depot, free initial vehicle load, and the three policies as special cases.

2.2.2 Interval-Based Inventory Representations

Günes Erdoğan et al. (2014) were the first to model station demand as an acceptability interval $[b_i, \bar{b}_i]$ rather than a fixed target. The interval can be derived from historical demand statistics or from forecasting models that estimate the minimum and maximum quantities required to absorb the next day's demand. Subsequent works have leveraged interval-based representations in branch-and-bound algorithms (Kadri et al., 2016b), tolerance-based formulations for faulty bikes (Y. Wang and W.Y. Szeto, 2018; Bruno A Neumann-Saavedra and Cavagnini, 2024), disjunctive programming (Tero J. Ikonen et al., 2024), and dynamic settings (Liang et al., 2024c; Hua et al., 2022; Du et al., 2020b). The consensus in this stream is that demand intervals reduce unnecessary transfers and improve robustness, but the combination of interval acceptability with preemption and multi-visit routing remains under-explored.

2.2.3 Positioning of This Work

Table 12 summarises the main attributes of representative studies and positions our contribution. Compared to existing works, our paper is the first to combine (i) demand intervals, (ii) multiple visits to stations and to the depot, (iii) a free initial vehicle load, and (iv) all three rebalancing policies in a single formulation, together with a scalable matheuristic. The closest works are Hernandez-Perez et al. (2018) and Hernández-Pérez and Salazar-González (2022), who study the SD1PDTSP without demand intervals; Günes Erdoğan et al. (2014) and Tero J. Ikonen et al. (2024), who address demand intervals but with restricted preemption rules; and Fábio Cruz et al. (2017) and Sin C. Ho and W.Y. Szeto (2017c), who

allow temporary buffering but not interval acceptability.

2.3 Practical Motivation

The problem studied in this work is inspired by the operational needs of the Bicimia bike-sharing system, which serves the city of Brescia, Italy. This system is operated by Brescia Mobilità, the city's public transportation company, and represents a typical example of a station-based bike-sharing service. Bicimia covers both the central area of Brescia and several surrounding neighbourhoods, with a total of 86 stations (85 for bicycles and one central depot) distributed across an area of approximately 55 square kilometers. It is noted that Our snapshot of the Bicimia system, used to derive the real-data instances, contains 85 stations and one central depot; the system now operates about 98 stations, but throughout this paper we use the 85-station snapshot consistently for all analyses.

The stations in the Bicimia network are heterogeneous in capacity, with an average of around 10 docks per station. The smallest stations offer 6 docks, while the largest can accommodate up to 30. Among these, 37 stations are equipped with 10 docks, making them the most common type. In contrast, 7 smaller stations, each with 6 docks, are mostly located in the outskirts. Altogether, the system provides 827 docking points and maintains an active fleet of around 400 bicycles at any given time, some of which are in use while others are parked at stations. Figure 3 illustrates the geographic distribution of the bike stations and the location of the central depot. In this figure, the depot is marked with a house symbol, and each bike station is represented with a bike symbol.

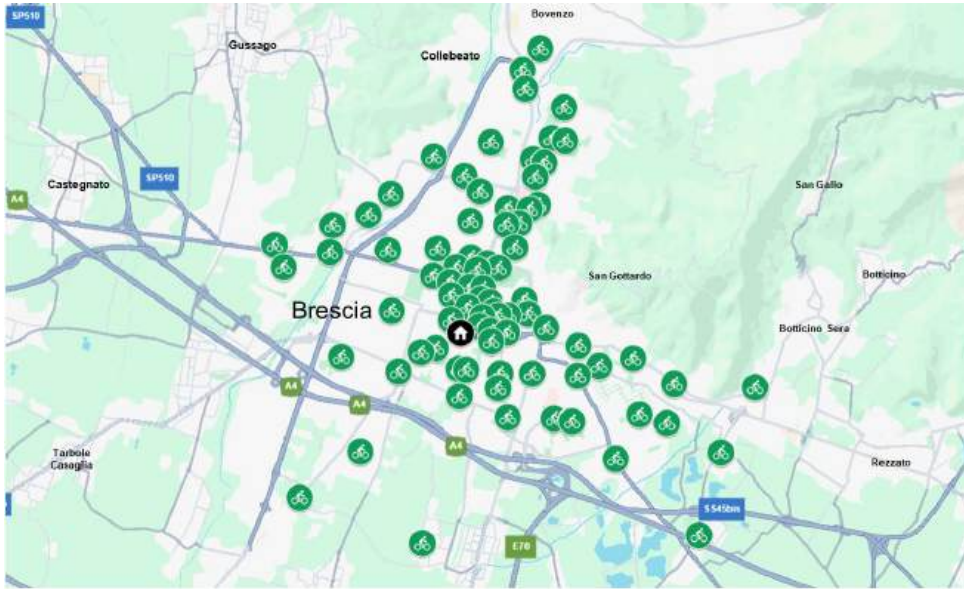


Figure 3: Locations of the depot (house symbol) and stations (bike symbols)

In the modelled setting considered in this work, overnight rebalancing is performed by one redistribution vehicle that transfers bicycles between stations to ensure that, by the beginning of the next operational day, the inventory at each station lies within an acceptability interval that absorbs the expected daily demand. This setting is adopted in the BSS rebalancing literature and is the one we analyse here. The operational cost is dominated by the time the vehicle spends travelling between stations and

Table 12: Summary of reviewed bike-sharing rebalancing papers.

Authors	Setting	Demand	Interval	Policy	Vehicle	Multi depot	Initial load	Objective	Method
Raviv et al. (2013b)	Static	Det.		w/o preemption	Hom.		Variable	Weighted cost	Time-indexed MILP
Chemla et al. (2013a)	Static	Det.		Preemption	Single		Variable	Weighted cost	Branch-and-cut, tabu search
Gişes Erdoğan et al. (2014)	Static	Det.	✓	Non-split	Single		No	Operational cost	Branch-and-cut
Dell'Amico et al. (2014a)	Static	Det.		Non-split	Hom.		Variable	Travel cost	Compact / arc-flow MILP
Gişes Erdoğan et al. (2015)	Static	Det.		Preemption	Single		Variable	Travel cost	Branch-and-cut
Kadri et al. (2016b)	Static	Det.		Non-split	Single		NA	Waiting time	Branch-and-bound
Alvarez-Valdes et al. (2016)	Static	Stoch.		w/o preemption	Hom.	✓	Fixed	Weighted cost	Multistart
Dell'Amico et al. (2016)	Static	Det.		w/o preemption	Hom.		Variable	Travel cost	Destroy & repair
Fábio Cruz et al. (2017)	Static	Det.		Preemption	Single		Variable	Weighted cost	Iterated local search
Sin C. Ho and W.Y. Szeto (2017c) and Sin C. Ho and Wai Yuen Szeto (2017)	Static	Det.		Preemption	Hom.		Variable	Weighted cost	Hybrid LNS
J. Schuijbroek et al. (2017)	Static	Det.		Preemption	Single		Variable	Weighted cost	Branch-and-cut
Hernandez-Perez et al. (2018)	Static	Det.		3 policies	Single		Variable	Travel + handling	VNS + branch-and-cut
Bulhões et al. (2018a)	Static	Det.		w/o preemption	Single		Variable	Travel cost	ILS
W.Y. Szeto and C. Shui (2018)	Static	Det.		Non-split	Hom.		Variable	Weighted cost	Lagrangian relaxation
Bruck et al. (2019b)	Static	Det.		w/o preemption	Single		Variable	Travel cost	PSO
Brinkmann et al. (2019b)	Dynamic	Det.		Non-split	Hom.		Variable	Unmet demand	Two-phase heuristic
Du et al. (2020b)	Static	Det.	✓	w/o preemption	Hom.		No	Tour length	ABC, MILP
Hua et al. (2022)	Static	Stoch.		w/o preemption	Hom.		Variable	Weighted cost	MILP
Gleditsch et al. (2022)	Dynamic	Det.		w/o preemption	Hom.		Fixed	Weighted cost	Column generation
Hernández-Pérez and Salazar-González (2022)	Static	Det.		3 policies	Single		Variable	Travel + handling	Branch-and-cut
M. Chen and W.Y. Szeto (2023b)	Static	Det.		Non-split	Hom.		Variable	Weighted cost	MILP
Bruno A. Neumann-Suavedra and Cavagnini (2024)	Static	Det.		Non-split	Hom.	✓	Variable	Weighted cost	Column generation
Tero J. Ikonen et al. (2024)	Static	Det.	✓	w/o preemption	Single		Variable	Operational cost	GA, disjunctive prog.
Liang et al. (2024a) and Liang et al. (2024c)	Dynamic	Stoch.	✓	w/o preemption	Hom.		Var.	Unmet demand	MILP
This study	Static	Det.	✓	3 policies	Single	✓	Variable	Travel + handling	AILS metaheuristic

loading or unloading bicycles. In practice, revisiting the same station can reduce total cost when its required transfer exceeds the current vehicle load; sometimes it is even useful to temporarily collect bicycles from a sink station, or to deliver bicycles to a source station, if doing so improves the route. Capturing these operational features in a deterministic optimisation model is the goal of the present work

2.4 Problem Description and Formulation

A layout is the triple $L = (G, U, S)$ describing the station network, station capacities, and initial inventories. The directed graph is $G = (V, A)$, where $V = \{0\} \cup I$, node 0 is the depot, and $I = \{1, \dots, n\}$ is the station set. An arc $(i, j) \in A$ has travel time τ_{ij} . The vectors $U = (u_i)_{i \in I}$ and $S = (s_i)_{i \in I}$ contain station capacities and initial inventories.

Station i has capacity u_i , initial inventory s_i , a minimum required bicycle stock b_i , and a minimum required number of vacant docks d_i . Its acceptable final inventory is therefore $[b_i, \bar{b}_i]$, where $\bar{b}_i := u_i - d_i$. Before rebalancing, stations are classified as

- **balanced** if $s_i \in [b_i, \bar{b}_i]$;
- a **sink** if $s_i < b_i$; and
- a **source** if $s_i > \bar{b}_i$.

For a completed solution, let $\hat{s}_i \in [b_i, \bar{b}_i]$ be the final inventory and define the realised signed transfer $q_i := s_i - \hat{s}_i$: $q_i > 0$ is a net pickup and $q_i < 0$ is a net delivery. Defining the policies through q_i removes an ambiguity created by interval targets and is used in the theorem below.

In the context of loading operations, t^L denotes the time required to load each bike onto the vehicle, and t^U the time needed to unload each bike from the vehicle.

The parameter m_{vis} denotes the maximum number of visits allowed at any station or at the depot.

To model multiple visits, we construct an expanded graph $G' = (V', A')$. Let $N_i = \{i_{(1)}, \dots, i_{(m_{\text{vis}})}\}$ be an ordered set of m_{vis} nodes representing potential visits to location $i \in V$, wherein node $i_{(l)}$ represents the l -th visit to location $i \in V$ by the vehicle, following the notation introduced in Hernandez-Perez et al. (2018). The set $V' = \cup_{i \in V} N_i$ is the node-set of the graph $G' = (V', A')$ where A' contains the arcs connecting nodes, with travel time $\tau_{i_{(l)}j_{(l')}} = \tau_{ij}$ for $i, j \in V$, and $1 \leq l, l' \leq m_{\text{vis}}$, inheriting travel times from the graph G .

A vehicle of capacity Q starts at the depot with a freely chosen initial load, visits nodes to load or unload bicycles, may revisit both stations and the depot, and finally returns to the depot. Under the most flexible policy it may temporarily move bicycles in the direction opposite to the realised net transfer q_i .

Rebalancing policies. We consider the three policies described in Section 2.1, illustrated in Figure 4.

Let $k_{i,l} := v_{i_{(l)}}^+ - v_{i_{(l)}}^-$ be the signed transfer at visit l , positive for pickup. Under the *non-split* policy (Fig. 4a), at most one $k_{i,l}$ is nonzero and it equals q_i . Under the *without-preemption* policy (Fig. 4b), multiple visits are allowed but every nonzero $k_{i,l}$ has the sign of q_i and $\sum_l k_{i,l} = q_i$. Under the *with-preemption* policy (Fig. 4c), intermediate transfers may have either sign, provided that $\sum_l k_{i,l} = q_i$. Thus only the with-preemption policy permits temporary buffering; all policies preserve the selected final inventory $\hat{s}_i = s_i - q_i$.

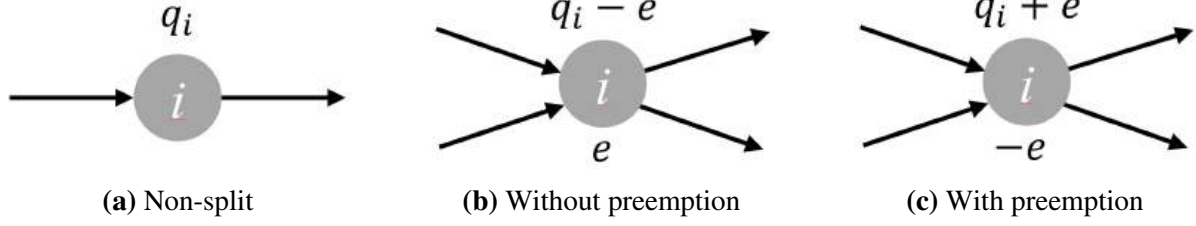


Figure 4: Illustration of the three rebalancing policies.

The Static Bike Rebalancing Problem with Preemption (SBRP-P) aims to find a minimum cost route in which a single vehicle performs a series of pickup and delivery operations ensuring that the following constraints are not violated:

- The route starts and ends at the depot;
- The vehicle capacity is not exceeded (and negative loads are not allowed) at any point of the route;
- Each station or the depot can be visited at most m_{vis} times;

And also, we assume that:

- the vehicle starts its tour with an initial load which is a decision variable taking values in $\{0, \dots, Q\}$;
- multiple visits to a station and the depot are allowed;
- it could be possible to deliver bikes to (pick up bikes from) a source (sink) station;
- travel time includes an expected time for parking near the station;
- service time for loading or unloading one bike is also considered.

The cost function includes both the routing cost (total travel time) and the flow cost (total operational time required for loading and unloading bikes).

2.4.1 An illustrative example

Figure 5 shows a small instance with five stations: three sink stations, one source station and one balanced station. Table 13 lists the relevant data; the vehicle capacity is $Q = 4$ and each loading or unloading operation requires one time unit per bike.

Table 13: Travel times (in minutes) and station data of the illustrative instance.

	capacity	init. stock	b_i	$\bar{b}_i$	depot	St. 1	St. 2	St. 3	St. 4	St. 5
depot						5	5	10	8	11
Station 1	4	0	3	4	7		7	10	9	13
Station 2	5	1	4	5	3	6		6	5	8
Station 3	4	1	3	4	9	11	7		6	8
Station 4	4	4	0	1	6	9	4	7		6
Station 5	5	3	1	5	9	12	7	8	6	

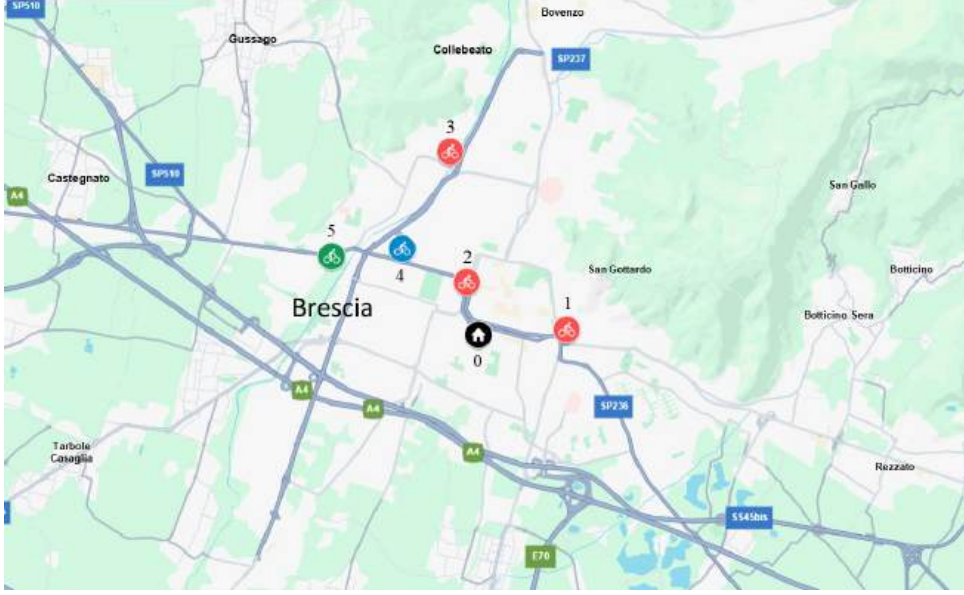


Figure 5: Five-station illustrative instance.

Table 14: One representative route for each policy.

Policy	Route (loading operations)	Routing cost	Flow cost	Total
Non-split	$0(+3) \rightarrow 1(-3) \rightarrow 4(+4) \rightarrow 3(-2) \rightarrow 5(+1) \rightarrow 2(-3) \rightarrow 0$	39	16	55
Without preemption	$0(+4) \rightarrow 1(-3) \rightarrow 4(+1) \rightarrow 3(-2) \rightarrow 4(+3) \rightarrow 2(-3) \rightarrow 0$	34	16	50
With preemption	$0(+4) \rightarrow 1(-3) \rightarrow 2(+1) \rightarrow 3(-2) \rightarrow 4(+4) \rightarrow 2(-4) \rightarrow 0$	31	18	49

Table 14 reports the best route obtained under each policy. Under the non-split policy, every station must be served in a single visit; this forces the vehicle to alternate between sink and source stations, due to the characteristics of the example, since two consecutive sink stations cannot be served with a single load. Under the without-preemption policy, station 4 is visited twice and the vehicle splits its pickup, saving one routing leg. Under the with-preemption policy, the vehicle *picks up* a bicycle at station 2 (a sink station) so that the load fits the subsequent pickup at station 4; the bicycle is later returned to station 2 in the second visit. The total cost decreases from 55 (non-split) to 50 (without preemption) to 49 (with preemption), a relative improvement of about 11% and 2%, respectively.

2.4.2 Mathematical formulation

In the following, the mathematical formulation for SBRP-P is presented. All variables are defined with respect to the expanded graph $G' = (V', A')$. Let x_{ij} be a binary variable equal to 1 if the vehicle travels directly from node i to node j , and 0 otherwise, while y_i denotes a binary variable for node i visit by the vehicle.

The integer variables v_i^+ and v_i^- track how many bikes are loaded from and unloaded to the node i by the vehicle, respectively. The number of bikes in the vehicle immediately after departing from node i and traveling toward node j is denoted by f_{ij} .

For each station i and visit copy $i_{(l)}$, the integer variable $g_{i_{(l)}}$ is the station inventory immediately after service at that visit. We define $g_{i_{(0)}} := s_i$, so Constraints (25)–(26) are also well-defined for $l = 1$. The

integer variable z_{ij} gives the ordinal position of arc $(i, j) \in A'$ in the route.

Table 15: Notation for the static model.

Sets and indices	
$V = \{0, 1, \dots, n\}$	Locations, comprising depot 0 and station set $I = \{1, \dots, n\}$.
A	Arcs between physical locations.
$N_i = \{i_{(1)}, \dots, i_{(m_{\text{vis}})}\}$	Potential visit copies of location i .
$V' = \bigcup_{i \in V} N_i; A'$	Node and arc sets of the expanded graph $G' = (V', A')$.
$i, j; l, l'$	Location or visit-copy indices; visit indices $1 \leq l, l' \leq m_{\text{vis}}$.
Parameters	
τ_{ij}	Travel time from location i to location j .
$u_i; s_i$	Capacity and initial inventory of station i .
$b_i; d_i; \bar{b}_i = u_i - d_i$	Minimum bicycles, minimum vacant docks, and maximum acceptable bicycles.
$Q; m_{\text{vis}}$	Vehicle capacity; maximum number of visits to a location.
t^L, t^U	Per-bicycle loading and unloading times.
Decision variables	
$x_{ij}; y_i$	Arc-use and visit indicators on G' .
$v_i^+; v_i^-$	Bicycles loaded onto and unloaded from the vehicle at visit copy i .
f_{ij}	Vehicle load after departing i towards j .
$g_{i(l)}$	Station inventory after visit copy $i_{(l)}$, with $g_{i(0)} := s_i$.
z_{ij}	Ordinal position of selected arc (i, j) in the route.

$$\min z = \sum_{i \in V'} \sum_{j \in V'} \tau_{ij} x_{ij} + \sum_{i \in V' \setminus \{0_{(1)}\}} (t^L v_i^+ + t^U v_i^-) \quad (7)$$

Subject to

$$y_{0_{(1)}} = 1 \quad (8)$$

$$\sum_{j \in V'} x_{ij} = y_i \quad \forall i \in V' \quad (9)$$

$$\sum_{j \in V'} x_{ij} - \sum_{j \in V'} x_{ji} = 0 \quad \forall i \in V' \quad (10)$$

$$\sum_{j \in V'} f_{ij} - \sum_{j \in V'} f_{ji} = v_i^+ - v_i^- \quad \forall i \in V' \setminus \{0_{(1)}\} \quad (11)$$

$$\sum_{j \in V'} f_{0_{(1)}j} = v_{0_{(1)}}^+ \quad (12)$$

$$f_{ij} \leq Q x_{ij} \quad \forall i, j \in V' \quad (13)$$

$$\sum_{j \in V'} z_{0_{(1)}j} = 1 \quad (14)$$

$$\sum_{j \in V'} z_{ij} - \sum_{j \in V'} z_{ji} = y_i \quad \forall i \in V' \setminus \{0_{(1)}\} \quad (15)$$

$$\sum_{j \in V'} z_{j0_{(1)}} = \sum_{i \in V'} \sum_{j \in V'} x_{ij} \quad (16)$$

$$z_{ij} \leq \mathcal{M} x_{ij} \quad \forall i, j \in V' \quad (17)$$

$$y_{i_{(l+1)}} \leq y_{i_{(l)}} \quad \forall i \in I, 1 \leq l \leq m_{\text{vis}} - 1 \quad (18)$$

$$\sum_{j \in V'} z_{i_{(l)}j} \leq \sum_{j \in V'} z_{i_{(l+1)}j} + (1 - \sum_{j \in V'} x_{i_{(l+1)}j}) \mathcal{M} \quad \forall i \in I, 1 \leq l \leq m_{\text{vis}} - 1 \quad (19)$$

$$g_{i(l)} \leq s_i + v_{i(l)}^- - v_{i(l)}^+ + (1 - \sum_{j \in V'} x_{i(l)j})\mathcal{M} \quad \forall i \in I \quad (20)$$

$$g_{i(l)} \geq s_i + v_{i(l)}^- - v_{i(l)}^+ - (1 - \sum_{j \in V'} x_{i(l)j})\mathcal{M} \quad \forall i \in I \quad (21)$$

$$g_{i(l)} \leq g_{i(l-1)} + v_{i(l)}^- - v_{i(l)}^+ + (1 - \sum_{j \in V'} x_{i(l)j})\mathcal{M} \quad \forall i \in I, 2 \leq l \leq m_{\text{vis}} \quad (22)$$

$$g_{i(l)} \geq g_{i(l-1)} + v_{i(l)}^- - v_{i(l)}^+ - (1 - \sum_{j \in V'} x_{i(l)j})\mathcal{M} \quad \forall i \in I, 2 \leq l \leq m_{\text{vis}} \quad (23)$$

$$0 \leq g_{i(l)} \leq u_i y_{i(l)} \quad \forall i \in I, 1 \leq l \leq m_{\text{vis}} \quad (24)$$

$$v_{i(l)}^- \leq u_i - g_{i(l-1)} \quad \forall i \in I, 1 \leq l \leq m_{\text{vis}} \quad (25)$$

$$v_{i(l)}^+ \leq g_{i(l-1)} \quad \forall i \in I, 1 \leq l \leq m_{\text{vis}} \quad (26)$$

$$\sum_{1 \leq l \leq m_{\text{vis}}} (v_{i(l)}^- - v_{i(l)}^+) + s_i \geq b_i \quad \forall i \in I \quad (27)$$

$$\sum_{1 \leq l \leq m_{\text{vis}}} (v_{i(l)}^+ - v_{i(l)}^-) + u_i - s_i \geq d_i \quad \forall i \in I \quad (28)$$

$$f_{ij} \geq 0, x_{ij} \in \{0, 1\}, z_{ij} \in \mathbb{Z}^+ \quad \forall i \in V', j \in V' \quad (29)$$

$$g_{i(l)} \geq 0, y_i \in \{0, 1\}, v_i^+, v_i^- \in \mathbb{Z}^+ \quad \forall i \in V' \quad (30)$$

The objective of the problem (7) is to minimise the sum of the total working time of the vehicle including the total travel time and total loading and unloading time. Constraint (8) forces the departure of the vehicle from the depot. Constraints (9) link each visit copy to at most one outgoing arc; repeated physical visits are represented by distinct copies in N_i . Constraints (10) establish connectivity requirements. Constraints (11) enforce load conservation at station i . Constraints (12) outline the initial load of the vehicle. Constraints (13) force the model to find loads that do not exceed the vehicle capacity. Within the set of Constraints (14) to (17), the model ensures the precise positioning of visited nodes along the route. Constraint (18) enforces the prerequisite that the $(l + 1)th$ visit to node i is feasible if the $(l)th$ visit is performed. Constraints (19) enforce ordering constraints, ensuring that the assigned number to the outgoing arc of the $(l + 1)th$ visit to node i exceeds the number assigned to the outgoing arc of the $(l)th$ visit to node i . Constraints (20) through (23) are designed to compute the quantity of bikes present at node i subsequent to the completion of the rebalancing operation. Constraints (24) impose non-negativity of bikes at each node, coupled with a restriction that the count at each node remains below its capacity. Constraints (25) and (26) guarantee the number of loaded and unloaded bikes on the vehicle at each station, respectively. Constraints (27) and (28) compel the model to confine the number of bikes at each station within the specified minimum and maximum intervals. Constraints (29) and (30) define the variable domains.

The displayed formulation is the with-preemption model. For a fixed realised net transfer q_i , the without-preemption variant additionally restricts every $k_{i,l} = v_{i(l)}^+ - v_{i(l)}^-$ to have the sign of q_i ; the non-split variant further permits at most one nonzero $k_{i,l}$. These restrictions are imposed when the three policies are compared.

2.4.3 An equivalence theorem under unbounded capacity

A natural question is whether the with-preemption policy can strictly outperform the without-preemption policy. The next result fixes the final inventory selected at every station and states all depot and handling-time assumptions used by the transformation.

Theorem. Assume $Q = +\infty$, $t^L, t^U \geq 0$, the initial vehicle load is a free decision whose loading at the initial depot is not charged, and the terminal vehicle load is unrestricted. Fix any feasible with-preemption solution and its final station inventories $(\hat{s}_i)_{i \in I}$. There exists a feasible without-preemption solution with the same visit sequence, the same final station inventories, and total cost no larger. Consequently, if z_{PR}^* and z_{WP}^* are the optimal values under with- and without-preemption, then

$$z_{\text{PR}}^* = z_{\text{WP}}^*.$$

Proof. Every without-preemption solution is feasible with preemption, so $z_{\text{PR}}^* \leq z_{\text{WP}}^*$. For the converse, consider a feasible with-preemption solution and keep its ordered visit sequence fixed. At visit l to station i , write the pickup and delivery quantities as $p_{i,l} \geq 0$ and $d_{i,l} \geq 0$, and let

$$q_i := \sum_l (p_{i,l} - d_{i,l}) = s_i - \hat{s}_i.$$

Construct new station operations by performing $\max\{q_i, 0\}$ pickups or $\max\{-q_i, 0\}$ deliveries at the first visit to station i , and zero transfer at every later visit. The station inventory therefore moves directly from s_i to $\hat{s}_i$ at its first visit. Both values lie in $[0, u_i]$, so station-capacity feasibility and the prescribed final inventory are preserved. All nonzero transfers have the sign of q_i , hence the transformed operations satisfy the without-preemption policy.

