UNIVERSITÀ DI PARMA

UNIVERSITÀ DEGLI STUDI DI PARMA

DOTTORATO DI RICERCA IN
"TECNOLOGIE DELL'INFORMAZIONE"

CICLO XXXVIII

Designing and evaluating bicycle networks from a sustainable urban mobility perspective

Cordinatore e tutore:
Chiar.mo Prof. Marco Locatelli

Co-Tutore:
Chiar.mo Prof. Anand Subramanian

Dottorando: Rafael Maranhão Rego Praxedes

Anni Accademici 2022/2023 - 2024/2025

ABSTRACT

The promotion of sustainable transportation modes, such as cycling, has become a crucial aspect of urban planning and design. Designing an efficient bicycle network is a complex optimization problem that requires the analysis of various factors, including infrastructure layout, connectivity, safety, and accessibility. Moreover, the fastest routes are not always the preferred options for cyclists, who may consider multiple criteria when choosing routes from their origins to their destinations. Therefore, this research aims to propose an optimization strategy for evaluating and designing bicycle networks, increasing, consequently, bicycle use in the cities. The proposed strategy is validated through experiments using real-world data collected from the bicycle network in the city of Parma, Italy.

The problem addressed in this research can be divided into the following two parts. First, we conduct an analysis to identify cyclists' preferences regarding a set of road characteristics, namely route length, safety, and practicability. Based on these preferences, cyclists can be categorized into different profiles, each associated with specific weights they assign to these characteristics when selecting routes. To this end, we propose two mathematical formulations and corresponding algorithms, depending on the type of data used in the identification process. One formulation is applied when cyclist flow data are known. These flows can be collected, for instance, from cameras across the city. The other formulation is used when cyclists' paths for a set of origin and destination pairs are available. This type of information can be obtained, for example, through bike-sharing services. The proposed formulations and algorithms were validated using both randomly generated data and real-world data from Parma.

Next, given a set of possible interventions in the bicycle network (e.g., constructing new bicycle lanes, improving the quality of existing ones, etc.) and the cyclist profiles identified in the first part of the problem, we select the best combination of interventions that minimizes the overall cost perceived by the cyclists traveling between different origins and destinations. The selection of interventions is subject to demand fulfillment and budget constraints. To address this problem, we propose a mixed-integer linear programming formulation and a set of optimization methods, including a branch-and-bound algorithm, two heuristics based on solving knapsack problems using dynamic programming, and an enumeration mechanism. Computational experiments were conducted using both randomly generated and real-world instances, the latter based on data collected from Parma. The city was represented as a graph comprising more than 40,000 nodes and 95,000 arcs. An intervention set of 10 intervention types was defined, with each type including multiple interventions affecting different regions of the city. The results demonstrated that the proposed approaches efficiently solve the studied problem and provide valuable managerial insights.

Finally, as an ongoing work, we aim to develop a strategy to analyze the impact of the proposed interventions on motorized vehicle users. Given the previously identified cyclist profiles in Parma and the set of possible interventions to the city's cycling network, the goal is to determine the most effective interventions to implement, considering not only the overall cost perceived by cyclists but also the effects on motorized traffic.

Keywords: Bicycle network design. Identification of cyclists' route choice criteria. Combinatorial optimization. Parma cycling network.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This research proposes the application of optimization strategies to address two key problems related to bicycle networks. The first concerns the evaluation of cycling networks by identifying cyclists' route choice behavior. The second, based on the estimated behavior, focuses on improving the quality of existing bicycle networks or designing new ones where they do not yet exist, to better accommodate cyclists' preferences and promote increased bicycle use. In both cases, the network is modeled as a graph, where arcs represent road segments and nodes correspond to intersections.

The goal of the first problem is to understand how bicycle users choose their routes. Indeed, they are not merely concerned with minimizing travel distance, but are also influenced by other factors, such as road safety and environmental considerations. Hence, the problem consists in identifying the cyclists' preferences (referred to as cyclists' weights) for a set of road characteristics, including distance, safety, and practicability, and in determining the fraction of users associated with each weight. As a result, the costs of the arcs in the graph representing the network are expressed as a convex combination of costs associated with the considered road characteristics, called basic costs, weighted by cyclists' preferences for those characteristics. Note that the identification is based either on traffic flow observations on (a subset of) the network arcs or on the knowledge of a set of paths followed by a sample of users.

The second problem refers to selecting, from a set of possible interventions that can be applied to the cycling network, those that minimize the perceived cost for cyclists. Examples of these interventions include building new bicycle lanes or paths, improving the pavement quality of existing lanes, and other similar actions. This selection considers the set of road features and cyclists' preferences for such features estimated on the previous problem, subject to budget and demand constraints.

The relevance of studying cycling networks is evidenced by the extensive literature on the topic. In recent years, cycling has become one of the most widely used sustainable modes of transport, as it produces no pollution and contributes to improving quality of life [72]. Consequently, research on cycling network design is essential for governments and companies aiming to implement well-planned infrastructure within real-world constraints.

Example of works addressing problems related to bicycle networks can be found in Laporte et al. [61, 62], who surveyed the literature on station location, fleet dimensioning, station sizing, rebalancing incentives, and vehicle repositioning in station-based car and bicycle sharing systems. Other surveys, such as those by Buehler and Dill [15] and Cueva et al. [25], focused on cycling levels and on the key factors influencing the design of bike-sharing systems, respectively. In addition, an important classification was proposed by Shui and Szeto [106], who categorized all problems related to bicycle-sharing systems, referred to as bicycle-sharing service planning problems, into three levels: strategic, tactical, and operational (see Figure 1.1).

The strategic level focuses on long-term decisions regarding network infrastructure, such as bikeway design, bicycle station planning, and fleet sizing. The tactical level involves decisions related to maintaining the efficiency of the bicycle-sharing system, while the operational level concerns its daily operations. One example of an optimization problem related to the tactical and operational levels is the bicycle relocation problem [14], which is considered tactical when dealing with static relocation, and operational when the decisions to move bicycles among the pickup and drop-off points are made dynamically.

Given the above, both problems addressed in this research belong to the strategic level of bicycle-sharing service planning, as they are related to the infrastructural planning of bicycle networks. Therefore, the main contributions of this research are

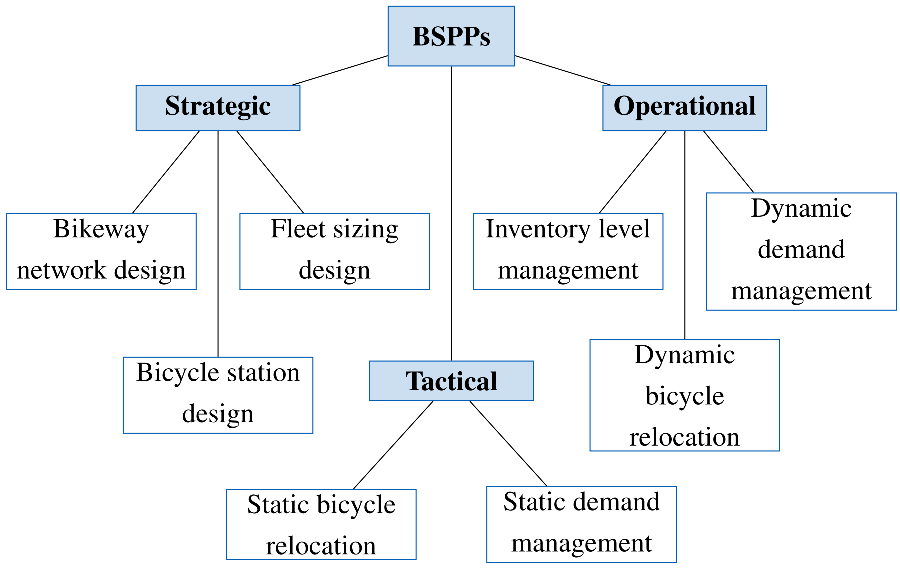


Figure 1.1: Classification schema proposed by Shui and Szeto [106].

presented separately for each problem, with the contributions related to the first problem listed as follows.

- For the scenario where identification is based on traffic flow observations, we propose a flow-based mathematical formulation that minimizes the error between calculated and observed arc flows. In this model, cyclists' weights are estimated not from route choice observations, but considering only observed flows.

- For the scenario where identification is based on the knowledge of a set of paths, we introduce a trajectory-based mathematical formulation that minimizes the cost difference between the observed routes and the minimum-cost paths for each OD pair.

- We extend the flow-based and path-based formulations by introducing a stochastic component into the definition of the basic costs, to account for the heterogeneous perception of road features among different users.

An advantage of the flow-based formulation is that it requires observations from only a subset of the network (a subset of arcs), which can be obtained, for example, via strategically placed cameras in the city. Another advantage is its broader representation of the cycling population, as it includes all cyclists traveling on the observed roads, not just users of specific mobile applications or survey respondents.

Regarding the trajectory-based formulation, similar to path-based logit models, it requires knowledge of observed and alternative routes. Unlike those models, which depend on choice set generation algorithms to construct alternatives, these routes are generated implicitly in the proposed formulation, by solving multiple shortest-path (SP) problems between the same OD pairs with different cost values. These alternative routes arise from the different weights assigned by cyclists and from the fact that basic costs are modeled as random variables, for which sampled values lead to the identification of different SPs, even within the same user group.

For the second problem, the main contributions of this research are:

- We introduce a mathematical formulation that selects the optimal combination of interventions within a given budget, minimizing the total cost perceived by cyclists (i.e., maximizing the perceived quality of the cycling network) while accounting for a set of road characteristics and cyclists' weights.

- We propose two exact approaches to solve the mathematical formulation. The first method is a simple complete enumeration technique that is able to return an optimal solution but is computationally demanding. It can also be interpreted as a dynamic programming (DP) algorithm without a dominance rule among the states. The second method is a branch-and-bound (BB) framework that efficiently tackles large-size instances.

- We develop two heuristics based on the dynamic programming method for solving the classic knapsack problem. In both approaches, each intervention is treated as an item of a knapsack.

- We derive managerial insights from computational experiments with real data from the city of Parma, Italy, containing nearly 45,000 nodes and 100,000 arcs.

In the proposed exact approaches, each DP state or BB node is associated with a subset of interventions, and generating a new state or node implies adding or removing one intervention from this subset, respectively. Since the choice of which intervention to remove is fundamental to the performance of the BB method, we explore two branching strategies, binary and non-binary branching, and, in the former case, introduce different criteria for selecting the intervention to remove, based on the parameters of the problem. A comparison among them is performed to identify the combination of strategies that results in the best performance regarding CPU time.

The heuristic procedures differ in how the item profits are calculated and in the number of knapsack problem instances that must be solved. Our experiments revealed that both are capable of achieving high quality solutions but one outperforms the other in terms of CPU time. In addition, the primal bounds obtained by the heuristics are used as a warm start for the BB method. As indicated in Appendix E, the performance of the BB method is not significantly dependent on the warm start, suggesting that the method delivers reliable performance even when no strong initial upper bounds are provided.

It is worth noting that the benchmark based on data from Parma was developed in collaboration with a group of architects from the University of Parma, who conducted an extensive study on inspection-based cycling indicators, providing quantitative values for qualitative road features such as safety indices and practicability scores.

This thesis is structured as follows:

- Chapter 2 presents a literature review related to the studied problems, including works on route choice models and on inverse optimization, approaches used to estimate route choice behavior; and an extensive review on the main approaches used in cycling networks problems.

- Chapter 3 defines the problem of identifying cyclists' route choice behavior, introducing both flow-based and path-based formulations, the solution approaches used, and the computation experiments performed with synthetic and real data collected from Parma.

- Chapter 4 addresses the problem of selecting interventions under budget constraints to minimize cyclists' perceived costs, presenting the mathematical formulation, the two exact approaches, and the two heuristics based on dynamic programming for the knapsack problem. Moreover, this chapter also presents the computational experiments conducted using both generated and real-world instances, the latter corresponding to Parma.

- Chapter 5 introduces an ongoing work that extends the problem addressed in Chapter 4. In this extension, interventions applied to the cycling network not only benefit cyclists but also negatively affect motorized traffic users by increasing their perceived costs. Therefore, the selection of the best intervention combinations considers the trade-off between the two types of users in terms of cost minimization.

- Chapter 6 provides the concluding remarks of this research and discusses perspectives for future work.

Chapter 2

Literature review

This chapter presents a literature review on problems related to cycling network planning. Section 2.1 describes the main methodologies used to estimate cyclists' route choice behavior, which corresponds to the problem addressed in Chapter 3. Section 2.2 covers bikeway network planning, which relates to the problem tackled in Chapter 4, as well as approaches for managing multimodal networks in the context of network design.

2.1 Route choice behavior

Many factors influence cyclists' route choices in urban networks. Studies show that distance is not the sole objective function that cyclists wish to minimize when choosing their routes [15, 96]. For instance, Howard and Burns[52] show that the presence (or absence) of bike facilities has a significant influence on cyclists' route choices. Therefore, it is fundamental to establish quantitative methods for assessing qualitative features of road networks from the cyclist's perspective. One such approach is the Bicycle Level of Service (BLOS) [31], whose purpose is to assign quantitative values to several aspects of a road segment, according to cyclist–tailored metrics.

In the following decades, similar concepts were proposed, such as the Bicycle Compatibility Index (BCI) [49], which focuses on assessing the coexistence of motor vehicles and bicycles along roadways, and the concept of bikeability [120], which evaluates how a road network promotes (or discourages) the use of bikes. More recently, Weikl and Mayer[118] proposed a data-driven methodology to assess the quality of cycling networks, evaluating it with respect to safety, comfort, directness, coherence, and attractiveness. Each criterion is associated with sub-criteria and indicators, which are further divided into local, route, and network-level indicators. These sub-criteria and indicators are combined and weighted to define a score for each criterion and an overall score for the network quality. Furthermore, Grisiute et al.[43] present an ontology-based approach to structure and unify the various metrics used to evaluate cycling networks.

After defining methodologies to quantify the road attributes, the next step is to understand how these attributes are perceived by different cyclists and how they influence the route choice process. Route choice models are essential for investigating how travelers select paths through a transportation network, identifying which of routes are likely to be chosen based on a utility-maximization or cost-minimization process [11]. Hence, we present a summary of the literature on some classes of route choice models, namely logit models (Section 2.1.1) and latent class logit models (Section 2.1.2). In addition, we point out some examples of studies that use inverse optimization for estimating route choice behavior (Section 2.1.3).

2.1.1 Logit models

The Multinomial Logit (MNL) model is one of the most commonly used route choice models. It associates to each arc an utility function, composed by a sum of a deterministic quantity and a random variable. The random variables associated to different arcs are independent and identically distributed. However, a key limitation of MNL models is that they do not account for similarity among alternatives [92]. To address this issue, Cascetta et al.[19] propose the C-logit model, which introduces a commonality factor to measure the degree of similarity among route alternatives. Similarly, Ben-Akiva and Bierlaire[9] developed the Path-Size Logit (PSL) model, which incorporates a correction factor, named path-size factor, to reduce the choice probabilities of paths with extensive overlap.

MNL and PSL models have been widely applied to real-world cycling networks around the world, including cities such as Zurich, Switzerland [81, 79]; Seattle, United States [20]; Copenhagen, Denmark [93]; Winnipeg, Canada [103]; and Helsinki, Finland [58], among others. Examples of input data used to analyze cyclists' route choices include GPS data on cycling trajectories, collected through mobile applications, as well as preference surveys, conducted among bike-sharing users [97].

These models can be classified as path-based models, as they require both observed and alternative routes to estimate model parameters [80]. Due to the impossibility of enumerating all possible paths, especially in large networks, algorithms are employed to generate a representative and computationally feasible set of alternative routes.

Two commonly used algorithms for choice set generation are the Breadth-First Search with Link Elimination (BFSLE) algorithm [101] and the Labeling algorithm [10]. More recently, Ton et al.[113] proposed a Data-Driven for Path Identification (DDPI) approach, which constructs the choice set by aggregating all unique observed routes for origin-destination (OD) pairs using GPS data from Amsterdam, Netherlands. Another choice set generation procedure was developed by Lu et al.[73], who use GPS trajectory data from the bike-sharing system Social Bicycles in Hamilton, Canada. In their approach, bike trips were grouped by OD pairs, and a map-matching algorithm based on geographic information systems was applied to generate routes for each pair, taking into account several route attributes.

An alternative to traditional PSL models is considered by Sobhani et al.[107], who developed an approach that combines the Metropolis-Hastings (MH) method for alternative route choice set generation with the Expanded Path-Size Logit (EPSL) model introduced by Frejinger et al.[36]. Unlike traditional PSL models, the EPSL also accounts for the influence of non-sampled paths when correcting for overlap between alternative routes.

To overcome the need for generating a set of alternative routes, Fosgerau et al.[34] proposed the first connection between link-based route choice model and MNL models, by developing the Recursive Logit (RL) model. In this model, a route is represented as a sequence of link choices, where each choice maximizes a utility function, and the probability of choosing a link is determined using the MNL model. The main advantage of this approach over path-based models is that it computes utilities and choice probabilities recursively at the link level, making path enumeration or sampling unnecessary.

To address the issue of overlapping paths, link-additive correction terms, similar to path-size attribute, were proposed. These corrections do not require path enumeration or sampling of the choice set. An extension of the RL model, named Nested Recursive Logit (NRL) model, was introduced by Mai et al.[76]. In this extension, the utility function is scaled by a link-specific factor. Although this increases the computational complexity of calculating value functions, it allows representing more realistic route choice scenarios, particularly in large networks with overlapping paths.

Applications of RL and NRL models can be found in [124] and [127]. The former describes the development of an RL model to estimate the route choice behavior of e-scooter users, using GPS data collected from trips on the Virginia Tech Blacksburg campus in the United States. The latter focuses on the estimation of RL and NRL models for cyclists' route choice behavior, based on GPS data collected in the city of Eugene, United States. In both cases, the RL models were solved using the Broyden–Fletcher–Goldfarb–Shanno algorithm, while the NRL model in [127] was estimated using the Berndt–Hall–Hall–Hausman method.

The work of Meister et al.[80] presents a comparison between path-based (PSL) and link-based (RL) route choice models. The results of the comparison indicate that certain factors, such as cycling infrastructure, affect model estimates differently depending on the modeling approach. Moreover, the best-performing model changes depending on the evaluation metric used, particularly whether or not probabilistic aspects of route choice behavior were considered.

An important issue in route choice modeling is the heterogeneity among decision-makers. In the context of cyclists' route choice, different cyclists or groups of cyclists assign different levels of importance to road features, depending on their perceptions. One approach to consider heterogeneity is the use of mixed logit models [20, 79, 93, 97, 58, 70]. Rather than assuming uniform preferences among all cyclists, these models allow the weights associated with each road characteristic to vary according to a probability distribution. Most of these studies assume that these parameters follow continuous probability distributions, such as normal, lognormal, and triangular, with the normal distribution being the most commonly assumed [114]. In addition, they mainly use likelihood-based methods for estimation.

Some of the results obtained from these studies indicate that cyclists' preferences are influenced by factors such as distance, traffic volume, road speed limits, traffic signals, slopes, crossings, among others. In addition, cyclists have shown a preference for routes that are separated from motorized traffic and have minimal interaction with pedestrians. Table 2.1 summarizes the referenced studies related to logit models.

2.1.2 Latent class logit models

Similar to mixed logit models, Latent Class Logit (LCL) models are widely used in transportation research to model unobserved heterogeneity in route choice behavior. Unlike MNL models, which assume homogeneous preferences, LCL models divide the population into a finite number of latent classes, each with distinct utility parameters. This approach captures behavioral variations in a flexible and interpretable way, providing a semi-parametric alternative to continuous mixture models. However, discretizing the population into classes introduces a combinatorial aspect to the problem, making it more difficult to solve.

Key studies have demonstrated the value of LCL models for identifying meaningful user segments, often incorporating psychometric or socio-demographic data [12]. Compared to mixed logit, LCL offers a simpler interpretation and avoids strong distributional assumptions [42]. Empirical work such as the study by Shen[105] has shown that LCL performs well, especially when a small number of classes adequately captures heterogeneity. Despite these strengths, LCL models face limitations. Choosing the number of classes is often arbitrary and model selection depends on statistical criteria. Moreover, LCL models require predefined choice sets, making them sensitive to how alternatives are generated.

Table 2.1: Summary of studies using logit models for route choice estimation.

Reference	Problem	Method
Menghini et al. [81] Transportation Research Part A	Estimation of route choice behavior based on a large sample of GPS data	Multinomial logit model with path-size factor
Meister et al. [79] Transportation Research Part A	Estimation of route choice behavior	Mixed path-size logit models
Chen et al. [20] International Journal of Sustainable Transportation	Analysis of the impact of built environment features on cyclists' route preferences	Mixed logit models
Prato et al. [93] International Journal of Sustainable Transportation	Estimation of route choice behavior	Generalized mixed path-size logit model
Ryu et al. [103] International Journal of Sustainable Transportation	Bicycle traffic assignment problem	Multi-objective shortest path algorithm and path-size logit model
Reckermann et al. [97] Journal of Cycling and Micromobility Research	Estimation of route choice behavior	Mixed multinomial logit models
Khavarian et al. [58] Journal of Cycling and Micromobility Research	Estimation of route choice behavior for e-bike and r-bike users	Multinomial and mixed logit models
Rieser-Schüssler et al. [101] Transportmetrica A: Transport Science	Route choice set generation	Breadth-First Search with Link Elimination algorithm
Ben-Akiva et al. [10] International Symposium on Transportation and Traffic Theory	Route choice set generation	Labeling algorithm
Ton et al. [113] Travel Behaviour and Society	Route choice set generation	Data-driven approach for path identification
Lu et al. [73] Journal of Transport Geography	Route choice set generation	A map-matching algorithm based on geographic information systems
Liu et al. [70] Journal of Cycling and Micromobility Research	Estimation of route choice behavior	Mixed path-size logit model
Sobhani et al. [107] International Journal of Transportation Science and Technology	Estimation of route choice behavior for non-work-related cyclists trips	Metropolis-Hastings and Expanded Path-Size logit model
Frejinger et al. [36] Transportation Research Part B	Estimation of route choice behavior	Expanded path-size logit model
Fosgerau et al. [34] Transportation Research Part B	Estimation of route choice behavior	Recursive logit model
Mai et al. [76] Transportation Research Part B	Estimation of route choice behavior	Nested recursive logit model
Meister et al. [80] Journal of Cycling and Micromobility Research	Comparison between link-based and path-based logit models	Path-size logit and recursive logit models
Ben-Akiva and Bierlaire [9] Handbook of Transportation Science	Estimation of route choice behavior	Path-size logit model
Cascetta et al. [19] Proceedings of The 13th ISTTT	Estimation of route choice behavior	C-logit model
Zhang et al. [124] Transportation Research Part D	Estimation of route choice behavior based on GPS data for e-scooter users	Recursive logit model
Zimmermann et al. [127] Transportation Research Part C	Estimation of route choice behavior based on GPS data	Recursive logit and nested recursive logit models

2.1.3 Inverse optimization

Other techniques can be employed to understand route choice behavior, such as inverse optimization (IO). In this approach, given the solutions to a particular problem, the goal is to infer an optimization model (objective function costs or parameters values) for which these solutions are optimal. Applications of IO to estimate choice behaviors can be found in the studies by Zattoni Scroccaro et al.[122], Rönnqvist et al.[104], and Zhao and Liang[125].

2.2 Cycling networks

As part of the literature on bikeway design problems, Lin and Yu [67] introduced a bikeway classification that consists of bike paths, lanes, and routes. Therefore, we divide this brief literature review into bikeway network design with bike lanes (Section 2.2.1), with bike paths (Section 2.2.2), and with mixed bikeways (Section 2.2.3). Furthermore, we include papers that address bikeway network design along with bicycle station design (Section 2.2.4), as well as those related to multimodal networks (Section 2.2.5).

2.2.1 Bikeway network design with bike lanes

Sohn [108] applied a genetic algorithm (GA) to reduce the number of lanes for car drivers and dedicate them to cyclists. Another application of GA in the context of network design is found in Mesbah et al. [82], where the method was used to solve a network design problem, modeled as bilevel problem, that promotes a balance between cyclists and car users. GA was also applied by Doorley et al. [27] to maximize the health and environmental benefits of cycling in a network design problem. In addition, the authors proposed a mathematical formulation with equilibrium constraints.

Bagloee et al. [7] tackled the bicycle priority lanes design problem in congested networks, formulated as a bilevel problem and solved via a branch-and-bound approach. The following year, Bao et al. [8] introduced a mathematical formulation to balance the trade-off between the number of covered users and the length of the covered trips, using bicycle trajectory data. Similarly, He et al. [50] employed trajectory data for bicycle lane planning on large-scale scenarios.

Akbarzadeh et al. [3] analyzed taxi trip data to identify high-demand areas for bicycle infrastructure. The most frequent origins and destinations formed a graph, with edges representing travel links. After constructing the graph, regions with dense travel activity were identified, and points less than 4 km apart were selected. A bicycle network was then proposed to connect these points, aiming to minimize travel costs and total network length.

Guo et al. [45] investigated bicycle flow dynamics on wide roads by applying a heuristic-based model that considers centrifugal effects on bicycles. Lim et al. [65] addressed a bicycle lane network design problem aimed at improving network safety under budget constraints. Simultaneously, Liu et al. [72] combined a bilevel program with a logit model to determine the most suitable locations for new lanes in the existing road network. All these references are summarized in Table 2.2.

2.2.2 Bikeway network design with bike paths

Duthie and Unnikrishnan [28] and Zhu and Zhu [126] studied the problem of retrofitting existing network for cyclists. The former aimed to connect all origin-destination (OD) pairs through roadways that satisfy specific bounds of bicycle level of service. The latter focused on maximizing accessibility and bicycle level of service, while minimizing the number of intersections and construction costs.

Liu et al. [71] solved a bike path network design problem with the goal of maximizing the total route utilities of cyclists, where their route choice preferences were identified through a path-size logit (PSL) model, while Zuo and Wei [128] developed an approach based on multi-criteria decision analysis to improve bicycle connectivity while considering multiple factors. The latter method was applied to a case study in Uptown Cincinnati, Ohio, United States.

Agarwal et al. [2] addressed the bicycle superhighway problem, which aims to determine the optimal number and the location of connectors between new infrastructure and the existing network, with the goal of increasing bicycle share. More recently, Ospina et al. [89] proposed a two-phase approach based on mixed-integer

Table 2.2: Summary of the literature on bikeway design with bike lanes.

Reference	Problem	Method
Akbarzadeh et al. [3] Applied network science	Non-dominated bike networks design, based on taxi trips data	Modularity maximization method and bi-objective model
Bagloee et al. [7] Transportation Research Part A	Bicycle priority lane design in congested networks	Bilevel model and Branch-and-Bound approach
Bao et al. [8] Proceedings of the ACM SIGKDD	Bicycle lane planning based on real bike trajectories	Mathematical formulation and grid network expansion algorithm
Doorley et al. [27] International Journal of Sustainable Transportation	Maximizing the health and environment benefits of cycling	Mathematical model with equilibrium constraints and genetic algorithm
Guo et al. [45] Transportation Research Part C	Investigating bicycle flow dynamics on wide roads	Heuristic-based model that considers centrifugal effects
He et al. [50] IEEE Transactions on Knowledge and Data Engineering	Bicycle lane planning based on large-scale real world bike trajectories	Greedy network expansion algorithm
Lim et al. [65] Journal of Transportation Engineering, Part A: Systems	Bike network improvement problem	MIP formulation with piecewise linear penalty function and Benders decomposition
Liu et al. [72] Manufacturing & Service Operations Management	Planning based on fine-grained bike trajectory data	Bilevel model based on Multinomial Logit
Mesbah et al. [82] Transportation Research Record	Planning considering a balance between cyclists and car users	Bilevel model and genetic algorithm
Sohn [108] Transportation Research Part A	Road diet network design problem	Bilevel model and multi- objective genetic algorithm

linear programming (MILP) to solve the maximal covering bicycle network design problem. Finally, Güldü et al. [44] tackled the problem of improving bicycle connectivity in the city of Sivas, Turkey. Table 2.3 summarizes the papers reviewed in this section.

Table 2.3: Summary of the literature on bikeway design with bike paths.

Reference	**Problem**	**Method**
Agarwal et al. [2] Transportation Research Part A	Bicycle superhighway	Algorithm to identify connectors with the existing network
Duthie and Unnikrishnan [28] Journal of Transportation Engineering	Retrofitting existing roadway infrastructure for bicycles	Mathematical formulations
Güldü et al. [44] Environment, Development and Sustainability	Creating a bicycle road network by integrating a few disconnected bicycle routes	Hybrid Multi-Criteria Model-Based Network Analysis
Liu et al. [71] Transportation Research Part E	Maximizing the total route utilities, considering route choice behavior	Bilevel model and matheuristic
Mahfouz et al. [75] Journal of Transport Geography	Cycling infrastructure prioritization	Approach that combines distance decay, route calculation, and network analysis
Ospina et al. [89] Transportation Research Part A	Maximizing the coverage of cyclists and minimize the total cost	Two-phase approach using MILP formulation
Steinacker et al. [111] Nature computational science	Networks based on demand distribution	Specific algorithm based on cyclists' route choices
Zhu and Zhu [126] Transportation	Retrofitting existing cycling infrastructure	Multi-objective formulation and ε-constraint method
Zuo and Wei [128] Transportation Research Part A	Increasing bicycle connectivity and bicycle-transit connection	Multi-Criteria Decision Analysis approach

2.2.3 Bikeway network design with mixed bikeways

Hood et al. [51] and Broach et al. [13] analyzed GPS data to identify the criteria used by cyclists to choose their routes. In both studies, PSL models were applied. Ehrgott et al. [29] proposed a bi-objective model to determine a set of alternative routes for commuter cyclists, considering travel time and route suitability as selection criteria. Similarly, Hrnčíř et al. [53] solved a multi-criteria bicycle routing problem utilizing a heuristic-enabled multi-criteria label-setting algorithm with speedup heuristics. Moreover, Mauttone et al. [78] proposed a multi-commodity mixed-integer formulation, based on the fixed-charge network design problem, and a GRASP-based metaheuristic to solve a cycling network planning problem aimed at minimizing cyclists' travel distances.

Olmos et al. [88] proposed a data analysis framework that identifies the most suitable portions of a network for new bike infrastructure via percolation theory [110]. On the other hand, Liazos et al. [64] focused on network design for electric scooters based on geo-fencing planning by means of a GA approach. This method was also earlier employed by Nash et al. [87] to solve a holistic cycling network planing in Spennymoor, England.

Kon et al. [59] developed a data analysis method to extract bike-sharing mobility flows from OD trip data, which is useful for cycling infrastructure planning. Moreover, Giménez-Gaydou et al. [38] and Wang et al. [116] focused on cyclists' exposure to pollutants. The former proposed a method to estimate cyclists' effort and exposure, while the latter addressed route planning for walking and cycling, considering air pollution exposure. Very recently, Pirolo and Moscarelli [91] introduced a mathematical formulation employing a geographical information system for designing cycleways dedicated to tourism activities, considering the use of electric bikes. Table 2.4 outlines the principal contributions of the papers examined in this section.

2.2.4 Bikeway and bicycle station design

Lin and Yang [68] proposed a non-linear integer programming model to solve the problem of station location and bikeway design, considering bike paths. A similar problem was solved by Lin et al. [69], considering bike lanes and inventory levels, whereas Lin and Liao [66] aimed to maximize bikeway and station service coverage through a multi-objective programming model and the ε-constraint method.

Table 2.4: Summary of the literature on bikeway design with mixed bikeways.

Reference	Problem	Method
Broach et al. [13] Transportation Research Part A	Bicycle route choice problem based on GPS data	Path-size logit model
Ehrgott et al. [29] Transportation Research Part A	Determining the route choice set, considering travel time and suitability of a route	Model and bi-objective near shortest path algorithm
Giménez-Gaydou et al. [38] Transportation Research Part D	Estimating cyclists' effort and exposure to pollutants and urban areas	Cyclist specific effort method
Hood et al. [51] Transportation Letters	Bicycle route choice problem based on GPS data	Path-size logit model
Hrnčíř et al. [53] IEEE Transactions on Knowledge and Data Engineering	Multi-criteria bicycle routing problem	Model and Heuristic-Enabled Multi- Criteria Label-Setting algorithm
Kon et al. [59] Public Transport	Extracting mobility flows from bike-sharing trips data	Analytical method and open source tool for data analysis
Liazos et al. [64] Transportation Research Part D	Network design for electric scooters based on geo-fencing planning	Bi-level bi-objective model and Non-Dominated Sorting Genetic Algorithm
Mauttone et al. [78] Transportation Research Procedia	Cycling network planning to minimize travel distances	MIP formulation and GRASP-based metaheuristic
Nash et al. [87] Transportation Planning and Technology	Holistic cycle network planning	Spatio-temporal model and genetic algorithm
Olmos et al. [88] Transportation Research Part C	Data analysis to identify the candidates for adding new bike infrastructure	Percolation theory
Pirolo and Moscarelli [91] Regional Science Policy & Practice	Cycling network design to integrate cycle tourism and electric bikes	Mathematical formulation in a GIS environment
Wang et al. [116] Transportation Research Part D	Route planning for walking and cycling, considering air pollution exposure	Human exposure assessment model with a shortest path algorithm

Jin et al. [56] addressed the problem of determining the location of stations and bicycle lanes, as well as defining the number of docks per station, through a two-stage stochastic programming model. The same problem was studied by Cheng et al. [21], in which the authors presented a bilevel model and a heuristic approach based on a Markov model, aiming to minimize the total cost. Table 2.5 presents an overview of the works discussed in this section.

Table 2.5: Summary of the literature on bikeway and bicycle station design.

Reference	Problem	Method
Cheng et al. [21] Computers & Industrial Engineering	Determining the location of stations and bike lanes to minimize the total cost	Bilevel model and hyper-heuristic approach based on hidden Markov model
Jin et al. [56] Transportation Letters	Determining the location of stations, the cycle lanes, and the number of docks per each station	Two-stage stochastic programming model
Lin and Liao [66] Networks and Spatial Economics	Maximizing bikeway and station services coverage, bikeway suitability, and minimizing cyclist risk	Multi-objective programming model and e-constraint method
Lin and Yang [68] Transportation Research Part E	Station location, bikeway design with bike paths, and determining the travel paths for users	Non-linear integer programming model
Lin et al. [69] Computers & Industrial Engineering	Station location, bikeway design with bike lanes, the selection of paths of users, and inventory levels	Hub location inventory model, greedy-drop heuristic and procedures to calculate the costs

2.2.5 Multimodal networks

Chow and Sayarshad [22] proposed a multi-objective optimization framework for network design in the presence of coexisting transportation networks, establishing analogies with symbiotic organisms. Nair and Miller-Hooks [86] addressed the problem of determining the location and capacity of stations, along with vehicle inventory levels, in a vehicle-sharing system that may include bicycles.

Burke and Scott [16] developed a method to evaluate the impact of implementing wider bike facilities on car traffic. Caggiani et al. [17] also considered car traffic, but from a different perspective: employing the revenues generated from tolling private car users to fund the construction of a free-floating bike-sharing system.

Li et al. [63] presented a bilevel optimization model and a hybrid iterative-based solution method to design a bimodal transit network, composed of sparse express lines and dense local lines, that is fed by shared bicycles. Frank et al. [35] proposed a MIP formulation for the problem of locating multimodal mobility hubs to improve the accessibility to rural areas. These hubs are intended to supplement public transportation services such as trains and buses. Luo et al. [74] applied stochastic optimization to promote an intermodal network that integrates vehicles of mobility-on-demand transit vehicles with micromobility services, including free-floating bike-sharing and electric scooters.

Ye et al. [121] proposed a bilevel optimization model to determine the location and capacity of transfer infrastructure in multimodal networks, considering elastic demand. In this model, the upper level maximizes the usage of green transport modes, while the lower level is a combined mode-split and traffic assignment model that accounts for travelers' route choices through multinomial logit model. Similarly, Zhang and Liu [123] presented a bilevel formulation to enhance the integration between bike-sharing and metro systems. Moreover, Wang et al. [115] developed a tri-level optimization model and applied a Kriging-surrogate-based optimization algorithm to maximize the capacity of multimodal networks.

Aydin et al. [6] applied an integrated fuzzy multi-criteria decision-making method to determine optimal locations for mobility hubs in Istanbul, Turkey. Cai et al. [18] also addressed hub location optimization, aiming to promote modal shifts between different transport modes and, consequently, achieving a balanced use of the entire network. Wiedemann et al. [119] employed Pareto optimality to analyze trade-offs between car and bicycle usage in road networks. Finally, Fan et al. [33] tackled the combined mode split and traffic assignment problem, considering five transport modes: car, public transport, bicycle, car-to-transit and bike-to-transit. Table 2.6 presents a summary of all the papers reviewed in this section.

Table 2.6: Summary of the literature on multimodal networks.

Reference	Problem	Method
Aydin et al. [6] Sustainable Cities and Society	Location selection problem for the mobility hubs	Integrated fuzzy multi-criteria decision-making method
Burke and Scott [16] Journal of Transport Geography	Evaluating the impact on car traffic of employing wider bicycle facilities	Network Robustness Index
Caggiani et al. [17] Journal of Advanced Transportation	Free-floating bike sharing system development based on pricing strategy applied to private car users	Specific method, including an equity-based optimization model and genetic algorithm
Cai et al. [18] Journal of Advanced Transportation	Multimodal network design problem, promoting a balanced use of the entire network	Bilevel model, hybrid method of the successive average, and hybrid genetic search with advanced diversity control
Chow and Sayarshad [22] Transportation Research Part B	Network design problem in the presence of coexisting networks (symbiotic network)	Framework including multi-objective model, *e*-constraint method, and other algorithms
Fan et al. [33] Transportation Research Part E	Combined mode split and traffic assignment problem	Cross-nested logit model, path-based variational inequality model, and self-regulated averaging and path-based methods
Frank et al. [35] Journal of Transport Geography	Locating multimodal mobility hubs to improving rural accessibility	MIP formulations
Li et al. [63] Transportation Research Part E	Bimodal transit network design problem with shared bikes	Bilevel model and hybrid iterative-based solution method
Luo et al. [74] EURO Journal on Transportation and Logistics	Intermodal mobility network design under uncertainty	Two-stage stochastic program model and an iterative algorithm
Nair and Miller-Hooks [86] European Journal of Operational Research	Location and capacity of the stations and vehicle inventories in a Vehicle Sharing Program	Equilibrium network design model (bilevel model)
Wang et al. [115] Transportation Research Part E	Tri-level multimodal network design problem to maximize the network capacity	Bilevel model and Kriging-surrogate-based optimization algorithm
Wiedemann et al. [119] Transportation Research Part B	Trade-off between car and bicycle usage	Pareto optimality
Ye et al. [121] Transportation Research Part E	Determining the location and capacity of transfer infrastructure, considering elastic demand	Bilevel model and genetic algorithm
Zhang and Liu [123] Transportation Research Part D	Integration between bike sharing and metro systems	Bilevel model and Nash Bargaining game

Chapter 3

Identifying cyclists' route choice criteria

This chapter introduces the problem of identifying the criteria used by cyclists to select routes in bicycle networks. As mentioned in Chapter 1, this identification is based on either traffic flows observations or on knowledge of a set of paths. Hence, Section 3.1 formalizes the problem in the case where flow observations are available on a subset of arcs and describes the proposed flow-based formulation. Section 3.2 addresses the case where we assume knowledge of a set of paths followed by some users, and presents the corresponding trajectory-based formulation. Moreover, Section 3.3 proposes solution algorithms for the problems addressed in this chapter, while Section 3.4 reports the computational experiments performed using both synthetic data and real data collected from the city of Parma, Italy. Finally, Appendix A presents all data related to the real case of Parma and discuss how the basic costs were derived, while Appendix B collects the proofs of most of the propositions presented throughout this chapter. It is worth noting that the work presented in this chapter is also available as research papers (see Ardizzoni et al. [4, 5]).

3.1 Identification via known flows over a subset of arcs

We represent a bicycle network with a directed graph $G = (V,A)$. We denote the number of nodes and directed arcs by $n = |V|$ and $m = |A|$, respectively. The arc set represents the roads used by cyclists, and the node set the intersections. We assume that G is strongly connected. We denote by $W \subset V \times V$ a set of origin-destination (OD) pairs in the network. We associate to each pair $w = (o_w, d_w) \in W$ a demand value $u_w \in \mathbb{N}$, corresponding to the number of users that move from node o_w to node d_w, traveling within the network, at a given time of the day. Values u_w, with $w \in W$, may be known in advance but, in some cases, there might be the need to estimate at least some of them. We denote by Π_w the set of elementary directed paths from origin node o_w to destination node d_w. To each $\pi \in \Pi_w$, we associate the subset $A^w_\pi \subseteq A$ of arcs belonging to directed path π. We associate to each arc $(i, j) \in A$ a flow x^w_{ij}, corresponding to the amount of users, associated to pair w, traveling along the arc. For all $(i, j) \in A$, the total flow along arc (i, j) is denoted by

$$x_{ij} = \sum_{w \in W} x^w_{ij}. \tag{3.1}$$

We define the following vectors:

- $\mathbf{x} \in \mathbb{R}^{|A|}_+$, the vector whose components are the total flows x_{ij} along the arcs of the network;
- $\mathbf{x}^w \in \mathbb{R}^{|A|}_+$, $w \in W$, the vectors whose components are flows x^w_{ij} of users associated to OD pair $w \in W$ along the arcs of the network.

Moreover, we associate to each arc $(i, j) \in A$ a set of *basic costs* $\mathbf{c}_{ij} = (c^1_{ij}, \ldots, c^r_{ij})$, that represents a set of features associated to the road represented by the arc. Possible features may be, among others, the road length, its safety, its practicability. We indicate with $\mathbf{c} = \{\mathbf{c}_{ij}\}_{(i,j)\in A}$ the set of all the basic costs of the graph. The total cost of an arc is a convex combination of its r basic costs. We assume that this combination is different for each individual user. Indeed, each cyclist can choose the best route with a different criterion, giving more importance to some specific features. Since different features may have different units of measure and different orders of magnitude, we normalize the basic costs in such a way that $\sum_{(i,j)\in A} c_{ij}^{h_1} = \sum_{(i,j)\in A} c_{ij}^{h_2}$ for all $h_1, h_2 \in \{1,\dots,r\}$. We assume that each cyclist assigns a cost to an arc (i,j) of form

$$\sum_{h=1}^{r} c_{ij}^{h} p_h.$$

That is, the cost of each arc is a weighted sum of the basic costs. The coefficient vector $\mathbf{p} = (p_1,\dots,p_r)$ belongs to the r-dimensional unit simplex

$$\Delta_r = \left\{ p \in \mathbb{R}_+^r \;\middle|\; \sum_{h=1}^{r} p_j = 1 \right\}.$$

We call 'weights' the components of $\mathbf{p}$. They represent the relative importance that the cyclist assigns to the basic costs. In this approach, these weights fully define the cyclist's policy. Then, for fixed origin and destination nodes, we assume that each cyclist follows a route that minimizes the sum of the arcs costs. As said, for the same origin and destination nodes, different cyclists may choose different paths, because of their different policies.

First, we consider a simplified problem, where we assume that each user may choose among a finite set of policies, that is a finite set of weight vectors $\mathbf{p}$ (Section 3.1.1). Later on (Section 3.1.2), we address the problem where such set is also to be computed.

3.1.1 The case with known set of convex combinations/weights

We assume that users are divided in q different groups, indexed by set $Q = \{1,\dots,q\}$, and that each group is associated to a known weight vector. The overall set of weights $P = \{\mathbf{p}^\ell\}_{\ell\in Q}$ is a finite subset of the r-dimensional unit simplex Δ_r. The total cost of arc (i,j) for a user that belongs to group ℓ (associated to weight vector $\mathbf{p}^\ell$) is

$$\mathbf{c}_{ij}^\top \mathbf{p}^\ell = \sum_{h=1}^{r} c_{ij}^{h} p_h^\ell.$$

We associate a coefficient $\alpha_\ell \in [0,1]$ to every group $\ell \in Q$. It represents the fraction of the total users that belong to group ℓ. Since every user belongs to a group,

$\sum_{\ell \in Q} \alpha_\ell = 1$. The function $Q \to [0,1]$ that associates to ℓ coefficient α_ℓ can be interpreted as a probability mass function. That is, α_ℓ is the probability that a generic user belongs to group ℓ. We assume that the traffic is not congested. Hence, all vehicles moving from o_w to d_w follow the SP in G. However, the SP depends on the chosen weights, so it may be different for users belonging to different groups.

To each $\mathbf{p}^\ell \in P$, with $\ell \in Q$, and each $w \in W$, we associate a vector $\mathbf{x}^{w,\ell} \in \mathbb{R}^{|A|}$, whose components are flows $x_{ij}^{w,\ell}$ along the arcs of the network of users associated to OD pair $w \in W$ whose selected weight vector is $\mathbf{p}^\ell$. For all $(i,j) \in A$, and $w \in W$, we have the following constraints, linking variables $x_{ij}^{w,\ell}$ and x_{ij}^{w},

$$x_{ij}^{w} = \sum_{\ell=1}^{q} x_{ij}^{w,\ell}. \tag{3.2}$$

For each pair $w \in W$, for each node $i \in V$, and for each weight $\mathbf{p}^\ell \in P$, the following flow conservation constraints hold:

$$\sum_{(i,j)\in A} x_{ij}^{w,\ell} - \sum_{(j,i)\in A} x_{ji}^{w,\ell} = \alpha_\ell U_i^w, \tag{3.3}$$

where

$$U_i^w = \begin{cases} u_w & i = o_w \\ -u_w & i = d_w \\ 0 & i \in V \setminus \{o_w, d_w\}. \end{cases}$$

We further impose the non-negativity constraints

$$(\forall w \in W)\ (\forall \ell \in Q)\ \ x, x^w, x^{w,\ell} \geq 0. \tag{3.4}$$

Constraints (3.3) can be written in the matrix form

$$\mathbf{N}\mathbf{x}^{w,\ell} = \alpha_\ell \mathbf{U}^w,$$

where $\mathbf{N}$ is the node-arc incidence matrix of the graph, and $\mathbf{U}^w$ is the vector of the demand related to pair $w \in W$.

We assume that we do not measure all flows, but only a subset of them $\bar{A} \subset A$. We denote with $\bar{x}_{ij}, (i,j) \in \bar{A}$, the measured flows. Our aim is to estimate $\{\alpha_\ell\}_{\ell \in Q}$ from the knowledge of $\bar{x}_{ij}, (i,j) \in \bar{A}$.

To this end, we minimize the squared distance between measured and estimated flows, so that we end up with the following bilevel optimization problem:

$$g(P) = \min_{\alpha} \sum_{(i,j)\in\bar{A}} [x_{ij} - \bar{x}_{ij}]^2 \tag{3.5a}$$

s.t.

$$x_{ij} = \sum_{\ell=1}^{q} \sum_{w\in W} x_{ij}^{w,\ell} \qquad (i,j) \in \bar{A} \tag{3.5b}$$

$$\sum_{\ell=1}^{q} \alpha_\ell = 1 \tag{3.5c}$$

$$\alpha_\ell \geq 0 \qquad \ell \in Q \tag{3.5d}$$

$$\{\mathbf{x}^{w,\ell}\}_{\ell\in Q, w\in W} \in S(\alpha_1,\ldots,\alpha_q) \tag{3.5e}$$

$$S(\alpha_1,\ldots,\alpha_q) = \arg\min \sum_{\ell=1}^{q} \sum_{w\in W} \sum_{(i,j)\in A} \left(\mathbf{c}_{ij}^\top \mathbf{p}^\ell\right) x_{ij}^{w,\ell} \tag{3.6a}$$

s.t.

$$\mathbf{N}\mathbf{x}^{w,\ell} = \alpha_\ell \mathbf{U}^w \qquad \ell \in Q, w \in W \tag{3.6b}$$

$$\mathbf{x}^{w,\ell} \geq 0 \qquad \ell \in Q, w \in W. \tag{3.6c}$$

Note that the optimal value $g(P)$ of the bilevel problem depends on the set of weights P, assumed to be known in advance. For the upper level model, objective function (3.5a) aims to minimize the error between the calculated and measured flows. Constraints (3.5b) compute the total arc flows, while constraints (3.5c) and (3.5d) impose that the components of α are positive and sum to 1. Constraints (3.5e) impose that the estimated flows must be optimal with respect to the lower-level problem parameterized by the upper-level variables α_ℓ.

For the lower level model, objective function (3.6a) aims to minimize the total travel cost, subject to the fulfillment of the OD pairs demands guaranteed by constraints (3.6b). Constraints (3.6c) define the domain of the lower level variables.

Problem (3.5a)–(3.6c) can be solved quite efficiently. First we observe that the lower-level problem can be split into the following $q|W|$ subproblems: for each $\ell \in Q$ and each $w \in W$, solve:

$$\min \sum_{(i,j)\in A} \left(\mathbf{c}_{ij}^{\top}\mathbf{p}^{\ell}\right) x_{ij}^{w,\ell}$$
$$\mathbf{N}\mathbf{x}^{w,\ell} = \alpha_{\ell}\mathbf{U}^{w}$$
$$\mathbf{x}^{w,\ell} \geq 0.$$

The solution of this problem is obtained by first detecting the SP from o_w to d_w with cost of each arc (i,j) equal to $\mathbf{c}_{ij}^{\top}\mathbf{p}^{\ell}$. Once the SP has been detected, we send a flow equal to $\alpha_{\ell}u^{w}$ along the arcs of the path. More precisely, we proceed as follows. Let $S^{w,\ell} \subset A$ be a SP from o_w to d_w based on the weight vector $\mathbf{p}^{\ell}$. Additionally, let

$$f_{ij}^{w,\ell} = \begin{cases} u^{w} & (i,j) \in S^{w,\ell} \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

Next, we define a matrix $\mathbf{M}_{|A|\times q}$ whose elements correspond to the sum of flows for each $w \in W$, considering the arc (i,j) and the weight $\mathbf{p}^{\ell}$, that is,

$$(\forall \ell \in Q)\ (\forall (i,j) \in A)\ \ M_{(i,j),\ell} = \sum_{w\in W} f_{ij}^{w,\ell}. \tag{3.8}$$

Then,

$$(\forall (i,j) \in A)\ \ x_{ij} = \sum_{\ell=1}^{q} M_{(i,j),\ell}\alpha_{\ell}, \tag{3.9}$$

or, in matrix form $\mathbf{x} = \mathbf{M}\alpha$. If we denote by $\mathbf{M}_{\bar{A}}$ the submatrix obtained by considering only the rows of $\mathbf{M}$ in $\bar{A}$, the upper-level problem reduces to:

$$\begin{aligned} &\min_{\alpha} \|\mathbf{M}_{\bar{A}}\alpha - \bar{\mathbf{x}}\|^2 \\ &\sum_{\ell=1}^{q} \alpha_{\ell} = 1 \\ &\alpha \geq 0, \end{aligned} \tag{3.10}$$

which is a convex quadratic problem, solvable by different available solvers like, e.g., `Gurobi` [46].

Algorithm 1 summarizes the proposed procedure to compute g and the optimal probability distribution over a fixed set P of weights.

Algorithm 1: Solution algorithm for the evaluation of function g and computation of the optimal probability distribution for a fixed set P of weights.

1 **function** IDENTIFICATION(P)
2 **for** $w \in W$ *and* $\mathbf{p}^\ell \in P$ **do**
3   compute SP from o_w to d_w with cost $\mathbf{c}_{ij}^\top \mathbf{p}^\ell$ for each arc (i, j) ;
4 Compute matrix $\mathbf{M} \in \mathbb{Z}^{|A| \times |P|}$ with entries defined in (3.8), where values $f_{ij}^{w,\ell}$ are defined in (3.7) ;
5 Solve convex Quadratic Programming (QP) Problem (3.10) ;
6 **return** α_P, $g(P)$;

The overall complexity of Algorithm 1 is stated in the following proposition.

Proposition 1. *The complexity of Algorithm 1 is*

$$O\left(|W||P|\left(|A| + |V|\log|V|\right) + |P|^3 L\right),$$

where L is the bit size of the input of the convex QP.

Proof. The `For` loop at line 2 of Algorithm 1 requires the solution of $|W||P|$ SP problems, so that the complexity of the loop, if Dijkstra's algorithm is employed to solve the SP problems, is: $O\left(|W||P|\left(|A| + |V|\log(|V|)\right)\right)$. Computing matrix M at line 4 requires a time $O(|W||P||V|)$ since, for each $w \in W$ and each $\mathbf{p}^\ell \in P$, only the entries of column ℓ of matrix M associated to the arcs in the SP from o_w to d_w are updated, and the SP contains at most $|V|$ arcs. Finally, the convex QP problem solved at line 5 belongs to the class of problems for which it is shown that the computing time for their solution is $O\left(|P|^3 L\right)$ [84]. □

Note that this complexity result shows that for small $|P|$ values the major cost is represented by the solution of the SP problems, but as $|P|$ increases, the major cost becomes the solution of the convex QP problem.

3.1.2 The case of unknown weights

If the set P of weights is not known in advance, then a further optimization has to be performed, searching for a set P with the lowest possible value $g(P)$ (i.e., the lowest possible distance between observed and estimated flows). The value of g can be reduced by: (i) enlarging the set of weights and/or (ii) perturbing the current weights. Enlarging the set of weights allows reducing g because of the monotonicity property of g, proved in the following proposition (see Appendix B for the proof of the following results).

Proposition 2. *Let $P' \supset P$. Then, $g(P') \leq g(P)$.*

In fact, a rather similar proof can be applied to reduce the number of weights.

Corollary 3.1.1. *Let P be a set of weights and α be an optimal solution of the optimization problem* (3.10). *If $\bar{P} \subset P$ is such that, for all $\mathbf{p}^\ell \in P \setminus \bar{P}$, $\alpha_\ell = 0$, then $g(P) = g(\bar{P})$.*

According to Proposition 2, we can reduce g by expanding the set of weights. However, a large set of weights P has at least two drawbacks. The first one is that the complexity result stated in Proposition 1 shows that the computing times for the algorithm calculating value $g(P)$ grow as $|P|^3$. The second drawback is that, for the sake of interpretability, large $|P|$ values should be discouraged. Note that the result stated in Corollary 3.1.1 allows reducing the set of weights by discarding all weights with null probability.

An alternative way to reduce g is by keeping fixed the cardinality q of P and by perturbing the weights in P. Let Δ_r^q be the Cartesian product of Δ_r by itself for q times. Let $\mathscr{Q} = (\mathbf{p}^1, \dots, \mathbf{p}^q) \in \Delta_r^q \subset \mathbb{R}^{rq}$ be an ordered set. Then, we introduce the function

$$\bar{g} \, : \, \Delta_r^q \to \mathbb{R}_+,$$

defined as follows: if $P = \{\mathbf{p}^1, \dots, \mathbf{p}^q\}$, where $\mathbf{p}^\ell \in \Delta_r$, with $\ell \in Q$, then $\bar{g}(\mathscr{Q}) = g(P)$. A *vectorization* of set $P = \{\mathbf{p}^1, \dots, \mathbf{p}^q\}$ is a real vector $\mathscr{Q} = (\mathbf{p}^1, \dots, \mathbf{p}^q) \in \mathbb{R}^{rq}$. Note that any order of $(\mathbf{p}^1,\ldots,\mathbf{p}^q)$ fulfills the definition. Hence, the problem of identifying the best set of weights with fixed cardinality q can be reformulated as follows:

$$\min_{(\mathbf{p}^1,\ldots,\mathbf{p}^q)\in\Delta_r^q} \bar{g}(\mathbf{p}^1,\ldots,\mathbf{p}^q).$$

Unfortunately, we cannot employ gradient-based methods even to detect local minimizers of $\bar{g}$. Indeed, we will show that this function is not continuous and is piecewise constant. To this end, we first introduce an assumption.