Index all retained visits in route order by $r = 1, \dots, R$, and let k'_r be the transformed signed transfer, positive for pickup. Define the prefix sum $S_r := \sum_{h=1}^r k'_h$ and choose the free initial load

$$F' := -\min\{0, S_1, \dots, S_R\}.$$

The vehicle load after visit r is $F' + S_r \geq 0$. Because $Q = +\infty$, there is no upper-capacity violation. The route sequence, depot visits, and all policy constraints are unchanged; the terminal load may be carried back to the depot under the stated assumption.

It remains to compare handling costs without assuming $t^L = t^U$. Let $P_i := \sum_l p_{i,l}$ and $D_i := \sum_l d_{i,l}$. The original station handling cost is $t^L P_i + t^U D_i$, whereas the transformed cost is

$$t^L \max\{q_i, 0\} + t^U \max\{-q_i, 0\}.$$

If $q_i \geq 0$, then $P_i = q_i + D_i$ and the cost reduction is $(t^L + t^U)D_i \geq 0$. If $q_i < 0$, then $D_i = -q_i + P_i$ and the reduction is $(t^L + t^U)P_i \geq 0$. Summing over stations shows that handling cost does not increase, while routing cost is identical because the visit sequence is fixed. Thus a without-preemption solution of no larger cost exists for every with-preemption solution, giving $z_{\text{WP}}^* \leq z_{\text{PR}}^*$ and therefore equality. $\square$

Discussion. Theorem 2.4.3 delineates precisely when preemption can be useful: only when the vehicle capacity binds along the route. In practice, Q is small relative to the total quantity to be redistributed, so preemption can still produce non-trivial improvements. The theorem also motivates the algorithmic design of the AILS metaheuristic: because the structural difference between the two policies is concentrated in capacity-tight regions of the route, neighbourhood operators that focus on the inventory profile—rather than only on the visit sequence—are essential, and this is precisely what the NDP-aware operators do.

2.5 The Adaptive Iterated Local Search (AILS)

We develop an Adaptive Iterated Local Search (AILS) metaheuristic to solve the SBRP-P. Iterated Local Search (ILS) is a metaheuristic framework that repeatedly applies a local search starting from perturbed versions of the incumbent solution Lourenço et al. (2003); a classical limitation of ILS for SBRP is that its routing-only operators evaluate moves on travel time alone, ignoring that the flow cost may increase as a side-effect of an apparent routing improvement. The algorithm combines the diversification power of ILS with two key innovations: the *NDP-aware full optimisation* (Algorithm 4), which jointly minimises routing and flow costs in every neighbourhood move, correcting a limitation in which routing-only operators could improve travel time while inadvertently increasing transfer costs, and an adaptive perturbation strategy driven by Q-learning that dynamically selects the degree of solution disruption at each iteration. The overall framework is summarised in Algorithm 1.

The algorithm begins by constructing an initial solution x^0 (line 1) and improving it with the local search procedure (line 2).

At each iteration,

the current solution is perturbed (line 8), locally improved (line 10), and evaluated against both current and best-known solutions to compute a reward signal (line 17). If the resulting solution is better than the best solution, it becomes the new best solution (line 12).

The algorithm proceeds with the current solution in the next iteration if its cost does not exceed an acceptable deviation (θ) from the best solution. Otherwise, the best solution is considered. The algorithm ends when there is no improvement after Max_{rep} iterations (line 4).

2.5.1 Initial Solution Method

The initial solution is built through a sector-based greedy heuristic. The *visit set* $\mathcal{V} \subseteq V'$ contains the node copies that must be served to satisfy demand, including additional copies for stations whose demand exceeds the vehicle capacity.

Formally, for each sink station i with $b_i > s_i$ we add $\min\{[(b_i - s_i)/Q], m_{\text{vis}}\}$ copies to $\mathcal{V}$, and analogously for source stations.

The complementary *buffer set* $\mathcal{B} = V' \setminus \mathcal{V}$ contains optional copies that act as temporary buffers under the with-preemption policy.

The geometric centroid c of all locations V is computed and the polar angle α_i of every station with respect to c is calculated. The vehicle direction (clockwise or counter-clockwise) is chosen uniformly at random. The first node is selected from $\mathcal{V}$ by a roulette-wheel mechanism that favours nodes close to the depot. After each addition, the NDP returns the feasibility window $[\text{LB}, \text{UB}]$ on the vehicle load upon

Algorithm 1 Adaptive Iterated Local Search (AILS)

Require: Layout $\mathcal{L} = (G, U, S)$; Q ; Max_{rep} **Ensure:** Best solution x^*

```
1:  $x^0 \leftarrow \text{INITIALSOLUTION}(\mathcal{L})$   $\triangleright$  Algorithm 2: sector-based construction
2:  $x^{\text{best}} \leftarrow \text{FULLOPTNDP}(x^0)$   $\triangleright$  Algorithm 4: extraction-insertion, swap, reversal, Or-opt
3: Initialise  $Q(\lambda)$  table  $\mathbf{Q}(s, a) \leftarrow \frac{1}{1-\zeta}$  for all  $s \in S, a \in \mathcal{A}$   $\triangleright$  Section 2.5.4
4: while  $\text{ctr} < \text{Max}_{\text{rep}}$  do
5:    $s \leftarrow \text{GETSTATE}(\text{cost}(x^{\text{cur}}), \text{cost}(x^{\text{best}}), \text{ctr})$   $\triangleright$  Eq. (31): gap, stagnation, last-success
     indicators
6:    $a \leftarrow \varepsilon\text{-GREEDY}(\mathbf{Q}, s)$ 
7:    $(\rho, \delta) \leftarrow \text{DECODE}(a)$   $\triangleright$  operator type  $\rho$ , removal fraction  $\delta$ 
8:    $x^{\text{cur}} \leftarrow \text{PERTURB}(x^{\text{cur}}, \rho, \delta)$   $\triangleright$  Algorithm 5
9:   if  $x^{\text{cur}}$  is feasible then
10:     $x^{\text{cur}} \leftarrow \text{FULLOPTNDP}(x^{\text{cur}})$ 
11:    if  $\text{cost}(x^{\text{cur}}) < \text{cost}(x^{\text{best}})$  then
12:       $x^{\text{best}} \leftarrow x^{\text{cur}}$ 
13:       $\text{ctr} \leftarrow 0$ 
14:    end if
15:  end if
16:  if  $x^{\text{cur}}$  is infeasible or  $\text{cost}(x^{\text{cur}}) > (1 + \theta) \times \text{cost}(x^{\text{best}})$  then
17:     $x^{\text{cur}} \leftarrow x^{\text{best}}$ 
18:  end if
19:   $\mathbf{Q}\text{-UPDATE}(\mathbf{Q}, s, a, x^{\text{cur}}, x^{\text{prev}}, x^{\text{best}})$   $\triangleright$  Eq. (33): eligibility-trace  $Q(\lambda)$ 
20:   $\text{ctr} \leftarrow \text{ctr} + 1$ 
21: end while
22: return  $x^{\text{best}}$ 
```

departing the last node.

At each subsequent step, an angular sector of width 3ϕ in the direction of travel and ϕ in the opposite direction (with $\phi = 15^\circ$ initially) is considered. An eligible node compatible with $[\text{LB}, \text{UB}]$ is chosen from $\mathcal{V}$ within these sectors via a roulette wheel biased toward nearby stations. If no eligible node exists, the sector width is doubled until the entire circle is covered. If still no node from $\mathcal{V}$ is available, a buffer node from $\mathcal{B}$ is inserted; the sector parameters are reset and the procedure continues. The pseudocode is given in Algorithm 2; Figure 6 illustrates one iteration.

2.5.2 Nested Dynamic Programming

Given a route $\mathcal{R} = \langle \mathcal{R}_1, \dots, \mathcal{R}_n \rangle$, in which repeated station indices represent repeated visits, the NDP computes signed transfers $k_1, \dots, k_n$ (positive for pickup and negative for delivery) that minimise handling cost subject to station intervals and vehicle capacity. The NDP receives the policy as an input. Under without-preemption, its transfer interval is intersected with $[0, +\infty)$ when the realised net transfer is positive and with $(-\infty, 0]$ when it is negative; under non-split, all but one visit transfer are additionally fixed to zero. With preemption imposes no sign restriction. The starting depot is included in the route, so exploring its feasible transfer determines the initial vehicle load endogenously.

The procedure maintains a *state pool*: at every position i , each state is a triple $(j, \mathbf{K}, c)$, where

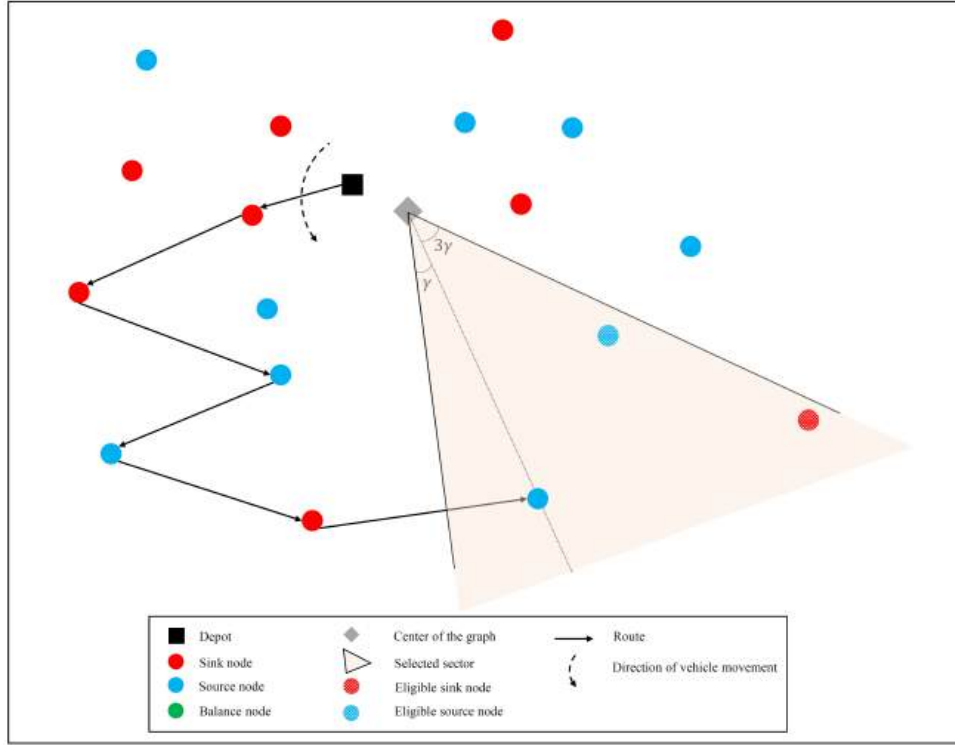


Figure 6: Sector-based node selection

$j \in \{0, \dots, Q\}$ is the on-board load, after performing the transfer at position i , $\mathbf{K} \in \mathbb{Z}^r$ records the cumulative net transfer at each of the r stations that appear *more than once* in $\mathcal{R}$ (stations appearing only once have a single transfer that is fully encoded in the load transition, so they need not be tracked in $\mathbf{K}$), and $c \in \mathbb{Z}_{\geq 0}$ is the accumulated flow cost.

When two states share the same $(j, \mathbf{K})$ at the same position, the one with the larger cost is dominated and discarded.

Two states with the same $(j, \mathbf{K})$ are merged, keeping the smaller cost.

The transfer bounds at the i -th position are tightened by combining the physical bounds (vehicle load and station capacity) with a *lookahead* that exploits the fact that the cumulative net transfer at station $\mathcal{R}_i$ must eventually reach the range $[s_{\mathcal{R}_i} - \bar{b}_{\mathcal{R}_i}, s_{\mathcal{R}_i} - b_{\mathcal{R}_i}]$ within the remaining visits. The complete derivation is reported in Appendix .1.

2.5.3 NDP-aware full optimisation

A central feature of AILS is replacing standard routing-only neighbourhood operators with NDP-aware equivalents. Routing-only operators (single-node relocation, swap, 2-opt, Or-opt) minimise travel time in isolation; changing the visit order may, however, increase transfer cost or render the route infeasible. The FULLOPTNDP procedure (Algorithm 4) evaluates every candidate move with the full NDP, so only moves that strictly reduce the *total* cost (routing plus flow) are accepted. To keep the local search efficient, a granular pre-filter excludes moves whose routing-only delta cannot improve the current best total cost even when assuming the minimum possible handling cost.

Algorithm 2 Initial Solution Construction

Require: Visit set $\mathcal{V}$, buffer set $\mathcal{B}$, depot 0, initial sector half-angle $\phi_0 = 15^\circ$

Ensure: Feasible route $\mathcal{R}$, or FAILURE

```
1: Compute centroid  $c$  and polar angles  $\{\alpha_i\}_{i \in \mathcal{V}}$  w.r.t.  $c$ 
2:  $\text{dir} \leftarrow \text{UNIFORM}(\{\text{CW}, \text{CCW}\})$ ;  $\phi \leftarrow \phi_0$ 
3:  $v_1 \leftarrow \text{ROULETTEWHEEL}(\mathcal{V}, 0)$ ;  $\mathcal{R} \leftarrow \langle 0, v_1 \rangle$ ;  $\mathcal{V} \leftarrow \mathcal{V} \setminus \{v_1\}$ 
4:  $[\text{LB}, \text{UB}] \leftarrow \text{NDP}(\mathcal{R})$ 
5: while  $\mathcal{V} \neq \emptyset$  do
6:    $\mathcal{E} \leftarrow \{v \in \mathcal{V} : \alpha_v \in \text{Sector}(\text{dir}, 3\phi, \phi), \text{ compatible with } [\text{LB}, \text{UB}]\}$ 
7:   if  $\mathcal{E} \neq \emptyset$  then
8:      $v^* \leftarrow \text{ROULETTEWHEEL}(\mathcal{E}, \text{last}(\mathcal{R}))$ ; append  $v^*$  to  $\mathcal{R}$ ;  $\mathcal{V} \leftarrow \mathcal{V} \setminus \{v^*\}$ ;  $\phi \leftarrow \phi_0$ 
9:   else if  $\phi < 90^\circ$  then
10:     $\phi \leftarrow 2\phi$ 
11:   else if  $\mathcal{B} \neq \emptyset$  then
12:     $b \leftarrow \text{ROULETTEWHEEL}(\mathcal{B}, \text{last}(\mathcal{R}))$ ; append  $b$  to  $\mathcal{R}$ ;  $\mathcal{B} \leftarrow \mathcal{B} \setminus \{b\}$ ;  $\phi \leftarrow \phi_0$ 
13:   else
14:     return FAILURE
15:   end if
16:    $[\text{LB}, \text{UB}] \leftarrow \text{NDP}(\mathcal{R})$ 
17: end while
18: return  $\mathcal{R} \cup \{0\}$ 
```

Routing pre-filter. Each candidate move admits a routing delta Δ_R computable in $O(1)$. Let $\Phi_{\min} = 2 \sum_{v \in I: s_v > \bar{b}_v} (s_v - \bar{b}_v)$ be a lower bound on the flow cost. A move is *discarded by the routing pre-filter* if $z_R + \Delta_R \geq \text{cost}(x^*) - \Phi_{\min} + \epsilon$, since no choice of transfers can then yield a total cost lower than the incumbent; here z_R is the current routing cost and ϵ is a numerical tolerance (set to 10^{-3} in our implementation).

2.5.4 $Q(\lambda)$ -guided Perturbation

The perturbation step selects, at every iteration, a destruction operator and a removal intensity through a $Q(\lambda)$ agent (Sutton and Barto, 1998). The agent maintains a value table $\mathbf{Q}(s, a)$ over a state space $\mathcal{S}$ of size 18 and an action space $\mathcal{A}$ of size 9.

State space. A state is defined by

$$s = 2(3g + d) + h. \quad (31)$$

It encodes three indicators:

$$g \in \{0, 1, 2\} \quad : \quad g = \begin{cases} 0, & \text{cost}(x^{\text{cur}}) \leq \text{cost}(x^{\text{best}}), \\ 1, & \text{cost}(x^{\text{cur}}) > \text{cost}(x^{\text{best}}) \text{ and } \text{cost}(x^{\text{cur}}) \leq (1 + \theta_s) \text{cost}(x^{\text{best}}), \\ 2, & \text{otherwise.} \end{cases}$$

$d \in \{0, 1, 2\}$: stagnation level, with thresholds $0.1 \text{ Max}_{\text{rep}}$ and $0.5 \text{ Max}_{\text{rep}}$ non-improving iterations.

$h \in \{0, 1\}$: 1 if the last iteration improved the current solution.

Action space. Each action $a = 3\rho + \delta$ encodes a destruction operator $\rho \in \{0, 1, 2\}$ and an intensity $\delta \in \{0, 1, 2\}$ corresponding to the removal of $\lfloor 0.03n \rfloor$, $\lfloor 0.06n \rfloor$ or $\lfloor 0.10n \rfloor$ nodes, where $n = |\mathcal{V}|$. The three operators are: *random* removal (uniform selection), *worst-cost* removal (largest detour cost), and *chain* removal (the seed is drawn at random and subsequent nodes are nearest neighbours of the previously removed node).

Reward and update. After each iteration the reward

$$r = \kappa \left(\omega_1 \frac{z^{\text{prev}} - z^{\text{cur}}}{z^{\text{prev}}} + \omega_2 \frac{z^*_{\text{prev}} - z^{\text{cur}}}{z^*_{\text{prev}}} + \omega_3 \mathbf{1}[\text{improvement}] \right) \quad (32)$$

is clipped to $[0, 1]$. The Q-table and the eligibility traces are updated by

$$e(s, a) += 1, \quad \delta_{\text{TD}} = r + \zeta \max_{a'} \mathbf{Q}(s', a') - \mathbf{Q}(s, a), \quad \mathbf{Q}(s'', a'') += \alpha_Q \delta_{\text{TD}} e(s'', a''), \quad (33)$$

followed by $e(s'', a'') \times= \zeta \lambda_Q$ for every (s'', a'') . Here s' is the state after action a , $\max_{a'} \mathbf{Q}(s', a')$ is the greedy one-step continuation value, and δ_{TD} is the temporal-difference error. The accumulating trace $e(s, a)$ assigns the current error not only to the latest pair but also to recently visited state–action pairs; multiplying all traces by $\zeta \lambda_Q$ makes that credit decay geometrically. Thus α_Q controls update magnitude, ζ controls the weight of future value, and λ_Q controls memory length. Equation (33) is applied once per AILS iteration after the perturbed solution has been accepted or rejected. The hyperparameters are calibrated in Section 2.6.2.

Perturbation procedure. Algorithm 5 formalises the perturbation. Buffer and balanced nodes are first stripped from the route. The selected operator removes $\text{act} = \lfloor (0.03 + 0.03\delta)n \rfloor$ forced-visit nodes, which are reinserted at the cheapest feasible positions, excluding their original positions to ensure genuine diversification. A *repair* step uses the NDP to detect infeasibility and inserts eligible buffer nodes ahead of any infeasible position until demand is restored. The repair fails (and the perturbed solution is discarded) only if the buffer set is exhausted without restoring feasibility.

2.6 Computational Results

In this section, we present the results of the computational tests performed to assess the performance of the mathematical model. The model was implemented in C++. The tests were performed on a laptop featuring an Intel Core i5 CPU processor @ 2.50 GHz and 12 GB of RAM, using CPLEX 20.1 as the MILP solver. Each execution was subject to a time limit of 3,600 CPU seconds. Section 2.6.1 provides details on the sets of randomly generated instances employed in our testing. The results obtained for the model are reported in Section 2.6.10.

2.6.1 Instance generation

We generated a controlled benchmark of asymmetric rebalancing instances in which instance size and vehicle parameters vary independently. The implementation used deterministic seeds derived from size

and replication index. The experimental data are publicly available in the dataset directory of the SBRPP GitHub repository. This release provides access to the instance data used in the computational study. The executable generator, AILS implementation, seed mapping, and per-run logs are not part of the data archive verified for this revision; exact end-to-end reproduction additionally requires those computational artefacts.

New instances were used because the standard static BRP benchmarks generally assume fixed target inventories or different visit, preemption, depot, and handling-cost rules. Applying AILS to them would therefore change the feasible set or objective and would not provide a controlled like-for-like comparison. The present generator varies capacity and visit structure while retaining the demand-interval and policy features studied here, and the same formulation is used by CPLEX and AILS. The public instance data support inspection and reuse of the test inputs, while direct comparison with previously published metaheuristics remains limited by the absence of a compatible public implementation baseline.

An instance is defined on $\mathcal{V}_0 = \{0\} \cup \mathcal{V}$, with depot node 0 and station set $\mathcal{V} = \{1, \dots, n\}$. Stations are placed uniformly at random in a bounded square region and the depot is set at their centroid, which lies inside the convex hull and avoids geographically degenerate layouts. Three size classes are considered,

$$\text{small: } n \in \{12, 16, 20\}, \quad \text{medium: } n \in \{40, 50, 60\}, \quad \text{large: } n \in \{100, 150, 200\}$$

with 20 independent replications per value of n , i.e. $9 \times 20 = 180$ synthetic instances in total. Travel times are made asymmetric yet metrically consistent: a base directed time $\tilde{\tau}_{ij} = \chi_{ij} d_{ij}$ (with d_{ij} the Euclidean distance and $\chi_{ij} \neq \chi_{ji}$ in general) is followed by a metric (shortest-path) closure, yielding a complete matrix $[\tau_{ij}]$ with $\tau_{ij} \neq \tau_{ji}$ that satisfies the directed triangle inequality $\tau_{ik} \leq \tau_{ij} + \tau_{jk}$. Each station i has a dock capacity C_i , an initial inventory $s_i \in [0, C_i]$, and a non-trivial target interval $[\underline{b}_i, \bar{b}_i] \subseteq [0, C_i]$ generated from C_i ; a station is *unbalanced* when $s_i \notin [\underline{b}_i, \bar{b}_i]$. Inventories are set so that rebalancing is always required, and the residual system imbalance is post-processed so that the depot can absorb it within capacity, which makes every instance feasible while remaining non-trivial. The handling time per bicycle is $t^L = t^U = 60$ s, and the vehicle capacity Q and the maximum number of visits m_{vis} are set per class.

The model supports three rebalancing policies, compared in Section 2.6.5: *non-split*, in which each station is visited exactly once; *non-preemptive* rebalancing with up to m_{vis} visits, in which a station's demand may be split across visits but bicycles may not be temporarily buffered; and *preemptive* rebalancing with up to m_{vis} visits, which additionally permits temporary buffering. By construction the non-split policy coincides with the single-visit case $m_{\text{vis}} = 1$.

2.6.2 Parameter tuning

The AILS contains several hyperparameters: the maximum number of non-improving iterations Max_{rep} ; the maximum removal fraction η ; the initial sector angle ϕ_0 ; the $Q(\lambda)$ learning rate α_Q , discount factor ζ , eligibility-trace coefficient λ_Q ; the reward scale κ ; the state-gap threshold θ_s ; and the reward weights $\omega_1, \omega_2, \omega_3$. We calibrated these parameters through a One-At-a-Time (OAT) analysis on a set of 27 tuning instances. Specifically, three instances were drawn uniformly at random from each of the nine instance-size groups described in Section 2.6.1, yielding 9 tuning instances per dataset class (small, medium, large) and 27 tuning instances in total. The metric used is the average percentage deviation from the best-known solution (BKD, %) over 10 independent runs per instance.

Starting from a default configuration informed by the reinforcement-learning literature, each parameter was varied within a candidate set while the others remained at their default values. Table 16 reports the average BKD on small, medium, and large instances; Table 17 summarises the final values together with their justification.

Table 16: Parameter tuning via One-At-a-Time (OAT) analysis on 27 calibration instances. The metric is the average percentage deviation from the best-known solution (BKD, %) over 10 runs per instance. The default value of each parameter is marked with “†”.

Parameter	Value	Small	Medium	Large	Avg BKD (%)	Avg time (s)
Learning rate α	0.10	0.00	0.51	1.82	0.78	2.87
	0.20 [†]	0.00	0.18	1.74	0.64	2.91
	0.30	0.01	0.44	2.02	0.82	2.98
Discount factor ζ	0.80	0.01	0.62	2.11	0.91	3.24
	0.90 [†]	0.00	0.18	1.74	0.64	2.91
	0.95	0.00	0.29	2.23	0.84	2.75
Eligibility trace λ	0.50	0.00	0.41	2.09	0.83	3.14
	0.70 [†]	0.00	0.18	1.74	0.64	2.91
	0.90	0.00	0.26	2.06	0.77	2.88
Reward scale κ	20	0.00	0.65	2.57	1.07	3.09
	50 [†]	0.00	0.18	1.74	0.64	2.91
	100	0.03	0.33	2.48	0.95	2.65
State gap θ_s	0.01 [†]	0.00	0.27	2.14	0.80	2.74
	0.02	0.00	0.18	1.74	0.64	2.91
	0.05	0.02	0.49	2.44	0.98	3.14
Max removal η	0.05	0.00	0.59	2.79	1.13	2.61
	0.10 [†]	0.00	0.18	1.74	0.64	2.91
	0.15	0.00	0.35	2.11	0.82	3.37
Max iterations Max_{rep}	500	0.01	0.59	2.19	0.93	2.67
	1000 [†]	0.00	0.18	1.74	0.64	2.91
	5000	0.00	0.18	1.73	0.64	3.26
Initial sector ϕ_0	10°	0.00	0.18	1.76	0.65	2.91
	15° [†]	0.00	0.18	1.74	0.64	2.91
	30°	0.01	0.20	1.76	0.66	2.90

Table 17: Final parameter settings of the AILS metaheuristic.

Component	Parameter	Symbol	Tested range	Selected	Justification
AILS	Max non-improving iterations	Max_{rep}	{500, 1000, 5000}	1000	Larger budgets yield negligible gains.
Perturbation	Max removal fraction	η	{0.05, 0.10, 0.15}	0.10	0.05 insufficient disruption; 0.15 destroys too much structure.
Constructive	Initial sector half-angle	ϕ_0	{10°, 15°, 30°}	15°	Doubles to 30°, 60°, 90° on failure.
Q(λ) agent	Learning rate	α_Q	{0.10, 0.20, 0.30}	0.20	0.10 too slow; 0.30 oscillates.
	Discount factor	ϕ	{0.80, 0.90, 0.95}	0.90	0.80 under-weights future; 0.95 negligible gain.
	Eligibility trace	λ_Q	{0.50, 0.70, 0.90}	0.70	Good credit assignment without instability.
	Reward scale	κ	{20, 50, 100}	50	Sufficient signal strength.
	State-gap threshold	θ_s	{0.01, 0.02, 0.05}	0.02	Best discrimination between near-optimal and suboptimal states.
	Reward weights	$\omega_1, \omega_2, \omega_3$	{0.30, ..., 0.60}	0.45, 0.45, 0.10	Symmetric local/global; small bonus for improvement.
	Initial exploration	ϵ_0	—	0.50	Decays to 0.05 over iterations.