Assumption 3.1.2. *For each $w \in W$, recall that Π_w is the finite collection of paths between o_w and d_w. For some $\pi \in \Pi_w$, let $\lambda_h(\pi) = \sum_{(i,j)\in\pi} c_{ij}^h$, $h \in \{1,\ldots,r\}$, be the cost of π with respect to the h-th basic cost. Then, we assume that for each $w \in W$, there do not exist two distinct paths $\pi,\pi' \in \Pi_w$ such that $\lambda_h(\pi) = \lambda_h(\pi')$ for all $h \in \{1,\ldots,r\}$.*

The assumption simply states that there do not exist two distinct paths equivalent with respect to all the r basic costs. If such two paths existed, they would be indistinguishable, since their cost would be the same for all possible weights $\mathbf{p} \in \Delta_r$. Now, we can prove the following lemma.

Lemma 3.1.3. *For each $w \in W$ and $\mathbf{p}^\ell \in \Delta_r$, let $S^{w,\ell}$ denote the set of SPs when the cost of each arc (i,j) is $\mathbf{c}_{ij}^\top \mathbf{p}^\ell$. Under Assumption 3.1.2, the set*

$$\Delta_r(w,\ell) = \left\{\mathbf{p}^\ell \in \Delta_r \;\middle|\; |S^{w,\ell}| > 1\right\},$$

has null measure in Δ_r.

In view of this lemma, we can prove the following proposition that basically states that function $\bar{g}$ is piecewise constant.

Proposition 3. *For all $\bar{\mathscr{Q}} = (\bar{\mathbf{p}}^1,\ldots,\bar{\mathbf{p}}^q) \in \Delta_r^q$ except those over a set of null measure over Δ_r^q, it holds that there exists $\delta > 0$ such that*

$$(\forall \mathscr{Q} \in I_\delta(\bar{\mathscr{Q}}))\;\; \bar{g}(\mathscr{Q}) = \bar{g}(\bar{\mathscr{Q}}),$$

where

$$I_\delta(\bar{\mathscr{Q}}) = \{\mathscr{Q} = (\mathbf{p}^1,\ldots,\mathbf{p}^q) \in \Delta_r^q \;|\; \|\mathbf{p}^\ell - \bar{\mathbf{p}}^\ell\| < \delta,\ \ell \in Q\},$$

is a neighborhood of $\bar{\mathscr{Q}}$.

As a consequence of Proposition 3, minimization of function $\bar{g}$ cannot be performed through a gradient search, since $\bar{g}$ is not even continuous. A combinatorial search through the constant pieces of $\bar{g}$ has to be performed.

3.1.3 A stochastic flow-based formulation

In the flow-based formulation, we described basic costs as deterministic quantities. This assumption may be not realistic, since different users may have a different perception of the basic costs of each arc. Indeed, even two users that are simply following a minimum distance path, may choose different paths with the same origin and destination, due to a different perception of the distance associated to each arc. Different perceptions can happen also for more qualitative road features, such as safety. To overcome this problem, in Problem (3.5a)–(3.6c) we assume that, for each arc $(i,j) \in A$, the perceived cost $\tilde{\mathbf{c}}_{ij}$ is a random variable with a normal distribution with mean $\mathbf{c}_{ij}$ and standard deviation equal to a fixed proportion of the mean, that is, $\sigma \cdot \mathbf{c}_{ij}$, with $\sigma \in (0,1)$:

$$\tilde{\mathbf{c}}_{ij} \sim \mathscr{N}\left(\mathbf{c}_{ij}, (\sigma \cdot \mathbf{c}_{ij})^2\right).$$

This stochastic model accounts for the variability in the users' perception of the basic costs. A higher variance implies greater heterogeneity in how different users perceive the same route attribute. Since the basic costs are random variables, users of the same group (that share the same weights associated to road features) may follow different paths for the same origin and destination nodes. From now on, we will denote the flow on each arc $(i,j) \in A$ as $x_{ij}(\tilde{\mathbf{c}})$, to highlight its dependence on the Gaussian random vector representing the cost.

The objective function of Problem (3.5a)–(3.6c) becomes

$$g(P) = \min_{\alpha} \mathbb{E}_{\tilde{\mathbf{c}}} \left[\sum_{(i,j) \in \bar{A}} [x_{ij}(\tilde{\mathbf{c}}) - \bar{x}_{ij}]^2 \right] \tag{3.11}$$

which is subject to the same constraints as in the original formulation, with the additional consideration that the flows $x_{ij}(\tilde{\mathbf{c}})$ are now stochastic, as they depend on the random cost $\tilde{\mathbf{c}}$.

For the simulations, we employ a Monte Carlo sampling approach. We generate N realizations of the cost vectors $\tilde{\mathbf{c}}$, each representing a possible user perception. We indicate each of them with $\tilde{\mathbf{c}}^{(s)}$, for $s \in \{1, \dots, N\}$. Note that we set the negative components of vectors $\tilde{\mathbf{c}}^{(s)}$ equal to 0. The generation of such negative components is extremely unlikely for small σ values, but we need to avoid them since Dijkstra's algorithm is not guaranteed to solve SP problems with negative costs. Alternatively, we could employ the Bellman-Ford algorithm to solve SP problems with negative costs, but this algorithm is computationally more expensive than Dijkstra's algorithm. The objective function (3.11) becomes

$$g(P) = \min_{\alpha} \frac{1}{N} \sum_{s=1}^{N} \left(\sum_{(i,j) \in \bar{A}} \left[x_{ij}(\tilde{\mathbf{c}}^{(s)}) - \bar{x}_{ij} \right]^2 \right). \tag{3.12}$$

For each sample $s \in \{1, \dots, N\}$, as in the deterministic case, the lower-level problem can be split into the following $q|W|$ subproblems: for each $\ell \in Q$ and each $w \in W$, solve:

$$\begin{aligned} \min \sum_{(i,j) \in A} & \left((\tilde{\mathbf{c}}_{ij}^{(s)})^\top \mathbf{p}^\ell \right) x_{ij}^{w,\ell} \\ \mathbf{N}\mathbf{x}^{w,\ell} &= \alpha_\ell \mathbf{U}^w \\ \mathbf{x}^{w,\ell} &\geq 0. \end{aligned}$$

The solution of this problem is obtained by first detecting the SP from o_w to d_w with cost of each arc (i, j) equal to $(\mathbf{c}_{ij}^{(s)})^\top \mathbf{p}^\ell$. Then, for each sample we denote with $\mathbf{M}^{(s)}$ the matrix defined as in (3.8), and the upper-level problem reduces to:

$$\begin{aligned} \min_{\alpha} \ & \frac{1}{N} \sum_{s=1}^{N} \|\mathbf{M}_{\bar{A}}^{(s)} \alpha - \bar{\mathbf{x}}\|^2 \\ & \sum_{\ell=1}^{q} \alpha_\ell = 1 \\ & \alpha \geq 0, \end{aligned} \tag{3.13}$$

which is again a convex quadratic problem, solvable by different available solvers. Therefore, the overall problem problem can be solved using Algorithm 1, with the only difference that lines 3–4 must be executed N times, once for each sample. From Proposition 1 it follows that the complexity of this algorithm is

$$O\left(N|W||P|\left(|A|+|V|\log|V|\right)+|P|^3L\right),$$

where L is the bit size of the input of the convex QP.

3.2 Identification via a set of known routes

In this section, we make a different assumption on the available data. We assume to know a subset of the cyclists' paths. For instance, we can obtain this information from the analysis of GPS data. As before, the objective is the identification of the weights/route choice criteria for different user groups. However, we need to consider a different objective function, that takes into account the actual individual users routes, and not the overall flows on a subset of the road segments.

Let Π be the set of directed paths of graph G, and let $\bar{\Pi} = \{\pi_i\}_{i\in\{1,\dots,t\}} \subset \Pi$ be the set of measured paths. For a path $\pi = (v_1,\dots,v_m) \in \Pi$, denote by $w_\pi = (v_1, v_m) \in V \times V$ the couple consisting of the initial and final nodes of π, i.e., $o_w = v_1$ and $d_w = v_m$. Couple w_π represents the OD pair associated to path π. Let $P = \{\mathbf{p}^\ell\}_{\ell\in Q} \subset \Delta_r$ be a finite set of coefficient vectors.

We introduce a distance $d(\pi,\mathbf{p})$ between a path $\pi \in \Pi$ and a coefficient vector $\mathbf{p} \in \Delta_r$. Namely,

$$d(\pi,\mathbf{p}) = \frac{\text{cost of path } \pi - \text{cost of the optimal path for the OD couple } w_\pi}{\text{cost of the optimal path for the OD couple } w_\pi}, \quad (3.14)$$

where the cost of each arc is computed by weighting the basic costs with coefficient vector p. By definition, $d(\pi,\mathbf{p}) \geq 0$. Moreover, $d(\pi,\mathbf{p}) = 0$ only if path π is a minimum cost path for the OD couple w_π.

To formally define d, first define function $c : \Pi \times \Delta_r \to \mathbb{R}$ as

$$c(\pi,\mathbf{p}) = \sum_{(i,j)\in\pi} \mathbf{c}_{ij}^\top \mathbf{p}.$$

Quantity $c(\pi, \mathbf{p})$ represents the cost of path π, computed by weighting the basic costs with coefficient vector $\mathbf{p}$. Then, define function $c^* : (V \times V) \times \Delta_r \to \mathbb{R}$ as

$$c^*(w, \mathbf{p}) = \min_{\pi \in \Pi \mid w_\pi = w} c(\pi, \mathbf{p}).$$

Quantity $c^*(w, \mathbf{p})$ represents the cost of the optimal path the for the OD pair w, computed by weighting the basic costs with coefficient vector p. Finally, define distance $d : \Pi \times \Delta_r \to \mathbb{R}$ as

$$d(\pi, \mathbf{p}) = \frac{c(\pi, \mathbf{p}) - c^*(w_\pi, \mathbf{p})}{c^*(w_\pi, \mathbf{p})}. \tag{3.15}$$

We define a function $f : \mathscr{P}(\Delta_r) \to \mathbb{R}$, that assigns a cost to each coefficient vector set $\hat{P} \subset \Delta_r$ as

$$f(\hat{P}) = \sum_{\pi_i \in \bar{\Pi}} \min_{\mathbf{p} \in \hat{P}} d(\pi_i, \mathbf{p}). \tag{3.16}$$

In this way, $f(\hat{P})$ represents the sum of minimum distances of all known paths from the coefficient vectors in $\hat{P}$. We are interested in the problem of minimizing $f(\hat{P})$, with a bound on the maximal cardinality of $\hat{P}$, that is

$$\min_{\hat{P} \subset \Delta_r \mid |\hat{P}| \leq k} f(\hat{P}). \tag{3.17}$$

We can also consider the approximation of problem (3.17), in which $\hat{P} \subset \bar{\Delta}_r$, where $\bar{\Delta}_r$ is a finite discretization of Δ_r, that can be defined as follows

$$\bar{\Delta}_r = \{(k_1 \delta, \dots, k_{q-1} \delta, 1 - \sum_{i=1}^{q-1} k_i \delta) \in \Delta_r \mid k_i \in \mathbb{N}, i \in \{1, \dots, q-1\}\}, \tag{3.18}$$

where $\delta > 0$ is the discretization step. Thus, the approximated problem becomes

$$\min_{\hat{P} \subset \bar{\Delta}_r \mid |\hat{P}| \leq k} f(\hat{P}). \tag{3.19}$$

In the following we show that the problem of minimizing f over a finite set $\bar{\Delta}_r$ is an instance of a k-median problem. In a k-median problem, we assume to have a set of demands I and a set of facilities J. The demands represent the users that need to access a given service. The facilities provide this service. We define a distance function $f : I \times J \to \mathbb{R}$, such that $e(i, j)$ represents the cost of serving the demand i with the facility j. The k-median problem consists in choosing k facilities in order to minimize the cost for serving all demands. That is, we want to solve

$$\min_{\hat{J} \subset J \mid |\hat{J}| \leq k} \sum_{i \in I} \min_{j \in \hat{J}} e(i, j). \tag{3.20}$$

The k-median problem was first introduced in the works by Hakimi [47, 48] and has been applied in various fields, such as cluster analysis, vehicle routing, and network topology design, among others [40]. The first mathematical programming formulation of the problem was proposed by ReVelle and Swain [100], which presented an integer programming model. This formulation had a significant impact on the literature, serving as base for many exact approaches to solving the problem (see, e.g., the paper by Church and Wang [23]). In addition, other formulations have been proposed, such as the one by Elloumi [30], which also considers the selection of the closest facility to serve each customer.

Due to its complexity, which was proven to be NP-Hard [57], several heuristic and metaheuristic approaches have been proposed to solve the k-median problem, as indicated in the paper by Mladenović et al. [83]. For instance, greedy methods start with an empty set of open facilities and add facilities one by one, choosing the ones that most reduce the total cost, until the desired number of k facilities is reached. Other heuristic approaches for the k-median problem include the alternate method [77], the dual ascent heuristic (DUALOC), based on the relaxed dual of the integer programming formulation of the problem, and proposed in the works by Eugénia Captivo [32] and Galvão [37]. Classical metaheuristic procedures, such as GRASP [99] and Simulated Annealing [85], have also been applied to solve the k-median problem. For more details about the problem, we refer, e.g., to the researches by Church and Wang [23], Goldengorin et al. [40], and Mladenović et al. [83]. Now, we can make the following remark.

Remark 3.2.1. *Problem* (3.17) *is a specific instance of Problem* (3.20)*, with demand set* $I = \{1, \ldots, t\}$ *(the index set of the paths* $\bar{\Pi}$*), facilities* $J = \{1, \ldots, s\}$ *(the index set of the weights in* $\bar{\Delta}_r$*), and cost function* $e(i, j) = d(\pi_i, p^j)$*.*

Therefore, function f defined in (3.16) can also be rewritten as

$$f(\hat{P}) = \sum_{\pi_i \in \bar{\Pi}} \min_{\mathbf{p}^\ell \in \hat{P}} e(i,\ell). \tag{3.21}$$

In the following, we will employ matrix $E \in \mathbb{R}^{|\bar{\Pi}|} \times \mathbb{R}^{|\hat{P}|}$, whose (i,j)-entry is $E_{i,j} = e(i,j)$.

For the sake of completeness, we give the short proofs of two properties (monotonicity and supermodularity) of function f, shared by the objective functions of all k-median problems.

Proposition 4. *Let $P_1 \subseteq P_2$. Then, $f(P_2) \leq f(P_1)$.*

Proof. It is enough to observe that if $P_1 \subseteq P_2$, then for all $\pi_i \in \bar{\Pi}$

$$\min_{\mathbf{p}^\ell \in P_2} e(i,\ell) \leq \min_{\mathbf{p}^\ell \in P_1} e(i,\ell),$$

and so, by definition (3.21), it follows that $f(P_2) \leq f(P_1)$. □

Proposition 5. *Function f is a supermodular function, i.e., it satisfies the following condition:*

$$f(P_1) - f(P_1 \cup \{\mathbf{p}^{\bar{\ell}}\}) \geq f(P_2) - f(P_2 \cup \{\mathbf{p}^{\bar{\ell}}\}). \tag{3.22}$$

for any $P_1 \subseteq P_2 \subseteq P$ and $\mathbf{p}^{\bar{\ell}} \in P \setminus P_2$.

For the proof of this proposition, see Appendix B.

Now, let $\mathscr{Q} = (\mathbf{p}^1, \ldots, \mathbf{p}^k) \in \Delta_r^k \subset \mathbb{R}^{rk}$ be an ordered set. Define a function $\tilde{f} : \Delta_r^k \to \mathbb{R}$ by $\tilde{f}(\mathbf{p}^1, \ldots, \mathbf{p}^k) = f(\{\mathbf{p}^1, \ldots, \mathbf{p}^k\})$. It is possible to prove an approximation result, for which we first need to prove two lemmas. The first one states that for any element of Δ_r^k, there exists an element belonging to the discretized set, which differs from it by at most a constant that is directly proportional to $\sqrt{rk}$ and to the discretization step δ.

Lemma 3.2.2. *For any $\mathscr{Q} \in \Delta_r^k$, there exists $\tilde{\mathscr{Q}} \subset \bar{\Delta}_r^k$ such that $\|\mathscr{Q} - \tilde{\mathscr{Q}}\| \leq \delta\sqrt{rk}$.*

Proof. It is enough to observe that for each $\mathbf{p} \in \Delta_r$ there exists $\tilde{\mathbf{p}} \in \bar{\Delta}_r$ such that $\|\mathbf{p} - \tilde{\mathbf{p}}\| \leq \delta\sqrt{r}$. □

The second one states that for any weight vector $\mathbf{p}$, the total cost of a path with this weight is always at least as large as the minimum basic cost of that path.

Lemma 3.2.3. *For each $\mathbf{p} \in \Delta_r$ and each path π, it holds that:*

$$\sum_{(i,j)\in\pi} \mathbf{c}_{ij}^\top \mathbf{p} \geq \min_{h\in\{1,\ldots,r\}} \sum_{(i,j)\in\pi} c_{ij}^h.$$

Proof. We have that

$$\sum_{(i,j)\in\pi} \mathbf{c}_{ij}^\top \mathbf{p} = \sum_{h=1}^{r} p_h \left(\sum_{(i,j)\in\pi} c_{ij}^h \right).$$

Then,

$$\sum_{(i,j)\in\pi} \mathbf{c}_{ij}^\top \mathbf{p} \geq \min_{\mathbf{p}\in\Delta_r} \sum_{(i,j)\in\pi} \mathbf{c}_{ij}^\top \mathbf{p} \geq \min_{h\in\{1,\ldots,r\}} \sum_{(i,j)\in\pi} c_{ij}^h.$$

□

Now, let us denote with W the set of all possible OD pairs, and, for a given OD pair $w \in W$, let Π_w be the set of all paths associated to such pair. For $\pi \in \bar{\Pi}$, we denote with w_π the corresponding OD pair, and we set:

$$c_{\max}(\pi) = \max_{h\in\{1,\ldots,q\}} \sum_{(i,j)\in\pi} c_{ij}^h, \qquad c_{\min}(\pi) = \min_{\pi'\in\Pi_{w_\pi}} \min_{h\in\{1,\ldots,q\}} \sum_{(i,j)\in\pi'} c_{ij}^h.$$

Note that, assuming that all costs associated with the arcs are positive, we have that $c_{\min}(\pi) > 0$. Finally, let us set

$$M = \max_{\pi\in\bar{\Pi}} \frac{c_{\max}(\pi)}{c_{\min}(\pi)}. \tag{3.23}$$

Now, we are ready to introduce the approximation result, which shows that the difference between the optimal value of the original problem and that of the approximated problem is bounded by a constant that is directly proportional to the discretization step, the number of paths, and the factor $\sqrt{rk}$.

Proposition 6. *Let us assume that* $\mathbf{c}_{ij} > 0$ *for all arcs* $(i, j) \in A$. *Let* $P^\star$ *be an optimal solution of problem* (3.17) *and* $\bar{P} \subset \bar{\Delta}_r$ *be an optimal solution of problem* (3.19). *Let us denote with* δ *the discretization step of* $\bar{\Delta}_r$. *Then:*

$$f(\bar{P}) - f(P^\star) \leq |\bar{\Pi}| M \, \delta \sqrt{rk}, \tag{3.24}$$

where M *is defined in* (3.23).

3.2.1 A stochastic trajectory-based formulation

As already done in Section 3.1.3 for the flow-based formulation, we now model the basic costs as a random variable, since each user may have a different perception. Again, we assume that costs follow a Gaussian distribution with mean $\mathbf{c}_{ij}$ and standard deviation equal to a fixed proportion of the mean, that is, $\sigma \cdot \mathbf{c}_{ij}$, with $\sigma \in (0, 1)$:

$$\tilde{\mathbf{c}}_{ij} \sim \mathcal{N}\left(\mathbf{c}_{ij}, (\sigma \cdot \mathbf{c}_{ij})^2\right).$$

From now on, we indicate the distance defined in (3.15) with $d(\pi, \mathbf{p}, \tilde{\mathbf{c}})$ to highlight the dependence on the random cost variable. Therefore, the objective function f defined in (3.16) now becomes:

$$f(\hat{P}) = \mathbb{E}_{\tilde{\mathbf{c}}}\left[\sum_{\pi_i \in \bar{\Pi}} \min_{\mathbf{p} \in \hat{P}} d(\pi, \mathbf{p}, \tilde{\mathbf{c}})\right]. \tag{3.25}$$

As in Section 3.1.3, for the simulations we adopt a Monte Carlo sampling approach by generating N realizations of the cost vectors. We indicate each realization with $\tilde{\mathbf{c}}^{(s)}$, for $s \in \{1, \ldots, N\}$. In this case, the objective function of the stochastic trajectory-based formulation is function $f : \mathscr{P}(\Delta_r) \to \mathbb{R}$, that assigns a cost to each coefficient vector set $\hat{P} \subset \Delta_r$ as

$$f(\hat{P}) = \frac{1}{N} \sum_{s=1}^{N} \left(\sum_{\pi_i \in \bar{\Pi}} \min_{\mathbf{p} \in \hat{P}} \tilde{d}(\pi, \mathbf{p}, \tilde{\mathbf{c}}^{(s)})\right). \tag{3.26}$$

It is easy to prove that the function just defined is monotone and supermodular. Indeed, this is defined as the sum of monotone and supermodular functions.

3.3 Solution approaches

This section describes an algorithm for the identification via known flows over a subset of arcs (the flow-based formulation discussed in Section 3.1), and two procedures for the identification via a set of known routes (the trajectory-based formulation discussed in Section 3.2).

3.3.1 An algorithm for the identification via known flows over a subset of arcs

We propose Algorithm 2 for the identification problem when flows over a subset of arcs are available. The algorithm exploits the properties of objective function g proved in Section 3.1.2.

At first, we consider a sparse grid of weights P_1 over Δ_r (line 1), and we compute $g(P_1)$, as well as the associated vector of probabilities α^{P_1}, through Algorithm 1 (line 2). Next, we enter the loop at lines 3–10. Following Corollary 3.1.1, at line 4 we consider a reduced set of weights $\bar{P}$, obtained by removing from P_t weights with probability not larger than a given threshold ε.

Next, at line 5, following Proposition 2, we perturb the current weights in $\bar{P}$ through the procedure PERTURBATION, whose second argument $\frac{1}{2^t}$ defines the size of the perturbation (decreasing as the iteration counter t increases). The newly generated weights are collected in the set P^{new}. Different possible implementations of the procedure PERTURBATION are possible. For the case discussed in Section 3.4, with $r = 3$, the procedure PERTURBATION returns the following set:

$$P^{\text{new}} = \left\{ \mathbf{p} + (i, j, -i-j)\frac{1}{2^t} \in \Delta_3 \ \middle|\ \mathbf{p} \in \bar{P},\ i, j \in \{-1, 0, 1\},\ i \neq 0 \vee j \neq 0 \right\}.$$

At line 6 we identify a set of weights $P^{\text{del}} \subset P_t \setminus \bar{P}$ to be discarded. These are weights with negligible probability and distance from $\bar{P}$ not lower than $\frac{1}{2^{t-1}}$. At line 7 we define the new collection P_{t+1}, and at line 8 we run Algorithm 1 to compute $g(P_{t+1})$ and the vector of probabilities $\alpha^{P_{t+1}}$. At the end of each iteration we double the value of threshold ε (line 9). The stopping condition of the loop (see line 10) is fulfilled either if ε becomes larger than a predefined tolerance value tol_1, or when the improvement from one iteration to the next is not large enough, i.e., when the new value of g is larger than a fraction tol_2 times the value of g at the previous iteration.

Procedure FIND CLUSTERS(P_t, k), called at line 11, returns the set B of the centroids of the k clusters identified within the collection of weights P_t returned by the loop. It also returns in vector *radius* the radii of the identified clusters. The FIND CLUSTERS(P_t, k) function can be based on various clustering algorithms, such as k-means with Euclidean distance, or the Hierarchical clustering. The latter only requires the set of weights P_t as input, whereas the former also requires specifying the number k of clusters into which the data should be partitioned.

Note that for a given cluster $C \subset P_t$, its centroid is $b_C = \sum_{\mathbf{p}\in C} \alpha_{\mathbf{p}}^{P_t} \mathbf{p}$, i.e., the centroid is computed by taking into account also the probabilities associated to the weights $\mathbf{p}$. The collection of weights is thus reduced from P_t to B, and at line 12 we employ the set of centroids to initialize a reduced set of weights P^*. The last part of the algorithm (`For` loop at lines 13–17) is a consequence of Proposition 3. Indeed, such proposition suggests the need for a combinatorial search in the neighborhood of the current weights to try to reduce g by refining each member of P^* through a local search.

At line 14, we associate to each weight $\mathbf{p}^\ell \in P^*$ a value ρ initialized with the radius of the corresponding cluster. Next, we enter a `While` loop. At each iteration, we perform, at line 16, a local search within the neighborhood $N_\rho(\mathbf{p}^\ell)$ of $\mathbf{p}^\ell$. The neighborhood can be defined in a discrete manner by considering all weight vectors in Δ_3 that lie within radius ρ from p^ℓ:

$$N_\rho(\mathbf{p}^\ell) = \left\{ \mathbf{p}^\ell + (i, j, -i-j)\rho \in \Delta_3 \;\middle|\; i, j \in \{-1, 0, 1\} \right\}.$$

The local search algorithm (see Algorithm 3) explores the neighboring weights in order to find one that decreases the objective function value (or establishes that none exists). If the local search identifies a new weight $\mathbf{p} \in N_\rho(\mathbf{p}^\ell)$ such that $g(P^* \cup \{\mathbf{p}\} \setminus \{\mathbf{p}^\ell\}) < g(P^*)$, then P^* is updated into $P^* \cup \{\mathbf{p}\} \setminus \{\mathbf{p}^\ell\}$. Otherwise, we halve the search radius ρ. Proposition 3 states that function g is constant in a sufficiently small neighborhood of P^*. Thus, the loop is stopped as soon as the search radius falls below a given threshold tol_3. By Proposition 1, a single iteration of Algorithm 3 has complexity $O\left(N|W||P|\left(|A|+|V|\log|V|\right)+|P|^3L\right)$ since it solves Problem (3.10) once for each weight in the neighborhood and $|N_\rho(\mathbf{p}^\ell)| \leq 6$. The number of iterations of the local search (loop at lines 15–17) is finite, but it depends on how many times an improved solution is detected at line 5 of Algorithm 3, which cannot be established in advance. However, according to our experiments, the computational impact of the local search is negligible, while the major computational cost comes from the computation of the loop at lines 3–10 of Algorithm 2.

Algorithm 2: Cyclists' route choice identification algorithm

Input: $G=(V,A)$, W, $[\bar{x}_{ij},(i,j)\in\bar{A}]$, $[u^w, w\in W]$, k, $tol_1, tol_2, tol_3 \in [0,1]$

Output: α^{P^*}, P^*

1 Initialize P_1, $\varepsilon \leftarrow 10^{-5}$, $t \leftarrow 1$;

2 $[\alpha^{P_1}, g(P_1))] \leftarrow$ IDENTIFICATION(P_1) ;

3 **repeat**

4     $\bar{P} \leftarrow \{p \in P_t \mid \alpha_p^{P_t} > \varepsilon\}$;

5     $P^{\text{new}} \leftarrow$ PERTURBATION($\bar{P}, \frac{1}{2^t}$) ;

6   P^{del} is the set of weights in $P_t \setminus \bar{P}$ such that the minimum distance from $\bar{P}$ is greater than $\frac{1}{2^{t-1}}$;

7     $P_{t+1} \leftarrow P_t \cup P^{\text{new}} \setminus P^{\text{del}}$;

8     $[\alpha^{P_{t+1}}, g(P_{t+1}))] \leftarrow$ IDENTIFICATION(P_{t+1}) ;

9     $\varepsilon \leftarrow 2\varepsilon$, $t \leftarrow t+1$;

10 **until** $\varepsilon < tol_1$ *and* $g(P_{t+1}) \leq tol_2 \cdot g(P_t)$;

11 $[B, radius] \leftarrow$ FIND CLUSTERS(P_t, k) ;

12 $P^* \leftarrow B$;

13 **for** $\mathbf{p}^\ell \in P^*$ **do**

14     $\rho \leftarrow radius(\ell)$;

15   **while** $\rho > tol_3$ **do**

16         $[P_{out}, \rho_{out}] \leftarrow$ LOCAL SEARCH($P^*, \mathbf{p}^\ell, \rho$) ;

17         $P^* \leftarrow P_{out}$, $\rho \leftarrow \rho_{out}$;

Algorithm 3: LOCAL SEARCH

Input: P^*, $\mathbf{p}^\ell$, ρ

Output: P_{out}, ρ_{out}

$N_\rho(\mathbf{p}^\ell) \leftarrow$ PERTURBATION$(\{\mathbf{p}^\ell\}, \rho)$;

$P_{out} \leftarrow P^*$, $\rho_{out} \leftarrow \rho$;

for $\mathbf{p} \in N_\rho(\mathbf{p}^\ell)$ **do**

  $\bar{P} \leftarrow P^* \cup \{\mathbf{p}\} \setminus \{\mathbf{p}^\ell\}$;

  if $g(\bar{P}) < g(P_{out})$ **then**

    $P_{out} \leftarrow \bar{P}$

if $P_{out} = P^*$ **then**

  $\rho_{out} \leftarrow \rho/2$

3.3.2 Greedy approach for the identification via a set of known routes

We consider the trajectory-based formulation described in Section 3.2. To solve it, we first discretize the feasible region. In particular, we define the finite discretization $\bar{\Delta}_r$ of Δ_r as in (3.18). We present Algorithm 4 to detect a suboptimal solution of problem (3.19). Recall that any optimal solution of problem (3.19) is also an approximate solution of the original problem (3.17) as stated in Proposition 6.

First, we compute the cost $e(i,\ell)$ for each route $\pi_i \in \bar{\Pi}$ and for each weight $\mathbf{p}^\ell \in \bar{\Delta}_r$ (line 1). This requires calculating the SP associated to the OD pair of each route $\pi_i \in \bar{\Pi}$ a number of times equal to $|\bar{\Delta}_r|$. Moreover, in case the objective function (3.21) is employed to take into account randomness of the basic costs, then the same calculations must be repeated for all the N realizations of the random vectors associated to the basic costs. Then, we proceed with the greedy approach. This is one of the most classical approaches for solving k-median problems. It comes in two distinct versions: the ascent and descent version. In the former, we start with an empty set of weights, and at each iteration we add the weight which minimizes f when added to the current set of weights. We stop as soon as the prefixed number k of weights is reached. In the descent version, we start with the whole set of weights $\bar{\Delta}_r$, and at each iteration we remove the weight which minimizes f when removed from the current set of weights. Also in this case we stop as soon as the prefixed number k of weights is reached.

In the greedy *ascent* approach, we start by choosing the single weight which minimizes function f, that is, we solve the optimization problem (3.19) with $k=1$. Then, we add new weights one at a time. In particular, we proceed as follows (see Algorithm 4). We initialize $P_0 = \varnothing$ (line 2), i.e., we start with an empty set of weights. Then, the algorithm repeats the following assignment for $i \in \{0,\dots,k-1\}$ (`For` loop at lines 3–4)

$$P_{i+1} = P_i \cup \left\{ \arg \min_{\mathbf{p}^\ell \in \bar{\Delta}_r \setminus P_i} f(P_i \cup \{\mathbf{p}^\ell\}) \right\}.$$

In other words, at iteration i we add to the current set of weights P_i the weight in $\bar{\Delta}_r \setminus P_i$ which minimizes function f when added to P_i. The suboptimal solution obtained by this procedure is $P := P_k$ (line 5). It is possible to improve the solution by performing a local search (lines 6–11). Finally, we assign a weight in P to each path in Π (line 12).

In the greedy *descent* approach, we initialize the set of weights with $\bar{\Delta}_r$, and at each iteration, we remove one weight until we reach the required cardinality. More precisely, in Algorithm 4, we need to modify line 2, where we set $P_0 = \bar{\Delta}_r$, and in the `For` loop (lines 3–4), set P_{i+1} is defined as follows

$$P_{i+1} = P_i \setminus \left\{ \arg \min_{\mathbf{p}^\ell \in P_i} f(P_i \setminus \{\mathbf{p}^\ell\}) \right\} \qquad \text{for } i \in \{1,\dots,(|\bar{\Delta}_r| - k - 1)\},$$

i.e., at iteration i we remove from the current set of weights P_i the weight in P_i which minimizes function f when removed from P_i. Furthermore, the suboptimal solution obtained by this procedure is $P := P_{|\bar{\Delta}_r|-k}$ (line 5). Since function f is a supermodular function, the greedy descent algorithm for the minimization problem provides an approximation guarantee, as proved by Il'ev [54]. In particular, denoting with P^* the optimal solution of problem (3.19), we have that:

$$f(P) \leq \left(\frac{e^t - 1}{t} \right) f(P^*),$$

where e is the base of the natural logarithm, and $t \in [0, +\infty)$ is a constant related to the problem, referred to as *steepness*.

Algorithm 4: Greedy ascent approach for the identification via a set of known paths

---

Input: $G = (V, A)$, W, $\bar{\Pi}$, ρ, r, k

Output: P

1 Compute cost matrix E ;

2 Initialize $P_0 = \varnothing$;

3 **for** $i \in \{0, \ldots, k-1\}$ **do**

4     $P_{i+1} \leftarrow P_i \cup \{\arg\min_{\mathbf{p}^\ell \in \bar{\Delta}_r \setminus P_i} f(P_i \cup \{\mathbf{p}^\ell\})\}$;

5 $P \leftarrow P_k$;

6 **repeat**

7     $OptValue \leftarrow f(P)$;

8   **for** $\mathbf{p}^\ell \in P$ **do**

9         $[P_{out}, \rho_{out}] \leftarrow$ LOCAL SEARCH$(P, \mathbf{p}^\ell, \rho)$;

10         $P \leftarrow P_{out}$;

11 **until** $f(P) < OptValue$;

12 $P \leftarrow$ ASSIGN WEIGHT TO PATH$(\bar{\Pi}, P)$;

Concerning the complexity of Algorithm 4, we have already observed that computing the cost matrix E at line 1 requires the solution of $N|\bar{\Pi}||\bar{\Delta}_r|$ SP problems, so that the complexity of this operation, if Dijkstra's algorithm is employed to solve the SP problems, is: $O\left(N|\bar{\Pi}||\bar{\Delta}_r|\left(|A| + |V|\log(|V|)\right)\right)$ (recall that N is the number of the realizations of the cost vectors). The greedy algorithm (loop at lines 3–2) requires a time $O(k|\bar{\Delta}_r|)$ for the greedy ascent version, and $O(|\bar{\Delta}_r|^2)$ for the greedy descent version. A more detailed analysis of the sensitivity of the algorithm to variations in the cardinality of the discretization set $\bar{\Delta}_r$ is provided in Appendix C. Finally, the number of iterations of the local search (loop at lines 6–11) is not fixed. Trivial upper bounds for the number of iterations are available. For instance, since the number of improvements cannot be larger than all possible subsets of $\bar{\Delta}_r$ with cardinality k, an upper bound is $\binom{|\bar{\Delta}_r|}{k}$. But such bounds are much larger than the number of observations observed in practice. In fact, according to our experiments, the computational impact of the local search is negligible, while the major computational cost comes from the computation of the cost matrix E.

3.4 Computational experiments

This section describes the computational experiments performed with Algorithms 2 and 4 (ascent and descent versions), both implemented in `MATLAB`. The first algorithm was tested on grid graphs with 1600 nodes using synthetic data, while the other two algorithms were evaluated on the road network of the city of Parma, using data provided by a bike-sharing service. All experiments were performed on a machine equipped with an Intel® Core™ i7-4510U processor (2.60 GHz) and 16 GB of RAM.

3.4.1 Results for the synthetic instances

For the flow-based formulation we do not have available data coming from a real road network. Therefore, to test Algorithm 2, we used synthetic data, for a grid network $G = (V,A)$ with 1600 nodes, and a set W made up of 1000 OD pairs. For each $w \in W$, we set the demand value $u_w = 10$. We considered the case $r = 3$, i.e., we employed three distinct basic cost functions c^1, c^2, c^3, corresponding, e.g., to distance, safety and environmental conditions costs. Each component of these vectors is an integer randomly generated in $[5,20]$. Such vector costs have been normalized in such a way that the sum of their components is the same for all of them, i.e., $\sum_{(i,j)\in A} c^1_{ij} = \sum_{(i,j)\in A} c^2_{ij} = \sum_{(i,j)\in A} c^3_{ij}$.

We generated 100 instances for which an optimal set of weights P_{ref} and the related vector of probabilities $\alpha^{P_{\text{ref}}}$ are known in advance. To this end, P_{ref} is made up of five distinct weights $\mathbf{p} \in \Delta_3$. The distance between each pair of weights in P_{ref} is not lower than 0.05 (weights are far enough from each other). We randomly generate the vector of probabilities $\alpha^{P_{\text{ref}}}$, imposing that each component of this vector is not lower than 0.05 (later, we will also consider cases with groups having small probabilities).

Through (3.7)–(3.9) with $P = P_{\text{ref}}$ and $\alpha = \alpha_{\text{ref}}$, we are able to compute the values $\bar{x}_{ij}$ for all $(i, j) \in A$.

Next, we randomly select a subset of arcs $\bar{A} \subset A$, with cardinality $|\bar{A}| = 0.4|A|$. It turns out that $g(P_{\text{ref}}) = 0$, so that P_{ref} and $\alpha^{P_{\text{ref}}}$ are the target solutions for these instances. We set the following tolerance values $tol_1 = 0.01$, $tol_2 = 0.85$, and $tol_3 = 0.005$, and we run Algorithm 2. We solve the Shortest Path problems through the `Matlab` routine implementing Dijkstra's algorithm, while we solve convex QPs through `Gurobi` called via `Yalmip`. We employ the `Matlab` routine for Hierarchical Clustering to identify clusters.

The results of the simulations show that the final values of the objective function ($g(P^*)$) are much lower than the initial ones ($g(P_1)$). Indeed, while the average of the latter is 861060, the average of the former decreases to 267.74. Given that the graph under consideration comprises 6240 edges, and that the objective function is defined as the sum of squared deviations over the observed edges (which correspond to 40% of the total), it follows that the mean squared error per edge is 0.1. This suggests that Algorithm 2 is quite effective in reducing the values of function g.

Note that the largest improvement is due to the first `While` loop. In fact, when, at the end of this loop, we replace the final set P_t with the set B of centroids of its clusters, we even have a small increase of g, but such increase is counterbalanced by the advantage of having significantly reduced the cardinality of the set of weights. With the final local search we are able to further refine the set of weights and slightly reduce the value of g.

Figure 3.1 displays the distribution of the Euclidean distances between the target weights P_{ref} and the weights P^* returned by the algorithm. Specifically, the distance between the two sets of weights is computed by solving an assignment problem that pairs each weight in one set with a corresponding weight in the other set. Once the optimal matching is determined, the Euclidean distances between the paired weights are evaluated, and the overall distance is defined as the L_2 norm of the resulting distance vector. With one exception, such distances are rather small. The computational costs of the algorithm are mainly due to the calls of the procedure IDENTIFICATION. The main steps of this procedure require the solution of SP problems and of convex

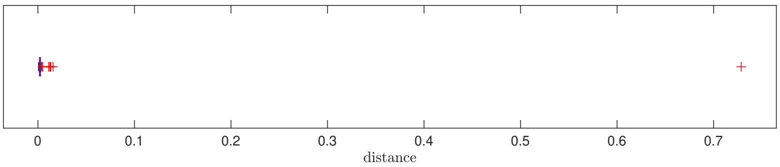


Figure 3.1: Distribution of the distances between the real weights and the weights returned by Algorithm 2.

QPs. Figure 3.2 compares the cumulative time needed by the solutions of SP problems (upper picture) and by the solutions of convex QPs (lower picture). It is clear that for these instances computing times are mostly determined by the solutions of the SP problems.

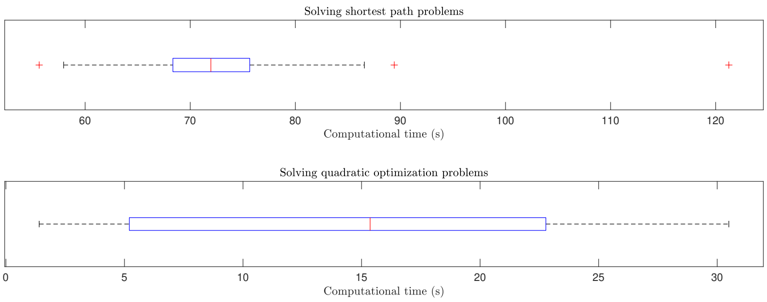


Figure 3.2: Computational time to solve the SP problems compared to the computational time to solve the convex QPs.

We have also explored the effect of class imbalance on the stability of the inferred weights and class shares. To this purpose, we performed a series of tests on 100 instances generated as follows. Again, P_{ref} is made up of five distinct weights $\mathbf{p} \in \Delta_3$ and the distance between each pair of weights in P_{ref} is not lower than 0.05. We randomly generate the vector of probabilities $\alpha^{P_{\text{ref}}}$, but, this time, imposing that each component of this vector is not lower than 0.01, and that at least one component is less than or equal to 0.05. In this way, for each instance we considered at least one user group of rather small size. From the obtained results, we can observe that in some cases the algorithm fails to identify the small-sized group, although in most cases it succeeds. However, the distance between the obtained solution and the optimal one remains very small, as shown in Figure 3.3. By comparing this figure with Figure 3.1, we observe that the median of the distances slightly increases and that more outliers appear, although overall the distance remains low. It should be noted that the employed distance is the Euclidean one, which does not take into account the percentages associated with each group, but depends solely on the position of the groups within the simplex Δ_3.

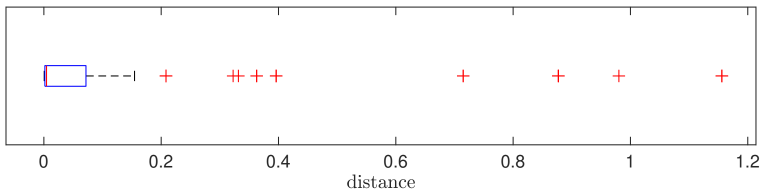


Figure 3.3: Distributions of the distances between real weights and weights returned by Algorithm 2 in the case with groups of very small size.

We also performed some tests to evaluate the impact of the choice of parameter k in Algorithm 2. To carry out this test, we generated 10 different instances on the same graph, varying the costs, the observed arcs, and the OD pairs. However, we kept the number of route choice criteria (i.e., the cardinality of P_{ref}) equal to 5 across all instances. We solved each instance with Algorithm 2 by varying k from 1 to 10. In Figure 3.4 we report the final values of the objective function. Note that they decrease up to $k = 5$, and then remain approximately constant, which shows that, as expected, increasing k beyond the cardinality of P_{ref} does not give any advantage.

Finally, we conducted a series of sensitivity tests to assess the impact of the relative size of subset $\bar{A}$ with respect to the full set of arcs A, on the accuracy of the results. The size of the subset of arcs $\bar{A}$ reflects how many observations of street flow are available, which in turn affects the inference of the route choice criteria. In these

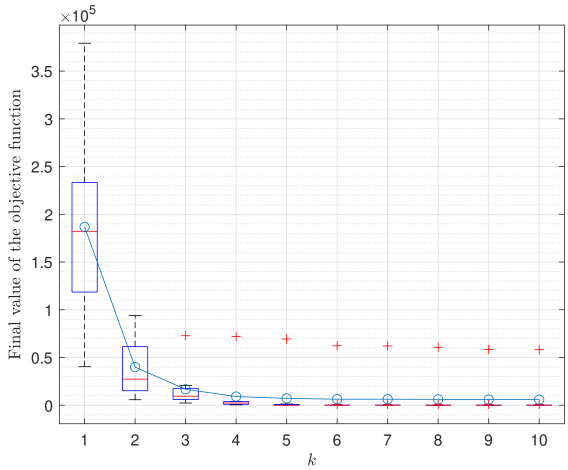


Figure 3.4: Objective function values for different numbers of clusters k.

tests, we used a sequence of nested subsets of A (each one contained within the next larger set) with sizes corresponding to percentages of the total number of arcs in A ranging from 1% to 50%. Figure 3.5 shows the distributions of the distances between the optimal solutions and those obtained with the algorithm, as the cardinality of set $\bar{A}$ varies. We observe that starting from a size of $\bar{A}$ equal to 10% of the total number of links in A, the estimated weights are very close to the optimal set of weights P_{ref} (the distance is always below 0.1), while as the percentage of observed flows falls below 10% of the total number of arcs, the distance between estimated and optimal set of weights becomes more significant.

Stochastic problem

We perform some experiments to evaluate the impact of introducing randomness in the basic costs to account for the different users' perception of such costs. To this end we solve the stochastic formulation presented in Section 3.2.1 where costs are modeled as random variables with a Gaussian distribution of the form $\tilde{\mathbf{c}}_{ij} \sim \mathcal{N}\left(\mathbf{c}_{ij}, (\sigma \cdot \mathbf{c}_{ij})^2\right)$. We test three cases with increasing variability $\sigma \in \{0.05, 0.1, 0.3\}$.

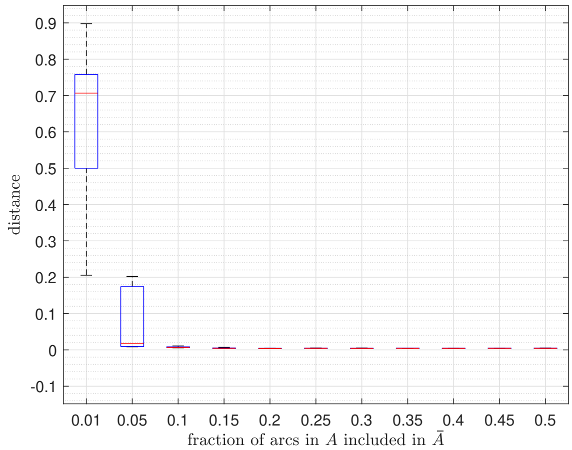


Figure 3.5: Distributions of the distances between real weights and weights returned by Algorithm 2 with varying $|\bar{A}|$.

The tests are performed on a 625 node grid network $G = (V, A)$, using the same configuration specified in Section 3.4.1. To generate the observed flows, as before we define in advance the optimal set of weights P_{ref} and the related vector of probabilities $\alpha^{P_{\text{ref}}}$, and assume that users choose the shortest path connecting their respective OD pair, based on the deterministic arc costs $\mathbf{c}_{ij}$.

To solve these instances we used Algorithms 1 and 2 with $N = 50$ perturbations. In Figure 3.6 we show the distances between the solutions returned by these algorithms and the optimal ones. The figure shows that as the standard deviation increases, the distance from the optimal solutions grows slightly, but the average value remains between 0.1 and 0.2.Overall these tests confirm the robustness of the solutions, which do not deviate significantly from the optimal ones even when we increase the variability of the basic costs.

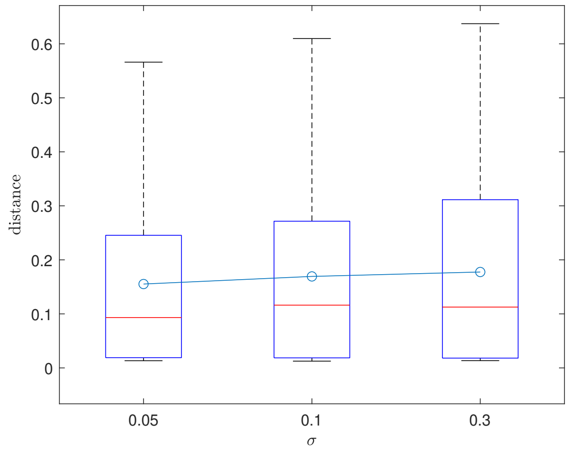


Figure 3.6: Distributions of the distances between real weights and weights returned by Algorithm 2 with varying σ.

3.4.2 Results for the Parma instances

We test Algorithm 4 to evaluate the behavior of users over the road network of the city of Parma, Italy. This is represented by a graph with 7953 nodes and 11649 edges. After giving a quantitative evaluation of different properties of the road network as described in Appendix A (namely, for each road segment we evaluated its distance, safety, and practicability), we use the information on the paths followed by bike-sharing users, and through these data we formulate the optimization problem described in Section 3.2. Next, we solved this problem through Algorithm 4 (both ascent and descent greedy approaches), using the finite discretization $\bar{\Delta}_3$ of Δ_3 as in (3.18) with discretization step $\delta = \frac{1}{32}$. More specifically, we gathered approximately 2000 user routes from the bike-sharing system of the bike-sharing provider `Dott`, a company operating in Parma. Figure 3.7 showd the paths provided by `Dott`. The color of the roads is based on the number of users that have passed through them. White represents roads where none of the users from the sample have passed, while darker colors are used for higher numbers of users, up to a maximum of 200 users.

From now on, we will denote by $\mathbf{p}^1$, $\mathbf{p}^2$, and $\mathbf{p}^3$ the weights associated exclusively with a single one of the three basic costs. Specifically:

- $\mathbf{p}^1 = (1,0,0)$ represents a criterion based solely on distance.
- $\mathbf{p}^2 = (0,1,0)$ represents a criterion based solely on safety.
- $\mathbf{p}^3 = (0,0,1)$ represents a criterion based solely on practicability.

Some pre-processing was required to remove outliers. Outliers are represented, e.g., by users utilizing bike-sharing for leisure rides around the city, with starting and ending point of their journeys nearly in the same location. Such users are not selecting a 'shortest' path, where the path is shortest with respect to a suitable combination of the three key indicators (length, safety, and practicability) that we have taken into account. We define an outlier as a path π_i such that $e(i,1) > 1$, where $\mathbf{p}^1$ is the weight associated to the distance. In other words, a path is classified as an outlier if its length exceeds twice the length of the SP. We denote this set of outliers with O_1:

$$O_1 = \{\pi_i \mid e(i,1) > 1\}.$$

According to this definition, 8.3% of the available paths were categorized as outliers and therefore excluded from the study on user decision-making criteria.

Another set of outliers O_2 consists of users who select the best route based on indicators other than distance, safety, and practicability. For instance, a user might choose a route based on how well-lit it is, the presence of parks, pollution levels, or other factors not considered in this study. These routes should therefore be excluded, as they are not relevant for identifying behavior based on the costs considered in our analysis. Within the same group of outliers we also include users who are not 'optimizing' some cost function but are simply roaming without a specific objective. More precisely, we define O_2 as follows:

$$O_2 = \left\{\pi_i \mid \min_{\ell \mid \mathbf{p}^\ell \in \bar{\Delta}_r} e(i,\ell) > 1\right\}.$$

After the removal of outliers, the subset of routes, denoted as $\bar{\Pi}$, contains 1715 routes.

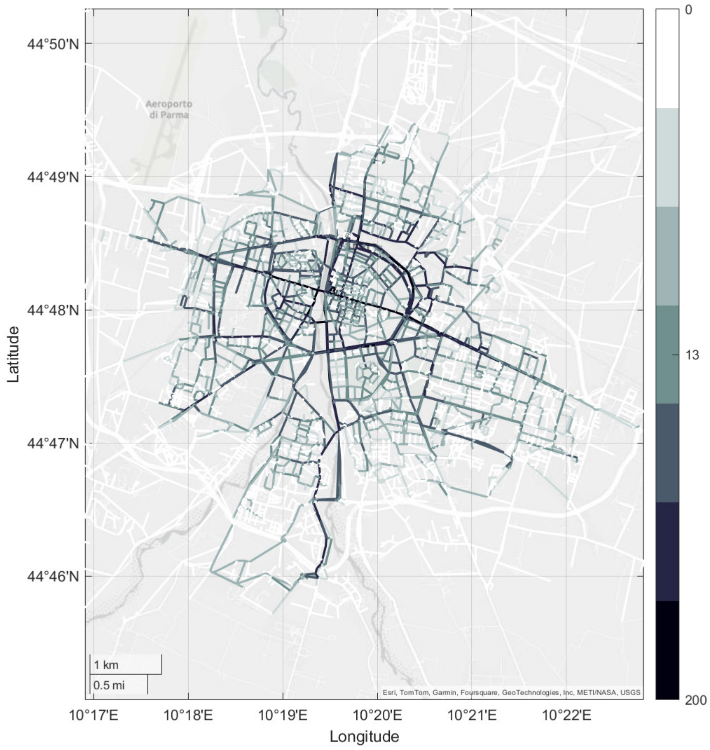


Figure 3.7: Paths provided by company Dott in Parma, Italy.

One important aspect we needed to address was the selection of the number k of weights to identify. This problem is conceptually similar to determining the number of clusters in a clustering task. In fact, choosing the number of weights corresponds to deciding how many behavioral groups the users should be divided into. Clearly, the best fit to the data would be obtained by assigning one weight vector per user (i.e., dividing users into $|\bar{\Pi}|$ groups), but this is not our goal. Instead, we aim to group users in a way that clusters together those with similar route choice behavior. The objective is to strike a balance between model accuracy and interpretability, by identifying a small number of representative behaviors that still allow us to achieve a good value of the objective function.

To this end, we used `dendrogram`, a `Matlab` function which generates a visual representation of a hierarchical clustering result, showing how clusters are formed at each step by merging or splitting data points. It takes as input a linkage matrix, typically produced by the `linkage` function, which encodes pairwise distances between data points. A dendrogram consists of many U-shaped lines that connect data points in a hierarchical tree. The height of each U represents the distance between the two data points being connected. By cutting the dendrogram at different heights, we can determine the optimal number of clusters. Figure 3.8 shows the dendrogram obtained on the weights in $\bar{P}$. Note that cutting the dendrogram at a height between 0.045 and 0.075 results in exactly $k = 9$ clusters. The only parameter required for this hierarchical clustering is the height h at which the dendrogram is cut. In our case, we chose $h = 0.05$.

Once we have chosen the value of k, we can proceed by solving problem (3.20). First, we solved it using the greedy ascent approach (Algorithm 4). In Figure 3.9 we show the value of the objective function for $k \in \{3, \dots, 20\}$ before and after applying local search. In this case, the local search does not significantly improve the solution, as the greedy approach already closely approximates the final solution of the algorithm. We also notice that the final objective function value decreases fast up to $k = 9$, and then it still decreases but more slowly. This is a further confirm that setting $k = 9$ is a good choice in terms of interpretability. Note that the final value of the objective function is relatively high (ranging from 260 to 280) because it represents the sum of

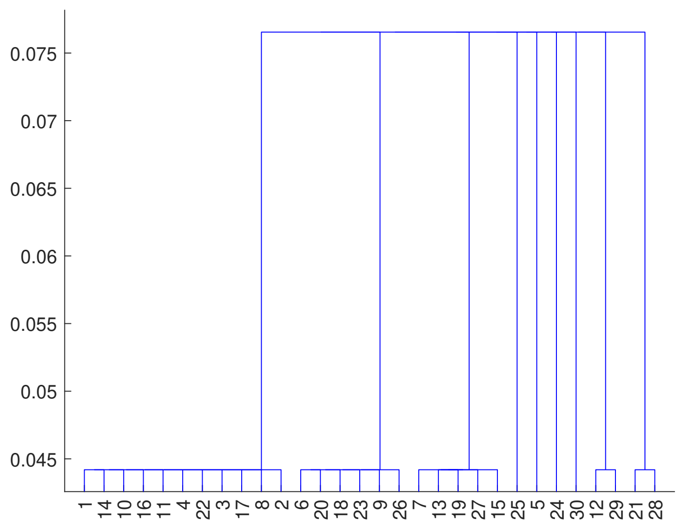


Figure 3.8: Dendrogram.