The OAT analysis shows that the algorithm is largely robust on small instances (BKD close to zero for almost every configuration) and that the sensitivity grows with the instance size, especially on large instances. The selected configuration achieves an average BKD of 0.64% across the three size classes and is therefore adopted in all subsequent experiments.

The following controlled ablation complements parameter tuning by isolating the contribution of the principal AILS components under a common experimental design.

2.6.3 Impact of the Main AILS Components on Algorithmic Performance

To quantify the contribution of the main components of AILS, we conducted a controlled ablation study involving four alternative configurations. Each configuration was executed independently 10 times on every instance, using the same stopping criterion and parameter settings as the complete algorithm. The same best-known reference values were used to calculate BKD for all configurations. Table 18 reports the average BKD and computational time over the 10 independent runs. Lower BKD values indicate better solution quality.

In *AILS without $Q(\lambda)$* , the learning-based action-selection mechanism was replaced by a non-learning selection policy, while the same set of perturbation operators and removal intensities was retained. In *AILS with random perturbation*, the perturbation operator and its intensity were selected uniformly at random. In *AILS without NDP awareness*, the proposed neighbourhood structure was retained, but candidate moves were evaluated without incorporating the NDP-based flow cost into the move-selection criterion. Finally, in *AILS with standard local search*, the proposed local-optimisation procedure was replaced by conventional relocate, swap, 2-opt, and Or-opt neighbourhoods, while the remaining components of AILS were kept unchanged.

Table 18: Component ablation results averaged over 10 independent runs. Lower BKD values indicate better solution quality.

Configuration	Small BKD (%)	Medium BKD (%)	Large BKD (%)	Avg. BKD (%)	Avg. time (s)
Full AILS	0.00	0.18	1.74	0.64	2.91
AILS without $Q(\lambda)$	0.27	1.41	3.74	1.81	2.82
AILS with random perturbation	0.10	1.07	2.65	1.27	2.64
AILS without NDP awareness	2.34	2.68	5.84	3.62	2.36
AILS with standard local search	0.51	0.82	2.01	1.11	2.41

The results in Table 18 show that the complete AILS achieves the best solution quality in every instance class. It obtains BKD values of 0.00%, 0.18%, and 1.74% on the small, medium, and large instances, respectively, resulting in an overall average BKD of 0.64%. Every ablated configuration produces a higher BKD in all three size classes. The results therefore provide consistent numerical evidence that each examined component contributes positively to the performance of the complete algorithm.

The most pronounced deterioration in solution quality occurs when NDP-aware move evaluation is disabled. This configuration obtains an average BKD of 3.62%, with values of 2.34%, 2.68%, and 5.84% on the small, medium, and large instances, respectively. It is the weakest configuration in terms of average performance and remains the weakest variant in every size class. This result is consistent with the integrated structure of the problem. A routing move that reduces travel time does not necessarily reduce the complete objective because the resulting visit sequence can increase loading and unloading costs or restrict the set of feasible transfer decisions. Consequently, routing-only move evaluation can guide the search toward routes that appear attractive from a travel perspective but are inferior once their

associated flow decisions are included. The results confirm that incorporating the NDP into the evaluation of neighbourhood moves is essential for obtaining high-quality solutions.

The configuration without $Q(\lambda)$ produces an average BKD of 1.81%. Its BKD increases from 0.27% on the small instances to 1.41% on the medium instances and 3.74% on the large instances. This progressive deterioration indicates that state-dependent action selection becomes more important as the instance size and the number of possible route structures increase. Although the complete set of perturbation actions remains available, the non-learning policy cannot adapt its choice of operator and removal intensity to the current condition of the search.

Uniform random perturbation obtains an average BKD of 1.27%, with BKD values of 0.10%, 1.07%, and 2.65% on the small, medium, and large instances, respectively. This variant performs better than the non-learning policy without $Q(\lambda)$ in all three size classes. Therefore, when learning is unavailable, random action selection provides more effective diversification than the non-learning policy used in the previous variant. Randomisation allows the algorithm to explore different combinations of perturbation operators and removal intensities and avoids repeatedly applying an unsuitable fixed or deterministic action.

Nevertheless, random perturbation remains consistently inferior to the complete AILS. Its BKD is higher than that of the complete method for small, medium, and large instances. This comparison demonstrates that the advantage of $Q(\lambda)$ is not merely due to the availability of multiple perturbation actions or to stochastic diversification. Instead, learning from the observed effectiveness of previous actions and selecting actions according to the current search state leads to better solution quality than uninformed random selection.

The standard-local-search configuration achieves an average BKD of 1.11%, making it the best-performing ablated configuration. Its BKD values are 0.51%, 0.82%, and 2.01% for the small, medium, and large instances, respectively. These results show that conventional neighbourhood operators remain effective for this problem. However, the standard local search is consistently inferior to the complete AILS in every size class. The proposed local-optimisation procedure therefore provides an additional and systematic improvement over conventional relocate, swap, 2-opt, and Or-opt moves.

The relatively close performance of standard local search and the complete method on the large instances also provides useful information about the relative contribution of the algorithmic components. It suggests that conventional neighbourhoods can still generate competitive routes when they are combined with the remaining AILS components. Nevertheless, the specialised local-optimisation procedure improves the results consistently, while the larger deteriorations observed after removing NDP awareness or $Q(\lambda)$ indicate that these two components make stronger contributions to overall solution quality.

The running-time results reveal an expected trade-off between computational effort and solution quality. The complete AILS requires 2.91 seconds on average, whereas the ablated configurations require between 2.36 and 2.82 seconds. The configuration without $Q(\lambda)$ requires 2.82 seconds, indicating that the learning procedure introduces only limited computational overhead. Random perturbation reduces the average time to 2.64 seconds but produces solutions of lower quality than the complete method. Removing NDP awareness gives the shortest running time, 2.36 seconds, but also produces the highest average BKD. Standard local search requires 2.41 seconds and provides the best quality among the ablated variants, although it does not match the solution quality of the complete AILS.

Overall, the ablation study supports three main conclusions. First, NDP-aware move evaluation is the

most influential component in terms of average solution quality because it explicitly coordinates routing moves with the associated transfer decisions. Second, $Q(\lambda)$ -guided perturbation outperforms both the non-learning and random selection policies, confirming the value of state-dependent operator and intensity selection. Third, the proposed local-optimisation procedure provides a consistent improvement over standard local search. Although the complete AILS requires slightly more computational time, its average running time remains below three seconds and it achieves the best solution quality in every instance class. The additional computational effort is therefore justified by the improvement in the complete routing–flow objective.

2.6.4 Impact of the maximum number of visits

We isolate the effect of m_{vis} by re-solving the twenty $n = 12$ instances with the exact model for $m_{\text{vis}} \in \{1, 2, 3, 4\}$, all else fixed (Table 19).

Table 19: Per-instance effect of the maximum number of visits m_{vis} on the $n = 12$ random instances (preemption policy, exact model). NFS: no feasible solution.

Instance	$m_{\text{vis}}=1$		$m_{\text{vis}}=2$		$m_{\text{vis}}=3$		$m_{\text{vis}}=4$		Instance	$m_{\text{vis}}=1$		$m_{\text{vis}}=2$		$m_{\text{vis}}=3$		$m_{\text{vis}}=4$	
	Obj.	Time	Obj.	Time	Obj.	Time	Obj.	Time		Obj.	Time	Obj.	Time	Obj.	Time	Obj.	Time
Ins-12-1	138	0.26	138	0.72	138	1.82	138	2.27	Ins-12-2	201	0.08	197	1.24	197	4.47	197	7.97
Ins-12-3	171	0.53	171	7.31	171	15.73	171	20.12	Ins-12-4	150	0.23	150	4.37	150	11.87	150	14.63
Ins-12-5	215	0.11	204	1.64	204	3.80	204	5.44	Ins-12-6	207	0.08	199	1.09	199	4.45	199	7.24
Ins-12-7	156	0.37	156	1	156	3.70	156	5.77	Ins-12-8	193	0.14	182	0.83	182	2.69	182	2.33
Ins-12-9	180	0.15	171	0.75	171	3.16	171	3.96	Ins-12-10	207	0.16	184	0.59	174	1.14	174	1.66
Ins-12-11	195	0.15	191	0.80	191	2.91	191	3.61	Ins-12-12	210	0.12	201	1.11	201	3.23	201	5.33
Ins-12-13	173	0.12	167	0.97	167	2.25	167	3.68	Ins-12-14	183	0.19	178	1.13	178	1.98	178	3.04
Ins-12-15	NFS	–	197	7.64	195	23.78	195	46.20	Ins-12-16	162	0.81	162	2.06	162	2.06	162	10.00
Ins-12-17	214	0.12	209	1.73	209	2.88	209	4.85	Ins-12-18	189	0.57	189	7.07	189	20.84	189	33.82
Ins-12-19	183	0.17	177	1.25	177	2.96	177	5.28	Ins-12-20	168	0.08	167	0.99	167	2.70	167	4.24

The number of visits exposes a clear trade-off among quality, feasibility, and effort. Permitting more visits relaxes the capacity coupling between stations served on one route and lowers the objective, but with sharply diminishing returns: moving from $m_{\text{vis}} = 1$ to $m_{\text{vis}} = 2$ improves 13 of 20 instances, raising m_{vis} to 3 improves only 2 further instances, and $m_{\text{vis}} = 4$ improves none. Running time, by contrast, grows monotonically with m_{vis} (it roughly quadruples from $m_{\text{vis}} = 1$ to $m_{\text{vis}} = 4$), since each additional permitted visit enlarges the search space. This saturation mirrors a known structural property of split-demand one-commodity pickup-and-delivery routing: Hernandez-Perez et al. (2018) report that for the SD1PDTSP optimal solutions seldom require more than two visits per customer — a conclusion already drawn by Salazar-González and Santos-Hernández (2017) — and accordingly fix $m_{\text{vis}} = 2$. Our experiments confirm that the same behaviour holds in the rebalancing setting, where the bulk of the achievable improvement is already captured at $m_{\text{vis}} = 2$. We adopt $m_{\text{vis}} = 3$ as the default because it absorbs the small residual gains observed in a minority of instances at negligible additional cost, while $m_{\text{vis}} = 4$ provides no further benefit. A second, operational observation is that $m_{\text{vis}} = 1$ is not merely the weakest setting but a fragile one: a single visit per station leaves one instance with no feasible plan, whereas $m_{\text{vis}} \geq 2$ restores feasibility throughout. These findings are derived from the random $n = 12$ instances.

2.6.5 Comparison of rebalancing policies

The ranking follows the temporal flexibility each policy grants. The preemptive policy attains the lowest objective and is never worse than the non-preemptive one (strictly better on 8 of 20 instances); the non-preemptive policy is intermediate; and the non-split policy — which forbids splitting and so reduces to a single visit per station — is the most restrictive and yields the largest objective. The running-time order is reversed: non-split is by far the fastest (0.23 s on average against several seconds for the split variants), because forbidding multiple visits and buffering yields a much smaller feasible region and a tighter formulation. The practical message is a genuine trade-off rather than a dominance relation. First, the marginal value of preemption is real but modest — about 0.6% on average over the non-preemptive policy — so preemption is worth its extra modelling and solving cost mainly when small cost reductions are operationally significant. Second, and more decisively, the non-split policy is the only one that can fail to admit a feasible plan (one NFS instance), so the ability to split demand across visits is what guarantees feasibility on tightly constrained instances. All policy conclusions are drawn from the random $n = 12$ instances.

Table 20: Per-instance comparison of the three policies on the $n = 12$ random instances (exact model). NFS: no feasible solution.

Instance	Non-split		Non-preemption		Preemption		Instance	Non-split		Non-preemption		Preemption	
	Obj.	Time	Obj.	Time	Obj.	Time		Obj.	Time	Obj.	Time	Obj.	Time
Ins-12-1	138	0.26	138	2.19	138	1.82	Ins-12-2	201	0.08	201	1.38	197	4.47
Ins-12-3	171	0.53	171	14.59	171	15.73	Ins-12-4	150	0.23	150	13.02	150	11.87
Ins-12-5	215	0.11	206	3.63	204	3.80	Ins-12-6	207	0.08	200	2.91	199	4.45
Ins-12-7	156	0.37	156	4.34	156	3.70	Ins-12-8	193	0.14	182	1.67	182	2.69
Ins-12-9	180	0.15	173	1.91	171	3.16	Ins-12-10	207	0.16	174	1.16	174	1.14
Ins-12-11	195	0.15	191	1.90	191	2.91	Ins-12-12	210	0.12	201	3.96	201	3.23
Ins-12-13	173	0.12	169	3.39	167	2.25	Ins-12-14	183	0.19	178	1.64	178	1.98
Ins-12-15	NFS	–	197	7.64	195	23.78	Ins-12-16	162	0.81	162	5.49	162	7.20
Ins-12-17	214	0.12	209	3.24	209	2.88	Ins-12-18	189	0.57	189	18.53	189	20.84
Ins-12-19	183	0.17	183	3.11	177	2.96	Ins-12-20	168	0.08	168	2.38	167	2.70

2.6.6 Validation against the exact method

Small instances are the only class for which the exact model is consistently tractable, and we use them to validate AILS (Table 21; full data in Table 27).

Two facts stand out. First, the exact method scales poorly: all twenty $n = 12$ instances are solved in a few seconds, but at $n = 16$ the average solving time rises by roughly two orders of magnitude with one instance unsolved, and at $n = 20$ six of twenty instances reach the time limit. This reflects the combinatorial growth of the routing component, whose feasible visit–route configurations proliferate super-linearly in n once travel times are asymmetric. Second, AILS matches the CPLEX objective on all 60 small instances — thereby certifying optimality on the 53 that CPLEX proves optimal — in at most a third of a second, three to four orders of magnitude faster than the exact method. On this class the best, average, and worst AILS runs coincide (zero standard deviation in Table 27), showing that the optimum is reached in every run rather than only on average.

Table 21: Exact method vs. AILS on the small random instances (20 replications per n); CPLEX time is averaged over instances solved within the limit. AILS reproduces the CPLEX objective on every instance.

n	CPLEX (exact)		AILS	
	#opt/20	avg. time (s)	avg. obj.	avg. time (s)
12	20	6.3	178.9	0.09
16	19	672.25	238.9	0.14
20	14	1260.19	299.1	0.33

Table 22: AILS scalability on the random benchmark (20 replications per n). “best→avg” / “best→worst” are the average relative gaps of the mean and worst run to the best run.

Class	n	avg. best	mean std	best→avg (%)	best→worst (%)	avg. time (s)
Small	12	178.9	0.00	0.00	0.00	0.09
	16	238.9	0.00	0.00	0.00	0.14
	20	299.1	0.00	0.00	0.00	0.33
Medium	40	587.4	1.86	0.26	0.84	0.72
	50	736.2	2.43	0.30	0.90	1.04
	60	885.2	2.65	0.27	0.82	1.32
Large	100	1740.7	7.41	0.43	1.22	2.03
	150	2600.7	25.46	0.68	1.61	3.59
	200	3568.8	16.42	0.50	1.39	8.67

2.6.7 Scalability and robustness

Beyond $n = 20$ the exact model is no longer competitive, so the medium and large classes are solved with AILS alone (Table 22; per-instance data in Tables 28–29).

AILS scales smoothly: even $n = 200$ instances are solved in under nine seconds on average, with running time growing gradually in n rather than exploding as for the exact method. The dispersion columns quantify robustness, an aspect often left implicit: across all medium and large instances the mean run never exceeds the best by more than 0.7%, and even the worst of the ten runs stays within 1.6% of the best. The absolute standard deviation grows only mildly with size and remains small relative to the objective (e.g. about 16 on values near 3,570 at $n = 200$). For a stochastic method this near-deterministic stability is operationally valuable: a single run can be trusted to deliver a solution very close to the best attainable, which matters when rebalancing plans must be produced under a tight dispatch deadline. These scalability and robustness conclusions are drawn on the synthetic benchmark, where size is varied with all other characteristics held under control.

2.6.8 Validation on Established Benchmark Instances from the Literature

The purpose-built instances in Section 2.6.1 are needed to test the complete problem studied here, namely non-degenerate demand intervals, positive handling costs, repeated visits to both stations and the depot,

and all three rebalancing policies. Most established SBRP benchmarks omit at least some of these features and therefore cannot replace the main test bed without changing either the feasible set or the objective. Nevertheless, evaluation on an established benchmark is valuable for assessing external comparability. We therefore also tested AILS on the 100 real-world configurations reported in Tables 5 and 6 of Güneş Erdoğan et al. (2015). These configurations were originally derived from the benchmark set of Dell’Amico et al. (2014a); they cover 17 cities, contain between 13 and 59 stations, use vehicle capacities between 10 and 30 bicycles, and consider the two inventory scales $\alpha \in \{1, 3\}$.

Alignment of the problem definitions. The classical preemptive SBRP is a special case of the model studied in this thesis. For each station i , the benchmark initial inventory and station capacity were retained, while its exact target inventory q_i was represented by the singleton interval

$$[b_i, \bar{b}_i] = [q_i, q_i].$$

The original travel-cost matrix, vehicle capacity Q , and depot-balancing convention were preserved. The with-preemption policy was selected, and the handling-time coefficients were set to $t^L = t^U = 0$, so that the AILS objective coincided exactly with the travel-cost objective used by Güneş Erdoğan et al. (2015). The default visit limit $m_{\text{vis}} = 3$ was retained. Although the published model does not impose this explicit visit limit, AILS recovered the published best-known value in every tested configuration, showing that this limit did not prevent the recovery of any reported best-known solution in this experiment. AILS used the parameter setting reported in Section 2.6.2, without instance-specific retuning. For each configuration, its objective value and computation time were recorded on the experimental platform described at the beginning of Section 2.6.

Comparison measures. Let z^{AILS} , z^{BKS} , z^{LB} , and z^{H} denote, respectively, the AILS objective, the published best-known solution, the published lower bound, and the solution returned by the constructive heuristic of Güneş Erdoğan et al. (2015). The AILS gap to the best-known solution is computed as

$$100 \frac{z^{\text{AILS}} - z^{\text{BKS}}}{z^{\text{BKS}}},$$

whereas the improvement over the published heuristic is

$$100 \frac{z^{\text{H}} - z^{\text{AILS}}}{z^{\text{H}}}.$$

Thus, a positive improvement denotes a reduction in travel cost. Table 23 reports unweighted means over the configurations in each group.

AILS matched the published best-known solution in all 100 configurations and returned a lower objective value than the published constructive heuristic in every case. For $\alpha = 1$, all 50 best-known values equal their respective lower bounds. Consequently, the AILS solutions are certified optimal for all configurations in this group. For $\alpha = 3$, 49 of the 50 matches are similarly certified. The sole exception is the RioDeJaneiro configuration with $|V| = 55$ and $Q = 20$: AILS obtained the published incumbent

$$z^{\text{AILS}} = z^{\text{BKS}} = 366,036,$$

Table 23: AILS results on the real-world preemptive SBRP configurations of Güneş Erdoğan et al. (2015).

Setting	Inst.	V	AILS = BKS	Certified	Mean AILS-BKS gap (%)	Published heuristic mean gap to LB (%)	Mean improvement over heuristic (%)	AILS time (s)	
								Mean	Max.
$\alpha = 1$ (Table 5)	50	13–59	50/50	50/50	0.00	45.10	30.30	0.090	0.48
$\alpha = 3$ (Table 6)	50	13–59	50/50	49/50	0.00	34.99	25.25	0.159	0.75
All configurations	100	13–59	100/100	99/100	0.00	40.05	27.77	0.125	0.75

Notes. BKS: best-known solution; LB: best-known lower bound. “Certified” counts configurations for which the AILS objective also equals the published lower bound. The published heuristic gap is the percentage deviation from the LB reported in Tables 5 and 6 of Güneş Erdoğan et al. (2015). The AILS computation times are descriptive results from the present implementation and are not used to claim a cross-machine speedup.

whereas the published lower bound is 356,429.5, leaving a gap of 2.70%. This result is therefore an incumbent match and not an optimality certificate.

The travel-cost reduction relative to the published constructive heuristic averages 30.30% for $\alpha = 1$ and 25.25% for $\alpha = 3$, with an overall average of 27.77%. The published heuristic itself has mean gaps of 45.10% and 34.99% from the lower bound in the two groups. The mean AILS computation times are 0.090 and 0.159 seconds, respectively, and the largest recorded time is 0.75 seconds.

These experiments complement, rather than replace, the purpose-built benchmark. They demonstrate that AILS reproduces the published reference solution quality when the proposed model is restricted to the classical preemptive SBRP. The custom instances remain necessary for evaluating the additional demand-interval, handling-cost, depot, and policy features of the full problem. Two limitations should nevertheless be noted. First, the baseline reported in Tables 5 and 6 of Güneş Erdoğan et al. (2015) is a constructive heuristic rather than a metaheuristic. Second, the published and present computation times were obtained using different hardware, software, and implementation environments. We therefore base the substantive comparison on common objective values and published lower bounds and do not claim either a hardware-normalised speedup or general superiority over all SBRP metaheuristics.

2.6.9 Real-Data Instances

To assess the performance of the AILS algorithm on realistic scenarios, we constructed 120 new instances derived from the BICIMIA data sheet, spanning the period from January 1, 2019, to December 31, 2019. The data sheet comprises 85 stations distributed across Brescia, all managed by the BICIMIA company. These 120 instances are grouped into groups with homogeneous values of $|I|$, using a vehicle with a fixed capacity of 14. Each subset includes 20 randomly generated instances, categorised as follows.

- Small-sized instances: 20 instances for each of $|I| = 12$, $|I| = 16$, and $|I| = 20$.
- Medium-sized instances: 20 instances for each of $|I| = 40$, $|I| = 50$.
- Full-network instances: 20 instances for $|I| = 85$.

For each instance, a random day from 2019 was selected, and stations were chosen such that at least 60% of the selected nodes were either sink or source nodes. The actual distances between the chosen stations are routing-based (shortest road-network distance, computed with ArcGIS) and are converted to travel times via a typical urban speed of 30 km/h. Each instance is labeled “Ins- n - x ,” where n denotes the number of stations, and x refers to the instance number.

2.6.10 Results on Small Real-world Data Instances

To assess the performance of the proposed AILS algorithm, we began by evaluating it on a collection of small-scale instances containing 12, 16, and 20 stations. Each group consists of 20 problem instances derived from real-world data, leading to a total of 60 instances. As a benchmark, we computed the exact solutions of the associated mathematical formulation using IBM ILOG CPLEX.

Table 24 compares the final objective values reported by CPLEX and AILS. A CPLEX value is an exact optimum only when the solver certified optimality before termination; when the 3,600-second limit was reached it is a time-limited incumbent. The tabulated “Gap” values are retained from the original computational report, but they must not be read as certified optimality gaps on time-limited rows.

For the 12- and 16-station groups, the reported mean differences are 0.03% and 0.04%, with mean AILS runtimes of 0.12 and 0.22 seconds. For 20 stations, the reported mean difference is 0.10% and mean AILS time is 0.30 seconds, but several CPLEX runs reach the time limit. These observations show close agreement with exact or time-limited CPLEX outputs on this test set and a large runtime difference under the stated hardware and settings; they do not establish a general optimality guarantee for AILS.

Table 24: Results on small real data instance

Instance	CPLEX		AILS			Instance	CPLEX		AILS			Instance	CPLEX		AILS		
	Obj.	Time (s)	Obj.	Gap(%)	Time(s)		Obj.	Time (s)	Obj.	Gap(%)	Time(s)		Obj.	Time (s)	Obj.	Gap(%)	Time(s)
Ins-12-1	97	18.22	97	0.00	0.13	Ins-16-1	186	133.24	186	0.00	0.18	Ins-20-1	117	993.44	117	0.00	0.29
Ins-12-2	131	28.34	131	0.00	0.10	Ins-16-2	145	1712.58	145	0.00	0.19	Ins-20-2	152	178.51	152	0.97	0.31
Ins-12-3	113	18.14	113	0.00	0.10	Ins-16-3	106	413.09	106	0.00	0.17	Ins-20-3	207	1786.30	209	0.00	0.32
Ins-12-4	129	22.25	129	0.00	0.12	Ins-16-4	114	110.37	114	0.00	0.24	Ins-20-4	113	70.82	113	0.00	0.28
Ins-12-5	144	511.08	145	0.69	0.12	Ins-16-5	137	447.64	137	0.00	0.26	Ins-20-5	127	115.71	127	0.49	0.30
Ins-12-6	168	460.3	168	0.00	0.12	Ins-16-6	146	1486.21	146	0.00	0.14	Ins-20-6	204	3600.00	205	0.00	0.33
Ins-12-7	99	15.71	99	0.00	0.09	Ins-16-7	150	2465.84	150	0.00	0.20	Ins-20-7	212	689.71	212	0.00	0.33
Ins-12-8	108	23.14	108	0.00	0.13	Ins-16-8	118	282.99	118	0.00	0.23	Ins-20-8	155	3600.00	155	0.52	0.26
Ins-12-9	147	18.76	147	0.00	0.11	Ins-16-9	137	308.84	137	0.00	0.19	Ins-20-9	193	245.37	194	0.00	0.30
Ins-12-10	109	21.34	109	0.00	0.11	Ins-16-10	103	90.51	103	0.00	0.23	Ins-20-10	209	3600.00	209	0.00	0.25
Ins-12-11	113	18.73	113	0.00	0.09	Ins-16-11	156	998.97	156	0.00	0.25	Ins-20-11	174	476.45	174	0.00	0.32
Ins-12-12	116	28.35	116	0.00	0.10	Ins-16-12	122	401.52	122	0.00	0.24	Ins-20-12	137	434.46	137	0.00	0.28
Ins-12-13	114	3.03	114	0.00	0.12	Ins-16-13	143	245.52	143	0.00	0.25	Ins-20-13	157	197.31	157	0.00	0.35
Ins-12-14	205	608.1	205	0.00	0.14	Ins-16-14	116	164.84	117	0.86	0.24	Ins-20-14	165	347.44	165	0.00	0.30
Ins-12-15	117	13.59	117	0.00	0.15	Ins-16-15	173	1079.80	174	0.58	0.28	Ins-20-15	167	195.81	167	0.00	0.28
Ins-12-16	103	10.57	103	0.00	0.14	Ins-16-16	121	107.60	121	0.00	0.27	Ins-20-16	173	228.35	173	0.00	0.24
Ins-12-17	91	25.95	91	0.00	0.11	Ins-16-17	102	91.31	102	0.00	0.19	Ins-20-17	184	219.46	184	0.00	0.34
Ins-12-18	128	46.52	128	0.00	0.10	Ins-16-18	138	241.92	138	0.00	0.26	Ins-20-18	172	158.98	172	0.00	0.23
Ins-12-19	127	15.41	127	0.00	0.12	Ins-16-19	136	1313.95	136	0.00	0.23	Ins-20-19	152	115.31	152	0.00	0.35
Ins-12-20	134	18.14	134	0.00	0.12	Ins-16-20	142	2399.79	142	0.00	0.24	Ins-20-20	160	126.01	160	0.10	0.29
Average		96.28		0.03	0.12			724.83		0.04	0.22			869.63		0.10	0.30

2.6.11 Results on Medium and Large Real-world data Instances

For the 40-station group, CPLEX returned a feasible incumbent in eight of 20 runs and reached the 3,600-second limit in each of those eight without certifying optimality; it returned no feasible incumbent in the other 12 runs. AILS returned a feasible solution for every tested 40- and 50-station instance, with reported mean runtimes of 1.15 and 1.60 seconds. On the eight 40-station rows with a CPLEX incumbent, the AILS objective is 6.9% lower on average. This is a relative difference from time-limited incumbents, not a negative optimality gap and not evidence that the AILS solutions are optimal. The comparison is specific to CPLEX 20.1, the stated formulation, time limit, hardware, and parameter settings.