Table 3.1: Weights for cyclists in Parma found with the deterministic formulation.

Choice Criteria	Percentage of users	Weights		
		distance	safety	practicability
$\mathbf{p}^1$	13.2 %	0.97	0.03	0
$\mathbf{p}^2$	15.3 %	0	1	0
$\mathbf{p}^3$	20.6 %	0	0	1
$\mathbf{p}^4$	12.3 %	0.25	0.75	0
$\mathbf{p}^5$	11.2 %	0	0.69	0.31
$\mathbf{p}^6$	8 %	0.16	0.09	0.75
$\mathbf{p}^7$	7.3 %	0.72	0	0.28
$\mathbf{p}^8$	6.7 %	0.53	0.47	0
$\mathbf{p}^9$	5.6 %	0.47	0.03	0.5

Table 3.2: Weights for cyclists in Parma found with the stochastic formulation ($\sigma = 0.1$).

Choice Criteria	Percentage of users	Weights		
		distance	safety	practicability
$\mathbf{p}^1$	13.8%	1	0	0
$\mathbf{p}^2$	16.7%	0	1	0
$\mathbf{p}^3$	18.8%	0	0	1
$\mathbf{p}^4$	7.4%	0.16	0.84	0
$\mathbf{p}^5$	9.2%	0	0.66	0.34
$\mathbf{p}^6$	9.4%	0	0.22	0.78
$\mathbf{p}^7$	8.4%	0.63	0	0.38
$\mathbf{p}^8$	9.7%	0.41	0.59	0
$\mathbf{p}^9$	6.6%	0	0.78	0.22

Table 3.3: Weights for cyclists in Parma found with the stochastic formulation ($\sigma = 0.3$).

Choice Criteria	Percentage of users	Weights		
		distance	safety	practicability
$\mathbf{p}^1$	13.3 %	1	0	0
$\mathbf{p}^2$	17.2 %	0	1	0
$\mathbf{p}^3$	16.2 %	0	0	1
$\mathbf{p}^4$	9.9 %	0.19	0.81	0
$\mathbf{p}^5$	9.4 %	0	0.66	0.34
$\mathbf{p}^6$	8 %	0	0.38	0.62
$\mathbf{p}^7$	8 %	0.72	0	0.28
$\mathbf{p}^8$	9.5 %	0.53	0.47	0
$\mathbf{p}^9$	8 %	0.31	0	0.69

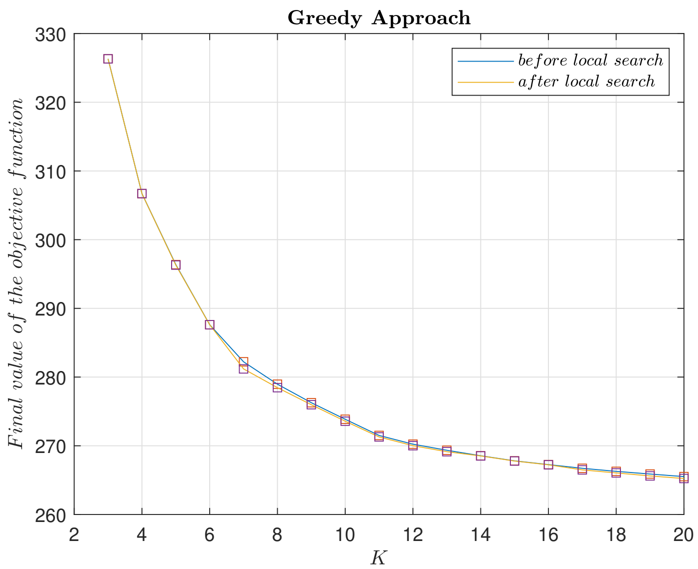


Figure 3.9: Objective function values of the greedy approach before and after performing local search.

the relative errors across all the routes considered. If we consider the average relative error of a single route, this is equal to 0.15, which can be considered as a low error.

The error is not equal to 0 for different reasons. First, we should recall that we are assuming that users follow SPs, but we should not forget that they do not run SP solvers to select them, so that deviations from optimal routes are possible. Next, we introduced randomness in the definition of the basic cost values (distance, safety and practicability) so that a path which is optimal with respect to a realization of the basic costs and, thus, has a null error with respect to such realization, may not be optimal with respect to another realization and, thus, has a strictly positive error with respect to that realization. Finally, we restricted the attention to three features, but other factors may influence the users' decisions (for example, the presence of shade during summer or lighting at night). Just to confirm the importance of considering multiple features influencing the choice of the cyclists, we can compare our solution considering three features with the one where cyclists are assumed to base their choice on a single feature (namely, distance). In such case, the value of the objective function would be approximately 738, which corresponds to an average relative error per single route of about 0.43, thus much larger than the 0.15 average error per single route obtained when considering all three features.

Figure 3.10a illustrates the weights generated by Algorithm 4, both before and after the application of local search, which appear to be very similar. This confirms that the greedy approach provides a solution that is already very close to the final outcome, even before local refinement.

Note that in this figure, we report along the x-axis the weight p_1^ℓ associated to distance, and along the y-axis the weight p_2^ℓ associated to safety. The weight p_3^ℓ associated to practicability is obtained by the following formula $p_3^\ell = 1 - p_1^\ell - p_2^\ell$. The same final results are reported in Table 3.1. The first column lists the weights/route choice criteria ($\mathbf{p}^\ell$) that have been identified, the second column lists the percentage of users with that behavior, and the last three columns list the values that define the corresponding three weights (p_1^ℓ, p_2^ℓ, p_3^ℓ) related to distance, safety, and practicability, respectively. The first three route choice criteria ($\mathbf{p}^1$, $\mathbf{p}^2$, $\mathbf{p}^3$) are related to the individual basic costs, either distance, or safety, or practicability ($\mathbf{p}^1$ is not exactly $(1,0,0)$, but very close to it). The last six criteria, on the other hand, are derived as various convex combinations, where the overall weight is predominantly distributed between two basic costs: distance and safety, distance and practicability, practicability and safety. In fact, from Figure 3.10a we observe that all final weights are distributed along or close to the edges of Δ_3, indicating that only a small percentage of the users adopts selection criteria which are a combination of all three basic costs. This result is interesting to interpret. We believe that, for most users, it is cognitively simpler and more intuitive to prioritize a smaller number of features when selecting a route. From our simulations, we observe that while there are cyclists who combine all three attributes (distance, safety, and practicability), such behaviors are less frequent. The most common behaviors tend to reflect a dominant preference for a single attribute, such as minimizing distance for those in a hurry, or maximizing safety or practicability for those with less time pressure. This aligns with the idea that decision-making is often guided by the most salient or personally relevant criterion. This is our current interpretation of the estimated weights. However, as part of future work, we plan to design and administer a questionnaire to a representative sample of users in order to further validate these findings and compare them with stated user preferences.

We also solved the problem with the greedy descent approach. The final results are similar to those of the greedy ascent. However, as shown in Figure 3.11, the ascent approach provides solutions with a lower final value of the objective function. The difference between the two greedy approaches is significantly reduced after applying the local search.

In Table 3.4, we present the computing times required to solve the problem instance with $k = 9$ using the two different algorithms. Although these computing times have to be taken with some care, since improved implementations are indeed possible, it appears quite evident that the fastest algorithm is the greedy ascent, which solves the problem in less than three seconds. The larger number of iterations of the greedy descent approach with respect to the greedy ascent approach ($|\bar{\Delta}_r| - k$ versus k) strongly penalizes the former.

An interesting outcome of these simulations is that for many cyclists the safety of the roads and their good maintenance are very important factors, not less impor-

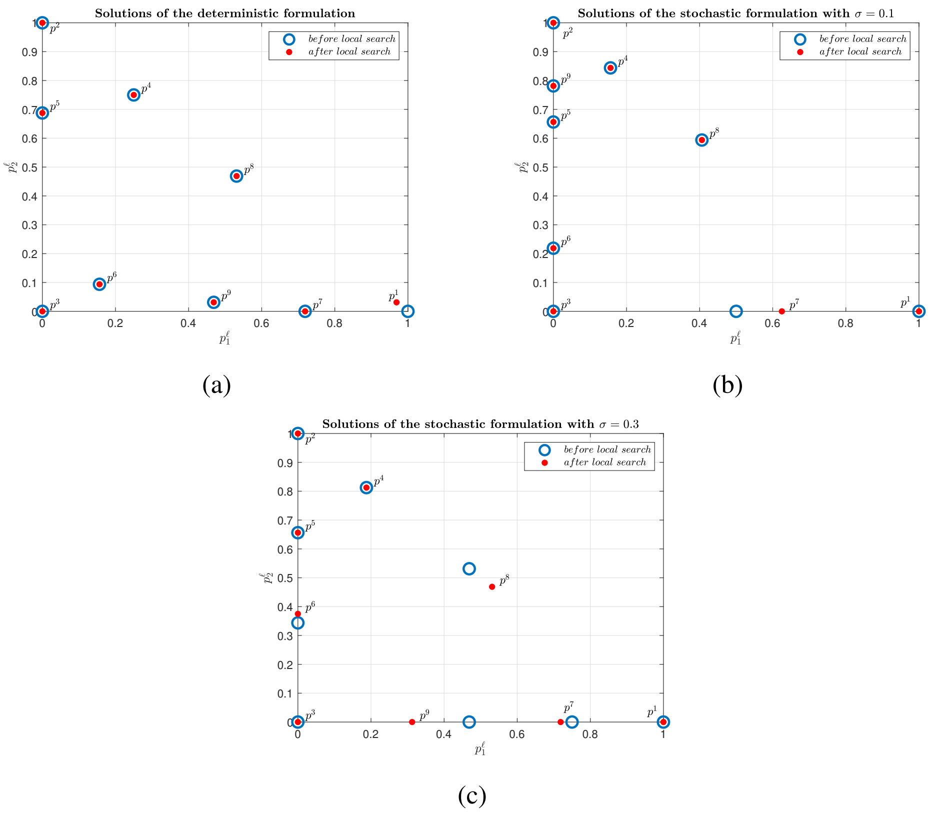


Figure 3.10: Weights obtained by the greedy ascent approach with $k = 9$ before and after performing local search: (a) deterministic formulation, (b) stochastic formulation with $\sigma = 0.1$, and (c) stochastic formulation with $\sigma = 0.3$.

Table 3.4: Computing times of the two approaches.

Approach	Greedy Ascent	Greedy Descent
Computational time (s)	2.86	767.60

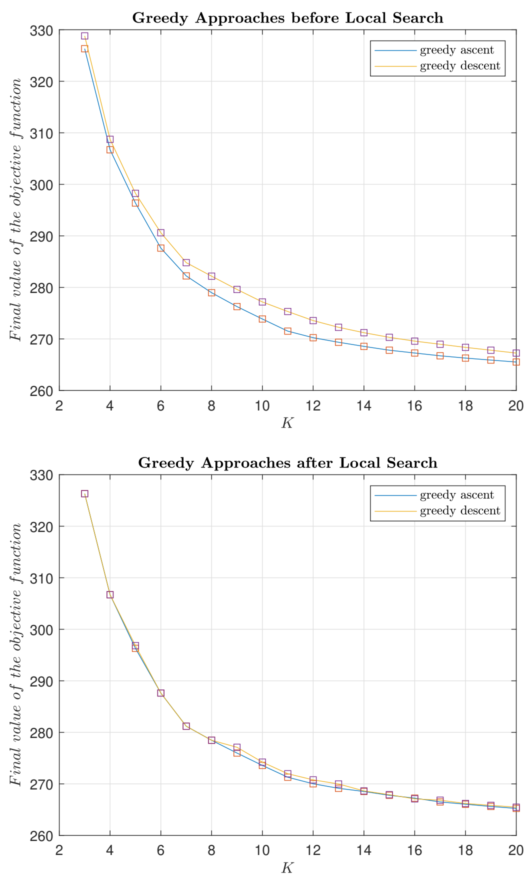


Figure 3.11: Comparison between greedy ascent and descent, both before and after applying local search.

tant than the length of a route. As already pointed out, this fact must be taken into account when planning new interventions to improve the quality of the bike-network, stressing the fact that improving existing infrastructures is as relevant as planning new bike-network segments to reduce distances.

Stochastic problem

As for the flow-based formulation, we have also performed some experiments to evaluate the impact of introducing randomness in the basic costs to account for the different users' perception of such costs, i.e., we addressed the problem arising from the stochastic formulation presented in Section 3.1.3. These experiments allowed us to consolidate the previously obtained results. Indeed, despite introducing perturbations to the arc costs perceived by users, the outcomes remain similar. In particular, the most frequent route choice criteria are confirmed, with percentages very close to those obtained in the deterministic case.

More in detail, we considered a Gaussian distribution for the basic costs $\tilde{\mathbf{c}}_{ij} \sim \mathscr{N}\left(\mathbf{c}_{ij}, (\sigma \cdot \mathbf{c}_{ij})^2\right)$ with $\sigma \in \{0.1, 0.3\}$. Tables 3.2 and 3.3 report the respective solutions. From these tables we observe that the choice criteria associated with higher user percentages remain nearly unchanged, while only those linked to lower percentages exhibit significant variations. Specifically, notable variations are observed, both with $\sigma = 0.1$ and with $\sigma = 0.3$, for what concerns the choice criteria $\mathbf{p^6}$ and $\mathbf{p^9}$, whose corresponding percentages are both low (below 10%). These differences are more easily visualized in the corresponding Figures 3.10b and 3.10c. In particular, for the former case ($\sigma = 0.1$), note that $\mathbf{p^9}$ changes its position completely, moving closer to $\mathbf{p^5}$, while $\mathbf{p^6}$ shifts less, and its distance component becomes null. For the latter case ($\sigma = 0.3$), note that $\mathbf{p^9}$ gets closer to $\mathbf{p^3}$, and its percentage increases at the expense of the percentage associated with $\mathbf{p^3}$ (decreasing from 20.6% to 16.2%). As for $\sigma = 0.1$, $\mathbf{p^6}$ shifts towards a point with null distance component. Except for these variations, the cyclists' behavior obtained with different variability values remains substantially the same. This consistency reinforces the robustness of the study and supports drawing the same conclusions as previously stated.

Chapter 4

Optimizing cycling network design within budget constraints

This chapter presents the cost reduction problem in bicycle networks. We formalized the problem (see Section 4.1) and proposed a mathematical formulation for it (see Section 4.2). To address this problem, we developed four solution approaches: two exact and two heuristics, which are described in Section 4.3. Finally, we performed computational experiments with all proposed methods, considering both randomly generated and real-world instances, the latter based on data collected from the city of Parma, Italy (see Section 4.4). Note that some of the definitions presented in Section 4.1 have already been introduced in Chapter 3. However, they are also included in this chapter to make it self-contained and, consequently, facilitate the definition of the problem addressed here. Furthermore, the papers by Praxedes et al. [94, 95] are outcomes of the research described in this chapter.

4.1 Problem definition

A road network is represented as a directed graph $G = (V, A)$, where the set of vertices V corresponds to intersections, and the set of arcs A corresponds to roads that are accessible to cyclists, including both dedicated bikeways and shared-use streets. Each arc $(i,j) \in A$ is associated with a cost vector $c_{0_{ij}} = (c^1_{0_{ij}}, \dots, c^r_{0_{ij}})$, where $c^h_{0_{ij}}$ represents the cost on arc (i,j) with respect to the road characteristic $h \in \{1,\dots,r\}$, where the number of road features is denoted by r. These characteristics can include length, traffic capacity, safety indicators, pavement condition, among others. Cyclists can be classified into distinct categories based on the importance that they assign to each road feature when selecting a path to follow. Hence, the cost of an arc, for each cyclist category, is defined as follows:

$$\sum_{h=1}^{r} c^h_{0_{ij}} p^\ell_h \quad \ell \in \{1,\dots,q\},$$

where p^ℓ_h represents the importance assigned to road feature h by cyclists category ℓ. Each category $\ell \in \{1\dots,q\}$ is associated with a weight vector $p^\ell = (p^\ell_1,\dots,p^\ell_r)$, where q denotes the number of profiles, $p^\ell_h \geq 0$ for $h \in \{1,\dots,r\}$, and $\sum_{h=1}^r p^\ell_h = 1$. Moreover, p^ℓ is chosen by a fraction α_ℓ of the whole population of cyclists. Since different features may have different units of measure, a normalization procedure of costs c_0 is necessary. For further details on the definition of the cyclist profiles with the related weights p, and the fractions α, we refer the reader to Chapter 3. In the current chapter, we consider all these values as input data.

Furthermore, let $W \subset \{V \times V\}$ be a set of OD pairs, where each pair $w = (o_w, d_w)$ has an associated demand $u_w \in \mathbb{N}$. Using the demand values, we define the difference between the incoming and outgoing flows of cyclists traveling from o_w to d_w at node $i \in V$ as follows:

$$U^w_i = \begin{cases} u_w & i = o_w \\ -u_w & i = d_w \\ 0 & i \in V \setminus \{o_w, d_w\}. \end{cases}$$

The main goal of this problem is to apply interventions to the network aiming at minimizing the costs perceived by cyclists, which include their preferences for a set of road features. Hence, let K denote the set of possible interventions, where each intervention $k \in K$ affects a subset $A_k \subset A$ of arcs. Note that the definition of the set K and the corresponding sets A_k should be provided by infrastructure planners, who must specify the interventions they intend to implement in their urban planning. The purpose of the studied problem is not to determine what they should do, but rather, given their intended actions, to identify which interventions should be prioritized under a limited budget. For the case study of Parma (Section 4.4.2), due to the lack of available information regarding these sets, we define K based on a set of intervention types described in Table 4.11. The corresponding arc sets A_k are constructed following the procedure detailed in Appendix F, which aims to select the most frequently used arcs in the city.

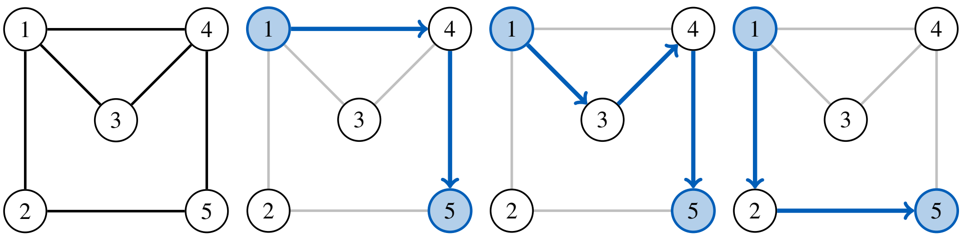
The effect of intervention k on road feature h for arc (i, j) is represented by parameter ϕ^h_{ijk}. The new arc cost, when k is applied, is calculated as $c^h_{0_{ij}} - \phi^h_{ijk}$. We remark that cost reductions must fulfill the requirement that costs are positive even when all interventions are performed. In addition, applying intervention k on arc (i, j) incurs a monetary cost τ^k_{ij}, referred to as the building cost. Therefore, given a budget B, the objective of the problem is to determine a subset K^* of interventions and the flows on each arc (i, j) that minimizes the overall cost perceived by cyclists, subject to budget and demand constraints. Table 4.1 summarizes all the parameters of the problem, while an example of the problem is illustrated in Figure 4.1 and described in what follows.

Consider a graph G representing a road network with five vertices ($V = \{1, 2, \ldots, 5\}$) and only one OD pair ($W = \{(o_1, d_1) = (1, 5)\}$). Figure 4.1a illustrates the graph G, and Figures 4.1b, 4.1c, and 4.1d show the three possible elementary routes from node 1 to node 5, respectively. In this example, we have a single cyclist profile containing only one cyclist ($q = 1$ and $\alpha_1 = 1$), and we consider two road characteristics ($r = 2$): distance and safety. The cyclist assigns weights of 0.35 to distance and 0.65 to safety, respectively ($p^1 = \{p^1_1, p^1_2\} = \{0.35, 0.65\}$). Furthermore, given a budget B = €500,000, suppose we can apply two different interventions, k_1 and k_2, to the network, which affect only the safety criterion. The values for all the parameters in this example are summarized in Table 4.2, where the column *Distance* (m) indicates the length of the arcs in meters, while the columns *Safety* and *Building cost* ($\times$1000 €) present the values of safety criterion and the monetary cost in euros for each arc, respectively, considering each possible combination of applied interventions.

Note that, in Table 4.2, the arcs affected by the applied interventions in each

Table 4.1: Summary of data and parameters for the cost reduction problem.

Bicycle network	
V	Set of vertices;
A	Set of arcs;
W	Set of OD pairs;
$o_w(d_w)$	Origin (destination) vertex associated with OD pair $w \in W$;
u_w	Demand associated with OD pair $w \in W$;
Interventions	
K	Set of possible interventions;
A_k	Subset of arcs affected by intervention $k \in K$;
τ_{ij}^k	Cost of applying intervention $k \in K$ on arc $(i, j) \in A_k$ (building cost);
r	Number of road characteristics;
$c_{0_{ij}}^h$	Cost perceived by the cyclist related to road characteristic $h \in \{1,\ldots,r\}$ on arc $(i, j) \in A$;
ϕ_{ijk}^h	Cost reduction on arc $(i, j) \in A$ associated with road characteristic $h \in \{1,\ldots,r\}$, due to intervention $k \in K$;
B	Budget available;
Cyclist profiles	
q	Number of cyclist profiles;
p_h^ℓ	Weight assigned by a cyclist of profile $\ell \in \{1,\ldots,q\}$ to the road characteristic $h \in \{1,\ldots,r\}$;
α_ℓ	Fraction of the total cyclist population that belongs to profile $\ell \in \{1,\ldots,q\}$.



(a) Graph. (b) Route 1. (c) Route 2. (d) Route 3.

Figure 4.1: Problem example for only one OD pair and cyclist.

Table 4.2: Parameter values for the cost-reduction problem example considering a single OD pair.

Arc	Distance (m)	Safety					Building cost (×1000 €)			
		∅	$\{k_1\}$	$\{k_2\}$	$\{k_1,k_2\}$		∅	$\{k_1\}$	$\{k_2\}$	$\{k_1,k_2\}$
(1,3); (3,1)	890	0.90	**0.55**	0.90	**0.55**		0	**350**	0	**350**
(1,4); (4,1)	1220	0.25	0.25	0.25	0.25		0	0	0	0
(4,3); (3,4)	765	0.48	0.48	0.48	0.48		0	0	0	0
(1,2); (2,1)	510	0.68	0.68	0.68	0.68		0	0	0	0
(2,5); (5,2)	635	0.49	0.49	**0.28**	**0.28**		0	0	**280**	**280**
(5,4); (4,5)	1980	0.30	**0.12**	0.30	**0.12**		0	**125**	0	**125**
Max	1980	0.90	0.68	0.90	0.68	**Total**	0	475	280	755

scenario have their safety and building cost values highlighted. Moreover, a lower safety cost value indicates a safer arc. Since we are comparing values of different magnitudes and units (meters and safety scores), we normalize each road feature by scaling its values relative to the respective maximum, as shown in Table 4.3.

Table 4.3: Normalized parameter values for the cost-reduction problem example considering a single OD pair.

Arc	Distance	Safety					Building cost (×1000 €)			
		∅	$\{k_1\}$	$\{k_2\}$	$\{k_1,k_2\}$		∅	$\{k_1\}$	$\{k_2\}$	$\{k_1,k_2\}$
(1,3); (3,1)	0.45	1.00	**0.81**	1.00	**0.81**		0	**350**	0	**350**
(1,4); (4,1)	0.62	0.28	0.37	0.28	0.37		0	0	0	0
(4,3); (3,4)	0.39	0.53	0.71	0.53	0.71		0	0	0	0
(1,2); (2,1)	0.26	0.76	1.00	0.76	1.00		0	0	0	0
(2,5); (5,2)	0.32	0.54	0.72	**0.31**	**0.41**		0	0	**280**	**280**
(5,4); (4,5)	1.00	0.33	**0.18**	0.33	**0.18**		0	**125**	0	**125**
Max	1.00	1.00	1.00	1.00	1.00	**Total**	0	475	280	755

In this example, given the available budget B, it is not possible to apply interventions k_1 and k_2 simultaneously. Hence, for the scenarios that do not exceed the budget, we compute the total cost of each elementary route, considering the weight vector p^1 associated with the cyclist, as described in Table 4.4.

Therefore, the route with the minimum cost, considering both distance and safety

Table 4.4: Route costs for the cost-reduction problem example considering one OD pair.

Route	Total cost perceived by the cyclist along the path		
	$\emptyset$	$\{k_1\}$	$\{k_2\}$
1	$0.35\times\mathbf{1.62}+0.65\times\mathbf{0.61}\approx\mathbf{0.96}$	$0.35\times\mathbf{1.62}+0.65\times\mathbf{0.55}\approx\mathbf{0.92}$	$0.35\times\mathbf{1.62}+0.65\times\mathbf{0.61}\approx\mathbf{0.96}$
2	$0.35\times\mathbf{1.84}+0.65\times\mathbf{1.86}\approx\mathbf{1.85}$	$0.35\times\mathbf{1.84}+0.65\times\mathbf{1.70}\approx\mathbf{1.75}$	$0.35\times\mathbf{1.84}+0.65\times\mathbf{1.86}\approx\mathbf{1.85}$
3	$0.35\times\mathbf{0.58}+0.65\times\mathbf{1.30}\approx\mathbf{1.05}$	$0.35\times\mathbf{0.58}+0.65\times\mathbf{1.72}\approx\mathbf{1.32}$	$0.35\times\mathbf{0.58}+0.65\times\mathbf{1.07}\approx\mathbf{0.90}$

aspects, is *Route 3* after intervention k_2 is applied. As a result, the optimal subset of interventions for this example is $K^* = \{k_2\}$, and the cyclist will follow the path corresponding to *Route 3*. As already mentioned, the example in Figure 4.1 refers to the case of only one OD pair and one cyclist. However, the problem accounts for all OD pairs present in the bicycle network and multiple cyclist profiles, where the number of cyclists associated with a specific OD pair $w \in W$ and profile $\ell \in \{1,\dots,q\}$ is determined by the product between the demand u_w and the fraction α_ℓ.

4.2 Mathematical formulation

Let $x_{ij}^{w\ell}$ be a non-negative continuous variable representing the flow of cyclists with profile $\ell \in \{1,\dots,q\}$ on arc $(i,j) \in A$, traveling from o_w to d_w, with $w = (o_w, d_w) \in W$. Moreover, let c_{ij}^h denote a non-negative continuous variable representing the cost associated with road characteristic $h \in \{1,\dots,r\}$ on arc $(i,j) \in A$, perceived by the cyclists after the application of the interventions. Binary variable y_k takes value 1 if intervention $k \in K$ is selected to be applied, and 0 otherwise. Table 4.5 summarizes the variables of the problem.

We can write a mathematical formulation for the problem as follows:

$$\min \sum_{w\in W} \sum_{(i,j)\in A} \sum_{\ell=1}^{q} \sum_{h=1}^{r} c_{ij}^h p_h^\ell x_{ij}^{w\ell} \tag{4.1a}$$

Table 4.5: Summary of variables for the cost-reduction problem.