Table 26 reports AILS objective values and CPU times for 20 instances with 85 stations. The mean runtime is 3.31 seconds and every run produced a feasible solution. No exact bound, certified optimum,

compatible published heuristic baseline, or component ablation is available for this group. The table therefore supports a runtime-and-feasibility statement for the tested implementation only; it does not establish solution quality relative to an optimum or a “best available” method.

Table 25: Results on medium-sized real data instances

Instance	CPLEX		AILS			Instance	AILS	
	Obj.	Time (s)	Obj.	gap (%)	Time (s)		Obj.	Time (s)
Ins-40-1	—	—	287	—	1.03	Ins-50-1	250	1.27
Ins-40-2	—	—	273	—	1.37	Ins-50-2	385	1.32
Ins-40-3	—	—	247	—	1.60	Ins-50-3	418	1.37
Ins-40-4	—	—	280	—	1.20	Ins-50-4	417	1.52
Ins-40-5	—	—	350	—	1.35	Ins-50-5	370	1.99
Ins-40-6	346	3600.00	327	-5.49	0.90	Ins-50-6	268	1.89
Ins-40-7	—	—	298	—	1.12	Ins-50-7	394	1.33
Ins-40-8	283	3600.00	280	-1.41	1.11	Ins-50-8	315	1.32
Ins-40-9	—	—	302	—	1.27	Ins-50-9	227	2.04
Ins-40-10	295	3600.00	261	-11.53	1.11	Ins-50-10	236	1.29
Ins-40-11	—	—	235	—	1.48	Ins-50-11	261	1.96
Ins-40-12	211	3600.00	204	-4.27	1.11	Ins-50-12	282	1.63
Ins-40-13	233	3600.00	199	-14.59	0.95	Ins-50-13	244	1.95
Ins-40-14	—	—	295	—	1.32	Ins-50-14	257	1.50
Ins-40-15	232	3600.00	210	-9.48	0.81	Ins-50-15	296	1.49
Ins-40-16	—	—	302	—	1.01	Ins-50-16	257	1.38
Ins-40-17	—	—	247	—	0.96	Ins-50-17	288	1.88
Ins-40-18	260	3600.00	256	-1.54	1.36	Ins-50-18	393	1.58
Ins-40-19	—	—	208	—	1.00	Ins-50-19	273	1.27
Ins-40-20	—	—	306	—	0.85	Ins-50-20	305	2.01
Average		3600.00		-6.9	1.15			1.60

Table 26: Results on large-sized real data instances

Instance	AILS		Instance	AILS	
	Obj.	Time (s)		Obj.	Time (s)
Ins-85-1	307	3.47	Ins-85-11	322	3.76
Ins-85-2	309	2.63	Ins-85-12	399	4.05
Ins-85-3	419	2.62	Ins-85-13	369	2.85
Ins-85-4	324	2.78	Ins-85-14	341	2.88
Ins-85-5	350	4.38	Ins-85-15	430	3.73
Ins-85-6	346	2.65	Ins-85-16	345	2.86
Ins-85-7	298	3.06	Ins-85-17	339	3.92
Ins-85-8	289	3.13	Ins-85-18	247	3.70
Ins-85-9	424	3.35	Ins-85-19	424	2.89
Ins-85-10	363	3.54	Ins-85-20	268	3.89
Average					3.31

2.7 Conclusions and Future Research

This study addressed the single-vehicle Static Bike Rebalancing Problem with demand intervals, repeated visits to stations and the depot, a free initial load, and non-split, without-preemption, and with-preemption policies. It contributed a common flow-based MILP, a formal equivalence theorem for

the unbounded-capacity case, and an AILS in which NDP optimises the transfer plan for every candidate route and $Q(\lambda)$ adapts the perturbation strategy.

The computational evidence has several layers. On the 60 small random instances, AILS matches every CPLEX objective; 53 matches are certified optima and seven are incumbent comparisons. The ablation study, averaged over ten independent runs, gives the lowest mean BKD to the complete AILS (0.64%); removing NDP awareness produces the largest deterioration (3.62%), followed by removing $Q(\lambda)$ (1.81%), random perturbation (1.27%), and standard local search (1.11%). On 100 established literature configurations, AILS matches every published best-known solution, and 99 matches equal published lower bounds. On medium and large purpose-built instances, the evidence concerns feasible objective values, runtime, and run-to-run dispersion whenever no exact certificate is available. The policy tests on the stated $n = 12$ set are consistent with the theorem: preemption has value only in capacity-constrained cases, but the observed magnitudes are instance-specific rather than general guarantees.

Future work should extend AILS to multiple vehicles and heterogeneous fleets, integrate it into rolling-horizon dynamic control, and model uncertainty in the demand intervals. A dedicated polyhedral study and branch-and-cut method could strengthen exact benchmarking. The equivalence theorem may also be sharpened into quantitative bounds on the value of preemption as a function of demand and vehicle capacity, in the spirit of Nowak et al. (2008).

Algorithm 3 Nested Dynamic Programming (NDP)

Require: Route $\mathcal{R} = \langle \mathcal{R}_1, \dots, \mathcal{R}_n \rangle$; station data $(s_i, u_i, b_i, \bar{b}_i)_{i \in I}$; vehicle capacity Q ; times t^L, t^U

Ensure: Minimum flow cost z_{flow} ; or $+\infty$ if infeasible

```

1: Identify repeated stations  $\mathcal{P} = \{v \in I : \text{count}(v, \mathcal{R}) > 1\}$ ; set  $r \leftarrow |\mathcal{P}|$ 
2: Record first/last position and total count for every node
3: Initialise pool: one state ( $j=0, \mathbf{K}=\mathbf{0}, c=0$ )
4: for  $i = 1$  to  $n$  do
5:    $v \leftarrow \mathcal{R}_i$ ; seenAfter  $\leftarrow$  prior occurrences of  $v$ 
6:   newPool  $\leftarrow \emptyset$ 
7:   for all states  $(j, \mathbf{K}, c)$  in current pool do
8:      $K^{\text{old}} \leftarrow \mathbf{K}[v]$  (0 if  $v \notin \mathcal{P}$ )
9:      $g_i \leftarrow s_v - K^{\text{old}}$  ▷ Running inventory at  $v$  before this visit
10:    Compute effective bounds  $[lb_i^{\text{eff}}, ub_i^{\text{eff}}]$  via Eqs. (34)–(36)
11:    for all  $k_i \in [lb_i^{\text{eff}}, ub_i^{\text{eff}}]$  do
12:       $j' \leftarrow j + k_i$ 
13:      if  $j' < 0$  or  $j' > Q$  then
14:        continue
15:      end if
16:       $K'^{\text{new}} \leftarrow K^{\text{old}} + k_i$ ;  $g' \leftarrow s_v - K'^{\text{new}}$ 
17:      if  $g' < 0$  or  $g' > u_v$  then
18:        continue ▷ Station physical capacity
19:      end if
20:      if  $i = \text{lastPos}(v)$  and  $g' \notin [b_v, \bar{b}_v]$  then
21:        continue ▷ Demand not satisfied at final visit
22:      end if
23:       $c' \leftarrow c + t^L \max(k_i, 0) + t^U \max(-k_i, 0)$  ▷ Loading and unloading cost
24:      Update  $\mathbf{K}' \leftarrow \mathbf{K}$  with  $\mathbf{K}'[v] \leftarrow K'^{\text{new}}$ 
25:      newPool.RELAX( $j', \mathbf{K}', c'$ ) ▷ Insert or update: keep min  $c'$  per  $(j', \mathbf{K}')$ 
26:    end for
27:  end for
28:  Current pool  $\leftarrow$  newPool
29:  if pool is empty then
30:    return  $+\infty$  ▷ No feasible continuation
31:  end if
32: end for
33: return  $\min_{(j, \mathbf{K}, c) \in \text{pool}} c$ 

```

Algorithm 4 NDP-Aware Full Optimisation (FULLOPTNDP)

Require: Current route $\mathcal{R}$; NDP oracle

Ensure:

Ensure: Route $\mathcal{R}'$ with $\text{cost}(\mathcal{R}') \leq \text{cost}(\mathcal{R})$

```
1: repeat
2:   improved  $\leftarrow$  FALSE
3:                                      $\triangleright$  N1: Single-node extraction and reinsertion
4:   for each non-depot node  $v$  at position  $p$  in  $\mathcal{R}$  do
5:     if  $v$  is a forced-visit node with exactly  $\text{NR}[v]$  remaining copies then
6:       skip  $\triangleright$  Cannot remove mandatory copy
7:     end if
8:      $\mathcal{R}' \leftarrow \mathcal{R} \setminus \{v\}$ ; evaluate NDP on  $\mathcal{R}'$ 
9:     if  $\text{cost}(\mathcal{R}') < \text{cost}(\mathcal{R})$  then
10:       $\mathcal{R} \leftarrow \mathcal{R}'$ ; improved  $\leftarrow$  TRUE; break
11:    end if
12:  end for
13:                                      $\triangleright$  N2: Single-node relocation (extraction + best insertion)
14:  for each node  $v$  at position  $p$ , each insertion position  $q \neq p, p-1$  do
15:    Apply routing pre-filter; if fails, skip
16:    Evaluate NDP on  $\mathcal{R}$  with  $v$  moved from  $p$  to  $q$ 
17:    if cost improves then
18:      Accept move; improved  $\leftarrow$  TRUE; break
19:    end if
20:  end for
21:  for all pairs  $(a, b)$  with  $a < b$ ,  $\mathcal{R}[a] \neq 0$ ,  $\mathcal{R}[b] \neq 0$  do  $\triangleright$  N3: Node swap
22:     $\Delta_R \leftarrow \text{O}(1)$  routing delta of swapping  $\mathcal{R}[a]$  and  $\mathcal{R}[b]$ 
23:    if  $z_R + \Delta_R \geq \text{cost}(x^{\text{best}}) - \Phi_{\min} + \epsilon$  then
24:      continue  $\triangleright$  Routing pre-filter: cannot beat best even with minimum flow
25:    end if
26:     $z' \leftarrow \text{NDP}(\mathcal{R} \text{ with } \mathcal{R}[a] \leftrightarrow \mathcal{R}[b])$ 
27:    if  $z' < z(\mathcal{R}) - \epsilon$  then
28:      Apply swap;  $z(\mathcal{R}) \leftarrow z'$ ; improved  $\leftarrow$  TRUE;  $z_R \leftarrow \text{RoutingCost}(\mathcal{R})$ 
29:    end if
30:  end for
31:  for all (from, to) pairs for Or-opt-1 relocation do  $\triangleright$  N4: Or-opt-1 relocation
32:    Apply routing pre-filter; if passes, evaluate with NDP
33:    if total cost strictly improves then
34:      Apply relocation; improved  $\leftarrow$  TRUE
35:    end if
36:  end for
37:  for all (from, to) pairs for Or-opt-2 relocation do  $\triangleright$  N5: segment reversal / 2-opt
38:    Apply routing pre-filter; if passes, evaluate with NDP
39:    if total cost strictly improves then
40:      Apply relocation; improved  $\leftarrow$  TRUE
41:    end if
42:  end for
43: until  $\neg$  improved
44: return  $\mathcal{R}$ 
```

Algorithm 5 Perturbation

Require: Current route $\mathcal{R}$; operator ρ ; removal count act; buffer set $\mathcal{C}$

Ensure: Perturbed (and repaired) route $\mathcal{R}$

- 1: Remove all buffer and balanced nodes from $\mathcal{R}$
 - 2: Select act nodes from the stripped route using operator ρ
 - 3: **for all** removed nodes v in random order **do**
 - 4: Reinsert v at the cheapest feasible position, excluding its original predecessor/successor
 - 5: **end for**
 - 6: $(\text{status}, i_{\text{inf}}) \leftarrow \text{NDP}(\mathcal{R})$
 - 7: **while** $\text{status} = \text{INFEASIBLE}$ and $\mathcal{C} \neq \emptyset$ **do**
 - 8: $c \leftarrow \text{ROULETTEWHEEL}(\mathcal{C}, i_{\text{inf}})$; insert c immediately before i_{inf}
 - 9: $\mathcal{C} \leftarrow \mathcal{C} \setminus \{c\}$; $(\text{status}, i_{\text{inf}}) \leftarrow \text{NDP}(\mathcal{R})$
 - 10: **end while**
 - 11: **return** $\mathcal{R}$
-

.1 Nested Dynamic Programming: Derivation

This appendix details the derivation of the NDP procedure used by the AILS to compute, given a route $\mathcal{R} = \langle \mathcal{R}_1, \dots, \mathcal{R}_n \rangle$, the optimal loading and unloading plan that minimises the total flow cost while satisfying the demand intervals and vehicle capacity.

Transfer bounds at the i -th visit. Let $K_{\mathcal{R}_i}^{\text{old}}$ denote the cumulative net transfer at station $\mathcal{R}_i$ before the i -th visit ($K_{\mathcal{R}_i}^{\text{old}} = 0$ at the first visit) and let $g_i = s_{\mathcal{R}_i} - K_{\mathcal{R}_i}^{\text{old}}$ denote the running inventory of station $\mathcal{R}_i$ at the moment of the visit (here a positive transfer denotes a pickup). Given the on-board load j just before the visit, the *physical* bounds on the transfer k_i at visit i are

$$lb_i = \max(g_i - u_{\mathcal{R}_i}, -j), \quad ub_i = \min(g_i, Q - j). \quad (34)$$

The first equality ensures that the resulting station inventory does not exceed the station capacity $u_{\mathcal{R}_i}$ and that the on-board load remains non-negative; the second ensures that the running inventory remains non-negative and that the on-board load does not exceed the vehicle capacity.

Tightening at the last visit. At the *last* visit $\ell(\mathcal{R}_i)$ of station $\mathcal{R}_i$, the final inventory must lie within the demand interval $[b_{\mathcal{R}_i}, \bar{b}_{\mathcal{R}_i}]$. This translates into the additional bounds

$$k_i \in [g_i - \bar{b}_{\mathcal{R}_i}, g_i - b_{\mathcal{R}_i}]. \quad (35)$$

Lookahead tightening. For non-last visits, an additional tightening exploits the fact that, in at most rem remaining visits, the cumulative net transfer at station $\mathcal{R}_i$ must eventually reach the range $[s_{\mathcal{R}_i} - \bar{b}_{\mathcal{R}_i}, s_{\mathcal{R}_i} - b_{\mathcal{R}_i}]$. Since at each visit the transfer is bounded by the vehicle capacity, this gives

$$k_i \in [s_{\mathcal{R}_i} - \bar{b}_{\mathcal{R}_i} - \text{rem} \cdot Q - K_{\mathcal{R}_i}^{\text{old}}, s_{\mathcal{R}_i} - b_{\mathcal{R}_i} + \text{rem} \cdot Q - K_{\mathcal{R}_i}^{\text{old}}]. \quad (36)$$

The effective bound on k_i at visit i is the intersection of (34)–(36). In Equation (34), $g_i - u_{\mathcal{R}_i}$ prevents a delivery from exceeding station capacity and $-j$ prevents the vehicle load from becoming negative; g_i prevents a pickup from making station inventory negative and $Q - j$ prevents the vehicle load from exceeding capacity. Equation (35) then forces the last visit to terminate inside the demand interval. Equation (36) removes transfers from which the remaining visits could not possibly reach that interval, even if each later transfer used the full vehicle capacity. An empty intersection immediately proves that the current state has no feasible continuation. Policy-specific sign restrictions are intersected with the same bounds, so feasibility and policy compliance are checked before a successor state is created.

State pool and recursion. The NDP maintains a state pool. At every position i , each state is a triple $(j, \mathbf{K}, c)$, where $j \in \{0, \dots, Q\}$ is the on-board load, $\mathbf{K} \in \mathbb{Z}^r$ records cumulative net transfer at each repeated station, and $c \in \mathbb{R}_{\geq 0}$ is accumulated handling cost. The initial state is $(0, \mathbf{0}, 0)$ at the starting depot. For an admissible signed transfer k_i , the successor has $j' = j + k_i$, the corresponding component

of $\mathbf{K}$ is increased by k_i , and

$$c' = c + t^L \max\{k_i, 0\} + t^U \max\{-k_i, 0\}.$$

This expression is valid whether or not $t^L = t^U$. States with the same $(j', \mathbf{K}')$ have identical feasible continuations, so retaining only the one with minimum c' is a valid dominance rule. The optimal flow cost is the minimum terminal-state cost.

.2 Detailed per-instance results

Table 27: Per-instance results on the small random instances. CPLEX columns give the optimal objective and time; TL marks the time limit. AILS columns are over the independent runs.

Inst.	CPLEX							Inst.	CPLEX							Time
	Obj.	Time	Best	Avg.	Worst	Std	Time		Obj.	Time	Best	Avg.	Worst	Std	Time	
Ins-12-1	138	1.82	138	138	138	0	0.05	Ins-12-2	197	4.47	197	197	197	0	0.10	
Ins-12-3	171	15.73	171	171	171	0	0.07	Ins-12-4	150	11.87	150	150	150	0	0.05	
Ins-12-5	204	3.80	204	204	204	0	0.12	Ins-12-6	199	4.45	199	199	199	0	0.06	
Ins-12-7	156	3.70	156	156	156	0	0.08	Ins-12-8	182	2.69	182	182	182	0	0.11	
Ins-12-9	171	3.16	171	171	171	0	0.20	Ins-12-10	174	1.14	174	174	174	0	0.03	
Ins-12-11	191	2.91	191	191	191	0	0.14	Ins-12-12	201	3.23	201	201	201	0	0.08	
Ins-12-13	167	2.25	167	167	167	0	0.14	Ins-12-14	178	1.98	178	178	178	0	0.09	
Ins-12-15	195	26.31	195	195	195	0	0.03	Ins-12-16	162	7.20	162	162	162	0	0.03	
Ins-12-17	209	2.88	209	209	209	0	0.11	Ins-12-18	189	20.84	189	189	189	0	0.03	
Ins-12-19	177	2.96	177	177	177	0	0.09	Ins-12-20	167	2.70	167	167	167	0	0.16	
Ins-16-1	232	4.23	232	232	232	0	0.29	Ins-16-2	230	TL	230	230	230	0	0.08	
Ins-16-3	207	150.97	207	207	207	0	0.06	Ins-16-4	243	6.41	243	243	243	0	0.21	
Ins-16-5	257	32.59	257	257	257	0	0.12	Ins-16-6	227	3197.09	227	227	227	0	0.07	
Ins-16-7	247	1276.22	247	247	247	0	0.13	Ins-16-8	212	1693.72	212	212	212	0	0.04	
Ins-16-9	252	24.88	252	252	252	0	0.12	Ins-16-10	223	1015.46	223	223	223	0	0.08	
Ins-16-11	261	11.59	261	261	261	0	0.23	Ins-16-12	260	5.38	260	260	260	0	0.15	
Ins-16-13	248	445.66	248	248	248	0	0.09	Ins-16-14	250	55.64	250	250	250	0	0.18	
Ins-16-15	244	13.06	244	244	244	0	0.31	Ins-16-16	241	33.85	241	241	241	0	0.05	
Ins-16-17	237	7.56	237	237	237	0	0.18	Ins-16-18	234	35.88	234	234	234	0	0.20	
Ins-16-19	258	1212.05	258	258	258	0	0.06	Ins-16-20	215	622.85	215	215	215	0	0.09	
Ins-20-1	328	TL	328	328	328	0	0.16	Ins-20-2	283	699.05	283	283	283	0	0.19	
Ins-20-3	287	TL	287	287	287	0	0.15	Ins-20-4	332	98.75	332	332	332	0	0.32	
Ins-20-5	303	26.78	303	303	303	0	0.36	Ins-20-6	259	145.90	259	259	259	0	0.12	
Ins-20-7	303	30.20	303	303	303	0	0.53	Ins-20-8	317	125.01	317	317	317	0	0.57	
Ins-20-9	297	33.52	297	297	297	0	0.28	Ins-20-10	285	TL	285	285	285	0	0.14	
Ins-20-11	272	TL	272	272	272	0	0.15	Ins-20-12	301	59.84	301	301	301	0	0.48	
Ins-20-13	292	TL	292	292	292	0	0.20	Ins-20-14	281	TL	281	281	281	0	0.34	
Ins-20-15	276	1944.36	276	276	276	0	0.25	Ins-20-16	335	101.91	335	335	335	0	0.64	
Ins-20-17	309	105.05	309	309	309	0	0.44	Ins-20-18	295	111.92	295	295	295	0	0.45	
Ins-20-19	309	12.11	309	309	309	0	0.38	Ins-20-20	318	109.47	318	318	318	0	0.38	

Table 28: Per-instance AILS results on the medium random instances.

Inst.	AILS					Inst.	AILS				
	Best	Avg.	Worst	Std	Time		Best	Avg.	Worst	Std	Time
Ins-40-1	579	580	584	1.89	0.35	Ins-40-2	580	581.20	588	2.53	0.88
Ins-40-3	586	587.60	590	1.35	0.35	Ins-40-4	560	560.40	563	0.97	0.93
Ins-40-5	617	619.70	627	3.89	0.69	Ins-40-6	575	575	575	0	0.26
Ins-40-7	594	596	601	3.23	1.05	Ins-40-8	578	579.10	581	0.99	0.21
Ins-40-9	612	612.90	617	1.91	0.98	Ins-40-10	604	604.20	605	0.42	0.44
Ins-40-11	576	579.20	583	2.86	1.36	Ins-40-12	617	617.60	619	0.97	0.18
Ins-40-13	562	564.20	568	2.49	0.77	Ins-40-14	606	607.90	611	2.13	1.19
Ins-40-15	566	566.20	568	0.63	0.48	Ins-40-16	586	587.90	591	1.66	0.71
Ins-40-17	591	595.10	603	4.01	1.28	Ins-40-18	593	593	593	0	0.38
Ins-40-19	603	606.10	609	2.13	0.88	Ins-40-20	562	564.30	570	3.13	1.04
Ins-50-1	720	722.70	731	4.08	1.52	Ins-50-2	735	737.40	743	3.20	0.91
Ins-50-3	714	714.80	718	1.69	0.54	Ins-50-4	740	742.40	745	1.71	0.99
Ins-50-5	709	714.60	718	2.50	1.87	Ins-50-6	737	738.50	740	1.58	0.72
Ins-50-7	769	769.40	773	1.26	0.44	Ins-50-8	716	719.50	724	3.17	1.08
Ins-50-9	744	747.80	756	4.13	1.94	Ins-50-10	765	765.40	768	0.97	0.39
Ins-50-11	718	719	724	1.94	0.99	Ins-50-12	752	755	761	3.53	2.03
Ins-50-13	740	741.40	742	0.97	0.25	Ins-50-14	772	774.90	778	2.56	0.89
Ins-50-15	723	726.50	732	3.72	1.86	Ins-50-16	763	767	771	3.56	1.76
Ins-50-17	729	730.10	734	1.66	0.83	Ins-50-18	715	716.20	722	2.25	0.41
Ins-50-19	747	747.60	751	1.26	0.47	Ins-50-20	717	719.50	726	2.88	0.83
Ins-60-1	870	872.30	878	2.87	1.40	Ins-60-2	856	856.90	861	1.66	0.42
Ins-60-3	886	888.40	893	2.22	2.07	Ins-60-4	964	967.50	975	3.81	1.32
Ins-60-5	895	896.30	899	1.49	0.53	Ins-60-6	873	875.80	881	3.36	0.79
Ins-60-7	848	852.20	859	3.29	2.16	Ins-60-8	911	912.60	916	2.22	1.17
Ins-60-9	865	868.20	872	2.90	1.29	Ins-60-10	884	888	895	4.06	2.23
Ins-60-11	916	917.80	919	1.55	0.71	Ins-60-12	861	864.40	870	3.66	2.16
Ins-60-13	885	888	894	2.98	1.40	Ins-60-14	903	904.90	910	2.73	1.65
Ins-60-15	867	868.60	872	1.58	0.59	Ins-60-16	872	874.40	878	2.63	1.06
Ins-60-17	886	887	890	1.70	0.68	Ins-60-18	866	869.50	878	4.30	2.20
Ins-60-19	917	917.70	922	1.64	1.65	Ins-60-20	879	880.40	886	2.32	0.86

Table 29: Per-instance AILS results on the large random instances.

Inst.	AILS					Inst.	AILS				
	Best	Avg.	Worst	Std	Time		Best	Avg.	Worst	Std	Time
Ins-100-1	1738	1746.50	1758	6.74	2.46	Ins-100-2	1790	1797.50	1811	7.18	2.98
Ins-100-3	1706	1710.20	1722	5.27	1.51	Ins-100-4	1690	1696.20	1707	6.55	2.59
Ins-100-5	1761	1762.10	1768	2.23	1.47	Ins-100-6	1773	1775.30	1782	3.74	1.54
Ins-100-7	1726	1730.10	1747	6.85	2.46	Ins-100-8	1698	1708.10	1720	7.32	1.65
Ins-100-9	1746	1754.10	1772	9.15	2.01	Ins-100-10	1720	1729.40	1743	8.17	2.25
Ins-100-11	1722	1728.30	1740	6.50	1.38	Ins-100-12	1779	1787.80	1806	9.53	1.83
Ins-100-13	1744	1754.40	1767	8.34	2.11	Ins-100-14	1825	1835.40	1854	8.60	2.70
Ins-100-15	1713	1724.30	1739	8.79	1.42	Ins-100-16	1760	1769.50	1781	8.61	1.97
Ins-100-17	1755	1764.10	1789	12.91	1.45	Ins-100-18	1686	1694.20	1713	9.15	3.04
Ins-100-19	1726	1732	1740	5.31	1.77	Ins-100-20	1755	1764.20	1778	7.35	2.04
Ins-150-1	2608	2623.90	2644	15.29	4.30	Ins-150-2	2592	2606.90	2631	22.77	3.01
Ins-150-3	2630	2643.80	2663	40.06	4.00	Ins-150-4	2559	2569.60	2593	37.91	4.67
Ins-150-5	2627	2639.80	2660	18.33	3.91	Ins-150-6	2595	2612.40	2633	14.42	2.81
Ins-150-7	2613	2628.30	2651	39.50	3.53	Ins-150-8	2539	2555.20	2587	23.88	3.34
Ins-150-9	2512	2578.40	2618	39.77	4.07	Ins-150-10	2620	2642.40	2672	13.05	2.38
Ins-150-11	2646	2646	2646	30.35	3.47	Ins-150-12	2573	2590.80	2616	35.64	2.92
Ins-150-13	2638	2653.90	2681	28.85	3.70	Ins-150-14	2592	2606.40	2638	16.44	3.12
Ins-150-15	2598	2620.80	2642	22.34	4.75	Ins-150-16	2648	2658.70	2681	17.14	3.06
Ins-150-17	2619	2631.50	2655	23.60	2.67	Ins-150-18	2570	2591.40	2613	29.80	3.24
Ins-150-19	2633	2644.60	2669	19.70	3.93	Ins-150-20	2602	2623	2657	20.43	4.94
Ins-200-1	3540	3565.70	3609	21.36	8.95	Ins-200-2	3567	3577.90	3607	13.82	5.67
Ins-200-3	3544	3565.70	3611	23.30	9.56	Ins-200-4	3562	3579.40	3625	19.45	10.40
Ins-200-5	3570	3584.90	3628	19.30	6.05	Ins-200-6	3484	3497.20	3530	14.56	9.32
Ins-200-7	3637	3656.30	3682	13.24	6.52	Ins-200-8	3581	3596.50	3620	13.25	5.83
Ins-200-9	3575	3590.30	3617	13.60	9.67	Ins-200-10	3633	3658.90	3703	21.90	9.78
Ins-200-11	3612	3631.10	3668	20.18	6.53	Ins-200-12	3622	3637.80	3664	12.59	10.82
Ins-200-13	3496	3513.40	3533	14.42	10.43	Ins-200-14	3618	3631	3661	15.36	5.81
Ins-200-15	3602	3622.20	3642	15.12	9.83	Ins-200-16	3592	3613.60	3628	10.65	10.60
Ins-200-17	3520	3538.60	3572	17.37	6.66	Ins-200-18	3502	3519.80	3543	13.88	10.84
Ins-200-19	3493	3512.50	3547	18.41	9.97	Ins-200-20	3627	3641.40	3679	16.58	10.07

Chapter 3

Dynamic Bike Rebalancing Problem with Demand Intervals

3.1 Introduction

Bike-sharing systems (BSSs) have become a cornerstone of sustainable urban mobility, providing an affordable, low-emission, and flexible option for short and first-/last-mile trips in dense metropolitan areas. Their adoption has accelerated dramatically over the past two decades: the number of operational systems worldwide grew from only a handful in the early 2000s to several hundred within roughly a decade, with the largest concentrations emerging in Asia, Europe, and North America (Fishman, 2016b). By substituting for a share of private-car trips, BSSs are associated with lower greenhouse-gas emissions, reduced road congestion, public-health benefits from increased physical activity, and tighter integration with mass-transit networks (Fishman, 2016b; Laporte et al., 2018b). Together with advances in information and communication technologies, these benefits have transformed bike sharing from a niche initiative into a mainstream instrument of contemporary transportation planning.