Continuous variables	
$x_{ij}^{w\ell}$	Cyclists' flow on arc $(i,j) \in A$, associated with OD pair $w \in W$ and cyclist profile $\ell \in \{1,\ldots,q\}$;
c_{ij}^{h}	Cost on arc $(i,j) \in A$ associated with road characteristic $h \in \{1,\ldots,r\}$ after the application of the interventions.
Binary variables	
y_k	Indicates whether intervention $k \in K$ is applied.

s.t.

$$\sum_{\substack{j\in V:\\(i,j)\in A}} x_{ij}^{w\ell} - \sum_{\substack{j\in V:\\(j,i)\in A}} x_{ji}^{w\ell} = \alpha_\ell U_i^w \qquad i \in V, \ell \in \{1,\ldots,q\}, w \in W \tag{4.1b}$$

$$\sum_{k\in K} \sum_{(i,j)\in A_k} \tau_{ij}^k y_k \leq B \tag{4.1c}$$

$$c_{ij}^h = c_{0_{ij}}^h - \sum_{k\in K} \phi_{ijk}^h y_k \qquad (i,j) \in A, h \in \{1,\ldots,r\} \tag{4.1d}$$

$$y_k \in \{0,1\} \qquad k \in K \tag{4.1e}$$

$$x_{ij}^{w\ell} \geq 0 \qquad (i,j) \in A, \ell \in \{1,\ldots,q\}, w \in W \tag{4.1f}$$

$$c_{ij}^h \geq 0 \qquad (i,j) \in A, h \in \{1,\ldots,r\}. \tag{4.1g}$$

Objective function (4.1a) minimizes the overall cost perceived by the cyclists. Constraints (4.1b) ensure the flow conservation and demand fulfillment, where U_i^w represents the difference between the total incoming and outgoing flows at node $i \in V$ for OD pair $w \in W$. Constraints (4.1c) impose that the interventions must be selected without exceeding the budget available. Constraints (4.1d) compute the costs on the arcs based on the selected interventions. Finally, constraints (4.1e)–(4.1g) define the domain of the variables.

Note that formulation (4.1a)–(4.1g) is non-linear, as it involves a product of variables in objective function (4.1a). However, by substituting c_{ij}^h by means of constraints (4.1d), we observe that the nonlinearity arises from the product of a continuous variable $x_{ij}^{w\ell}$, and a binary variable y_k, as indicated in the following

$$\min \sum_{w \in W} \sum_{(i,j) \in A} \sum_{\ell=1}^{q} \sum_{h=1}^{r} \left(c_{0_{ij}}^{h} p_h^{\ell} x_{ij}^{w\ell} - \sum_{k \in K} \phi_{ijk}^{h} p_h^{\ell} y_k x_{ij}^{w\ell} \right). \quad (4.2)$$

This type of product can be easily linearized by introducing auxiliary variables $z_{ij}^{w\ell k}$ to substitute the product, which satisfy the following constraints:

$$z_{ij}^{w\ell k} \leq M y_k \quad (i,j) \in A,\ \ell \in \{1,\ldots,q\},\ w \in W,\ k \in K \quad (4.3a)$$

$$z_{ij}^{w\ell k} \leq x_{ij}^{w\ell} \quad (i,j) \in A,\ \ell \in \{1,\ldots,q\},\ w \in W,\ k \in K \quad (4.3b)$$

$$z_{ij}^{w\ell k} \geq 0 \quad (i,j) \in A,\ \ell \in \{1,\ldots,q\},\ w \in W,\ k \in K, \quad (4.3c)$$

where M is a sufficiently large constant, which, for instance, could be $M = \alpha_\ell u_w$. Note that constraints $z_{ij}^{w\ell k} \geq x_{ij}^{w\ell} - M(1 - y_k)$, for $(i,j) \in A$, $\ell \in \{1,\ldots,q\}$, $w \in W$, and $k \in K$ (which guarantee that $z_{ij}^{w\ell k} = x_{ij}^{w\ell}$ when $y_k = 1$) are not necessary, since we are minimizing the objective function and the coefficients of variables $z_{ij}^{w\ell k}$ in (4.2) are negative, so that the value of these variables is always taken as large as possible at optimal solutions. Hence, when $y_k = 1$, we have $z_{ij}^{w\ell k} = x_{ij}^{w\ell}$, without the need for these additional constraints. Figure 4.2 illustrates a data flowchart for the proposed optimization problem. Note that $\bar{\Pi} \subset \Pi$ denotes the set of measured paths within the set of directed paths Π of graph G, which is required by the procedure described in Chapter 3 to identify cyclist route choice behavior.

4.3 Solution approaches

This section describes the four proposed approaches to solve the problem: a dynamic programming algorithm (Section 4.3.1), a branch-and-bound method (Section 4.3.2), and two heuristics (Sections 4.3.3 and 4.3.4) that use the dynamic programming for the knapsack problem. For simplicity in the notation used to explain the proposed approaches, we define the following vectors for the parameters and variables introduced in Sections 4.1 and 4.2. Specifically, let $c_0 = [c_{0_{ij}}^{h}, (i,j) \in A, h \in \{1,\ldots,r\}]$ and $c = [c_{ij}^{h}, (i,j) \in A, h \in \{1,\ldots,r\}]$ be the vectors of costs perceived by the cyclists before

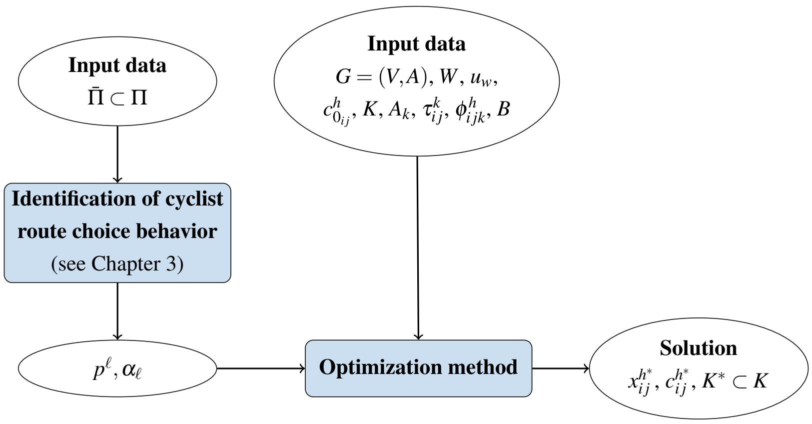


Figure 4.2: Data flow chart of the proposed bikeway network design problem.

and after interventions, respectively. In addition, let $p = [p^\ell, \ell \in \{1,\ldots,q\}]$ denote the vector of cyclist weight vectors, and $\alpha = [\alpha_\ell, \ell \in \{1,\ldots,q\}]$ the vector of cyclist fractions associated with the cyclist profiles. Also, consider $\tau = [\tau^k_{ij}, k \in K, (i,j) \in A_k]$ and $\phi = [\phi^h_{ijk}, (i,j) \in A, k \in K, h \in \{1,\ldots,r\}]$ as the vectors of building costs and cost reductions, respectively. Finally, let $x = [x^{w\ell}_{ij}, (i,j) \in A, w \in W, \ell \in \{1,\ldots,q\}]$ represent the vector of flow variables, and $U = [U^w_i, i \in V, w \in W]$ the vector representing the differences between incoming and outgoing flows. It is worth mentioning that all these parameters will be referred to as $data$.

4.3.1 Dynamic Programming

The first method consists of an explicit enumeration of feasible solutions, where each state is associated with a subset of interventions that satisfies the budget constraint. This procedure can be seen as a dynamic programming algorithm without a dominance rule among the states. The absence of a dominance rule is due to the need of keeping track all possible combinations of interventions, as will be explained later in this section.

Before describing the method in further details, we first provide the following relevant definitions. Let S be the set of states. Associated with each state $s \in S$, we have a subset $K_s \subset K$ of interventions and, thus, a feasible solution of the problem. Therefore, we can define the states as follows:

$$s = [\sigma_s, F(K_s, x_{K_s}), K_s] \tag{4.4}$$

where: $\sigma_s \in \{1, \ldots, |K|\}$ indicates the intervention being considered at the current state; x_{K_s} denotes the flows corresponding to the optimal solution of problem (4.1a)–(4.1g) when $y_k = 1$ for $k \in K_s$, and $y_k = 0$ otherwise; and $F(K_s, x)$ represents the optimal cost of such problem, given by:

$$F(K_s, x) = \sum_{w \in W} \sum_{(i,j) \in A} \sum_{\ell=1}^{q} \sum_{h=1}^{r} \left(c^h_{0_{ij}} p^\ell_h x^{w\ell}_{ij} - \sum_{k \in K_s} \phi^h_{ijk} p^\ell_h x^{w\ell}_{ij} \right), \tag{4.5}$$

which is equivalent to objective function (4.1a) for K_s. We also introduce sets $S_0 \subset S$ and $S_f \subset S$ of initial and final states, respectively. In particular, S_0 contains the single state $s_0 = [1, F(\{\}, x_{\{\}}), \{\}]$, while final states are all those for which $\sigma_s = |K| + 1$. Next, we denote with $T(K_s)$ the total building cost of all interventions in K_s, which can be computed as:

$$T(K_s) = \sum_{k \in K_s} \sum_{(i,j) \in A_k} \tau^k_{ij}. \tag{4.6}$$

For each state $s \in S \setminus S_f$, we can decide to add intervention σ_s to K_s if the residual budget given by $B - T(K_s \cup \{\sigma_s\})$ is nonnegative. Otherwise, intervention σ_s cannot be applied given the interventions in K_s. If the residual budget is insufficient for the current intervention, a new state $s' = [\sigma_{s'}, F(K_{s'}, x_{K_{s'}}), K_{s'}]$ is generated, where $\sigma_{s'} = \sigma_s + 1$, $x_{K_{s'}} = x_{K_s}$, and $K_{s'} = K_s$. Conversely, when it is possible to apply σ_s, states s' and $s'' = [\sigma_{s''}, F(K_{s''}, x_{K_{s''}}), K_{s''}]$ are created, where $\sigma_{s''} = \sigma_s + 1$ and $K_{s''} = K_s \cup \{\sigma_s\}$.

Given the above, the proposed method, denoted as *cost reduction model – dynamic programming* (CRM-DP), starts with a queue of states containing only state $s_0 = [1, F(\{\}, x_{\{\}}), \{\}]$. It then expands states that satisfy the budget constraint, checking whether there is sufficient budget to add the current intervention to the existing set. When a final state $s_f \in S_f$ is reached, it is not enqueued but rather compared with the current best solution. If its cost is lower, the best solution is updated. Otherwise, this final state is ignored. At the end, the method returns the final state with the smallest cost, which corresponds to the optimal solution.

The pseudocode for CRM-DP is presented in Algorithm 5. In this pseudocode, and in what follows, $\texttt{shortestPaths}(data, K_s)$ refers to solving formulation (4.1a)–(4.1g) with $y_k = 1$ for each $k \in K_s$ and $y_k = 0$ otherwise. In other words, it involves solving the following formulation:

$$\min \sum_{w\in W} \sum_{(i,j)\in A} \sum_{\ell=1}^{q} \sum_{h=1}^{r} c_{ij}^{h} p_h^{\ell} x_{ij}^{w\ell} \tag{4.7a}$$

s.t.

$$\sum_{\substack{j\in V:\\(i,j)\in A}} x_{ij}^{w\ell} - \sum_{\substack{j\in V:\\(j,i)\in A}} x_{ji}^{w\ell} = \alpha_\ell U_i^w \qquad i \in V, \ell \in \{1,\dots,q\}, w \in W \tag{4.7b}$$

$$x_{ij}^{w\ell} \geq 0 \qquad (i,j) \in A, \ell \in \{1,\dots,q\}, w \in W, \tag{4.7c}$$

by computing the shortest paths for each OD pair $w \in W$ and cyclist weights vector $p^\ell, \ell \in \{1,\dots,q\}$, using Dijkstra's algorithm [26].

In Algorithm 5, lines 2–5 create the initial state s_0 and add it to the state queue Q. The main loop runs from line 6 to line 24. Specifically, line 7 removes the state from the top of Q. From line 8 to 10, the state corresponding to not applying the current intervention is created and then checked to determine whether it is final or not. In the latter case, it is added to Q. In the former one, line 12 checks whether its solution cost is lower than the cost of the current best solution $f_{\min}$, and update the best solution if that is the case (lines 13–14). Furthermore, the same procedure is followed from line 15 to line 24 for the state corresponding to applying the current intervention. The only differences here are indicated in line 16, where it is checked whether there is enough budget to apply the current intervention, and line 17, where the flow vector x and the cost vector c are calculated, respectively. Lines 25-27 return the flow vector x^*, the cost vector c^*, and the set of interventions K^*, which are all associated with the optimal solution.

The following numerical examples provide more details on how the method works. The first, *Example 1*, illustrates how the states are generated and the optimal solution is obtained, considering a small instance. The second, *Example 2*, clarifies the neces-

Algorithm 5: CRM-DP

function `CRM-DP`$(data)$
$x_{\{\}}, c_{\{\}} \leftarrow$ `shortestPaths`$(data, \{\})$;
$s_0 \leftarrow [1, F(\{\}, x_{\{\}}), \{\}]$;
$f_{min} \leftarrow F(\{\}, x_{\{\}})$;
Q.`enqueue`$(\{s_0\})$;
while $|Q| > 0$ **do**
    $s_{cr} \leftarrow Q$.`dequeue`$()$;
    $s' \leftarrow [\sigma_{s_{cr}} + 1, F(K_{s_{cr}}, x_{K_{s_{cr}}}), K_{s_{cr}}]$;
    if $\sigma_{s_{cr}} < |K|$ **then**
        Q.`enqueue`$(\{s'\})$;
    else
        $f_{sol} \leftarrow min(f_{min}, F(K_{s_{cr}}, x_{K_{s_{cr}}}))$;
        if $f_{sol} \neq f_{min}$ **then**
            $s_{sol} \leftarrow s'$, $f_{min} \leftarrow f_{sol}$;
    $K_{s''} \leftarrow K_{s_{cr}} \cup \{\sigma_{s_{cr}}\}$;
    if $B - T(K_{s''}) \geq 0$ **then**
        $x_{K_{s''}}, c_{s''} \leftarrow$ `shortestPaths`$(data, K_{s''})$;
        $s'' \leftarrow [\sigma_{s_{cr}} + 1, F(K_{s''}, x_{K_{s''}}), K_{s''}]$;
        if $\sigma_{s_{cr}} < |K|$ **then**
            Q.`enqueue`$(\{s''\})$;
        else
            $f_{sol} \leftarrow min(f_{min}, F(K_{s''}, x_{K_{s''}}))$;
            if $f_{sol} \neq f_{min}$ **then**
                $s_{sol} \leftarrow s''$, $f_{min} \leftarrow f_{sol}$;
$K^* \leftarrow K_{s_{sol}}$;
$x^*, c^* \leftarrow$ `shortestPaths`$(data, K^*)$;
return x^*, c^*, K^* ;

sity of tracking all possible subset of interventions instead of using a dominance rule among the states based on the residual budget.

Example 1: Consider $B = 6$ and $K = \{1,2,3,4\}$. In addition, let $K^{(i,j)} \subset K$ denote the subset of interventions that affect the arc $(i, j) \in A$. Table 4.6 indicates the values for all parameters in this example, while the objective function values (see Eq. (4.5)) are provided during its description. Let s^* be the state corresponding to the optimal solution, with its cost value given by $\mathscr{F}^*$, which is initially set to $F(\{\},x_{\{\}})$. The method starts by building the initial state $s_0 = [1, F(\{\},x_{\{\}}), \{\}]$ and adding it to the state queue Q.

Table 4.6: Cost reduction problem instance.

Arc	**Basic costs $c^h_{0_{ij}}$**		**Building costs τ^k_{ij}**				**Cost reductions ϕ^h_{ijk}**		
(i, j)	$c^1_{0_{ij}}$	$c^2_{0_{ij}}$	τ^1_{ij}	τ^2_{ij}	τ^3_{ij}	τ^4_{ij}	$K^{(i,j)}$	ϕ^1_{ijk}	ϕ^2_{ijk}
(1, 2)	16.34	8.02	1.50	0.00	0.00	0.00	{1}	6.14	2.98
(1, 3)	36.54	72.05	1.25	0.00	0.00	0.00	{1}	26.85	42.57
(2, 1)	64.17	15.42	0.00	1.78	0.00	0.00	{2}	25.30	8.98
(2, 4)	66.60	42.59	0.15	0.00	0.00	0.00	{1}	46.09	27.87
(3, 1)	66.72	65.14	0.00	0.00	1.05	0.00	{3}	56.43	57.48
(3, 4)	25.84	17.72	0.00	0.00	1.02	0.00	{3}	5.30	8.11
(4, 2)	29.04	28.83	0.00	0.00	0.00	2.44	{4}	3.14	2.97
(4, 3)	72.21	68.98	0.00	0.00	1.03	0.00	{3}	48.13	27.28
–//–	–//–	**Total:**	**2.90**	**1.78**	**3.10**	**2.44**	–//–	–//–	–//–
–//–	**OD pairs W**		**Cyclist weights p^ℓ**				–//–	–//–	–//–
–//–	(o_w, d_w)	u_w	ℓ	p^ℓ_1	p^ℓ_2	α_ℓ	–//–	–//–	–//–
–//–	(3, 2)	2	1	0.31	0.69	0.06	–//–	–//–	–//–
–//–	(1, 3)	5	2	0.28	0.72	0.30	–//–	–//–	–//–
–//–	(2, 3)	4	3	0.94	0.06	0.21	–//–	–//–	–//–
–//–	–//–	–//–	4	0.64	0.36	0.01	–//–	–//–	–//–
–//–	–//–	–//–	5	0.32	0.68	0.42	–//–	–//–	–//–

State queue (Q)
$s_0 = [1, F(\{\},x_{\{\}}), \{\}]$

In state s_0, after removing it from the top of Q, we can apply the intervention 1, as $B - T(\{1\}) = 3.10 \geq 0$. Consequently, the states $s_1 = [2, F(\{\},x_{\{\}}), \{\}]$ and

$s_2 = [2, F(\{1\}, x_{\{1\}}), \{1\}]$ are generated and enqueued in Q.

State queue (Q)
$s_1 = [2, F(\{\}, x_{\{\}}), \{\}]$
$s_2 = [2, F(\{1\}, x_{\{1\}}), \{1\}]$

In s_1, we can apply the intervention 2, as $B - T(\{2\}) = 4.22 \geq 0$ and then the new states $s_3 = [3, F(\{\}, x_{\{\}}), \{\}]$ and $s_4 = [3, F(\{2\}, x_{\{2\}}), \{2\}]$ are generated and enqueued in Q.

State queue (Q)
$s_2 = [2, F(\{1\}, x_{\{1\}}), \{1\}]$
$s_3 = [3, F(\{\}, x_{\{\}}), \{\}]$
$s_4 = [3, F(\{2\}, x_{\{2\}}), \{2\}]$

Similarly, in s_2, we can apply the intervention 2, as $B - T(\{1,2\}) = 1.32 \geq 0$. Hence, the states $s_5 = [3, F(\{1\}, x_{\{1\}}), \{1\}]$ and $s_6 = [3, F(\{1,2\}, x_{\{1,2\}}), \{1,2\}]$ are generated and enqueued in Q.

State queue (Q)
$s_3 = [3, F(\{\}, x_{\{\}}), \{\}]$
$s_4 = [3, F(\{2\}, x_{\{2\}}), \{2\}]$
$s_5 = [3, F(\{1\}, x_{\{1\}}), \{1\}]$
$s_6 = [3, F(\{1,2\}, x_{\{1,2\}}), \{1,2\}]$

The intervention 3 can be applied in s_3, s_4, and s_5, since $B - T(\{3\}) = 2.90 \geq 0$, $B - T(\{2,3\}) = 1.12 \geq 0$, and $B - T(\{1,3\}) = 0 \geq 0$, respectively. However, it cannot be applied in s_6, as $B - T(\{1,2,3\}) = -1.78 < 0$. Therefore, after exploring those states, we have the following state queue:

State queue (Q)
$s_7 = [4, F(\{\}, x_{\{\}}), \{\}]$
$s_8 = [4, F(\{3\}, x_{\{3\}}), \{3\}]$
$s_9 = [4, F(\{2\}, x_{\{2\}}), \{2\}]$
$s_{10} = [4, F(\{2,3\}, x_{\{2,3\}}), \{2,3\}]$
$s_{11} = [4, F(\{1\}, x_{\{1\}}), \{1\}]$
$s_{12} = [4, F(\{1,3\}, x_{\{1,3\}}), \{1,3\}]$
$s_{13} = [4, F(\{1,2\}, x_{\{1,2\}}), \{1,2\}]$

In s_7, the states $s_{14} = [5, F(\{\}, x_{\{\}}), \{\}]$ and $s_{15} = [5, F(\{4\}, x_{\{4\}}), \{4\}]$ can be generated, since $B - T(\{4\}) = 3.56 \geq 0$. However, as they are final states, they will not be enqueued in Q. Therefore, we set:

$$\mathscr{F}^* = \min\{\mathscr{F}^*, F(\{\}, x_{\{\}}), F(\{4\}, x_{\{4\}})\},$$
$$\mathscr{F}^* = \min\{755.65, 755.65, 749.57\},$$
$$\mathscr{F}^* = 749.57 \text{ and } s^* = s_{15}.$$

State queue (Q)
$s_8 = [4, F(\{3\}, x_{\{3\}}), \{3\}]$
$s_9 = [4, F(\{2\}, x_{\{2\}}), \{2\}]$
$s_{10} = [4, F(\{2,3\}, x_{\{2,3\}}), \{2,3\}]$
$s_{11} = [4, F(\{1\}, x_{\{1\}}), \{1\}]$
$s_{12} = [4, F(\{1,3\}, x_{\{1,3\}}), \{1,3\}]$
$s_{13} = [4, F(\{1,2\}, x_{\{1,2\}}), \{1,2\}]$

In s_8, we can apply the intervention 4 as $B - T(\{3,4\}) = 0.46 \geq 0$. Hence, we generate the final states $s_{16} = [5, F(\{3\}, x_{\{3\}}), \{3\}]$ and $s_{17} = [5, F(\{3,4\}, x_{\{3,4\}}), \{3,4\}]$, and we set:

$$\mathscr{F}^* = \min\{\mathscr{F}^*, F(\{3\}, x_{\{3\}}), F(\{3,4\}, x_{\{3,4\}})\},$$
$$\mathscr{F}^* = \min\{749.57, 671.43, 671.43\},$$
$$\mathscr{F}^* = 671.43 \text{ and } s^* = s_{16}.$$

State queue (Q)
$s_9 = [4, F(\{2\}, x_{\{2\}}), \{2\}]$
$s_{10} = [4, F(\{2,3\}, x_{\{2,3\}}), \{2,3\}]$
$s_{11} = [4, F(\{1\}, x_{\{1\}}), \{1\}]$
$s_{12} = [4, F(\{1,3\}, x_{\{1,3\}}), \{1,3\}]$
$s_{13} = [4, F(\{1,2\}, x_{\{1,2\}}), \{1,2\}]$

Repeating the process for s_9, we have the states $s_{18} = [5, F(\{2\}, x_{\{2\}}), \{2\}]$ and $s_{19} = [5, F(\{2,4\}, x_{\{2,4\}}), \{2,4\}]$ as $B - T(\{2,4\}) = 1.78 \geq 0$. Since they are final states, they will not be enqueued. We set:

$$\mathscr{F}^* = \min\{\mathscr{F}^*, F(\{2\}, x_{\{2\}}), F(\{2,4\}, x_{\{2,4\}})\},$$
$$\mathscr{F}^* = \min\{671.43, 690.96, 684.87\},$$
$$\mathscr{F}^* = 671.43 \text{ and } s^* = s_{16}.$$

State queue (Q)
$s_{10} = [4, F(\{2,3\}, x_{\{2,3\}}), \{2,3\}]$
$s_{11} = [4, F(\{1\}, x_{\{1\}}), \{1\}]$
$s_{12} = [4, F(\{1,3\}, x_{\{1,3\}}), \{1,3\}]$
$s_{13} = [4, F(\{1,2\}, x_{\{1,2\}}), \{1,2\}]$

In s_{10}, on the other hand, only the final state $s_{20} = [5, F(\{2,3\}, x_{\{2,3\}}), \{2,3\}]$ is generated, since $B - T(\{2,3,4\}) = -1.32 < 0$ and, therefore, the intervention 4 cannot be applied. For that reason, we set:

$$\mathscr{F}^* = \min\{\mathscr{F}^*, F(\{2,3\}, x_{\{2,3\}})\},$$
$$\mathscr{F}^* = \min\{671.43, 631.53\},$$
$$\mathscr{F}^* = 631.53 \text{ and } s^* = s_{20}.$$

State queue (Q)
$s_{11} = [4, F(\{1\}, x_{\{1\}}), \{1\}]$
$s_{12} = [4, F(\{1,3\}, x_{\{1,3\}}), \{1,3\}]$
$s_{13} = [4, F(\{1,2\}, x_{\{1,2\}}), \{1,2\}]$

We can apply the intervention 4 in s_{11} as $B - T(\{1,4\}) = 0.66 \geq 0$. Consequently, the final states $s_{21} = [5, F(\{1\}, x_{\{1\}}), \{1\}]$ and $s_{22} = [5, F(\{1,4\}, x_{\{1,4\}}), \{1,4\}]$ are generated. Hence, we set:

$$\mathscr{F}^* = \min\{\mathscr{F}^*, F(\{1\}, x_{\{1\}}), F(\{1,4\}, x_{\{1,4\}})\},$$
$$\mathscr{F}^* = \min\{631.53, 434.89, 428.80\},$$
$$\mathscr{F}^* = 428.80 \text{ and } s^* = s_{22}.$$

State queue (Q)
$s_{12} = [4, F(\{1,3\}, x_{\{1,3\}}), \{1,3\}]$
$s_{13} = [4, F(\{1,2\}, x_{\{1,2\}}), \{1,2\}]$

However, in s_{12}, the intervention 4 cannot be applied, since $B - T(\{1,3,4\}) = -2.44 < 0$. Hence, we have only the final state $s_{23} = [5, F(\{1,3\}, x_{\{1,3\}}), \{1,3\}]$, and we set:

$$\mathscr{F}^* = \min\{\mathscr{F}^*, F(\{1,3\}, x_{\{1,3\}})\},$$
$$\mathscr{F}^* = \min\{428.80, 340.75\},$$
$$\mathscr{F}^* = 340.75 \text{ and } s^* = s_{23}.$$

State queue (Q)
$s_{13} = [4, F(\{1,2\}, x_{\{1,2\}}), \{1,2\}]$

Lastly, in s_{13}, only the final state $s_{24} = [5, F(\{1,2\}, x_{\{1,2\}}), \{1,2\}]$ is generated as $B - T(1,2,4) = -1.12 < 0$. We set:

$$\mathscr{F}^* = \min\{\mathscr{F}^*, F(\{1,2\}, x_{\{1,2\}})\},$$
$$\mathscr{F}^* = \min\{340.75, 370.19\},$$
$$\mathscr{F}^* = 340.75 \text{ and } s^* = s_{23}.$$

Since Q is empty, the method stops and, therefore, the optimal solution is applying the interventions $K^* = \{1,3\}$, whose total cost perceived by the cyclists is $\mathscr{F}^* = F(\{1,3\}, x_{\{1,3\}}) = 340.75$.

Example 2: Consider a graph $G = (V,A)$, with set of vertices $V = \{1,2,3,4\}$ and set of arcs $A = \{(1,2),(2,3),(3,4),(1,4),(2,4)\}$ (see Figure 4.3a). In this example, we assume a single road feature (e.g., distance), one cyclist, and a single OD pair $w = (1,4)$. Moreover, consider a set $K = \{1,2,3,4\}$ of interventions and the following values for the parameters regarding basic costs and cost reductions.

Arc	**Basic costs**	**Cost reductions**	
(i,j)	$c^1_{0_{ij}}$	$K^{(i,j)}$	ϕ^1_{ijk}
(1, 2)	40	{1}	10
(2, 3)	50	{2}	20
(3, 4)	20	{4}	15
(1, 4)	100	∅	0
(2, 4)	80	{3}	40

The optimal path followed by the cyclist in the no-intervention scenario is illustrated in Figure 4.3b, with a cost of $F(\{\}, x_{\{\}}) = 100$. Suppose that we are exploring the states (4, $F(\{1,3\}, x_{\{1,3\}})$, $\{1,3\}$) and (4, $F(\{1,2\}, x_{\{1,2\}})$, $\{1,2\}$), with solution costs of $F(\{1,3\}, x_{\{1,3\}}) = 70$ (see Figure 4.3c) and $F(\{1,2\}, x_{\{1,2\}}) = 80$ (see Figure 4.3d), respectively.