Despite this success, BSSs face a fundamental operational challenge: the demand for bicycles and for vacant docks is highly uneven in both space and time. Commuting, shopping, and leisure flows cause some stations to be depleted of bikes while others saturate to capacity, so that renters cannot find a bicycle and returners cannot find a free dock. Both situations translate directly into lost demand and degraded quality of service, which in turn suppress ridership and threaten the long-term viability of the system (Raviv and Kolka, 2013b; Laporte et al., 2018b). To counteract these imbalances, operators perform *rebalancing*, that is, relocating bicycles among stations and a depot using one or more dedicated vehicles. Rebalancing is thus not a peripheral support activity but a core operational decision that largely determines whether a BSS can remain reliable under realistic demand fluctuations, and it is consistently identified as one of the central planning problems in the shared-mobility literature (Laporte et al., 2018b; C. S. Shui and W. Y. Szeto, 2020b).

Operational rebalancing couples two decisions that are difficult to separate in practice. The first is an inventory decision: how many bicycles should be available at each station, or, more generally, what range of inventories should be considered acceptable. The second is a routing and transfer decision: which stations should a vehicle visit, in what sequence, and how many bicycles should it pick up or

deliver. Research has traditionally concentrated on *static* (typically overnight) planning, in which a vehicle restores stations to predetermined target levels before the next operating day. The static problem was first formalised as a one-commodity pickup-and-delivery routing problem and shown to be computationally hard, which motivated exact branch-and-cut algorithms and standardised benchmark formulations (Chemla et al., 2013b; Dell’Amico et al., 2014c). A key conceptual refinement replaced single target inventories with target *ranges*: Güneş Erdoğan et al. (2014) introduced the static bicycle relocation problem with demand intervals, requiring each station’s post-repositioning inventory to lie between a lower and an upper bound rather than at a single value, and solved it exactly through a Benders decomposition and a branch-and-cut scheme. Parallel work emphasized station-specific service levels and developed scalable heuristics and exact algorithms for single- and multi-vehicle settings (Raviv et al., 2013c; Iris A. Forma et al., 2015b; Sin C. Ho and W. Y. Szeto, 2017b; Bulhões et al., 2018b).

Static planning is intrinsically limited, however, when demand changes materially during the operating day: a plan that is optimal at night can become obsolete within hours once rentals and returns evolve. This limitation has driven a shift toward *dynamic* rebalancing, in which repositioning decisions are revised during the day in response to the observed system state and anticipated short-term demand, with the explicit aim of reducing lost demand as it arises rather than merely reaching an end-of-day configuration (Ghosh et al., 2017). A growing body of work formulates the dynamic problem through Markov decision processes, approximate dynamic programming, and multi-period mixed-integer programming, frequently embedded in rolling- or receding-horizon schemes that repeatedly re-solve a short look-ahead subproblem as new information becomes available (Legros, 2019b; X. Luo et al., 2022a; Liang et al., 2024b; Xiang Li et al., 2024b). The empirical evidence consistently indicates that anticipating future demand within such dynamic policies yields substantial reductions in lost demand relative to static or myopic strategies (Ghosh et al., 2017; Liang et al., 2024b).

A further operational complication is that the historical trip data used by these models are censored. Trip records contain successful rentals and returns, but they do not reveal prospective renters who abandon an empty station or riders whose return requests are blocked at a full station. Consequently, rates estimated directly from recorded trips understate latent demand precisely at the locations and times at which service reliability is most vulnerable. Although censored-demand estimation has been studied in bike-sharing and related shared-mobility settings (Negahban, 2019; Gammelli et al., 2020), its integration with risk-based inventory intervals and rolling-horizon vehicle routing remains limited. Ignoring this dependence can propagate a systematic estimation error through the entire decision process: understated rates yield insufficient inventory protection, and those distorted inventory targets subsequently guide routing and transfer decisions.

Despite the progress across these areas, several gaps therefore remain. First, many dynamic models represent demand through single point forecasts or fixed target inventories, which underrepresent both short-term demand uncertainty and the information lost at empty and full stations. Second, the estimation of latent station demand, the determination of service-oriented inventory targets, and the dynamic routing and repositioning decisions are commonly treated as separate tasks, even though the outputs of each task directly shape the next. Third, the field still lacks a unified yet operationally tractable framework that combines censoring-aware demand estimation, a probabilistically grounded characterisation of acceptable station inventories, and rolling-horizon control.

The present study addresses these gaps through four connected contributions.

- (C1) **Censoring-aware demand estimation and robust intervals.** CARI estimates latent station-period return and rental rates by correcting observed counts for boundary exposure in a continuous-time birth–death process. Bootstrap rate uncertainty is then propagated to inventory-level risk, producing robust station-period demand intervals.
- (C2) **Integrated dynamic inventory routing.** The intervals enter a rolling-horizon mixed-integer model that jointly determines routes, transfer quantities, vehicle loads, and station inventories. Its weighted service objective distinguishes deterministic operational loss from deviation outside the applicable demand interval.
- (C3) **A fleet-level solution workflow.** Vehicle-specific network partitioning limits the combinatorial burden. Within each cluster, a period-feasible VNS provides an incumbent, a fixed-route repair verifies model feasibility, a time-limited MIP refines the solution, and a final VNS pass exploits the refined incumbent.
- (C4) **A transparent computational assessment.** Event-driven simulation on 20 instances of 40 stations sampled from 2024 Capital Bikeshare data examines objective weights, rolling look-ahead length, CARI uncertainty width, alternative demand representations, and a no-rebalancing baseline, while reporting service, operational, and computational measures.

The contribution is the operational linkage of these components rather than the isolated introduction of censoring correction, demand intervals, VNS, or rolling-horizon control. Expected blocked demand from the birth–death model is used upstream to construct the intervals, whereas the routing model uses a deterministic weighted operational-loss proxy. Solver gaps are therefore interpreted only for a fixed cluster assignment and time limit. The reported computational results are averages from model-based simulations; without replication-level outputs, confidence intervals, or held-out operational validation, they support comparisons on the tested instances but not universal dominance claims. The remainder of the paper presents the problem description, solution method, and computational evaluation.

3.2 Literature Review

Research on bike-sharing rebalancing has matured into a substantial body of work spanning strategic, tactical, and operational decisions; comprehensive surveys are provided by Laporte et al. (2018b) and C. S. Shui and W. Y. Szeto (2020b), while Fishman (2016b) reviews the broader planning and behavioural context of bike-sharing systems. For the purposes of this study, the operational literature can be organised into six overlapping streams: static exact optimisation, station inventory and service-level modelling, scalable routing heuristics, stochastic demand modelling, dynamic real-time rebalancing, and latent-demand estimation. The streams differ mainly in how they represent demand uncertainty and observability, how tightly they couple inventory and routing decisions, and how much emphasis they place on operational tractability.

The earliest and most influential stream treats rebalancing as a static optimisation problem. Chemla et al. (2013b) cast the task as a generalised one-commodity pickup-and-delivery problem with a single capacitated vehicle, proved its combinatorial hardness, and developed exact branch-and-cut machinery for benchmark instances. Dell’Amico et al. (2014c) contributed additional mathematical formulations

and a standardised set of benchmark instances, clarifying the structural complexity of the problem and establishing a reference point for subsequent algorithmic work. A central refinement for the present paper is the move from single target inventories to inventory *intervals*: Güneş Erdoğan et al. (2014) required each station’s inventory after repositioning to lie within given lower and upper bounds rather than at a fixed value, and solved the resulting variant exactly. The multi-vehicle, multi-visit extension of the static problem was later studied by Bulhões et al. (2018b), who combined a branch-and-cut algorithm with an iterated local search metaheuristic on instances of up to two hundred stations. Collectively, these studies established the modelling language that still underpins most later research and demonstrated that even the static problem is difficult once vehicle capacities and routing are jointly considered.

A second stream focuses on how acceptable station inventories should be defined in the first place, linking redistribution decisions to station-level service quality. Raviv and Kolka (2013b) modelled a single station’s inventory as a continuous-time birth–death Markov process and introduced a user-dissatisfaction function that quantifies the expected penalty from unmet rental and return demand as a function of the initial inventory; this single-station, Markov-based characterisation is the direct methodological precursor to the demand-interval determination proposed in this paper. Raviv et al. (2013c) and Jasper Schuijbroek et al. (2017) extended the notion of station-specific service levels to network and routing contexts, while Datner et al. (2019b) explicitly accounted for the interactions among stations’ inventory levels and showed that neglecting these interactions degrades the resulting quality of service. This stream motivates representing each station’s target not as a single set point but as a range of inventories that keeps the combined risk of shortage and surplus within an acceptable tolerance. It also connects the present problem to the broader service-level inventory-control literature, in which stockout and overflow risks are translated into operational targets.

A third stream pursues scalable solution methods for realistically sized systems, because exact optimisation becomes intractable as the network grows. Jasper Schuijbroek et al. (2017) proposed a cluster-first, route-second heuristic that uses a polynomial-size clustering model to capture service-level feasibility while approximating routing cost, thereby decoupling the combinatorial explosion of the routing layer from the station-level inventory decision. Iris A. Forma et al. (2015b) developed a three-step math-heuristic for the static repositioning problem, and Sin C. Ho and W. Y. Szeto (2017b) proposed a hybrid large-neighbourhood search for the multi-vehicle variant that outperformed earlier math-heuristics at a fraction of the computation time. Gleditsch et al. (2024b) more recently introduced a column-generation heuristic for the dynamic problem and showed that rolling-horizon optimisation can deliver solution times compatible with real-time operation while substantially improving practice-based policies.

The routing literature offers a broader family of metaheuristics for this setting, including tabu search, large-neighbourhood search, and evolutionary methods. The present paper focuses on VNS because its alternating shaking and descent structure is straightforward to initialise from a feasible route and can be restarted naturally after a MIP refinement step. The general VNS framework was introduced by Mladenović and Hansen (1997) and has subsequently been applied across a wide range of routing problems. In the proposed framework, clustering limits the fleet-level combinatorial burden, VNS explores the route space within each vehicle-specific cluster, and the intermediate MIP stage refines the incumbent before the final VNS phase resumes the search.

A fourth stream introduces demand uncertainty explicitly, recognizing that deterministic models are

fragile when station-level demand varies substantially across days or intraday windows. Dell’Amico et al. (2018b) studied the bike-sharing rebalancing problem with stochastic demands, representing uncertainty through scenarios and solving the resulting stochastic program with integer L-shaped and branch-and-cut techniques. Maggioni et al. (2019b) proposed stochastic optimisation models for a bike-sharing problem with transshipment, penalising deviations from target inventory levels in addition to routing cost. This stream shows that uncertainty changes the structure of the optimisation itself: the policy must trade off current repositioning effort against expected downstream service failures, which makes the inventory objective fundamentally probabilistic.

The fifth stream addresses dynamic rebalancing directly. Ghosh et al. (2017) formalised dynamic repositioning as a sequential decision problem aimed at reducing lost demand during the day rather than only restoring end-of-day balance, redefining the objective from static feasibility to online service protection. Legros (2019b) modelled the dynamic problem as a Markov decision process with heterogeneous user-dissatisfaction costs and derived a tractable priority-based policy for deciding which station to rebalance and by how much. X. Luo et al. (2022a) applied approximate dynamic programming to dynamic intra-cell repositioning in free-floating systems, while Liang et al. (2024b) proposed a general multi-period mixed-integer programming framework together with a fine-grained simulator and identified the modelling assumptions that empirically yield the most effective rebalancing strategies. Closest to the dynamic structure of the present work, Xiang Li et al. (2024b) embedded a multi-period stochastic model within an improved rolling-horizon framework to handle uncertain demand. Compared with static approaches, these methods better reflect the operational reality of bike sharing, in which demand evolves continuously and decisions must be revised under time pressure; however, the demand estimates entering such models are often treated as directly observable or are not explicitly corrected for boundary censoring.

The sixth stream concerns the estimation of the latent demand hidden by empty and full stations. Negahban (2019) estimated real demand in bike-sharing systems when boundary states conceal unserved requests, and Gammelli et al. (2020) developed censored Gaussian-process models for latent shared-mobility demand. These contributions establish that successful trip counts alone need not be unbiased measures of the underlying request process. They primarily address the demand-estimation side of the problem, whereas the present paper carries this insight into an integrated inventory-routing workflow by correcting station-period rates before constructing demand intervals and before optimizing rolling-horizon rebalancing actions.

Across these streams, four consistent observations emerge. First, the problem is combinatorially hard even before uncertainty is introduced, which explains the continued reliance on exact formulations for benchmarking and on heuristics for implementation. Second, explicit service-level modelling has become standard, and pure cost minimisation is no longer regarded as sufficient for credible BSS operations. Third, censored observations can systematically bias the demand inputs used to construct inventory targets and evaluate rebalancing decisions. Fourth, the literature remains fragmented across deterministic routing, single-station inventory modelling, stochastic programming, dynamic control, and latent-demand estimation, and there is still no widely accepted framework that integrates censoring-aware rates and a probabilistically grounded characterisation of acceptable inventories with a dynamic rolling-horizon repositioning policy while preserving interpretability and computational efficiency.

This gap motivates the proposed DBRP-DI framework. Rather than treating each station’s target

as a single value or relying on an uncorrected point estimate of demand, CARI estimates latent rental and return rates from boundary-censored observations and propagates bootstrap rate uncertainty through a continuous-time birth–death model. The resulting robust station-period intervals limit worst-case expected blocked requests and are embedded in a single-vehicle mixed-integer formulation whose two weighted service components represent operational shortage and surplus loss and deviation from the interval. At the fleet level, vehicle-specific clustering and a VNS–MIP-refinement–VNS procedure make the rolling-horizon problem operationally manageable. In this way, DBRP-DI connects the interval-based static formulation of Güneş Erdoğan et al. (2014) and the Markov-based inventory analysis of Raviv and Kolka (2013b) with censored-demand estimation (Negahban, 2019; Gammelli et al., 2020) and the dynamic, multi-period rebalancing tradition exemplified by Ghosh et al. (2017), Liang et al. (2024b), and Xiang Li et al. (2024b).

3.3 Problem Description

The system is represented by the directed network $G = (V, A)$. The set of bike-sharing stations is $N = \{1, \dots, n\}$. If a depot or parking point is included in the routing network, it is denoted by node 0. For a vehicle cluster c , the corresponding physical routing set is written as $V_c = N_c \cup \{0\}$, where $N_c \subseteq N$ is the set of stations assigned to that cluster. Two artificial nodes, o and d , are used to represent the start and end of a vehicle route within a planning period. These artificial nodes are modelling devices and do not represent physical stations.

The global planning horizon is divided into the period set $P = \{1, \dots, P_{\max}\}$, where each period has duration $\bar{T}$. At each rolling decision epoch, the controller observes the current station inventories, vehicle positions, and vehicle loads, and then optimizes over a short look-ahead subset of P . Only the decisions associated with the first period of the rolling window are implemented before the horizon is shifted forward.

Each station $i \in N$ has capacity u_i and current inventory I_i^0 . During period p , latent return attempts and rental attempts are represented by independent Poisson processes. The underlying rates are not observed directly, because rentals are censored when a station is empty and returns are censored when a station is full. The CARI procedure described in Section 3.3.1 estimates censoring-adjusted rates $\hat{\lambda}_{ip}$ and $\hat{\mu}_{ip}$. The deterministic net drift used inside the routing model is evaluated as

$$R_{ip} = \hat{\lambda}_{ip} - \hat{\mu}_{ip}. \quad (37)$$

A positive value of R_{ip} indicates that, absent rebalancing, inventory tends to increase during the period; a negative value indicates that inventory tends to decrease.

The stochastic birth–death model is not embedded directly in the routing objective. Instead, it is used upstream to construct a service-oriented demand interval $[L_{ip}, H_{ip}]$ for each station-period pair. This interval is later passed to the routing model as the desirable end-of-period inventory range $[b_{ip}, \bar{b}_{ip}]$.

The rebalancing fleet consists of vehicles K . Vehicle $k \in K$ has capacity Q_k , initial load q_k^0 , and current physical position $\xi_k^0 \in \mathcal{V}$. Travel time on arc $(i, j) \in A$ is τ_{ij} , and τ^t is the handling time required to load or unload one bike. For distinct physical locations $i, j \in \mathcal{V}$, travel times are assumed to satisfy $\tau_{ij} > 0$. Travel distance is not included in the routing objective of this paper. It may still appear in descriptive data

or external reporting, but the proposed optimisation criterion contains only service-based terms.

At each rolling epoch, the fleet problem is decomposed into vehicle-specific station clusters. For a fixed cluster and a fixed vehicle, the routing model determines the sequence of visited locations and the number of bikes picked up or delivered in each period. The vehicle may start a period at the physical position inherited from the previous period, serve a sequence of stations, and end at a physical location that becomes the starting location of the next period. Inventory evolves from one period to the next by combining the vehicle transfer and the net demand drift.

Two service penalties guide the routing decisions. The first is the weighted operational-loss proxy (WOLP), which penalizes shortage and surplus implied by the deterministic net-demand trajectory. This term is a tractable surrogate for service loss; it is not the stochastic expectation of blocked requests. The birth–death expected blocked counts are used only when constructing demand intervals. The second penalty measures deviation from the CARI demand interval. The routing objective is therefore a weighted sum of WOLP and interval deviation. This separation avoids mixing routing convenience, travel distance, and stochastic-loss estimation in the same term.

The solution procedure mirrors the operational timing. First, a censoring-aware rate estimator constructs demand intervals from historical successful-trip data. Second, the current network is partitioned across vehicles. Third, a variable neighbourhood search (VNS) constructs a period-feasible route for each cluster. Fourth, the route is repaired into a complete feasible mixed-integer start, and a time-limited MIP refinement is launched from that start. If the refinement improves the incumbent but does not close the bound gap within the assigned time, the final incumbent is decoded and passed back to VNS for one additional improvement pass. This workflow is heuristic at the fleet level and should be interpreted as a controlled decision-support procedure rather than as a certificate of optimality for the joint fleet problem.

3.3.1 Demand interval construction

This subsection describes the CARI procedure used to construct the demand interval assigned to each station-period pair. The purpose of this interval is to define a range of desirable boundary inventories for the routing model, rather than forcing vehicle-routing decisions toward a single target inventory level.

For each station $i \in N$ and period $p \in P$, let $\mathcal{H}_{ip}$ denote the set of historical observations associated with that station-period pair. For each historical observation $\eta \in \mathcal{H}_{ip}$, let $l_{ip\eta}$ be the observed inventory at the beginning of the period, and let $n_{ip\eta}^{\text{ret}}$ and $n_{ip\eta}^{\text{rent}}$ denote the observed numbers of bike returns and bike rentals during the period, respectively. The corresponding censored observed rates are

$$\lambda_{ip}^{\text{obs}} = \frac{1}{|\mathcal{H}_{ip}|T} \sum_{\eta \in \mathcal{H}_{ip}} n_{ip\eta}^{\text{ret}}, \quad \mu_{ip}^{\text{obs}} = \frac{1}{|\mathcal{H}_{ip}|T} \sum_{\eta \in \mathcal{H}_{ip}} n_{ip\eta}^{\text{rent}}. \quad (38)$$

Because observed rentals and returns are censored when a station is empty or full, the observed rates may underestimate the underlying attempt rates. To account for this effect, station inventory is modelled as a finite-capacity birth–death process on states $\{0, \dots, u_i\}$. For candidate return and rental rates (λ, μ) ,

define the generator matrix $\mathbf{G}_i(\lambda, \mu)$ by

$$\begin{aligned} (\mathbf{G}_i)_{q,q+1} &= \lambda, & q &= 0, \dots, u_i - 1, \\ (\mathbf{G}_i)_{q,q-1} &= \mu, & q &= 1, \dots, u_i, \\ (\mathbf{G}_i)_{q,q} &= \begin{cases} -\lambda, & q = 0, \\ -(\lambda + \mu), & q = 1, \dots, u_i - 1, \\ -\mu, & q = u_i. \end{cases} \end{aligned} \quad (39)$$

All other entries of $\mathbf{G}_i(\lambda, \mu)$ are equal to zero.

For historical observation η , the transient probability vector is

$$\boldsymbol{\pi}_{ip\eta}(t; \lambda, \mu) = \mathbf{e}_{i_{p\eta}}^\top \exp(\mathbf{G}_i(\lambda, \mu)t), \quad (40)$$

where $\mathbf{e}_{i_{p\eta}}$ is the unit vector associated with the observed initial inventory. The time-averaged probabilities of being empty and full are

$$a_{ip\eta}^-(\lambda, \mu) = \frac{1}{\bar{T}} \int_0^{\bar{T}} \pi_{ip\eta,0}(t; \lambda, \mu) dt, \quad a_{ip\eta}^+(\lambda, \mu) = \frac{1}{\bar{T}} \int_0^{\bar{T}} \pi_{ip\eta,u_i}(t; \lambda, \mu) dt. \quad (41)$$

The average empty and full probabilities over historical observations are

$$\bar{a}_{ip}^-(\lambda, \mu) = \frac{1}{|\mathcal{H}_{ip}|} \sum_{\eta \in \mathcal{H}_{ip}} a_{ip\eta}^-(\lambda, \mu), \quad \bar{a}_{ip}^+(\lambda, \mu) = \frac{1}{|\mathcal{H}_{ip}|} \sum_{\eta \in \mathcal{H}_{ip}} a_{ip\eta}^+(\lambda, \mu). \quad (42)$$

The CARI-adjusted rates are obtained by a fixed-point correction. Starting from $\lambda_{ip}^{(0)} = \lambda_{ip}^{\text{obs}}$ and $\mu_{ip}^{(0)} = \mu_{ip}^{\text{obs}}$, the updates are

$$\lambda_{ip}^{(\ell+1)} = \frac{\lambda_{ip}^{\text{obs}}}{\max\{\epsilon, 1 - \bar{a}_{ip}^+(\lambda_{ip}^{(\ell)}, \mu_{ip}^{(\ell)})\}}, \quad \mu_{ip}^{(\ell+1)} = \frac{\mu_{ip}^{\text{obs}}}{\max\{\epsilon, 1 - \bar{a}_{ip}^-(\lambda_{ip}^{(\ell)}, \mu_{ip}^{(\ell)})\}}, \quad (43)$$

where $\epsilon > 0$ is a small numerical safeguard. The iteration stops when the relative change in both rates is below a prescribed tolerance; a maximum number of iterations is also imposed as a numerical safeguard. The resulting estimates are denoted by $\hat{\lambda}_{ip}$ and $\hat{\mu}_{ip}$. These estimates define the net drift used in the routing model, as stated in Eq. (37).

To account for estimation uncertainty, a bootstrap uncertainty set is constructed. Let $\{(\hat{\lambda}_{ip}^b, \hat{\mu}_{ip}^b)\}_{b=1}^{B_{\text{boot}}}$ be the rate estimates obtained from B_{boot} bootstrap resamples of $\mathcal{H}_{ip}$. For a desired joint confidence level $1 - \delta$, a Bonferroni-adjusted rectangular uncertainty set is defined as

$$\mathcal{U}_{ip} = [\lambda_{ip}^L, \lambda_{ip}^U] \times [\mu_{ip}^L, \mu_{ip}^U]. \quad (44)$$

Here, λ_{ip}^L and λ_{ip}^U are the empirical $\delta/4$ and $1 - \delta/4$ quantiles of the bootstrap estimates of λ , and μ_{ip}^L and μ_{ip}^U are the corresponding $\delta/4$ and $1 - \delta/4$ quantiles of the bootstrap estimates of μ . By the Bonferroni inequality, this construction provides a joint coverage of at least $1 - \delta$. In the implementation,

the uncertainty-set width is indexed by the standard-normal confidence parameter z_{CARI} , with $\delta = 2[1 - \Phi(z_{\text{CARI}})]$, where Φ is the standard normal distribution function. Thus, z_{CARI} controls the confidence level used to select the bootstrap quantiles; it is distinct from the model objective $\mathcal{J}$ and from the risk tolerance Γ_{ip} .

Next, CARI evaluates the risk associated with candidate boundary inventory levels. Let $q \in \{0, \dots, u_i\}$ denote a candidate inventory level for station i at the boundary associated with period p . For given rates (λ, μ) , define

$$\pi_i^q(t; \lambda, \mu) = \mathbf{e}_q^\top \exp(\mathbf{G}_i(\lambda, \mu)t). \quad (45)$$

The expected lost rental demand caused by an empty station and the expected lost return demand caused by a full station are

$$B_{ip}^-(q; \lambda, \mu) = \mu \int_0^{\bar{T}} \pi_{i,0}^q(t; \lambda, \mu) dt, \quad B_{ip}^+(q; \lambda, \mu) = \lambda \int_0^{\bar{T}} \pi_{i,u_i}^q(t; \lambda, \mu) dt. \quad (46)$$

The quantities $a_{ip\eta}^-$ and $a_{ip\eta}^+$ are time-averaged boundary-state probabilities used for censoring correction, whereas B_{ip}^- and B_{ip}^+ are expected blocked-demand counts over a period and therefore are not normalized by $\bar{T}$.