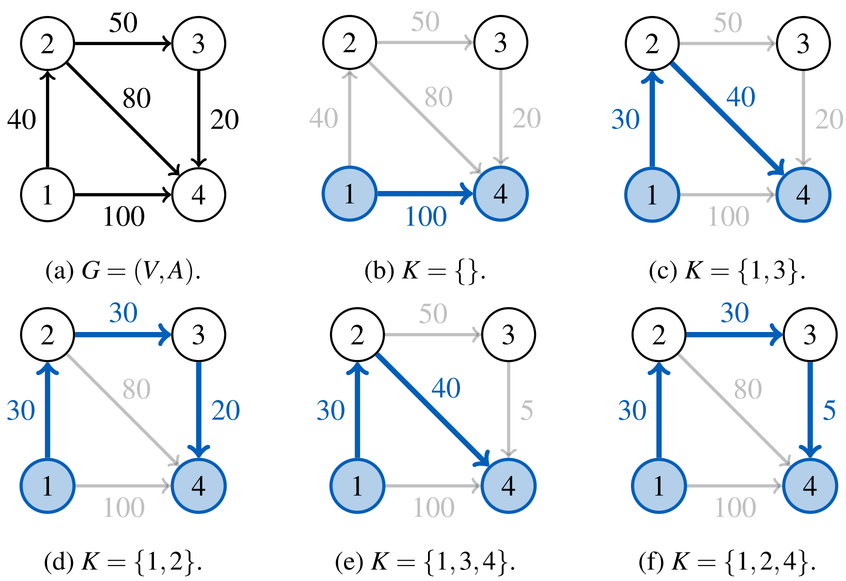


Figure 4.3: Example illustrating the necessity of tracking all possible subsets of interventions.

The main assumption in this example is that the subsets $\{1,2\}$ and $\{1,3\}$ result in the same residual budget, meaning that $T(\{1,3\}) = T(\{1,2\})$. Hence, if the dominance rule based on the residual budget were applied in CRM-DP, the state (4, $F(\{1,2\}, x_{\{1,2\}})$, $\{1,2\}$) would be eliminated, as $F(\{1,2\}, x_{\{1,2\}}) > F(\{1,3\}, x_{\{1,3\}})$. However, applying interventions $\{1,2,4\}$ results in a cost of $F(\{1,2,4\}, x_{\{1,2,4\}}) = 65$ (see Figure 4.3f), which is lower than the cost of $F(\{1,3,4\}, X_{\{1,3,4\}}) = 70$ when applying interventions $\{1,3,4\}$ (see Figure 4.3e). Therefore, the dominance rule based on residual budget is not valid for this problem, requiring the tracking of all possible subsets of interventions.

4.3.2 Branch-and-Bound

The second proposed method is based on the well-known branch-and-bound (BB) algorithm [60], widely used to solve combinatorial optimization problems. This algorithm consists of splitting the problem into subproblems (branching) and using bounds to eliminate unpromising solutions (bounding).

In the context of the problem we aim to solve, the minimum-cost solution corresponds to the subproblem in which all interventions are applied. Although this solution is not feasible due to budget constraint (4.1c), its cost serves as a lower bound for the optimal solution. In addition, at each level of the BB tree, removing one intervention results in a node whose solution cost is greater than or equal to the cost of its parent node.

Given the above, the proposed BB approach, referred to as the *cost reduction model – branch-and-bound* (CRM-BB), initializes the BB tree with a root node representing the application of all possible interventions. New nodes are generated according to a branching criterion (*branching strategy*) and, depending on the strategy, each node corresponds to either keeping or removing a specific intervention. The intervention considered at each node is selected based on a predefined criterion (*intervention selection strategy*). More precisely, the first non-analyzed intervention is selected from a sequence of interventions whose order depends on the chosen strategy.

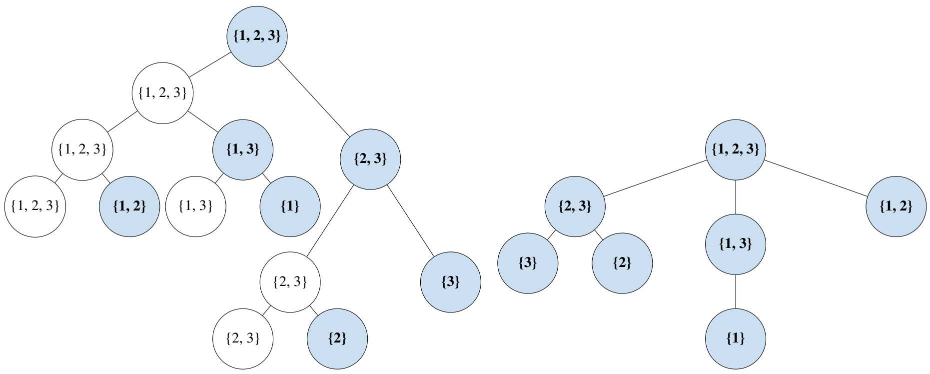
Since the set of selected interventions is predetermined at each node, as a result of the relaxed budget constraint and the adopted branching strategies, exploring a node consists of solving formulation (4.7a)–(4.7c). The solution cost obtained from solving a node represents its corresponding lower bound. Moreover, upper bounds are obtained when nodes containing a subset of interventions that satisfies the budget constraint are reached. Nodes of the BB tree are selected according to *node selection strategies*, which are based on breadth-first search (BFS), depth-first search (DFS), and best-bound search (BBS).

The proposed BB approach considers different strategies for branching and intervention selection. Figure 4.4 illustrates the two branching strategies considered in this work, namely binary branching and non-binary branching, through an example where $K = \{1,2,3\}$ and interventions are selected in ascending order of their identification numbers.

In binary branching, each node generates two child nodes: the left child represents keeping the selected intervention, while the right child represents removing it (see Figure 4.4a). In contrast, non-binary branching generates one child node for each intervention that has not been removed (see Figure 4.4b).

To avoid generating duplicate nodes with identical subsets of interventions in the non-binary case, we only remove interventions with identification numbers greater than that of the intervention removed in the parent node. For instance, in Figure 4.4b, the node corresponding to the subset $\{1,3\}$ is generated from the root node by removing intervention 2. Consequently, from this node, only intervention 3 is removed, resulting in the child node corresponding to the subset $\{1\}$.

All possible nodes in the two examples from Figure 4.4 are explicitly enumerated to compare the strategies with respect to lower bound computation. In both cases, the highlighted nodes indicate where new lower bounds are computed. In the binary case, these nodes correspond to the right child nodes (see Figure 4.4a), while all nodes in the non-binary case require lower bound computation (see Figure 4.4b).



(a) Binary BB tree. (b) Non-binary BB tree.

Figure 4.4: Branching strategies.

Regarding intervention selection strategies, interventions are ordered according to their identification numbers (i.e., $1,2,\ldots,|K|$) in the non-binary branching case. For binary branching, we propose four selection criteria: building costs (BC), cost reductions (CR), cost ratios (CRT), and intervention impacts (II), which are described in what follows:

- For the BC (building costs) criterion, interventions are ranked in non-ascending order based on their total building cost, $T(\{k\})$, where $\{k\}$ represents a subset containing only intervention k (see Eq. (4.6)). The intuition behind this criterion is to find feasible solutions more quickly by removing the most expensive interventions first.

- For the CR (cost reductions) criterion, interventions are ranked in non-ascending order according to the total cost reduction, calculated as

$$\Phi(k) = \sum_{(i,j)\in A_k} \sum_{h=1}^{r} \phi_{ijk}^{h},$$

  where $\Phi(k)$ represents the aggregated cost reduction associated with intervention k. The idea of this criterion is to analyze the interventions that have greater impact on the solution cost.

- For the CRT (cost ratios) criterion, interventions are ranked in non-ascending order according to the ratio of total cost reductions to total building costs, given by $\left(\Phi(k)/T(\{k\})\right)$. This third criterion allows the method to capture the trade-off between the two previous criteria, identifying interventions whose impact on the solution cost is high relative to their total building cost.

- For the II (intervention impacts) criterion, the impact of intervention $k \in K_s$ is then measured as the difference between the total cost of applying the intervention set K_s, $F(K_s, x_{K_s})$, and the total cost after applying the same intervention set without k, $F(K_s \setminus \{k\}, x_{K_s \setminus \{k\}})$, where x_{K_s} and $x_{K_s \setminus \{k\}}$ represent the optimal flow solutions for the respective scenarios. Interventions are ranked in non-ascending order of their impact. Due to the higher cost to evaluate this criterion with respect to the previous ones, the ranking is computed only for nodes located at three different depths of the BB tree: 25%, 50%, and 75%. The motivation for this criterion is to dynamically select the interventions with the greatest impact on the solution cost during the exploration, taking into account the subset of interventions already selected at each node, as opposed to statically sorting the set of interventions before starting the BB tree exploration.

In what follows, the criterion for selecting interventions will be included as part of the branching strategies for simplification. For example, binary branching with intervention selection based on building costs will be denoted as *building costs binary (BCB)*. In addition, the non-binary branching strategy with the intervention selection based on the identification number will be referred as *unbalanced branching strategy*. The pseudocode for CRM-BB is presented in Algorithm 6.

Algorithm 6: CRM-BB

```
function CRM-BB(data, initial_ub, branching_strategy, search_mode)
is_feasible ← FALSE ;
best_cost ← initial_ub ;
K_{n_b} ← {} ;
K_0 ← sortInterventions(K, branching_strategy) ;
c_0, x_0 ← shortestPaths(data, K_0) ;
n_0 ← (K_0, F(K_0, x_0), −1) ;
BB_tree.add(n_0) ;
while |BB_tree| > 0 do
    n_cr ← selectNode(search_mode) ;
    is_feasible ← checkFeasibility(K_{n_cr}) ;
    if is_feasible then
        if (F(K_{n_cr}, x_{K_{n_cr}}) < best_cost) or (F(K_{n_cr}, x_{K_{n_cr}}) =
          best_cost and |K_{n_cr}| < |K_{n_b}|) then
            K_{n_b} ← K_{n_cr} ;
            best_cost ← F(K_{n_cr}, x_{K_{n_cr}}) ;
    if F(K_{n_cr}, x_{K_{n_cr}}) ≤ best_cost then
        branching(n_cr) ;
K* ← K_{n_b} ;
x*, c* ← shortestPaths(data, K*) ;
return x*, c*, K* ;
```

In Algorithm 6, lines 2–7 initialize the root node and parameters of the method. In particular, line 3 uses the solution costs obtained through the heuristics developed in this research (see Sections 4.3.3 and 4.3.4) as initial upper bounds for the optimal solution, while line 5 sorts the initial set of interventions using the function `sortInterventions`$(K, branching\_strategy)$. The root node is then added to the BB tree in line 8. The main loop runs from line 9 to line 17. The function `selectNode`$(search\_mode)$ selects the current node (line 10) according to the chosen search strategy (BFS, DFS, or BBS), while the function `checkFeasibility`$(K_{n_{cr}})$ checks the feasibility of the node (line 11) by verifying whether the total building cost is less than or equal to the available budget. If the node is feasible, the best solution is updated either when the current solution cost is less than the best known cost, or, if the costs are equal, when the current solution has fewer interventions than the best solution (see line 13). Line 16 checks whether the current solution cost is less than or equal to the best cost. If so, branching is performed in line 17 using the function `branching`(n_{cr}). Note that the lower bound of each node is computed during the execution of `branching`(n_{cr}), when the node is created. Computing bounds at node initialization is necessary because the node's cost influences its position in the BB tree when the BBS strategy is used for node selection. Finally, lines 18–20 return the flow vector x^*, the cost vector c^*, and the set of interventions K^*, which represent the optimal solution.

After explaining the complete procedure, we illustrate the CRM-BB method using the example from Section 4.3.1 with the best strategy combination (BCB + BBS). Note that since interventions are sorted by the largest sum of building costs (see Table 4.6), they are removed in the sequence: $3-1-4-2$.

Node 0	
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	
Node 1	**Node 2**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,4\}, x_{\{1,2,4\}}) = 364.10$

We start by solving the root node, 0, which considers all possible interventions $\{1,2,3,4\}$. From this node, we generate two child nodes: node 1, by keeping all interventions, and node 2, which excludes intervention 3. Since we use BBS as the node selection strategy, node 1 is explored next, as it has the lowest cost among the open nodes.

Node 0	
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	
Node 1	**Node 2**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,4\}, x_{\{1,2,4\}}) = 364.10$
Node 3	**Node 4**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{2,3,4\}, x_{\{2,3,4\}}) = 631.53$

From node 1, we generate child nodes 3 and 4, where node 3 maintains the intervention set of node 1, and node 4 removes intervention 1. Since node 3 has the lowest cost among the open nodes, it is explored next.

Node 0	
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	
Node 1	**Node 2**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,4\}, x_{\{1,2,4\}}) = 364.10$
Node 3	**Node 4**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{2,3,4\}, x_{\{2,3,4\}}) = 631.53$
Node 5	**Node 6**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,3\}, x_{\{1,2,3\}}) = 299.92$

Following the same procedure, we generate child nodes 5 and 6 from node 3, where node 5 corresponds to the intervention set $\{1,2,3,4\}$ and node 6 to $\{1,2,3\}$. Note that nodes 5 and 6 have the same solution cost. Ties are resolved based on the number of interventions. Therefore, node 6 is the next to be explored, as it has fewer interventions than node 5.

Node 0	
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	
Node 1	**Node 2**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,4\}, x_{\{1,2,4\}}) = 364.10$
Node 3	**Node 4**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{2,3,4\}, x_{\{2,3,4\}}) = 631.53$
Node 5	**Node 6**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,3\}, x_{\{1,2,3\}}) = 299.92$
Node 7	**Node 8**
$F(\{1,2,3\}, x_{\{1,2,3\}}) = 299.92$	$F(\{1,3\}, x_{\{1,3\}}) = 340.75$

From node 6, we generate node 7, which includes intervention 2, and node 8, which excludes it. Considering the criterion of selecting the node with the lowest cost and that the ties are resolved based on the lowest number of interventions, the next node to be explored is node 7.

Node 0	
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	
Node 1	**Node 2**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,4\}, x_{\{1,2,4\}}) = 364.10$
Node 3	**Node 4**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{2,3,4\}, x_{\{2,3,4\}}) = 631.53$
Node 5	**Node 6**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,3\}, x_{\{1,2,3\}}) = 299.92$
Node 7	**Node 8**
$F(\{1,2,3\}, x_{\{1,2,3\}}) = 299.92$	$F(\{1,3\}, x_{\{1,3\}}) = 340.75$

In node 7, all possible interventions have already been removed. Therefore, no more child nodes can be generated from it. The next node to be explored is node 5 since it has the lowest cost among the open nodes.

Node 0	
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	
Node 1	**Node 2**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,4\}, x_{\{1,2,4\}}) = 364.10$
Node 3	**Node 4**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{2,3,4\}, x_{\{2,3,4\}}) = 631.53$
Node 5	**Node 6**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,3\}, x_{\{1,2,3\}}) = 299.92$
Node 7	**Node 8**
$F(\{1,2,3\}, x_{\{1,2,3\}}) = 299.92$	$F(\{1,3\}, x_{\{1,3\}}) = 340.75$
Node 9	**Node 10**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,3,4\}, x_{\{1,3,4\}}) = 340.75$

In node 5, the child nodes 9 and 10 are generated by keeping and removing intervention 2, respectively. Since node 9 has the lowest cost, it is explored next.

Node 0	
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	
Node 1	**Node 2**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,4\}, x_{\{1,2,4\}}) = 364.10$
Node 3	**Node 4**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{2,3,4\}, x_{\{2,3,4\}}) = 631.53$
Node 5	**Node 6**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,3\}, x_{\{1,2,3\}}) = 299.92$
Node 7	**Node 8**
$F(\{1,2,3\}, x_{\{1,2,3\}}) = 299.92$	$F(\{1,3\}, x_{\{1,3\}}) = 340.75$
Node 9	**Node 10**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,3,4\}, x_{\{1,3,4\}}) = 340.75$

Similar to node 7, node 9 is also a leaf node since there are no remaining interventions to remove. Nodes 8 and 10 have the same cost, but node 8 has fewer interventions. Therefore, node 8 is selected to be explored.

Node 0	
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	
Node 1	**Node 2**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,4\}, x_{\{1,2,4\}}) = 364.10$
Node 3	**Node 4**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{2,3,4\}, x_{\{2,3,4\}}) = 631.53$
Node 5	**Node 6**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,3\}, x_{\{1,2,3\}}) = 299.92$
Node 7	**Node 8**
$F(\{1,2,3\}, x_{\{1,2,3\}}) = 299.92$	$F(\{1,3\}, x_{\{1,3\}}) = 340.75$
Node 9	**Node 10**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,3,4\}, x_{\{1,3,4\}}) = 340.75$

Node 8 is a special node for two reasons. First, it is a leaf node because no remaining interventions can be removed. Second, it is a feasible node with respect to the budget constraints (4.1c). Therefore, node 8 becomes the node corresponding to the best solution. Node 10 is the next to be explored.

Node 0	
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	
Node 1	**Node 2**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,4\}, x_{\{1,2,4\}}) = 364.10$
Node 3	**Node 4**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{2,3,4\}, x_{\{2,3,4\}}) = 631.53$
Node 5	**Node 6**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,3\}, x_{\{1,2,3\}}) = 299.92$
Node 7	**Node 8**
$F(\{1,2,3\}, x_{\{1,2,3\}}) = 299.92$	$F(\{1,3\}, x_{\{1,3\}}) = 340.75$
Node 9	**Node 10**
$F(\{1,2,3,4\}, x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,3,4\}, x_{\{1,3,4\}}) = 340.75$

Node 10 is also a leaf node and, although it has the same solution cost as the best node, it is infeasible with respect to the budget constraints (4.1c). Therefore, our best solution is still the solution corresponding to node 8. Node 2 is explored next, as it has the lowest cost among the unexplored nodes.

Node 0	
$F(\{1,2,3,4\},x_{\{1,2,3,4\}}) = 299.92$	
Node 1	**Node 2**
$F(\{1,2,3,4\},x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,4\},x_{\{1,2,4\}}) = 364.10$
Node 3	**Node 4**
$F(\{1,2,3,4\},x_{\{1,2,3,4\}}) = 299.92$	$F(\{2,3,4\},x_{\{2,3,4\}}) = 631.53$
Node 5	**Node 6**
$F(\{1,2,3,4\},x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,3\},x_{\{1,2,3\}}) = 299.92$
Node 7	**Node 8**
$F(\{1,2,3\},x_{\{1,2,3\}}) = 299.92$	$F(\{1,3\},x_{\{1,3\}}) = 340.75$
Node 9	**Node 10**
$F(\{1,2,3,4\},x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,3,4\},x_{\{1,3,4\}}) = 340.75$

In node 2, child nodes could be generated, as there are interventions to be removed, specifically intervention 1 in this case. However, since

$$F(\{1,2,4\},x_{\{1,2,4\}}) = 364.10 > F(\{1,3\},x_{\{1,3\}}) = 340.75,$$

which corresponds to the best known solution, we know that all nodes generated from node 2 will have a cost greater than the best known solution cost. Therefore, we can prune this node. A similar situation occurs with node 4, the last open node. We could generate child nodes from it by removing intervention 4, but since

$$F(\{2,3,4\},x_{\{2,3,4\}}) = 631.53 > F(\{1,3\},x_{\{1,3\}}) = 340.75,$$

we can also prune this node.

Node 0	
$F(\{1,2,3,4\},x_{\{1,2,3,4\}}) = 299.92$	
Node 1	**Node 2**
$F(\{1,2,3,4\},x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,4\},x_{\{1,2,4\}}) = 364.10$
Node 3	**Node 4**
$F(\{1,2,3,4\},x_{\{1,2,3,4\}}) = 299.92$	$F(\{2,3,4\},x_{\{2,3,4\}}) = 631.53$
Node 5	**Node 6**
$F(\{1,2,3,4\},x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,2,3\},x_{\{1,2,3\}}) = 299.92$
Node 7	**Node 8**
$F(\{1,2,3\},x_{\{1,2,3\}}) = 299.92$	$F(\{1,3\},x_{\{1,3\}}) = 340.75$
Node 9	**Node 10**
$F(\{1,2,3,4\},x_{\{1,2,3,4\}}) = 299.92$	$F(\{1,3,4\},x_{\{1,3,4\}}) = 340.75$

As there are no more nodes to be explored, the method stops, and the optimal solution is to apply the interventions $K^* = \{1,3\}$, with a cost of $\mathscr{F}^* = F(\{1,3\},x_{\{1,3\}}) = 340.75$.

4.3.3 Heuristic approach based on an approximation as a knapsack problem

The third proposed method, named *cost reduction model – heuristic based on knapsack dynamic programming* (CRM-HKDP), is a procedure based on dynamic programming for solving the knapsack problem. In this approach, we disregard the possible dependency between interventions as outlined in *Example 2*, and we approximate the problem as a knapsack problem, where each item in the knapsack represents a possible intervention $k \in K$. The profit of each intervention/item is measured as the marginal gain of the intervention with respect to the no intervention scenario. Hence, the method consists of solving the following knapsack problem formulation:

$$\max \sum_{k \in K} (F(\{\},x_{\{\}}) - F(\{k\},x_{\{k\}}))y_k \tag{4.8a}$$

s.t.

$$\sum_{k \in K} \lceil T(\{k\}) \rceil y_k \leq B \tag{4.8b}$$

$$y_k \in \{0,1\} \quad k \in K, \tag{4.8c}$$

where the weight of item k corresponds to the total building cost (see Eq.(4.6)) of its associated intervention, $F(\{\},x_{\{\}})$ is the total cost with no interventions, and $F(\{k\},x_{\{k\}})$ is the total cost after applying intervention k. We can solve (4.8a)–(4.8c) by using the standard dynamic programming approach for the knapsack problem (hereafter referred to as SDPKP), whose complexity is $O(|K|B)$. Since it is necessary to compute $|W|q$ shortest paths using Dijkstra's algorithm, for each intervention $k \in K$, to build the knapsack items corresponding to (4.8a)–(4.8c), we proved the following complexity result for CRM-HKDP.

Proposition 7. *The overall complexity of CRM-HKDP is* $O(|K||W|q(|A|+|V|\log(|V|))+|K|B)$.

The pseudocode for CRM-HKDP is presented in Algorithm 7. In this algorithm, lines 2-10 initialize the first row and first column of the decision matrix, as well as the vector of items, which contains information about the interventions. The main loop runs from line 11 to line 21, where the algorithm checks, for each value of residual budget, whether an item should be added to the knapsack. After completing the decision matrix, the solution is retrieved in lines 22 to 25. Finally, based on the selected interventions, the flows and costs are computed in line 26 and the best solution found is returned in line 27.

The same numerical example (see Table 4.6) from Section 4.3.1 will be used to illustrate how CRM-HKDP works (see Table 4.6). As this approach is equivalent to SDPKP, the details of how the method works will be omitted. From the previous example, we have $f_0 = F(\{\},x_{\{\}}) = 755.65$, $F(\{1\},x_{\{1\}}) = 434.89$, $F(\{2\},x_{\{2\}}) = 690.96$, $F(\{3\},x_{\{3\}}) = 671.43$, and $F(\{4\},x_{\{4\}}) = 749.57$. Therefore, we can compute the weights and profits for each item, as indicated in Table 4.7.

Table 4.7: Values of weights and profits for the CRM-DP example.

Item	Weight	Profit
1	$\lceil T(\{1\}) \rceil = 3$	$f_0 - F(\{1\},x_{\{1\}}) = 320.76$
2	$\lceil T(\{2\}) \rceil = 2$	$f_0 - F(\{2\},x_{\{2\}}) = 64.69$
3	$\lceil T(\{3\}) \rceil = 4$	$f_0 - F(\{3\},x_{\{3\}}) = 84.22$
4	$\lceil T(\{4\}) \rceil = 3$	$f_0 - F(\{4\},x_{\{4\}}) = 6.08$

Algorithm 7: CRM-HKDP

```
function CRM-HKDP(data)
x_{}, c_{} ← shortestPaths(data, {}) ;
f_0 ← F({}, x_{}); items ← [ ] ;
matrix ← [ ]_{|K|+1×B+1}; matrix[0][0] ← (0, FALSE) ;
for k ← 1 to |K| do
    x_{k}, c_{k} ← shortestPaths(data, {k}) ;
    items.append(⌈T({k})⌉, f_0 − F({k}, x_{k})) ;
    matrix[k][0] ← (0, FALSE) ;
for b ← 1 to B do
    matrix[0][b] ← (0, FALSE) ;
for k ← 1 to |K| do
    for b ← 1 to B do
        if items[k−1].weight ≤ b then
            budget ← b − items[k−1].weight ;
            profit ← items[k−1].profit + matrix[k−1][budget].profit ;
            if profit > matrix[k−1][b].profit then
                matrix[k][b] ← (profit, TRUE) ;
            else
                matrix[k][b] ← (matrix[k−1][b].profit, FALSE) ;
        else
            matrix[k][b] ← (matrix[k−1][b].profit, FALSE) ;
b ← B ;
for k ← |K| to 0 do
    if matrix[k][b].decision = TRUE then
        K* ← K* ∪ {k}; b ← b − items[k−1].weight ;
x*, c* ← shortestPaths(data, K*) ;
return x*, c*, K* ;
```

The solution to this knapsack instance is selecting items 1 and 2, which, in the context of the studied problem, corresponds to the subset of interventions $K^* = \{1,2\}$ with cost $\mathscr{F}^* = F(\{1,2\}, x_{\{1,2\}}) = 370.19$.

4.3.4 Alternating method

The fourth proposed method, named *cost reduction model – alternating method* (CRM-AM), is similar to CRM-HKDP, since both methods propose approximations to the knapsack problem. In this case, instead of assuming intervention independence as in the first heuristic, CRM-AM considers a knapsack where the profits of the items associated to the interventions change during the execution of the method.

CRM-AM starts by solving formulation (4.7a)–(4.7c) for the scenario where no intervention is applied, obtaining the flow vector x. By fixing the flows in formulation (4.1a)–(4.1g), we derive a new formulation:

$$\min \sum_{w \in W} \sum_{(i,j) \in A} \sum_{\ell=1}^{q} \sum_{h=1}^{r} \left(c_{0_{ij}}^{h} p_h^{\ell} x_{ij}^{w\ell} - \sum_{k \in K} (\phi_{ijk}^{h} p_h^{\ell} x_{ij}^{w\ell}) y_k \right) \tag{4.9a}$$

s.t.

$$\sum_{k \in K} \sum_{(i,j) \in A_k} \tau_{ij}^{k} y_k \leq B \tag{4.9b}$$

$$y_k \in \{0,1\} \qquad k \in K, \tag{4.9c}$$

which is equivalent to the mathematical formulation of the knapsack problem, as the first term in the objective function is constant. We solve this formulation using SDPKP, after which we obtain a new set of interventions that minimizes the total cost perceived by cyclists. This procedure is repeated until the solutions obtained by solving formulations (4.7a)–(4.7c) and (4.9a)–(4.9c) have the same cost. We denote with I the number of iterations until cost convergence. Hence, the complexity of CRM-AM is $O(I|W|q(|A| + |V|\log(|V|)))$, for computing the $|W|q$ shortest paths in each iteration, plus the complexity $O(I|K|B)$ of solving I knapsack problems. Thus, we proved the following complexity result for CRM-AM.

Proposition 8. *The overall complexity of CRM-AM is*

$$O(I\,(|W|q\,(|A|+|V|\log(|V|))+|K|B)).$$

Compared to CRM-HKDP, CRM-AM outperforms the previous heuristic when the number of iterations I is small compared to the number of interventions $|K|$. The value of I cannot be established in advance and may vary from instance to instance. However, in practice, we observed that it generally holds that I is significantly lower than $|K|$.