The worst-case pointwise operational risk of inventory level q is

$$r_{ip}(q) = \max_{(\lambda, \mu) \in \mathcal{U}_{ip}} \left\{ B_{ip}^-(q; \lambda, \mu) + B_{ip}^+(q; \lambda, \mu) \right\}. \quad (47)$$

For a candidate interval $[L, H]$, define its worst-case interval risk as

$$C_{ip}(L, H) = \max_{q \in \{L, \dots, H\}} r_{ip}(q). \quad (48)$$

Given a risk tolerance Γ_{ip} , the demand interval is selected as the widest interval whose worst-case risk does not exceed the tolerance:

$$(L_{ip}, H_{ip}) \in \arg \max_{\substack{0 \leq L \leq H \leq u_i \\ L, H \in \mathbb{Z}}} \{H - L : C_{ip}(L, H) \leq \Gamma_{ip}\}. \quad (49)$$

If several intervals have the same maximum width, the interval with the smaller value of $C_{ip}(L, H)$ is selected. If a tie remains, the interval whose midpoint is closest to the minimum-risk inventory level is selected, where

$$q_{ip}^* \in \arg \min_{q \in \{0, \dots, u_i\}} r_{ip}(q). \quad (50)$$

If no interval satisfies the risk tolerance, the minimum-risk singleton interval $[q_{ip}^*, q_{ip}^*]$ is used. The tolerance Γ_{ip} is a service-level parameter fixed by the service specification; the computational experiments examine sensitivity to the uncertainty-set width through z_{CARI} .

The output of the demand-interval construction is linked to the routing model by setting

$$b_{ip} = L_{ip}, \quad \bar{b}_{ip} = H_{ip}. \quad (51)$$

If the upper bound is alternatively expressed through the minimum required number of empty docks, this

value can be recovered as

$$s_{ip} = u_i - H_{ip}, \quad \bar{b}_{ip} = u_i - s_{ip}. \quad (52)$$

3.3.2 Mathematical Formulation and Notation

The formulation is presented for a fixed vehicle cluster. Let $N_c \subseteq N$ denote the set of stations assigned to cluster c , and let $V_c = \{0\} \cup N_c$ denote the corresponding set of physical locations. For notational compactness, the vehicle assigned to this cluster is represented by its capacity Q , initial load F , and initial location V^0 . These correspond to the fleet-level quantities Q_k , q_k^0 , and ξ_k^0 for the selected vehicle k .

The model is defined over a consecutive period set P . Let $p_0 = \min P$ and $p_f = \max P$ denote the first and last periods in this set. In a rolling-horizon implementation, P is replaced by the current look-ahead window P_r , as described in Section 3.4.1. For each station $i \in N_c$ and period $p \in P$, the routing model uses the CARI-adjusted net rate R_{ip} defined in Eq. (37). The desired end-of-period demand interval is $[b_{ip}, \bar{b}_{ip}]$, where $b_{ip} = L_{ip}$ and $\bar{b}_{ip} = H_{ip}$ are obtained from the CARI procedure.

Two artificial nodes, o and d , are added to represent the beginning and end of the vehicle route in each period. The extended node set is $V'_c = V_c \cup \{o, d\}$. Let $\mathcal{A}'_c$ denote the set of feasible arcs in the extended graph. No arc enters o , and no arc leaves d . Artificial arcs from o and to d record the physical starting and ending locations of a period; they do not represent additional physical travel. Therefore, travel times on artificial arcs are set to zero.

The binary variable x_{ijp} equals 1 if the vehicle travels directly from node i to node j in period p . The binary variable y_{ip} equals 1 if node i is visited in period p . The non-negative integer variable v_{ip}^+ denotes the number of bikes loaded from node i onto the vehicle, while v_{ip}^- denotes the number of bikes unloaded from the vehicle to node i . The continuous variable f_{ijp} represents the vehicle load immediately after departing node i toward node j , and t_{ip} denotes the service start time at node i .

For each station $i \in N_c$, I_{ip} denotes the inventory immediately before vehicle service in period p ; if station i is not visited, it represents the inventory at the beginning of that period. Variables D_{ip} and E_{ip} denote the shortage and excess-bike quantities accumulated before service. Similarly, $\bar{I}_{ip}$, $\bar{D}_{ip}$, and $\bar{E}_{ip}$ denote the end-of-period inventory, shortage, and excess-bike quantities. The variable G_{ip} measures the deviation of the end-of-period inventory from $[b_{ip}, \bar{b}_{ip}]$.

Define the operational loss proxy as

$$\text{OLP}_{ip} = D_{ip} + E_{ip} + \bar{D}_{ip} + \bar{E}_{ip}. \quad (53)$$

The coefficients κ_p^{olp} and κ_p^{int} are generic period-dependent weights. Their rolling-horizon specification is given in Section 3.4.1.

Let

$$M_i^v = \begin{cases} Q, & i = 0, \\ \min\{Q, u_i\}, & i \in N_c, \end{cases}$$

and set $v_{op}^+ = v_{op}^- = 0$ and $M_o^v = 0$ for the artificial origin. A safe inventory-loss bound is

$$M_{ip}^I = u_i + |R_{ip}| \bar{T}. \quad (54)$$

For the time-propagation constraints, define

$$M_{ijp}^t = p\overline{T} + \tau_{ij} + \tau^t M_i^v. \quad (55)$$

The model is formulated as follows:

$$\min \mathcal{J} = \sum_{p \in \mathcal{P}} \sum_{i \in N_c} \left(\kappa_p^{\text{olp}} \text{OLP}_{ip} + \kappa_p^{\text{int}} G_{ip} \right). \quad (56)$$

$$y_{op} = y_{dp} = 1 \quad \forall p \in P, \quad (57)$$

$$\sum_{j:(o,j) \in \mathcal{A}'_c} x_{ojp} = 1 \quad \forall p \in P, \quad (58)$$

$$\sum_{i:(i,d) \in \mathcal{A}'_c} x_{idp} = 1 \quad \forall p \in P, \quad (59)$$

$$x_{oV^0p_0} = 1, \quad (60)$$

$$x_{idp} = x_{oi,p+1} \quad \forall i \in V_c, p = p_0, \dots, p_f - 1, \quad (61)$$

$$\sum_{j:(i,j) \in \mathcal{A}'_c} x_{ijp} = y_{ip} \quad \forall i \in V_c, p \in P, \quad (62)$$

$$\sum_{j:(j,i) \in \mathcal{A}'_c} x_{jip} = y_{ip} \quad \forall i \in V_c, p \in P, \quad (63)$$

$$\sum_{j:(i,j) \in \mathcal{A}'_c} f_{ijp} - \sum_{j:(j,i) \in \mathcal{A}'_c} f_{jip} = v_{ip}^+ - v_{ip}^- \quad \forall i \in V_c, p \in P, \quad (64)$$

$$f_{ijp} \leq Qx_{ijp} \quad \forall (i,j) \in \mathcal{A}'_c, p \in P, \quad (65)$$

$$\sum_{j:(o,j) \in \mathcal{A}'_c} f_{ojp_0} = F, \quad (66)$$

$$\sum_{j:(o,j) \in \mathcal{A}'_c} f_{oj,p+1} = \sum_{i:(i,d) \in \mathcal{A}'_c} f_{idp} \quad p = p_0, \dots, p_f - 1, \quad (67)$$

$$v_{ip}^+ + v_{ip}^- \leq M_i^v y_{ip} \quad \forall i \in V_c, p \in P, \quad (68)$$

$$t_{op} = (p-1)\bar{T} \quad \forall p \in P, \quad (69)$$

$$(p-1)\bar{T}y_{ip} \leq t_{ip} \leq p\bar{T}y_{ip} \quad \forall i \in V_c \cup \{d\}, p \in P, \quad (70)$$

$$t_{jp} \geq t_{ip} + \tau_{ij} + \tau^t (v_{ip}^+ + v_{ip}^-) - M_{ijp}^t (1 - x_{ijp}) \quad \forall (i,j) \in \mathcal{A}'_c, i \in V_c \cup \{o\}, p \in P, \quad (71)$$

$$\bar{I}_{i,p_0-1} = I_i^0 \quad \forall i \in N_c, \quad (72)$$

$$I_{ip} = \bar{I}_{i,p-1} + R_{ip} (t_{ip} - (p-1)\bar{T}y_{ip}) + D_{ip} - E_{ip} \quad \forall i \in N_c, p \in P, \quad (73)$$

$$D_{ip} + E_{ip} \leq M_{ip}^I y_{ip} \quad \forall i \in N_c, p \in P, \quad (74)$$

$$0 \leq I_{ip} \leq u_i \quad \forall i \in N_c, p \in P, \quad (75)$$

$$0 \leq I_{ip} - v_{ip}^+ + v_{ip}^- \leq u_i \quad \forall i \in N_c, p \in P, \quad (76)$$

$$\bar{I}_{ip} = I_{ip} - v_{ip}^+ + v_{ip}^- + R_{ip} (\bar{T} - t_{ip} + (p-1)\bar{T}y_{ip}) + \bar{D}_{ip} - \bar{E}_{ip} \quad \forall i \in N_c, p \in P, \quad (77)$$

$$0 \leq \bar{I}_{ip} \leq u_i \quad \forall i \in N_c, p \in P, \quad (78)$$

$$G_{ip} \geq b_{ip} - \bar{I}_{ip} \quad \forall i \in N_c, p \in P, \quad (79)$$

$$G_{ip} \geq \bar{I}_{ip} - \bar{b}_{ip} \quad \forall i \in N_c, p \in P, \quad (80)$$

$$x_{ijp} \in \{0, 1\} \quad \forall (i,j) \in \mathcal{A}'_c, p \in P, \quad (81)$$

$$y_{ip} \in \{0, 1\} \quad \forall i \in V'_c, p \in P, \quad (82)$$

$$v_{ip}^+, v_{ip}^- \in \mathbb{Z}_+ \quad \forall i \in V_c, p \in P, \quad (83)$$

$$f_{ijp} \geq 0 \quad \forall (i,j) \in \mathcal{A}'_c, p \in P, \quad (84)$$

$$t_{ip} \geq 0 \quad \forall i \in V'_c, p \in P, \quad (85)$$

$$I_{ip}, \bar{I}_{ip}, D_{ip}, E_{ip}, \bar{D}_{ip}, \bar{E}_{ip}, G_{ip} \geq 0 \quad \forall i \in N_c, p \in P. \quad (86)$$

The objective function (56) gives the generic weighted form of the cluster model. The first term penalizes the operational-loss proxy, and the second term penalizes deviations from the CARI demand interval. No travel-distance term is included in the objective function. Travel times enter the model only through feasibility constraints.

Constraints (57)–(59) activate the artificial origin and destination and ensure that each period has exactly one starting arc and one ending arc. Constraint (60) fixes the starting location of the first period in the current model horizon. Constraints (61) link consecutive periods by requiring the vehicle to start period $p + 1$ from the location at which it ended period p . Constraints (62) and (63) connect node visits to outgoing and incoming arcs. Constraint (64) conserves the vehicle load at each physical node. Constraints (65)–(67) enforce vehicle-capacity consistency, define the initial vehicle load, and carry the vehicle load between consecutive periods. Constraint (68) allows bike transfers only at visited locations and bounds the transferred quantity by the station and vehicle capacities.

Constraint (69) sets the start time of each period. Constraints (70) keep visit times within the corresponding period. Constraint (71) propagates service start times along selected arcs while accounting for travel time and bike-transfer time. With strictly positive travel or service times on physical arcs, this constraint also prevents disconnected subtours, because a selected cycle would require service start times to increase around a closed loop. Constraint (72) initializes station inventories for the current model horizon. Constraint (73) computes the inventory immediately before service. Constraint (74) links pre-service shortage and excess variables to station visits. Constraints (75) and (76) ensure feasible pre-service and post-service station inventories. Constraint (77) updates the end-of-period inventory after the transfer operation and the remaining within-period demand evolution. Constraint (78) bounds the end-of-period inventory. Constraints (79) and (80) define the deviation from the demand interval. Constraints (81)–(86) specify variable domains.

3.4 Rolling-Horizon Fleet Solution Method

3.4.1 Temporal decomposition and objective

At rolling iteration r , the controller observes station inventories, vehicle locations, and vehicle loads. It then constructs a look-ahead window

$$P_r = \{r, \dots, r + h_r - 1\}, \quad h_r = \min\{h_{RH}, P_{\max} - r + 1\}.$$

Only the decisions associated with period r are implemented; the horizon is then shifted forward, and the system state is updated using the realised events. Let $h = p - r + 1$ denote the relative position of period p in the current look-ahead window. Discount weights ω_h^{olp} and ω_h^{int} may be used to emphasize near-term service.

When the formulation in Section 3.3.2 is solved at rolling iteration r , the period set P is set to P_r , and the current observed station inventories, vehicle location, and vehicle load define I_i^0 , V^0 , and F , respectively. For a fixed vehicle cluster N_c , the generic weights in (56) are specified as

$$\kappa_p^{\text{olp},(r)} = \kappa_{\text{olp}} \omega_h^{\text{olp}}, \quad \kappa_p^{\text{int},(r)} = \kappa_{\text{int}} \omega_h^{\text{int}}, \quad h = p - r + 1. \quad (87)$$

When solving the rolling subproblem, the generic coefficients in (56) are set to $\kappa_p^{\text{olp}} = \kappa_p^{\text{olp},(r)}$ and $\kappa_p^{\text{int}} = \kappa_p^{\text{int},(r)}$ for $p \in P_r$. Accordingly, the rolling objective for cluster c is

$$\min \sum_{p \in P_r} \sum_{i \in N_c} \left[\kappa_{\text{olp}} \omega_h^{\text{olp}} \text{OLP}_{ip} + \kappa_{\text{int}} \omega_h^{\text{int}} G_{ip} \right]. \quad (88)$$

No travel-distance term is included in Eq. (88). Travel times remain part of the feasibility constraints, because a route that cannot be completed within a period is not operationally meaningful.

3.4.2 Spatial decomposition by reproducible network partitioning

The station assignment is an outer fleet heuristic, not a component of the single-vehicle model. For the current rolling window, let OLP_{ip}^0 and G_{ip}^0 be the no-service values obtained by propagating the current inventory through the deterministic clipping relations. The workload of station i is

$$w_i = \sum_{p \in P_r} \left(\kappa_{\text{olp}} \omega_h^{\text{olp}} \text{OLP}_{ip}^0 + \kappa_{\text{int}} \omega_h^{\text{int}} G_{ip}^0 \right). \quad (89)$$

For a tentative cluster C_k , define

$$\Phi(C_1, \dots, C_{|K|}) = \sum_{k \in K} \left| \sum_{i \in C_k} w_i - \frac{1}{|K|} \sum_{i \in N} w_i \right| + \rho_{\text{cl}} \sum_{k \in K} \sum_{i \in C_k} \tau_{\xi_k^0}^i. \quad (90)$$

The first term balances expected service workload, while the second discourages assigning a vehicle to a distant cluster at the start of the iteration.

At the first iteration, stations are sorted by polar angle around the depot and contiguous sweeps are evaluated over all cyclic starting positions; the sweep with the smallest Φ is retained. At later iterations, the previous assignment is reused as the starting solution. Newly active stations are inserted into the cluster with the smallest incremental Φ , and a best-improvement relocation and swap search is then applied until no accepted move decreases Φ . This procedure is deterministic for a fixed tie-breaking rule and random seed. It terminates on a finite set of partitions, but it does not provide a global fleet guarantee.

Algorithm 6 Rolling partition update

Require: current station states; vehicle positions ξ_k^0 ; previous clusters if available

- 1: Compute no-service values OLP_{ip}^0 and G_{ip}^0 for all active station-period pairs
- 2: Compute station workloads w_i from Eq. (89)
- 3: **if** first rolling iteration **then**
- 4: Generate angular sweep partitions and retain the partition with smallest Φ
- 5: **else**
- 6: Start from the previous partition and insert newly active stations by smallest incremental Φ
- 7: **end if**
- 8: **repeat**
- 9: Apply the best improving relocation or swap move under Eq. (90)
- 10: **until** no improving move is available

Ensure: vehicle-specific station clusters

3.4.3 Period-feasible VNS construction and improvement

For each cluster, VNS starts from a route that is feasible with respect to period length and vehicle capacity. Candidate service quantities are evaluated by replaying the route through the deterministic inventory dynamics. At a visited station i in period p , the replay computes the pre-service inventory $\tilde{I}_{ip}$ and the vehicle load ℓ_{ip} immediately before service. Candidate pickup and delivery quantities are bounded by

$$0 \leq q_{ip}^+ \leq \min\{Q - \ell_{ip}, \tilde{I}_{ip}\}, \quad 0 \leq q_{ip}^- \leq \min\{\ell_{ip}, u_i - \tilde{I}_{ip}\}. \quad (91)$$

The replay rejects any move that violates period duration, vehicle load, or station capacity.

The local search uses Relocate, Swap, 2-opt, Or-opt(2), and Or-opt(3). After an improving move, the search restarts from the first neighbourhood. The shaking phase applies s random relocations, then VND is run from the shaken solution. If the resulting route improves the WOLP–interval objective, the incumbent is updated and s is reset; otherwise s is increased cyclically up to $s_{\max}$. The random seed, $s_{\max}$, and the stopping limit are algorithm parameters and are reported with the computational design.

Algorithm 7 Period-feasible VNS for one cluster

Require: cluster stations; vehicle state; period window; demand intervals; time limit or iteration limit

- 1: Construct a period-feasible initial route by greedy insertion
- 2: Replay the route to assign feasible transfer quantities and evaluate WOLP plus interval deviation
- 3: Apply VND with Relocate, Swap, 2-opt, Or-opt(2), and Or-opt(3)
- 4: **while** stopping criterion is not met **do**
- 5: Shake the incumbent by s random relocations
- 6: Repair period feasibility and replay transfer quantities
- 7: Apply VND to the shaken route
- 8: **if** the replayed WOLP–interval objective improves **then**
- 9: Accept the route and reset $s \leftarrow 1$
- 10: **else**
- 11: Increase s cyclically up to $s_{\max}$
- 12: **end if**
- 13: **end while**

Ensure: incumbent VNS route and service quantities

3.4.4 Fixed-route repair, MIP refinement, and feedback VNS

The VNS output is a feasible routing-policy candidate, but the station sequence and replayed quantities are not treated as a complete certificate for the mixed-integer formulation. Therefore, the selected route binaries x and visit binaries y are fixed first. A short fixed-route repair subproblem then determines transfers, loads, timing variables, clipping regimes, and inventory variables under the full constraints of Section 3.3.2. Only a repair solution that satisfies all model constraints is used as the initial mixed-integer solution.

Starting from this feasible initial solution, the full single-vehicle cluster model is run for the allotted computation time. The refinement phase may improve the incumbent and may also return a valid bound

for the fixed cluster when the solver status provides one. Any reported gap is conditional on the current cluster assignment and the time-limited model run; it is not a guarantee for the original joint fleet problem.

If the time limit is reached before the bound gap is closed and the refinement incumbent differs from the final VNS solution, the incumbent is decoded into a route and service plan and passed back to VNS for one additional improvement pass. This final feedback pass is intended to exploit local route improvements around the refined solution while preserving period feasibility. The accepted solution for the cluster is the best feasible solution found across the initial VNS, repaired warm start, MIP refinement, and feedback VNS stages.

Algorithm 8 Rolling DBRP-DI solution workflow

Require: initial inventories; vehicle states; historical trip data; planning periods; rolling-window length

- 1: Construct CARI demand intervals for all station-period pairs
- 2: **for** each rolling iteration r **do**
- 3: Observe current inventories, vehicle positions, and vehicle loads
- 4: Update vehicle-specific clusters using Algorithm 6
- 5: **for** each vehicle cluster **do**
- 6: Run VNS to obtain a period-feasible incumbent route
- 7: Solve the fixed-route repair problem to obtain a feasible mixed-integer start
- 8: Start the full cluster model from the repaired solution and run it within the allotted time
- 9: **if** the time limit is reached before the bound gap is closed and the incumbent has changed **then**
- 10: Decode the model incumbent and run one feedback VNS pass
- 11: **end if**
- 12: **end for**
- 13: Implement only first-period decisions and update the system state by event-driven simulation
- 14: **end for**

Ensure: rolling sequence of vehicle routes and transfer decisions

3.4.5 State update and event-driven simulation

After first-period decisions are implemented, station inventories and vehicle positions are updated by chronological events. Travel and handling events are processed in route order. Rental and return attempts are sampled from the latent demand process for the simulated period; successful events change inventory, while events blocked by an empty or full station are counted as service failures. The next rolling iteration starts from the resulting physical state. This event-driven update is used for offline evaluation of the policy and is separate from the deterministic WOLP calculation used inside the routing model.

3.5 Computational Results

In this section, we present the outcomes of computational experiments conducted to evaluate the performance of the proposed mathematical model. The model was implemented in C++ and solved

using CPLEX 20.1 as the mixed-integer linear programming (MILP) solver. All experiments were performed on a laptop equipped with an Intel Core i5 processor running at 2.50 GHz and 12 GB of RAM. Details regarding the real-world data instances used for testing are provided in Section 3.5.1.

3.5.1 Data instances

Capital Bikeshare was selected because its public, year-long, time-stamped trip records provide repeated weekday observations over a large and geographically dense docked network, while road-network travel data are available for routing. These features make it a useful testbed for estimating station-period rates and constructing varied network samples. The study constructed 20 instances from trips recorded between 1 January and 31 December 2024. Because the source system has no operator depot in the trip data, a location near the Lincoln Memorial station was introduced as a synthetic depot; this modelling choice is not evidence of an actual depot or operating practice.

The analysis uses workdays from 08:00 to 16:00, divided into sixteen 30-minute periods, and classic bicycles only because the available fields did not support a consistent treatment of electric bicycles. Historical records were used to estimate station-period rental and return rates. Each instance contains 40 randomly selected stations, with initial stocks generated inside the first-period demand intervals; feasible flows were then simulated from the estimated rates as described in Section 3.4.5.

3.5.2 Experimental design and performance measures

The experiments are organised in five parts. First, the objective-function weights are calibrated on four randomly selected instances. Second, the effect of the rolling-horizon length is evaluated to compare myopic and look-ahead policies. Third, the sensitivity of the CARI interval-width parameter is tested. Fourth, three demand representations are compared: a point target demand, the benchmark interval demand representation, and the proposed robust demand interval. Finally, the selected robust-interval rebalancing policy is compared with a no-rebalancing baseline. The same performance measures are used throughout the experiments so that service quality, operational effort, and computational effort can be interpreted jointly.

3.5.3 Calibration of objective-function weights

The rolling objective contains two service components. The first-period coefficient of WOLP is fixed at $\kappa_1^{\text{olp}} = 1$. The initial coefficient of the interval-deviation term, κ_1^{int} , and the discount factors applied to later periods are selected from $\{0.1, 0.25, 0.5, 0.75, 1\}$. Table 30 reports the most informative configurations from the 25 tested combinations.

Table 30: Representative results from objective-weight calibration.

Configuration	κ_1^{olp}	ω^{olp} decay	κ_1^{int}	ω^{int} decay	WOLP	Realised loss
Lowest WOLP	1.00	0.10	0.10	0.10	0.4925	4.50
Lowest realised loss and distance	1.00	0.50	0.50	0.50	0.6025	3.50
Lowest computation time	1.00	0.10	0.75	0.10	0.6450	4.75

The configuration with $\kappa_1^{\text{olp}} = 1$, $\kappa_1^{\text{int}} = 0.5$, and discount factors of 0.5 for both objective components gives the lowest realised lost demand in the calibration set. It also gives the shortest average travelled distance among all tested parameter combinations. Although the setting with $\kappa_1^{\text{int}} = 0.1$ and discount factors of 0.1 gives the smallest WOLP, it does not minimise realised lost demand. This difference is important because WOLP is a deterministic operational proxy, whereas realised lost demand is measured after event-driven demand realisation. The remaining experiments therefore use $\kappa_1^{\text{olp}} = 1$, $\kappa_1^{\text{int}} = 0.5$, $\omega_{h+1}^{\text{olp}} = 0.5\omega_h^{\text{olp}}$, and $\omega_{h+1}^{\text{int}} = 0.5\omega_h^{\text{int}}$.

3.5.4 Effect of rolling-horizon length

Table 31 compares a myopic policy with two look-ahead windows. Moving from $h_{\text{RH}} = 1$ to $h_{\text{RH}} = 3$ reduces WOLP from 0.7325 to 0.6440 and reduces realised lost demand from 5.00 to 4.10. These correspond to reductions of 12.1% and 18.0%, respectively. The same setting also slightly reduces travelled distance. The number of operations increases by 3.6%, which is expected because the look-ahead policy performs more anticipatory rebalancing actions. Increasing the window to $h_{\text{RH}} = 5$ does not improve service and increases the average time per period sharply. Therefore, $h_{\text{RH}} = 3$ is used as the default rolling-horizon length.

Table 31: Impact of rolling-horizon length.

h_{RH}	WOLP	Realised loss	Distance	Operations	Time/period (s)
1	0.7325	5.00	1597.1225	2228.00	21.9350
3	0.6440	4.10	1558.7510	2309.20	8.2674
5	0.6825	5.25	1613.8375	2278.25	967.2100

The results support a moderate look-ahead policy. A one-period policy lacks information about near-future inventory pressure, while a longer window increases computational burden without producing a corresponding service benefit in these tests.

3.5.5 Sensitivity to the CARI interval-width parameter

The next experiment evaluates the CARI interval-width parameter, denoted by z_{CARI} . This parameter controls the conservativeness of the inventory interval. Table 32 shows that $z_{\text{CARI}} = 1.960$ produces the lowest realised lost demand, but its average computation time is 490.95 seconds per period. The value $z_{\text{CARI}} = 1.645$ gives a realised loss of 4.10, only 0.10 higher than the best observed value, while keeping the average computation time at 8.2674 seconds per period. This provides a more favourable service–time trade-off and is selected for the remaining experiments.

The sensitivity results also show that the smallest WOLP does not necessarily produce the smallest realised loss. For example, $z_{\text{CARI}} = 1.440$ gives the lowest WOLP but loses 5.70 demands on average. This confirms that the deterministic proxy should be interpreted together with event-driven service outcomes rather than used as the only tuning criterion.

Table 32: Sensitivity to the CARI interval-width parameter.

z_{CARI}	WOLP	Realised loss	Distance	Operations	Time/period (s)
1.282	0.657	5.40	1585.941	2345.8	6.6019
1.440	0.516	5.70	1585.306	2239.0	8.1002
1.645	0.644	4.10	1558.751	2309.2	8.2674
1.960	1.182	4.00	1535.219	2253.4	490.9500
2.326	0.750	5.10	1541.426	2264.1	744.6088
2.576	0.715	5.80	1590.596	2378.7	1146.8240

3.5.6 Comparison of demand representations

Table 33 compares three ways of representing demand in the rebalancing model. The point target-demand policy performs poorly because it drives the route toward a single inventory level and leaves little flexibility for stochastic imbalance. The benchmark interval policy is operationally cheaper and obtains the smallest WOLP, but it loses 8.0 demands on average. The proposed robust demand interval reduces realised lost demand to 4.1, which is a 48.8% reduction relative to the benchmark interval and a 78.6% reduction relative to the target-demand policy.

Table 33: Comparison of demand representations.

Demand representation	WOLP	Realised loss	Distance	Operations	Time/period (s)
Target demand	13.270	19.20	2618.700	6387.18	1161.8152
Benchmark demand interval	0.295	8.00	1201.030	1557.00	8.9111
Robust demand interval	0.644	4.10	1558.751	2309.20	8.2674

Relative to the target-demand policy, the robust demand interval reduces WOLP by 95.1%, travelled distance by 40.5%, and transfer operations by 63.8%. Compared with the benchmark demand interval, the robust interval requires more travel and more operations, but this additional effort is associated with substantially lower realised lost demand. This is the main computational evidence for using a robust interval rather than a point target or a narrower benchmark interval: it gives the routing policy enough operational flexibility while protecting service against realised demand variation.

3.5.7 Value of active rebalancing

The final experiment compares the selected robust-interval rebalancing policy with a no-rebalancing baseline. As shown in Table 34, without rebalancing the average realised lost demand is 213.9. With the proposed rebalancing policy, this value decreases to 4.1. This corresponds to a 98.1% reduction in realised lost demand.

The magnitude of this difference shows that active rebalancing is essential in the tested 40-station instances. At the same time, the selected rolling-horizon configuration keeps the average computational time per period within a range that is compatible with repeated decision making.

Table 34: Effect of active rebalancing.

Policy	Realised loss	Distance	Operations	Time/period (s)
Robust demand interval with rebalancing	4.10	1558.751	2309.2	8.2674
Without rebalancing	213.90	0.000	0.0	–

3.5.8 Summary of findings

The computational results provide four main insights. First, balancing the WOLP and interval-deviation components leads to better realised service than focusing only on the deterministic proxy. Second, a moderate look-ahead window, $h_{RH} = 3$, improves over the myopic policy without the computational burden observed for $h_{RH} = 5$. Third, the CARI interval-width parameter should be selected using both realised lost demand and computation time; the selected value $z_{CARI} = 1.645$ provides a practical compromise. Fourth, the proposed robust demand interval substantially improves realised service relative to the target-demand policy, the benchmark interval policy, and the no-rebalancing baseline.

3.6 Conclusion

This study developed a dynamic interval-based framework linking censoring-aware demand estimation, service-oriented inventory targets, and rolling-horizon rebalancing. CARI uses a finite-capacity birth–death process to correct observed station-period rates for empty- and full-station censoring, propagates bootstrap uncertainty, and constructs robust demand intervals. A mixed-integer model then coordinates routing, transfers, loads, timing, and inventory evolution. For a fleet, reproducible vehicle-specific clustering and a VNS–MIP-refinement–VNS workflow provide a tractable solution architecture; the MIP formulation uses explicit, instance-specific bounds rather than claiming to be Big-M-free.

On the 20 tested Capital Bikeshare-derived instances, each containing 40 stations, a three-period rolling window gives lower average realised loss than the one-period policy (4.1 versus 5.0) and avoids the high computation time observed with five periods. The robust demand interval records average realised loss of 4.1, compared with 8.0 for the benchmark interval and 19.2 for the point target, while the no-rebalancing baseline records 213.9. These comparisons also reveal trade-offs: the robust interval requires more travel and transfer activity than the benchmark interval, and the lowest deterministic WOLP does not always yield the lowest realised loss.

The findings are conditional on the stated data construction, parameter settings, clustering, solver limits, and event-driven simulation. Replication-level outputs, confidence intervals, stronger dynamic baselines, and held-out operational tests are not available in the present study, so the averages do not establish general statistical superiority. Future work should retain and publish run-level outputs, evaluate additional cities and out-of-sample periods, test alternative censoring and demand models, and compare the complete fleet workflow with stronger rolling-horizon and learning-based baselines.

Chapter 4

Conclusion

The **review** organised a large and fragmented literature through a unified six-tuple formalisation of the BRP and a taxonomy separating planning horizon, demand representation, fleet structure, operational scope, and rebalancing paradigm. Applied to 104 studies, this framework identifies a mature static deterministic stream but weaker reproducibility and cross-paper comparability in dynamic, stochastic, and learning-based research. The resulting agenda motivates the two methodological studies by emphasising richer demand representations, stronger benchmark practice, and closer integration of inventory and routing decisions.

The **static study** addressed a single-vehicle problem with acceptability intervals, repeated visits to stations and the depot, a free initial load, and three rebalancing policies. A common flow-based MILP represents the non-split, without-preemption, and with-preemption cases. The structural theorem proves equality of the with- and without-preemption optimal values under unbounded capacity, non-negative handling times, an uncharged free initial load, and unrestricted terminal load. AILS combines NDP-based transfer optimisation with $Q(\lambda)$ -guided perturbation. It matches all 60 small-instance CPLEX objectives, with 53 certified optima, and matches all 100 published best-known solutions in the literature validation, 99 of which equal published lower bounds. The ten-run ablation gives the complete AILS the lowest average BKD (0.64%) and shows the greatest deterioration when NDP awareness is removed. Larger-instance results establish feasibility, runtime, and stability for the tested implementation, but not optimality where no certificate exists.

The **dynamic study** linked CARI censoring correction, bootstrap rate uncertainty, robust station-period inventory intervals, rolling-horizon mixed-integer optimisation, vehicle-specific clustering, and a VNS–MIP-refinement–VNS workflow. Event-driven experiments were conducted on 20 instances containing 40 stations sampled from 2024 Capital Bikeshare data. In the reported averages, a three-period look-ahead reduces realised loss from 5.0 to 4.1 relative to the one-period policy. The robust interval records realised loss of 4.1, compared with 8.0 for the benchmark interval, 19.2 for a point target, and 213.9 without rebalancing. These comparisons are conditional on the stated model, samples, settings, and simulation design; they are not out-of-sample operational evidence.

Overarching insights and managerial implications

Demand intervals provide the central link between the static and dynamic studies. In the static setting they avoid forcing every station to a single target; in the dynamic setting they translate censored-demand

estimates and service risk into ranges that can be used by the routing model. The results also show that operational flexibility has a cost: repeated visits, NDP-aware evaluation, and anticipatory control can improve the tested objective or realised service, but may require more computation or transfer activity. The appropriate design therefore depends on vehicle capacity, service priorities, and the time available for decision-making.

For practice, three cautious implications follow. First, repeated visits can recover feasibility on capacity-tight instances, although feasibility is not guaranteed for every possible instance. Second, preemption is most relevant when vehicle capacity binds; under the theorem’s unbounded-capacity assumptions it has no value over the without-preemption policy. Third, in the dynamic experiments a moderate look-ahead provides a better service–time trade-off than either a myopic or a longer horizon, but the preferred horizon should be recalibrated for each operating environment.

Limitations

The static model uses one vehicle and deterministic intervals, and its largest tests lack exact certificates. Although the literature benchmark improves external comparability, it represents a restricted special case and does not support a hardware-normalised speed claim. The dynamic evaluation relies on independent Poisson return and rental processes with rates held constant within each period, a synthetic depot, fixed-size 40-station samples, vehicle clusters treated through an outer heuristic, and model-based event simulation. Replication-level outputs, confidence intervals, held-out dates or cities, and comparisons with stronger end-to-end dynamic baselines are not available. The review is also conditioned on its search protocol and cut-off date.

Future research

Future work should extend the static formulation and AILS to coordinated multi-vehicle and heterogeneous fleets, develop stronger exact methods, and derive quantitative bounds on the value of preemption. For the dynamic setting, priorities are to release run-level outputs and code, validate latent-demand estimates out of sample, compare against stronger rolling-horizon and learning-based baselines, and test larger networks and multiple cities. A further direction is to use the static AILS as a fast routing-and-loading subproblem solver within the dynamic framework while retaining uncertainty-aware intervals and operationally realistic dispatch constraints.

Closing remark

By moving from a structured review to static and dynamic formulations linked by demand intervals, this thesis provides a coherent modelling perspective and a set of evaluated algorithmic tools. Its main contribution is not a universal performance claim, but an explicit connection between demand representation, routing flexibility, uncertainty, and the strength of the evidence used to assess each method.

References

- Alvarez-Valdes, Ramon, Jose M. Belenguer, Enrique Benavent, Jose D. Bermudez, Facundo Muñoz, Enriqueta Vercher, and Francisco Verdejo (2016). “Optimizing the level of service quality of a bike-sharing system”. In: *Omega* 62, pp. 163–175.
- Banisharif-Dehkordi, Amir Mohammad, Ali Asghar Alesheikh, and Mohammad Sharif (2025). “Enhancing first- and last-mile connectivity: A study of bike-sharing and ferry service integration in New York city”. In: *Transportation Research Interdisciplinary Perspectives* 32, p. 101537. ISSN: 2590-1982. DOI: <https://doi.org/10.1016/j.trip.2025.101537>. URL: <https://www.sciencedirect.com/science/article/pii/S2590198225002167>.
- Benchimol, Mike, Pascal Benchimol, Benoît Chappert, Arnaud De La Taille, Fabien Laroche, Frédéric Meunier, and Ludovic Robinet (2011). “Balancing the stations of a self service “bike hire” system”. In: *RAIRO-Operations Research-Recherche Opérationnelle* 45.1, pp. 37–61.
- Brinkmann, Jan, Marlin W. Ulmer, and Dirk C. Mattfeld (2019a). “Dynamic Lookahead Policies for Stochastic-Dynamic Inventory Routing in Bike Sharing Systems”. In: *COMPUTERS & OPERATIONS RESEARCH* 106, pp. 260–279. ISSN: 0305-0548. DOI: 10.1016/j.cor.2018.06.004.
- Brinkmann, Jan, Marlin W. Ulmer, and Dirk C. Mattfeld (2019b). “Dynamic Lookahead Policies for Stochastic-Dynamic Inventory Routing in Bike Sharing Systems”. In: *Computers and Operations Research* 106, pp. 260–279.
- Bruck, Bruno P., Walton P. Coutinho, and Pedro Munari (2025). “The Robust Bike sharing Rebalancing Problem: Formulations and a branch-and-cut algorithm”. In: *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 325.1, pp. 67–80. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2025.02.029.
- Bruck, Bruno P., Fabio Cruz, Manuel Iori, and Anand Subramanian (2019a). “The Static Bike Sharing Rebalancing Problem with Forbidden Temporary Operations”. In: *TRANSPORTATION SCIENCE* 53.3, pp. 882–896. ISSN: 0041-1655. DOI: 10.1287/trsc.2018.0859.
- Bruck, Bruno P., Fábio Cruz, Manuel Iori, and Anand Subramanian (2019b). “The Static Bike Sharing Rebalancing Problem with Forbidden Temporary Operations”. In: *Transportation Science* 53, pp. 882–896.

- Bruck, Bruno P. and Anand Subramanian (2023). “Bike-Sharing Rebalancing Problems”. In: *Encyclopedia of Optimization*. Cham: Springer International Publishing, pp. 1–9.
- Bulhoes, Teobaldo, Anand Subramanian, Gunes Erdogan, and Gilbert Laporte (2018). “The static bike relocation problem with multiple vehicles and visits”. In: vol. 264. 2. AUG 02-07, 2015, pp. 508–523. DOI: 10.1016/j.ejor.2017.06.028.
- Bulhões, Teobaldo, Anand Subramanian, Güneş Erdoğan, and Gilbert Laporte (2018a). “The static bike relocation problem with multiple vehicles and visits”. In: *European Journal of Operational Research* 264, pp. 508–523.
- Bulhões, Teobaldo, Anand Subramanian, Güneş Erdoğan, and Gilbert Laporte (2018b). “The static bike relocation problem with multiple vehicles and visits”. In: *European Journal of Operational Research* 264.2, pp. 508–523. DOI: 10.1016/j.ejor.2017.06.028.
- Caggiani, Leonardo, Rosalia Camporeale, Michele Ottomanelli, and Wai Yuen Szeto (2018). “A modeling framework for the dynamic management of free-floating bike-sharing systems”. In: *TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES* 87, pp. 159–182. ISSN: 0968-090X. DOI: 10.1016/j.trc.2018.01.001.
- Cai, Yutong, Ghim Ping Ong, and Qiang Meng (2022a). “Dynamic bicycle relocation problem with broken bicycles”. In: *TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW* 165. ISSN: 1366-5545. DOI: 10.1016/j.tre.2022.102877.
- Cai, Yutong, Ghim Ping Ong, and Qiang Meng (2022b). “Dynamic bicycle relocation problem with broken bicycles”. In: *Transportation Research Part E: Logistics and Transportation Review* 165, p. 102846.
- Cavagnini, Rossana, Francesca Maggioni, Luca Bertazzi, and Mike Hewitt (2024). “A two-stage stochastic programming model for bike-sharing systems with rebalancing”. In: *EURO JOURNAL ON TRANSPORTATION AND LOGISTICS* 13. ISSN: 2192-4376. DOI: 10.1016/j.ejtl.2024.100140.
- Chang, Ximing, Jianjun Wu, Huijun Sun, Goncalo Homem de Almeida Correia, and Jianhua Chen (2021). “Relocating operational and damaged bikes in free-floating systems: A data-driven modeling framework for level of service enhancement”. In: *TRANSPORTATION RESEARCH PART A-POLICY AND PRACTICE* 153, pp. 235–260. ISSN: 0965-8564. DOI: 10.1016/j.tra.2021.09.010.
- Chang, Ximing, Jianjun Wu, Huijun Sun, and Xuedong Yan (2023). “A Smart Predict-then-Optimize method for dynamic green bike relocation in the free-floating system”. In: *TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES* 153. ISSN: 0968-090X. DOI: 10.1016/j.trc.2023.104220.
- Chemla, Daniel, Frederic Meunier, and Roberto Wolfler Calvo (2013a). “Bike sharing systems: Solving the static rebalancing problem”. In: *DISCRETE OPTIMIZATION* 10.2, pp. 120–146. ISSN: 1572-5286. DOI: 10.1016/j.disopt.2012.11.005.

- Chemla, Daniel, Frédéric Meunier, and Roberto Wolfler Calvo (2013b). “Bike sharing systems: Solving the static rebalancing problem”. In: *Discrete Optimization* 10.2, pp. 120–146. DOI: 10.1016/j.disopt.2012.11.005.
- Chen, Mingxuan and W. Y. Szeto (2023a). “A static green bike repositioning problem with heavy and light carriers”. In: *TRANSPORTATION RESEARCH PART D-TRANSPORT AND ENVIRONMENT* 118. ISSN: 1361-9209. DOI: 10.1016/j.trd.2023.103711.
- Chen, Mingxuan and W.Y. Szeto (2023b). “A static green bike repositioning problem with heavy and light carriers”. In: *Transportation Research Part D: Transport and Environment* 118, p. 103683.
- Chen, Qingxin, Chenyi Fu, Ning Zhu, Shoufeng Ma, and Qiao-Chu He (2023). “A target-based optimization model for bike-sharing systems: From the perspective of service efficiency and equity”. In: *TRANSPORTATION RESEARCH PART B-METHODOLOGICAL* 167, pp. 235–260. ISSN: 0191-2615. DOI: 10.1016/j.trb.2022.12.002.
- Chen, Qingxin, Shoufeng Ma, Chenyi Fu, Ning Zhu, and Qiao-Chu He (2025). “A robust satisficing multi-objective optimization approach for bike-sharing systems with heterogeneous user types”. In: *TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW* 202. ISSN: 1366-5545. DOI: 10.1016/j.tre.2025.104265.
- Chen, Qingxin, Shoufeng Ma, Hongming Li, Ning Zhu, and Qiao-Chu He (2024). “Optimizing bike rebalancing strategies in free-floating bike-sharing systems: An enhanced distributionally robust approach”. In: *TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW* 184. ISSN: 1366-5545. DOI: 10.1016/j.tre.2024.103477.
- Cheng, Yao, Junwei Wang, and Yan Wang (2021). “A user-based bike rebalancing strategy for free-floating bike sharing systems: A bidding model”. In: *TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW* 154. ISSN: 1366-5545. DOI: 10.1016/j.tre.2021.102438.
- Contardo, Claudio, Catherine Morency, and Louis-Martin Rousseau (2012). “Balancing a dynamic public bike-sharing system”. In: *CIRRELT*, pp. 1–24.
- Cruz, Fabio, Anand Subramanian, Bruno P. Bruck, and Manuel Iori (2017). “A heuristic algorithm for a single vehicle static bike sharing, rebalancing problem”. In: *COMPUTERS & OPERATIONS RESEARCH* 79, pp. 19–33. ISSN: 0305-0548. DOI: 10.1016/j.cor.2016.09.025.
- Cruz, Fábio, Anand Subramanian, Bruno P. Bruck, and Manuel Iori (2017). “A heuristic algorithm for a single vehicle static bike sharing rebalancing problem”. In: *Computers & Operations Research* 79, pp. 19–33.
- Cui, Jia, W. Y. Szeto, and Yue Wang (2025). “A static mixed bike repositioning problem with both man-powered bikes and e-bikes”. In: *TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW* 203. ISSN: 1366-5545. DOI: 10.1016/j.tre.2025.104275.

- Datner, Sharon, Tal Raviv, Michal Tzur, and Daniel Chemla (2019a). “Setting inventory levels in a bike sharing network”. In: *Transportation Science* 53.1, pp. 62–76.
- Datner, Sharon, Tal Raviv, Michal Tzur, and Daniel Chemla (2019b). “Setting inventory levels in a bike sharing network”. In: *Transportation Science* 53.1, pp. 62–76. DOI: 10.1287/trsc.2017.0790.
- Dell’Amico, Mauro, Eleni Hadjiconstantinou, Manuel Iori, and Stefano Novellani (2014a). “The bike sharing rebalancing problem: Mathematical formulations and benchmark instances”. In: *European Journal of Operational Research* 239, pp. 677–691.
- Dell’Amico, Mauro, Eleni Hadjicostantinou, Manuel Iori, and Stefano Novellani (2014b). “The bike sharing rebalancing problem: Mathematical formulations and benchmark instances”. In: *OMEGA-INTERNATIONAL JOURNAL OF MANAGEMENT SCIENCE* 45, pp. 7–19. ISSN: 0305-0483. DOI: 10.1016/j.omega.2013.12.001.
- Dell’Amico, Mauro, Eleni Hadjicostantinou, Manuel Iori, and Stefano Novellani (2014c). “The bike sharing rebalancing problem: Mathematical formulations and benchmark instances”. In: *Omega* 45, pp. 7–19. DOI: 10.1016/j.omega.2013.12.001.
- Dell’Amico, Mauro, Manuel Iori, Stefano Novellani, and Thomas Stutzle (2016). “A destroy and repair algorithm for the Bike sharing Rebalancing Problem”. In: *COMPUTERS & OPERATIONS RESEARCH* 71, pp. 149–162. ISSN: 0305-0548. DOI: 10.1016/j.cor.2016.01.011.
- Dell’Amico, Mauro, Manuel Iori, Stefano Novellani, and Anand Subramanian (2018a). “The Bike sharing Rebalancing Problem with Stochastic Demands”. In: *TRANSPORTATION RESEARCH PART B-METHODOLOGICAL* 118, pp. 362–380. ISSN: 0191-2615. DOI: 10.1016/j.trb.2018.10.015.
- Dell’Amico, Mauro, Manuel Iori, Stefano Novellani, and Anand Subramanian (2018b). “The bike sharing rebalancing problem with stochastic demands”. In: *Transportation Research Part B: Methodological* 118, pp. 362–380. DOI: 10.1016/j.trb.2018.10.015.
- Dell’Amico, Mauro, Manuel Iori, Stefano Novellani, and Thomas Stützle (2016). “A destroy and repair algorithm for the Bike sharing Rebalancing Problem”. In: *Computers & Operations Research* 71, pp. 149–162.
- Delle Donne, Diego, Alberto Santini, and Claudia Archetti (2025). “Integrating public transport in sustainable last-mile delivery: Column generation approaches”. In: *European Journal of Operational Research* 324.1, pp. 75–91.
- Du, Mingyang, Lin Cheng, Xuefeng Li, and Fang Tang (2020a). “Static rebalancing optimization with considering the collection of malfunctioning bikes in free-floating bike sharing system”. In: *TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW* 141. ISSN: 1366-5545. DOI: 10.1016/j.tre.2020.102012.
- Du, Mingyang, Lin Cheng, Xuefeng Li, and Fang Tang (2020b). “Static rebalancing optimization with considering the collection of malfunctioning bikes in free-floating bike sharing system”. In: *Transportation Research Part E* 141, p. 102047.

- Erdogan, Guenes, Maria Battarra, and Roberto Wolfler Calvo (2015). “An exact algorithm for the static rebalancing problem arising in bicycle sharing systems”. In: *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 245.3, pp. 667–679. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2015.03.043.
- Erdogan, Guenes, Gilbert Laporte, and Roberto Wolfler Calvo (2014). “The static bicycle relocation problem with demand intervals”. In: *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 238.2, pp. 451–457. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2014.04.013.
- Erdoğan, Günes, Gilbert Laporte, and Roberto Wolfler Calvo (2014). “The static bicycle relocation problem with demand intervals”. In: *European Journal of Operational Research* 243, pp. 291–305.
- Erdoğan, Güneş, Maria Battarra, and Roberto Wolfler Calvo (2015). “An exact algorithm for the static rebalancing problem arising in bicycle sharing systems”. In: *European Journal of Operational Research* 245.3, pp. 667–679.
- Erdoğan, Güneş, Gilbert Laporte, and Roberto Wolfler Calvo (2014). “The static bicycle relocation problem with demand intervals”. In: *European Journal of Operational Research* 238.2, pp. 451–457. DOI: 10.1016/j.ejor.2014.04.013.
- Fan, Rui-Na, Quan-Lin Li, Xiaole Wu, and Zhe George Zhang (2025). “Dockless bike-sharing systems with unusable bikes: removing, repair and redistribution under batch policies”. In: *ANNALS OF OPERATIONS RESEARCH* 348.1, pp. 181–219. ISSN: 0254-5330. DOI: 10.1007/s10479-022-05058-8.
- Fishman, Elliot (2016a). “Bikeshare: A review of recent literature”. In: *Transport reviews* 36.1, pp. 92–113.
- Fishman, Elliot (2016b). “Bikeshare: A review of recent literature”. In: *Transport Reviews* 36.1, pp. 92–113. DOI: 10.1080/01441647.2015.1033036.
- Forma, Iris A, Tal Raviv, and Michal Tzur (2015a). “A 3-step math heuristic for the static repositioning problem in bike-sharing systems”. In: *Transportation research part B: methodological* 71, pp. 230–247.
- Forma, Iris A., Tal Raviv, and Michal Tzur (2015b). “A 3-step math heuristic for the static repositioning problem in bike-sharing systems”. In: *Transportation Research Part B: Methodological* 71, pp. 230–247. DOI: 10.1016/j.trb.2014.10.003.
- Fu, Chenyi, Ning Zhu, Shoufeng Ma, and Ronghui Liu (2022). “A two-stage robust approach to integrated station location and rebalancing vehicle service design in bike-sharing systems”. In: *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 298.3, pp. 915–938. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2021.06.014.
- Fu, Chenyi, Ning Zhu, Michael Pinedo, and Shoufeng Ma (2025). “Station-based, free-float, or hybrid: An operating mode analysis of bike-sharing system”. In: *TRANSPORTATION RESEARCH PART B-METHODOLOGICAL* 191. ISSN: 0191-2615. DOI: 10.1016/j.trb.2024.103105.

- Gammelli, Daniele, Inon Peled, Filipe Rodrigues, Dario Pacino, Haci A. Kurtaran, and Francisco C. Pereira (2020). “Estimating latent demand of shared mobility through censored Gaussian processes”. In: *Transportation Research Part C: Emerging Technologies* 120, p. 102775. DOI: 10.1016/j.trc.2020.102775.
- Gammelli, Daniele, Yihua Wang, Dennis Prak, Filipe Rodrigues, Stefan Minner, and Francisco Camara Pereira (2022). “Predictive and prescriptive performance of bike-sharing demand forecasts for inventory management”. In: *TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES* 138. ISSN: 0968-090X. DOI: 10.1016/j.trc.2022.103571.
- Gao, Feng, Shaoying Li, Zhangzhi Tan, Zhifeng Wu, Xiaoming Zhang, Guanping Huang, and Ziwei Huang (2021). “Understanding the modifiable areal unit problem in dockless bike sharing usage and exploring the interactive effects of built environment factors”. In: *INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE* 35.9, pp. 1905–1925. ISSN: 1365-8816. DOI: 10.1080/13658816.2020.1863410.
- Gao, Xuehong and Gyu M. Lee (2019). “Moment-based rental prediction for bicycle-sharing transportation systems using a hybrid genetic algorithm and machine learning”. In: *COMPUTERS & INDUSTRIAL ENGINEERING* 128, pp. 60–69. ISSN: 0360-8352. DOI: 10.1016/j.cie.2018.12.023.
- Gaspero, Luca Di, Andrea Rendl, and Tommaso Urli (2016). “Balancing bike sharing systems with constraint programming”. In: *Constraints* 21, pp. 318–348.
- Ghosh, Supriyo, Pradeep Varakantham, Yuta Adulyasak, and Patrick Jaillet (2017). “Dynamic repositioning to reduce lost demand in bike sharing systems”. In: *Journal of Artificial Intelligence Research* 58, pp. 387–430. DOI: 10.1613/jair.5308.
- Gleditsch, Marte D., Kristine Hagen, Henrik Andersson, Steffen J. Bakker, and Kjetil Fagerholt (2022). “A column generation heuristic for the dynamic bicycle rebalancing problem”. In: *European Journal of Operational Research* 297, pp. 493–505.
- Gleditsch, Marte D., Kristine Hagen, Henrik Andersson, Steffen J. Bakker, and Kjetil Fagerholt (2024a). “A column generation heuristic for the dynamic bicycle rebalancing problem”. In: *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 317.3, pp. 762–775. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2022.07.004.
- Gleditsch, Marte D., Kristine Hagen, Henrik Andersson, Steffen J. Bakker, and Kjetil Fagerholt (2024b). “A column generation heuristic for the dynamic bicycle rebalancing problem”. In: *European Journal of Operational Research* 317.3, pp. 762–775. DOI: 10.1016/j.ejor.2022.07.004.
- Guo, Yuhan, Jinning Li, Linfan Xiao, Hamid Allaoui, Alok Choudhary, and Lufang Zhang (2024a). “Efficient inventory routing for Bike-Sharing Systems: A combinatorial reinforcement learning framework”. In: *Transportation Research Part E: Logistics and Transportation Review* 181, p. 103374.

- Guo, Yuhan, Jinning Li, Linfan Xiao, Hamid Allaoui, Alok Choudhary, and Lufang Zhang (2024b). “Efficient inventory routing for Bike-Sharing Systems: A combinatorial reinforcement learning framework”. In: *TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW* 182. ISSN: 1366-5545. DOI: 10.1016/j.tre.2024.103415.
- Haider, Zulqarnain, Alexander Nikolaev, Jee Eun Kang, and Changhyun Kwon (2018). “Inventory rebalancing through pricing in public bike sharing systems”. In: *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 270.1, pp. 103–117. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2018.02.053.
- Hernandez-Perez, Hipolito, Juan Jose Salazar-Gonzalez, and Beatriz Santos-Hernandez (2018). “Heuristic algorithm for the Split-Demand One-Commodity Pickup-and-Delivery Travelling Salesman Problem”. In: *Computers & Operations Research* 97, pp. 1–17. ISSN: 0305-0548.
- Hernández-Pérez, Hipólito and Juan-José Salazar-González (2004). “A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery”. In: *Discrete Applied Mathematics* 145, pp. 126–139.
- Hernández-Pérez, Hipólito and Juan-José Salazar-González (2022). “A branch-and-cut algorithm for the split-demand one-commodity pickup-and-delivery travelling salesman problem”. In: *European Journal of Operational Research* 297.2, pp. 467–483.
- Ho, Sin C and Wai Yuen Szeto (2017). “A hybrid large neighborhood search for the static multi-vehicle bike-repositioning problem”. In: *Transportation Research Part B: Methodological* 95, pp. 340–363.
- Ho, Sin C. and W. Y. Szeto (2014). “Solving a static repositioning problem in bike-sharing systems using iterated tabu search”. In: *TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW* 69, pp. 180–198. ISSN: 1366-5545. DOI: 10.1016/j.tre.2014.05.017.
- Ho, Sin C. and W. Y. Szeto (2017a). “A hybrid large neighborhood search for the static multi-vehicle bike-repositioning problem”. In: *TRANSPORTATION RESEARCH PART B-METHODOLOGICAL* 95, pp. 340–363. ISSN: 0191-2615. DOI: 10.1016/j.trb.2016.11.003.
- Ho, Sin C. and W. Y. Szeto (2017b). “A hybrid large neighborhood search for the static multi-vehicle bike-repositioning problem”. In: *Transportation Research Part B: Methodological* 95, pp. 340–363. DOI: 10.1016/j.trb.2016.11.003.
- Ho, Sin C. and W.Y. Szeto (2017c). “A hybrid large neighborhood search for the static multi-vehicle bike-repositioning problem”. In: *Transportation Research Part B* 95, pp. 340–363.
- Hu, Runqiu, W. Y. Szeto, and Sin C. Ho (2025). “Repositioning in bike sharing systems with broken bikes considering on-site repairs”. In: *TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW* 201. ISSN: 1366-5545. DOI: 10.1016/j.tre.2025.104155.

- Hua, Mingzhuang, Xuewu Chen, Jingxu Chen, Di Huang, and Long Cheng (2022). “Large-scale dockless bike sharing repositioning considering future usage and workload balance”. In: *Physica A* 605, p. 127991.
- Huang, Di, Xinyuan Chen, Zhiyuan Liu, Cheng Lyu, Shuaian Wang, and Xuewu Chen (2020). “A static bike repositioning model in a hub-and-spoke network framework”. In: *TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW* 141. ISSN: 1366-5545. DOI: 10.1016/j.tre.2020.102031.
- Huang, Ganxiang and Di Xu (2023). “The last mile matters: Impact of dockless bike-sharing services on traffic congestion”. In: *Transportation research part D: transport and environment* 121, p. 103836.
- Ikonen, Teemu J., Keijo Heljanko, and Iiro Harjunoski (2025). “A generalized disjunctive programming model for the static bike sharing rebalancing problem with demand intervals”. In: *OPTIMIZATION AND ENGINEERING* 26.3, pp. 1781–1813. ISSN: 1389-4420. DOI: 10.1007/s11081-024-09942-z.
- Ikonen, Tero J., Keijo Heljanko, and Iiro Harjunoski (2024). “A generalized disjunctive programming model for the static bike rebalancing problem with demand intervals”. In: *Optimization and Engineering*, pp. 1–30.
- Jiang, Zhoutong and Yanfeng Ouyang (2022). “Pricing and resource allocation under competition in a docked bike-sharing market”. In: *TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES* 143. ISSN: 0968-090X. DOI: 10.1016/j.trc.2022.103833.
- Jimenez-Merono, Enrique and Francesc Soriguera (2024). “Optimization of bike-sharing repositioning operations: A reactive real-time approach”. In: *EURO JOURNAL ON TRANSPORTATION AND LOGISTICS* 13. ISSN: 2192-4376. DOI: 10.1016/j.ejtl.2024.100138.
- Kadri, Ahmed Abdelmoumene, Imed Kacem, and Karim Labadi (2016a). “A branch-and-bound algorithm for solving the static rebalancing problem in bicycle-sharing systems”. In: *COMPUTERS & INDUSTRIAL ENGINEERING* 95, pp. 41–52. ISSN: 0360-8352. DOI: 10.1016/j.cie.2016.02.002.
- Kadri, Ahmed Abdelmoumene, Imed Kacem, and Karim Labadi (2016b). “A branch-and-bound algorithm for solving the static rebalancing problem in bicycle-sharing systems”. In: *Computers & Industrial Engineering* 95, pp. 41–52.
- Kaltenbrunner, Andreas, Rodrigo Meza, Jens Grivolla, Joan Codina, and Rafael Banchs (2010). “Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system”. In: *Pervasive and Mobile Computing* 6.4, pp. 455–466.
- Kaspi, Mor, Tal Raviv, and Michal Tzur (2016). “Detection of unusable bicycles in bike-sharing systems”. In: *Omega* 65, pp. 10–16.
- Laporte, Gilbert, Frederic Meunier, and Roberto Wolfler Calvo (2018a). “Shared mobility systems: an updated survey”. In: *ANNALS OF OPERATIONS RESEARCH* 271.1, pp. 105–126. ISSN: 0254-5330. DOI: 10.1007/s10479-018-3076-8.

- Laporte, Gilbert, Frédéric Meunier, and Roberto Wolfler Calvo (2018b). “Shared mobility systems: an updated survey”. In: *Annals of Operations Research* 271.1, pp. 105–126. DOI: 10.1007/s10479-018-3076-8.
- Legros, Benjamin (2019a). “Dynamic repositioning strategy in a bike-sharing system; how to prioritize and how to rebalance a bike station”. In: *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 272.2, pp. 740–753. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2018.06.051.
- Legros, Benjamin (2019b). “Dynamic repositioning strategy in a bike-sharing system; how to prioritize and how to rebalance a bike station”. In: *European Journal of Operational Research* 272.2, pp. 740–753. DOI: 10.1016/j.ejor.2018.06.051.
- Li, Xiang, Xianzhe Wang, and Ziyang Feng (2024a). “Dynamic repositioning in bike-sharing systems with uncertain demand: An improved rolling horizon framework”. In: *OMEGA-INTERNATIONAL JOURNAL OF MANAGEMENT SCIENCE* 126. ISSN: 0305-0483. DOI: 10.1016/j.omega.2024.103047.
- Li, Xiang, Xianzhe Wang, and Ziyang Feng (2024b). “Dynamic repositioning in bike-sharing systems with uncertain demand: An improved rolling horizon framework”. In: *Omega* 126, p. 103047. DOI: 10.1016/j.omega.2024.103047.
- Li, Xinyu, Yang Xu, Xiaohu Zhang, Wenzhong Shi, Yang Yue, and Qingquan Li (2023). “Improving short-term bike sharing demand forecast through an irregular convolutional neural network”. In: *TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES* 147. ISSN: 0968-090X. DOI: 10.1016/j.trc.2022.103984.
- Li, Yanfeng and Yang Liu (2021). “The static bike rebalancing problem with optimal user incentives”. In: *TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW* 146. ISSN: 1366-5545. DOI: 10.1016/j.tre.2020.102216.
- Li, Yanfeng, W. Y. Szeto, Jiancheng Long, and C. S. Shui (2016). “A multiple type bike repositioning problem”. In: *TRANSPORTATION RESEARCH PART B-METHODOLOGICAL* 90, pp. 263–278. ISSN: 0191-2615. DOI: 10.1016/j.trb.2016.05.010.
- Li, Yexin, Yu Zheng, and Qiang Yang (2018). “Dynamic Bike Reposition: A Spatio-Temporal Reinforcement Learning Approach”. In: AUG 19-23, 2018, pp. 1724–1733. DOI: 10.1145/3219819.3220110.
- Li, Yiming and Zhuo Sun (2023). “Repositioning shared bikes under low-carbon policies”. In: *COMPUTERS & INDUSTRIAL ENGINEERING* 184. ISSN: 0360-8352. DOI: 10.1016/j.cie.2023.109594.
- Liang, Jiaqi, Sanjay Dominik Jena, and Andrea Lodi (2024a). “Dynamic rebalancing optimization for bike-sharing systems: A modeling framework and empirical comparison”. In: *European Journal of Operational Research* 312, pp. 863–880.
- Liang, Jiaqi, Sanjay Dominik Jena, and Andrea Lodi (2024b). “Dynamic rebalancing optimization for bike-sharing systems: A modeling framework and empirical comparison”. In: *European Journal of Operational Research* 317.3, pp. 875–889. DOI: 10.1016/j.ejor.2024.04.037.

- Liang, Jiaqi, Maria Clara Martins Silva, Daniel Aloise, and Sanjay Dominik Jena (2024c). “Dynamic rebalancing for Bike-sharing systems under inventory interval and target predictions”. In: *EURO Journal on Transportation and Logistics* 13, p. 100104.
- Liang, Jiaqi, Maria Clara Martins Silva, Daniel Aloise, and Sanjay Dominik Jena (2024d). “Dynamic rebalancing for Bike-sharing systems under inventory interval and target predictions”. In: *EURO JOURNAL ON TRANSPORTATION AND LOGISTICS* 13. ISSN: 2192-4376. DOI: 10.1016/j.ejtl.2024.100147.
- Liu, Junming, Weiwei Chen, and Leilei Sun (2025). “A Data-Driven Optimization Framework for Static Rebalancing Operations in Bike Sharing Systems”. In: *INFORMS JOURNAL ON COMPUTING* 37.5. ISSN: 1091-9856. DOI: 10.1287/ijoc.2022.0182.
- Liu, Yang, Tao Feng, Zhuangbin Shi, and Mingwei He (2022). “Understanding the route choice behaviour of metro-bikeshare users”. In: *TRANSPORTATION RESEARCH PART A-POLICY AND PRACTICE* 166, pp. 460–475. ISSN: 0965-8564. DOI: 10.1016/j.tra.2022.11.006.
- Liu, Ying, W. Y. Szeto, and Sin C. Ho (2018). “A static free-floating bike repositioning problem with multiple heterogeneous vehicles, multiple depots, and multiple visits”. In: *TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES* 92, pp. 208–242. ISSN: 0968-090X. DOI: 10.1016/j.trc.2018.02.008.
- Lourengo, Helena R., Olivier C. Martin, and Thomas Stützle (2003). “Iterated Local Search”. In: *Handbook of Metaheuristics*. Ed. by Fred Glover and Gary A. Kochenberger. Boston, MA: Springer US, pp. 320–353. ISBN: 978-0-306-48056-0. DOI: 10.1007/0-306-48056-5_11. URL: https://doi.org/10.1007/0-306-48056-5_11.
- Lu, Chen, Linjie Gao, and Yuqiao Huang (2023). “Exploring travel patterns and static rebalancing strategies for dockless bike-sharing systems from multi-source data: a framework and case study”. In: *TRANSPORTATION LETTERS-THE INTERNATIONAL JOURNAL OF TRANSPORTATION RESEARCH* 15.4, pp. 336–349. ISSN: 1942-7867. DOI: 10.1080/19427867.2022.2051798.
- Luo, Hao, Fu Zhao, Wei-Qiang Chen, and Hua Cai (2020). “Optimizing bike sharing systems from the life cycle greenhouse gas emissions perspective”. In: *TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES* 117. ISSN: 0968-090X. DOI: 10.1016/j.trc.2020.102705.
- Luo, Xue, Li Li, Lei Zhao, and Jianfeng Lin (2022a). “Dynamic intra-cell repositioning in free-floating bike-sharing systems using approximate dynamic programming”. In: *Transportation Science* 56.4, pp. 799–826. DOI: 10.1287/trsc.2021.1122.
- Luo, Xue, Li Li, Lei Zhao, and Jianfeng Linc (2022b). “Dynamic Intra-Cell Repositioning in Free-Floating Bike-Sharing Systems Using Approximate Dynamic Programming”. In: *TRANSPORTATION SCIENCE* 56.4, pp. 799–826. ISSN: 0041-1655. DOI: 10.1287/trsc.2021.1122.
- Lv, Chang, Qiong Liu, Chaoyong Zhang, Yaping Ren, and Hao Zhou (2024). “A feature correlation reinforce clustering and evolutionary algorithm for the green bike-sharing reposition problem”.

- In: *COMPUTERS & OPERATIONS RESEARCH* 166. ISSN: 0305-0548. DOI: 10.1016/j.cor.2024.106627.
- Lv, Chang, Chaoyong Zhang, Kunlei Lian, Yaping Ren, and Leilei Meng (2020). “A hybrid algorithm for the static bike-sharing re-positioning problem based on an effective clustering strategy”. In: *TRANSPORTATION RESEARCH PART B-METHODOLOGICAL* 140, pp. 1–21. ISSN: 0191-2615. DOI: 10.1016/j.trb.2020.07.004.
- Lv, Chang, Chaoyong Zhang, Kunlei Lian, Yaping Ren, and Leilei Meng (2022). “A two-echelon fuzzy clustering based heuristic for large-scale bike sharing repositioning problem”. In: *TRANSPORTATION RESEARCH PART B-METHODOLOGICAL* 160, pp. 54–75. ISSN: 0191-2615. DOI: 10.1016/j.trb.2022.04.003.
- Ma, Guanhua, Bowen Zhang, Changjing Shang, and Qiang Shen (2021). “Rebalancing stochastic demands for bike-sharing networks with multi-scenario characteristics”. In: *INFORMATION SCIENCES* 554, pp. 177–197. ISSN: 0020-0255. DOI: 10.1016/j.ins.2020.12.044.
- Maggioni, Francesca, Matteo Cagnolari, Luca Bertazzi, and Stein W. Wallace (2019a). “Stochastic optimization models for a bike-sharing problem with transshipment”. In: *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 276.1, pp. 272–283. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2018.12.031.
- Maggioni, Francesca, Matteo Cagnolari, Luca Bertazzi, and Stein W. Wallace (2019b). “Stochastic optimization models for a bike-sharing problem with transshipment”. In: *European Journal of Operational Research* 276.1, pp. 272–283. DOI: 10.1016/j.ejor.2018.12.031.
- Mladenović, Nenad and Pierre Hansen (1997). “Variable neighborhood search”. In: *Computers & Operations Research* 24.11, pp. 1097–1100. DOI: 10.1016/S0305-0548(97)00031-2.
- Negahban, Ashkan (2019). “Simulation-based estimation of the real demand in bike-sharing systems in the presence of censoring”. In: *European Journal of Operational Research* 277.1, pp. 317–332. DOI: 10.1016/j.ejor.2019.02.013.
- Neumann-Saavedra, Bruno A and Rossana Cavagnini (2024). “Column-generation-based heuristics for integrating static rebalancing and faulty bike collection in bike-sharing systems”. In: *OR Spectrum*, pp. 1–35.
- Neumann-Saavedra, Bruno A. and Rossana Cavagnini (2025). “Column-generation-based heuristics for integrating static rebalancing and faulty bike collection in bike-sharing systems”. In: *OR SPECTRUM* 47.2, pp. 375–409. ISSN: 0171-6468. DOI: 10.1007/s00291-024-00789-5.
- Nowak, Maciek, ozlem Ergun, and Chelsea C. White III (2008). “Pickup and delivery with split loads”. In: *Transportation Science* 42, pp. 32–43.
- Ntamwiza, Jean Marie Vianney, Hannibal Bwire, and Alphonse Nkurunziza (2026). “Mapping a decade of bike-sharing research published between (2015–2025): A systematic and bibliometric review highlighting gaps and future trends”. In: *Transport Economics and Management*.

- Pal, Aritra and Yu Zhang (2017). “Free-floating bike sharing: Solving real-life large-scale static rebalancing problems”. In: *TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES* 80, pp. 92–116. ISSN: 0968-090X. DOI: 10.1016/j.trc.2017.03.016.
- Pan, Xiaoyi, Jinjun Tang, Tianjian Yu, Jianming Cai, Yong Xiong, and Fan Gao (2023). “Reposition optimization in the free-floating bike-sharing system considering transferring travels from urban rail transit”. In: *COMPUTERS & INDUSTRIAL ENGINEERING* 178. ISSN: 0360-8352. DOI: 10.1016/j.cie.2023.109127.
- Parada, Lucas, Jean-Francois Cote, and Michel Gendreau (2026). “A disaggregated integer L-shaped method for the bike sharing rebalancing problem with stochastic demands”. In: *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 329.2, pp. 436–446. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2025.07.041.
- Qiao, Jian, Mengying He, Niannian Sun, Pengfei Sun, and Ying Fan (2023). “Factors affecting the final solution of the bike-sharing rebalancing problem under heuristic algorithms”. In: *COMPUTERS & OPERATIONS RESEARCH* 159. ISSN: 0305-0548. DOI: 10.1016/j.cor.2023.106368.
- Raviv, Tal and Ofer Kolka (2013a). “Optimal inventory management of a bike-sharing station”. In: *Iie Transactions* 45.10, pp. 1077–1093.
- Raviv, Tal and Ofer Kolka (2013b). “Optimal inventory management of a bike-sharing station”. In: *IIE Transactions* 45.10, pp. 1077–1093. DOI: 10.1080/0740817X.2013.770186.
- Raviv, Tal, Michal Tzur, and Iris A. Forma (2013a). “Static repositioning in a bike-sharing system: models and solution approaches”. In: *EURO JOURNAL ON TRANSPORTATION AND LOGISTICS* 2.3, pp. 187–229. ISSN: 2192-4376. DOI: 10.1007/s13676-012-0017-6.
- Raviv, Tal, Michal Tzur, and Iris A. Forma (2013b). “Static repositioning in a bike-sharing system: models and solution approaches”. In: *EURO Journal on Transportation and Logistics* 2, pp. 25–37.
- Raviv, Tal, Michal Tzur, and Iris A. Forma (2013c). “Static repositioning in a bike-sharing system: models and solution approaches”. In: *EURO Journal on Transportation and Logistics* 2.3, pp. 187–229. DOI: 10.1007/s13676-012-0017-6.
- Regue, Robert and Will Recker (2014). “Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem”. In: *TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW* 72, pp. 192–209. ISSN: 1366-5545. DOI: 10.1016/j.tre.2014.10.005.
- Ricci, Miriam (2015). “Bike sharing: A review of evidence on impacts and processes of implementation and operation”. In: *Research in Transportation Business & Management* 15, pp. 28–38.
- Salazar-González, Juan-José and Beatriz Santos-Hernández (2017). “The one-commodity pickup-and-delivery traveling salesman problem with time windows”. In: *European Journal of Operational Research* 257, pp. 815–828.

- Schuijbroek, J., R. C. Hampshire, and W. J. van Hoes (2017). "Inventory rebalancing and vehicle routing in bike sharing systems". In: *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH* 257.3, pp. 992–1004. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2016.08.029.
- Schuijbroek, Jasper, Robert C. Hampshire, and Willem-Jan van Hoes (2017). "Inventory rebalancing and vehicle routing in bike sharing systems". In: *European Journal of Operational Research* 257.3, pp. 992–1004. DOI: 10.1016/j.ejor.2016.08.029.
- Shi, Ziyi, Meng Xu, Yancun Song, and Zheng Zhu (2024). "Multi-Platform dynamic game and operation of hybrid Bike-Sharing systems based on reinforcement learning". In: *TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW* 181. ISSN: 1366-5545. DOI: 10.1016/j.tre.2023.103374.
- Shu, Jia, Mabel C Chou, Qizhang Liu, Chung-Piaw Teo, and I-Lin Wang (2013). "Models for effective deployment and redistribution of bicycles within public bicycle-sharing systems". In: *Operations Research* 61.6, pp. 1346–1359.
- Shui, C. S. and W. Y. Szeto (2018). "Dynamic green bike repositioning problem - A hybrid rolling horizon artificial bee colony algorithm approach". In: *TRANSPORTATION RESEARCH PART D-TRANSPORT AND ENVIRONMENT* 60, pp. 119–136. ISSN: 1361-9209. DOI: 10.1016/j.trd.2017.06.023.
- Shui, C. S. and W. Y. Szeto (2020a). "A review of bicycle-sharing service planning problems". In: *TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES* 117. ISSN: 0968-090X. DOI: 10.1016/j.trc.2020.102648.
- Shui, C. S. and W. Y. Szeto (2020b). "A review of bicycle-sharing service planning problems". In: *Transportation Research Part C: Emerging Technologies* 117, p. 102648. DOI: 10.1016/j.trc.2020.102648.
- Si, Hongyun, Jian-gang Shi, Guangdong Wu, Jindao Chen, and Xianbo Zhao (2019). "Mapping the bike sharing research published from 2010 to 2018: A scientometric review". In: *Journal of cleaner production* 213, pp. 415–427.
- Silva, Maria Clara Martins, Daniel Aloise, and Sanjay Dominik Jena (2024). "Data-driven prioritization strategies for inventory rebalancing in bike-sharing systems". In: *OMEGA-INTERNATIONAL JOURNAL OF MANAGEMENT SCIENCE* 129. ISSN: 0305-0483. DOI: 10.1016/j.omega.2024.103141.
- Singla, Adish, Marco Santoni, Gabor Bartok, Pratik Mukerji, Moritz Meenen, and Andreas Krause (2015). "Incentivizing Users for Balancing Bike Sharing Systems". In: JAN 25-30, 2015, pp. 723–729.
- Soares Machado, Claudia A., Nicolas Patrick Marie de Salles Hue, Fernando Tobal Berssaneti, and Jose Alberto Quintanilha (2018). "An Overview of Shared Mobility". In: *SUSTAINABILITY* 10.12. DOI: 10.3390/su10124342.
- Sutton, Richard S. and Andrew G. Barto (1998). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

- Szeto, W. Y., Ying Liu, and Sin C. Ho (2016). “Chemical reaction optimization for solving a static bike repositioning problem”. In: *TRANSPORTATION RESEARCH PART D-TRANSPORT AND ENVIRONMENT* 47, pp. 104–135. ISSN: 1361-9209. DOI: 10.1016/j.trd.2016.05.005.
- Szeto, W. Y. and C. S. Shui (2018). “Exact loading and unloading strategies for the static multi-vehicle bike repositioning problem”. In: *TRANSPORTATION RESEARCH PART B-METHODOLOGICAL* 109, pp. 176–211. ISSN: 0191-2615. DOI: 10.1016/j.trb.2018.01.007.
- Szeto, W.Y. and C.S. Shui (2018). “Exact loading and unloading strategies for the static multi-vehicle bike repositioning problem”. In: *Transportation Research Part B* 109, pp. 176–211.
- Tang, Jinjun, Maoxin Ren, Ziyue Yuan, Jianming Cai, and Yunyi Liang (2025). “System-Wide Optimization of Free-Floating Bike-Sharing for Urban Rail Stations: A demand prediction and scheduling approach”. In: *COMPUTERS & INDUSTRIAL ENGINEERING* 204. ISSN: 0360-8352. DOI: 10.1016/j.cie.2025.111121.
- Tian, Zihao, Jing Zhou, W. Y. Szeto, Lixin Tian, and Wenbin Zhang (2020). “The rebalancing of bike-sharing system under flow-type task window”. In: *TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES* 112, pp. 1–27. ISSN: 0968-090X. DOI: 10.1016/j.trc.2020.01.015.
- Vallez, Carlos M., Mario Castro, and David Contreras (2021). “Challenges and Opportunities in Dock-Based Bike-Sharing Rebalancing: A Systematic Review”. In: *SUSTAINABILITY* 13.4. DOI: 10.3390/su13041829.
- Walker, Awnalisa and Soongeol Kwon (2024). “Risk-averse two-stage stochastic programming for the inventory rebalancing of bike-sharing systems”. In: *INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH* 31.2, pp. 749–779. ISSN: 0969-6016. DOI: 10.1111/itor.13388.
- Wang, Junwei and Yan Wang (2021). “A two-stage incentive mechanism for rebalancing free-floating bike sharing systems: Considering user preference”. In: *TRANSPORTATION RESEARCH PART F-TRAFFIC PSYCHOLOGY AND BEHAVIOUR* 82, pp. 54–69. ISSN: 1369-8478. DOI: 10.1016/j.trf.2021.08.003.
- Wang, Y. and W.Y. Szeto (2018). “Static green repositioning in bike sharing systems with broken bikes”. In: *Transportation Research Part D* 65, pp. 438–457.
- Wang, Yue and W. Y. Szeto (2018). “Static green repositioning in bike sharing systems with broken bikes”. In: *TRANSPORTATION RESEARCH PART D-TRANSPORT AND ENVIRONMENT* 65, pp. 438–457. ISSN: 1361-9209. DOI: 10.1016/j.trd.2018.09.016.
- Wang, Yue and W. Y. Szeto (2021a). “An enhanced artificial bee colony algorithm for the green bike repositioning problem with broken bikes”. In: *TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES* 125. ISSN: 0968-090X. DOI: 10.1016/j.trc.2020.102895.

- Wang, Yue and W.Y. Szeto (2021b). “An enhanced artificial bee colony algorithm for the green bike repositioning problem with broken bikes”. In: *Transportation Research Part C: Emerging Technologies* 129, p. 103259.
- Wang, Yue and WY Szeto (2021). “The dynamic bike repositioning problem with battery electric vehicles and multiple charging technologies”. In: *Transportation Research Part C: Emerging Technologies* 131, p. 103327.
- Warrington, Joseph and Dominik Ruchti (2019). “Two-stage stochastic approximation for dynamic rebalancing of shared mobility systems”. In: *TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES* 104, pp. 110–134. ISSN: 0968-090X. DOI: 10.1016/j.trc.2019.04.021.
- Wu, Aigui, Tong Shu, and Shouyang Wang (2024). “Joint optimization of the inventory routing problem considering the recycling of broken bikes in the bike-sharing system”. In: *RAIRO-OPERATIONS RESEARCH* 58.4, pp. 3069–3091. ISSN: 0399-0559. DOI: 10.1051/ro/2024109.
- Zhang, Dezhi, Wei Xu, Bin Ji, Shuangyan Li, and Yajie Liu (2020). “An adaptive tabu search algorithm embedded with iterated local search and route elimination for the bike repositioning and recycling problem”. In: *COMPUTERS & OPERATIONS RESEARCH* 123. ISSN: 0305-0548. DOI: 10.1016/j.cor.2020.105035.
- Zhang, Dong, Chuhang Yu, Jitamitra Desai, H. Y. K. Lau, and Sandeep Srivathsan (2017a). “A time-space network flow approach to dynamic repositioning in bicycle sharing systems”. In: *TRANSPORTATION RESEARCH PART B-METHODOLOGICAL* 103, pp. 188–207. ISSN: 0191-2615. DOI: 10.1016/j.trb.2016.12.006.
- Zhang, Dong, Chuhang Yu, Jitamitra Desai, HYK Lau, and Sandeep Srivathsan (2017b). “A time-space network flow approach to dynamic repositioning in bicycle sharing systems”. In: *Transportation research part B: methodological* 103, pp. 188–207.
- Zhang, J., M. Meng, and Z. W. Wang David (2019). “A dynamic pricing scheme with negative prices in dockless bike sharing systems”. In: *TRANSPORTATION RESEARCH PART B-METHODOLOGICAL* 127, pp. 201–224. ISSN: 0191-2615. DOI: 10.1016/j.trb.2019.07.007.
- Zhang, Runhao, Chi Xie, and Daniel Zhuoyu Long (2025). “A data-driven robust optimization model for repositioning problem in bike-sharing systems”. In: *TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW* 201. ISSN: 1366-5545. DOI: 10.1016/j.tre.2025.104192.
- Zhao, Rui, Zihao Tian, Lixin Tian, Wenshan Liu, and David Z. W. Wang (2025). “Research on rebalancing of large-scale bike-sharing system driven by zonal heterogeneity and demand uncertainty”. In: *TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES* 170. ISSN: 0968-090X. DOI: 10.1016/j.trc.2024.104933.
- Zhou, Jibiao, Yanyong Guo, Jian Sun, Erze Yu, and Rui Wang (2022). “Review of bike-sharing system studies using bibliometrics method”. In: *JOURNAL OF TRAFFIC*

AND TRANSPORTATION ENGINEERING-ENGLISH EDITION 9.4, pp. 608–630. ISSN: 2095-7564. DOI: 10.1016/j.jtte.2021.08.003.