Algorithm 8 describes the pseudocode for this method, where `knapsack`$(items)$ refers to solving formulation (4.9a)–(4.9c) using SDPKP. In this algorithm, line 2 initializes the current subset of interventions, $\bar{K}$, with the empty set. Lines 5 to 12 solve formulation (4.7a)–(4.7c) considering $\bar{K}$, whereas lines 13 to 21 solve formulation (4.9a)–(4.9c) using the flow vector x, obtained after solving (4.7a)–(4.7c). Line 22 checks whether the method has converged. If yes, the method stops and the solution is returned in line 26. Otherwise, the method starts a new iteration. Note that to prevent the method from looping indefinitely without cost convergence, a parameter MAX_ITER is used to limit the number of iterations. However, convergence was achieved in all computational experiments (see Section 4.4). Figure 4.5 presents the method in a flowchart.

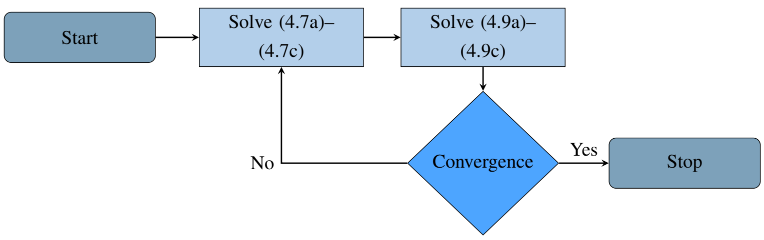


Figure 4.5: Cost reduction problem solved using an alternating approach.

Since we have already explained how to obtain the shortest paths and solve the knapsack problem (see Sections 4.3.1 and 4.3.3), we will not describe in detail how CRM-AM works for the example considered in the description of the three previous methods. Nevertheless, Figure 4.6 illustrates the solution obtained the method.

Algorithm 8: CRM-AM

```
function CRM-AM(data)
K̄ ← {} ;
count_iter ← 0 ;
while count_iter < MAX_ITER do
    x, c ← shortestPaths(data, K̄) ;
    cost_SP ← F(K̄, x) ;
    t_0 ← 0 ;
    for w ∈ W do
        for (i, j) ∈ A do
            for ℓ ← 1 to q do
                for h ← 1 to r do
                    t_0 ← t_0 + c^h_{0_{ij}} p^ℓ_h x^{wℓ}_{ij} ;
    for k ← 1 to |K| do
        profit_k ← 0 ;
        for w ∈ W do
            for (i, j) ∈ A do
                for ℓ ← 1 to q do
                    for h ← 1 to r do
                        profit_k ← profit_k + φ^h_{ijk} p^ℓ_h x^{wℓ}_{ij} ;
        items.append(k, ⌈T({k})⌉, profit_k) ;
    cost_KP, K̄ ← knapsack(items) ;
    if t_0 − cost_KP = cost_SP then
        K* ← K̄; x* ← x; c* ← c;
        break;
    count_iter ← count_iter + 1;
return x*, c*, K* ;
```

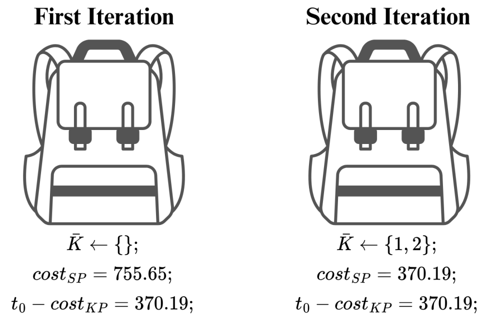


Figure 4.6: Solution provided by CRM-AM method for the considered example.

According to Figure 4.6, the solution obtained by CRM-AM involves applying the interventions $K^* = \{1,2\}$, with a total cost perceived by the cyclists equal to 370.19. Note that it required two iterations to reach this solution. Table 4.8 summarizes the solutions obtained by the four proposed methods.

Table 4.8: Summary of the solutions obtained by the four proposed methods.

Method	K^*	$F(K^*, x_{K^*})$
CRM-DP	{1,3}	340.75
CRM-BB	{1,3}	340.75
CRM-HKDP	{1,2}	370.19
CRM-AM	{1,2}	370.19

4.4 Computational experiments

This section describes the computational experiments conducted using the proposed methods. We ran experiments on a set of randomly generated instances (Section 4.4.1), to evaluate the performance of the methods, as well as on real instances (Section 4.4.2) based on the real data collected from the city of Parma, Italy. All experiments were performed on a machine with an Intel® Xeon® Silver 4316 processor with 2.30 GHz and 125 GB of RAM, running Linux Ubuntu 22.04. CPLEX 22.1.1 was used as a MIP solver. All algorithms were implemented in C++ (g++, version 11.4.0).

Furthermore, for CRM-BB, all branching strategies were evaluated, as described in Appendix D. Based on these tests, BCB proved to be the most effective and was therefore adopted in the computational experiments presented in this section.

4.4.1 Results for the randomly generated instances

We considered graphs $G = (V,A)$ representing grid networks of different sizes. More precisely, we selected sizes in $\{4,8,16,32,40\}$. Instances with graphs derived from grids with sizes of 4 or 8 are classified as small instances, those of size 16 as medium instances, and those of size 32 and 40 as large instances. Based on the grid size, the number of vertices is given by $|V| = (grid\_size)^2$ and the number of arcs by $|A| = 4 \times grid\_size \times (grid\_size - 1)$.

Regarding the OD pairs, we defined $|W| = \lceil 0.6 \times |V| \rceil$, with origins and destinations randomly selected from the set of vertices. Moreover, the demand (u_w) for each OD pair is an integer number randomly sampled between 1 and 50.

For the set of interventions, the number of available interventions $|K|$ was chosen from set $\{10,15,20\}$. The subset of arcs affected by each intervention (A_k) consists of a random number of arcs, ranging from 1 to $\lceil 0.5 \times |A| \rceil$. Each building cost ($\tau_{ij}^k$) is a real number randomly selected between 1 and 10, while the available budget (B) represents a random percentage between 30% and 80% of the total building costs (see Eq.(4.6) with $K_s = K$).

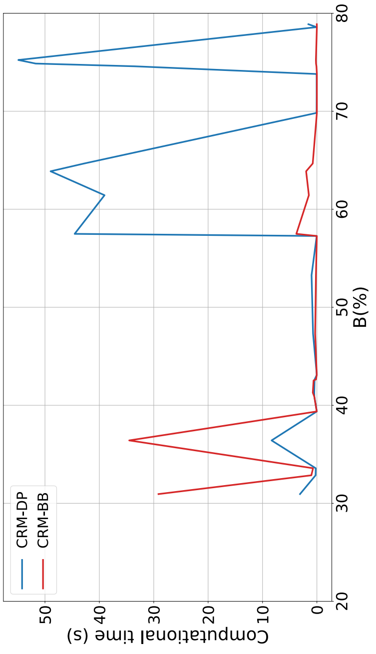
We selected between 3 and 5 road characteristics, that is, $r \in \{3,4,5\}$. For the vector of costs c_0, each cost $c_{0_{ij}}^h$ is a real number randomly selected between 1 and 100. Regarding cost-reduction vector ϕ, we first assigned a randomly chosen real number to each element ϕ_{ijk}^h if $(i,j) \in A_k$, or zero otherwise, for each intervention $k \in K$. Then, we normalized these values by the total cost reduction for arc (i,j) and road feature h, given by $\left(\sum_{k \in K^{(i,j)}} \phi_{ijk}^h\right)$, so that $\phi_{ijk}^h \in [0,1]$ and $\sum_{k \in K} \phi_{ijk}^h = 1$. After normalization, each cost-reduction value ϕ_{ijk}^h is then multiplied by $0.1 \times \lambda_{(i,j)}^h \times c_{0_{ij}}^h$, where $\lambda_{(i,j)}^h$ is a randomly chosen integer between 2 and 8, associated with arc (i,j) and road characteristic h. In this way, we guarantee that all costs, even after the application of all possible interventions, are positive. Recall that this is relevant since shortest paths can be computed through the Dijkstra's algorithm only if the arc costs are nonnegative.

To complete the instance set generation, we selected $q = 5$ random cyclist weights, whose sum of components equals to 1, and the Euclidean distances among them are greater than 1×10^{-5}. Note that, in this research, both the weight vectors p^{ℓ} and the cyclist fractions α_{ℓ} are assumed to be known. In Chapter 3, we discussed how to estimate them based on observed cyclists' behavior on a cycling network. The final set contains 135 instances (27 instances for each grid size), as we generated three different instances for each combination of grid size, number of interventions, and number of road characteristics.

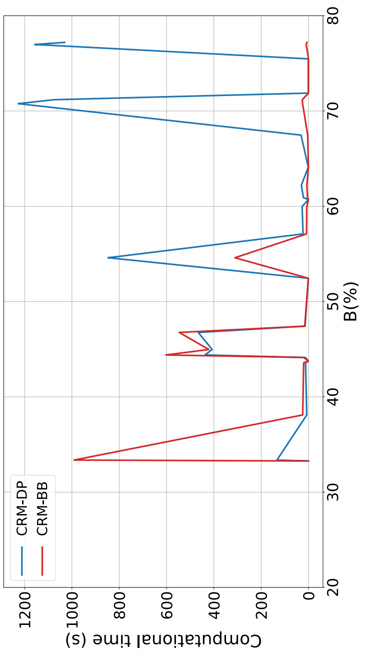
Before presenting the results for all randomly generated instances, we selected a subset of them and performed computational tests to compare the performance of CRM-DP and CRM-BB (with BCB and BBS as branching and node selection strategies, respectively). Specifically, we ran experiments on instances representing grid networks with grid sizes (GS) of up to 16, as shown in Figure 4.7.

It is important to highlight that CRM-DP can be suitable in cases where $B(\%)$ is lower than 50%, as illustrated in Figures 4.7a–4.7c. In such cases, CRM-DP outperformed CRM-BB in terms of computational time, since smaller budget values allow only a few interventions, leading to the optimal state being reached more quickly. In contrast, for these instances, feasible nodes in the BB tree tend to be located at deeper levels. Overall, however, the presence of an implicit enumeration mechanism makes CRM-BB the most suitable exact approach for this problem, as shown in Figure 4.7d, which highlights the number of instances for which each method is the best choice. Therefore, in the following experiments, we used only CRM-BB as the proposed exact approach, since employing a complete enumeration procedure for larger instances becomes intractable.

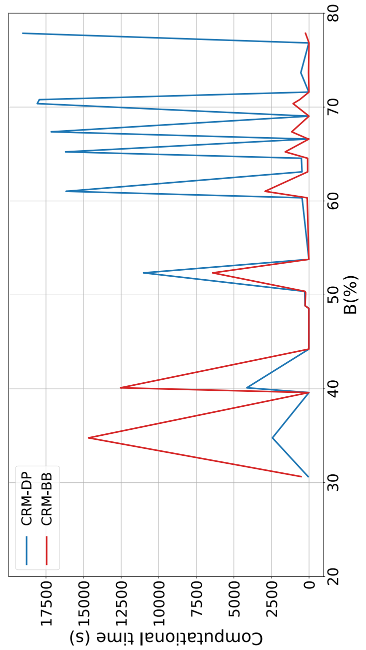
The results obtained for the randomly generated instances are presented in Tables 4.9 and 4.10. Each row of these tables reports the average results for the three instances that share the same characteristics with respect to the graph, OD pairs, intervention set, and number of road features (referred to as an instance group), as indicated in column *#inst*. Column *B(%)* represents the percentage of the total building cost, considering all interventions, that is covered by the available budget. This



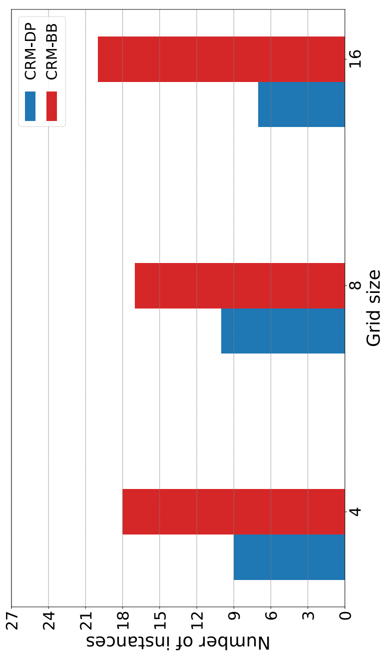
(a) Computation time by instance (GS = 4).



(b) Computation time by instance (GS = 8).



(c) Computation time by instance (GS = 16).



(d) Frequency of best performance.

Figure 4.7: Comparison between CRM-DP and CRM-BB.

parameter is calculated as $B(\%) = (B/T(K)) \times 100$ and refers to the average value over the three instances of each group. Column *#opt* indicates how many instances of each group were solved to optimality, while columns *Gap (%)* and *Time (s)* report the average optimality gaps and computational times, respectively. All experiments were performed considering a time limit of 16 hours (57600 seconds).

Table 4.9 presents a performance analysis of CRM-BB for each of the considered node selection strategies (BFS, DFS, and BBS). In this table, the gap is calculated as $Gap\ (\%) = (ub - lb)/ub$, where ub and lb denote the global upper and lower bounds obtained after executing the method, respectively. CPLEX was also used to solve the MILP reformulation of the problem for small and medium instances. However, the solver required excessive memory, with 100 GB proving insufficient to solve medium instances, as shown in Table 4.9.

From this performance analysis, we observe that all instances representing grid networks with sizes up to 16 were solved to optimality. For grid sizes of 32 and 40, optimal solutions were obtained for all instances with up to 15 interventions. For instances with 20 interventions, only two were solved to optimality when the grid size was 32, while no optimal solutions were found for the case with 20 interventions and grid size 40. The same results were observed for all three node selection strategies. For these large instances (grid sizes of 32 and 40), BBS significantly outperforms BFS and DFS in terms of average gaps. Moreover, BBS generally achieves better computational times than the other strategies.

Focusing on the results obtained with the heuristic procedures, Table 4.10 compares the performance of CRM-HKDP and CRM-AM. The column definitions are the same as in Table 4.9, with the addition of a new parameter, *Impr (%)*, which measures the percentage reduction, in cost, achieved by each heuristic, compared to the case with no interventions, $F(\{\}, x_{\{\}})$. This parameter is defined as

$$Impr(\%) = \frac{F(\{\}, x_{\{\}}) - F(K_m^*, x_m^*)}{F(\{\}, x_{\{\}})} \times 100,$$

where K_m^* and x_m^* denote the intervention sets and flow vector associated with the best solutions obtained by each method $m \in \{$CRM-HKDP, CRM-AM$\}$. Furthermore, the

Table 4.9: Summary of the results obtained for the generated instances with the branch-and-bound method.

Instances				CPLEX	CRM-BB (BFS)			CRM-BB (DFS)			CRM-BB (BBS)		
$\|V\|/\|A\|/\|W\|/\|K\|$	*r*	*B* (%)	*#inst*	*Time* (s)	*#opt*	*Gap* (%)	*Time* (s)	*#opt*	*Gap* (%)	*Time* (s)	*#opt*	*Gap* (%)	*Time* (s)
	3	61.412	3	1.906	3	0.000	0.014	3	0.000	0.014	3	0.000	0.014
16/48/10/10	4	64.594	3	2.015	3	0.000	0.014	3	0.000	0.013	3	0.000	0.014
	5	66.970	3	1.920	3	0.000	0.008	3	0.000	0.007	3	0.000	0.008
	3	35.771	3	5.515	3	0.000	0.700	3	0.000	0.677	3	0.000	0.736
16/48/10/15	4	47.738	3	5.332	3	0.000	0.218	3	0.000	0.209	3	0.000	0.223
	5	54.234	3	9.661	3	0.000	0.464	3	0.000	0.460	3	0.000	0.462
	3	55.747	3	32.064	3	0.000	9.926	3	0.000	9.236	3	0.000	10.256
16/48/10/20	4	67.711	3	15.259	3	0.000	0.952	3	0.000	0.861	3	0.000	0.946
	5	56.379	3	26.583	3	0.000	12.276	3	0.000	11.452	3	0.000	12.826
	3	60.812	3	96.661	3	0.000	0.378	3	0.000	0.378	3	0.000	0.369
64/224/39/10	4	62.878	3	99.054	3	0.000	0.307	3	0.000	0.298	3	0.000	0.291
	5	59.970	3	87.938	3	0.000	0.404	3	0.000	0.406	3	0.000	0.400
	3	52.036	3	440.837	3	0.000	12.640	3	0.000	12.453	3	0.000	12.299
64/224/39/15	4	57.748	3	606.893	3	0.000	10.856	3	0.000	10.639	3	0.000	10.261
	5	50.512	3	615.206	3	0.000	15.255	3	0.000	14.811	3	0.000	14.537
	3	58.926	3	4584.470	3	0.000	247.859	3	0.000	243.312	3	0.000	246.247
64/224/39/20	4	49.513	3	3956.888	3	0.000	551.597	3	0.000	539.624	3	0.000	539.468
	5	64.975	3	12017.193	3	0.000	201.775	3	0.000	193.754	3	0.000	193.964
	3	61.253	3	–//–	3	0.000	6.539	3	0.000	6.514	3	0.000	6.094
256/960/154/10	4	66.550	3	–//–	3	0.000	4.959	3	0.000	5.016	3	0.000	4.744
	5	44.130	3	–//–	3	0.000	12.396	3	0.000	12.361	3	0.000	12.058
	3	47.075	3	–//–	3	0.000	307.211	3	0.000	303.524	3	0.000	294.217
256/960/154/15	4	67.793	3	–//–	3	0.000	69.221	3	0.000	69.039	3	0.000	67.973
	5	54.592	3	–//–	3	0.000	215.737	3	0.000	213.483	3	0.000	209.351
	3	56.792	3	–//–	3	0.000	5944.437	3	0.000	5893.551	3	0.000	5770.820
256/960/154/20	4	72.004	3	–//–	3	0.000	702.579	3	0.000	686.385	3	0.000	670.239
	5	51.162	3	–//–	3	0.000	7549.701	3	0.000	7458.227	3	0.000	7293.532
	3	37.604	3	–//–	3	0.000	245.835	3	0.000	245.169	3	0.000	240.400
1024/3968/615/10	4	58.516	3	–//–	3	0.000	120.072	3	0.000	119.602	3	0.000	117.170
	5	51.119	3	–//–	3	0.000	163.848	3	0.000	164.270	3	0.000	160.838
	3	53.708	3	–//–	3	0.000	4581.731	3	0.000	4544.019	3	0.000	4433.812
1024/3968/615/15	4	61.863	3	–//–	3	0.000	2793.548	3	0.000	2784.835	3	0.000	2732.203
	5	37.228	3	–//–	3	0.000	7491.121	3	0.000	7453.703	3	0.000	7290.507
	3	46.919	3	–//–	0	36.677	57600.000	0	36.654	57600.000	0	10.082	57600.000
1024/3968/615/20	4	56.687	3	–//–	1	22.757	43606.623	1	22.757	43589.087	1	5.304	43387.748
	5	55.902	3	–//–	1	22.864	43702.481	1	22.855	43749.261	1	5.987	43544.516
	3	56.998	3	–//–	3	0.000	349.107	3	0.000	349.710	3	0.000	339.983
1600/6240/960/10	4	45.316	3	–//–	3	0.000	500.781	3	0.000	500.359	3	0.000	490.576
	5	60.913	3	–//–	3	0.000	282.558	3	0.000	282.835	3	0.000	276.314
	3	39.760	3	–//–	3	0.000	18630.853	3	0.000	18654.874	3	0.000	18046.156
1600/6240/960/15	4	50.602	3	–//–	3	0.000	12325.981	3	0.000	12293.713	3	0.000	11992.578
	5	60.779	3	–//–	3	0.000	7810.910	3	0.000	7783.660	3	0.000	7583.038
	3	56.776	3	–//–	0	31.101	57600.000	0	31.068	57600.000	0	8.343	57600.000
1600/6240/960/20	4	55.867	3	–//–	0	30.877	57600.000	0	30.893	57600.000	0	9.576	57600.000
	5	50.639	3	–//–	0	33.212	57600.000	0	33.218	57600.000	0	12.617	57600.000

gaps are calculated as

$$Gap(\%) = \frac{F(K_m^*, x_m^*) - F(K_{best}, x_{best})}{F(K_{best}, x_{best})} \times 100,$$

where K_{best} and x_{best} correspond to the intervention set and flow vector associated with the best feasible solution identified by CRM-BB, including the optimal solution when it is found. For cases where CRM-BB was not able to find feasible solutions for instances of a group, the gaps of the heuristics with respect to the exact method are calculated using the global lower bound instead of the cost of the best feasible solution. This occurs for instances with grid sizes of 32 and 40, both considering 20 interventions.

From Table 4.10, we observe that CRM-HKDP and CRM-AM were capable of finding solutions very close to the known optimal ones, with average gaps mostly below 1% for both heuristics. In addition, both methods provide very similar solutions, since the maximum difference in cost improvement averages (see columns *Impr (%)*) between the two heuristics is below 1 percentage point. Their performance is also similar with respect to the number of optimal solutions found, with 38 and 33 optimal solutions obtained by CRM-HKDP and CRM-AM, respectively. Despite this similarity in solution quality, CRM-AM outperforms CRM-HKDP in terms of computational time, especially in the experiments with large instances. This can be explained by the fact that the number of iterations until cost convergence, I, is smaller than the number of possible interventions, $|K|$, as discussed in the complexity analysis of both methods (see Propositions 7 and 8).

4.4.2 Results for the Parma instances

We considered a graph $G = (V, A)$ with 44820 nodes and 98578 arcs to represent the city of Parma, Italy (Figure 4.8a). The set of OD pairs W and their corresponding demands u_w, for $w \in W$, were obtained from census data on the citizens of Parma [55], resulting in a total of 3806 pairs. We selected three road features ($r = 3$) for the case studies of Parma: length ($h = 1$, also referred to as distance), safety ($h = 2$), and practicability ($h = 3$). The values of the basic costs $c^h_{0_{ij}}$, for $(i, j) \in A$ and $h \in \{1, \dots, r\}$, are

Table 4.10: Summary of the results obtained for the generated instances with the heuristic methods.

Instances				**CRM-HKDP**				**CRM-AM**			
$\|V\|/\|A\|/\|W\|/\|K\|$	*r*	*B* (%)	*#inst*	*Impr* (%)	*#opt*	*Gap* (%)	*Time* (s)	*Impr* (%)	*#opt*	*Gap* (%)	*Time* (s)
	3	61.412	3	39.130	1	1.337	<0.001	39.130	1	1.337	<0.001
16/48/10/10	4	64.594	3	35.879	2	0.163	<0.001	35.879	2	0.163	<0.001
	5	66.970	3	40.853	1	4.602	<0.001	41.817	1	2.602	<0.001
	3	35.771	3	31.203	1	0.361	0.001	31.203	1	0.361	<0.001
16/48/10/15	4	47.738	3	36.866	2	0.407	0.001	36.967	2	0.240	<0.001
	5	54.234	3	33.899	2	0.522	0.001	33.898	1	0.523	<0.001
	3	55.747	3	34.801	0	1.809	0.001	35.563	0	0.679	<0.001
16/48/10/20	4	67.711	3	42.339	0	0.855	0.001	42.674	2	0.268	<0.001
	5	56.379	3	37.353	1	0.967	0.001	37.580	1	0.599	<0.001
	3	60.812	3	35.491	1	1.063	0.015	35.204	0	1.415	0.003
64/224/39/10	4	62.878	3	35.728	1	0.756	0.012	35.800	1	0.646	0.002
	5	59.970	3	34.499	2	0.025	0.014	34.519	3	0.000	0.003
	3	52.036	3	32.194	1	0.174	0.016	32.119	0	0.274	0.003
64/224/39/15	4	57.748	3	31.678	0	1.300	0.018	31.879	0	1.063	0.003
	5	50.512	3	29.562	0	0.814	0.018	29.503	0	0.903	0.003
	3	58.926	3	35.401	0	0.790	0.025	35.648	1	0.394	0.003
64/224/39/20	4	49.513	3	29.747	0	0.989	0.026	29.866	0	0.801	0.004
	5	64.975	3	36.225	0	0.544	0.025	36.060	0	0.755	0.003
	3	61.253	3	34.645	0	1.083	0.217	34.783	1	0.816	0.038
256/960/154/10	4	66.550	3	36.862	0	0.650	0.210	36.699	0	0.952	0.037
	5	44.130	3	25.260	1	0.311	0.216	25.179	1	0.425	0.038
	3	47.075	3	28.363	0	0.531	0.309	28.217	0	0.712	0.039
256/960/154/15	4	67.793	3	38.380	1	0.583	0.302	38.342	1	0.614	0.039
	5	54.592	3	31.874	2	0.265	0.309	31.781	1	0.412	0.040
	3	56.792	3	32.954	1	0.505	0.393	32.851	0	0.624	0.040
256/960/154/20	4	72.004	3	40.395	1	0.383	0.394	40.491	1	0.222	0.041
	5	51.162	3	30.272	1	0.331	0.393	30.222	1	0.405	0.041
	3	37.604	3	22.755	2	0.005	3.657	22.755	2	0.005	0.632
1024/3968/615/10	4	58.516	3	33.116	0	0.503	3.518	33.448	1	0.005	0.614
	5	51.119	3	29.686	3	0.000	3.472	29.555	1	0.164	0.606
	3	53.708	3	30.979	2	0.152	5.075	30.899	0	0.269	0.628
1024/3968/615/15	4	61.863	3	35.275	1	0.284	5.014	35.405	2	0.067	0.622
	5	37.228	3	21.863	0	0.231	4.921	21.871	0	0.224	0.613
	3	46.919	3	26.767	0	11.665	6.566	26.846	0	11.543	0.636
1024/3968/615/20	4	56.687	3	32.247	0	6.115	6.598	32.407	0	5.855	0.646
	5	55.902	3	31.545	0	6.878	6.300	31.592	0	6.818	0.622
	3	56.998	3	32.857	1	0.148	9.249	32.866	1	0.138	1.587
1600/6240/960/10	4	45.316	3	26.495	1	0.348	8.915	26.466	0	0.382	1.540
	5	60.913	3	34.319	3	0.000	8.908	34.241	2	0.135	1.553
	3	39.760	3	23.706	0	0.387	12.586	23.632	0	0.485	1.551
1600/6240/960/15	4	50.602	3	30.040	2	0.076	12.707	29.917	1	0.246	1.578
	5	60.779	3	34.678	1	0.157	12.440	34.739	1	0.051	1.549
	3	56.776	3	31.771	0	10.515	16.584	31.740	0	10.566	1.614
1600/6240/960/20	4	55.867	3	31.528	0	11.416	16.322	31.573	0	11.343	1.589
	5	50.639	3	28.687	0	14.867	16.236	28.737	0	14.785	1.577

the same as those used in Chapter 3. Moreover, the weights $p^\ell = (p_1^\ell, \dots, p_h^\ell, \dots, p_r^\ell)$, together with the cyclist fractions α_ℓ for $\ell \in \{1, \dots, q\}$, correspond to those estimated in the previous chapter, which is a study on route choice criteria in Parma. Table 4.11 presents these weights, as well as the intervention types considered for the Parma instances.

Table 4.11: Cyclist weights and intervention types for the Parma instances.

Cyclist weights p^ℓ					**Intervention types**
ℓ	p_1^ℓ	p_2^ℓ	p_3^ℓ	α_ℓ	Add a bike crossing (ABC)
1	0.97	0.03	0.00	13.2%	Add protective elements (APE)
2	0.00	1.00	0.00	15.3%	Build a bike lane (BBL)
3	0.00	0.00	1.00	20.6%	Build a bike path (BBP)
4	0.00	0.69	0.31	11.2%	Improvement pavement maintenance (IPM)
5	0.47	0.03	0.50	5.6%	Install markings (IM)
6	0.25	0.75	0.00	12.3%	Install signage (IS)
7	0.16	0.09	0.75	8.0%	Modify lane dimensions and remove bottlenecks (MDB)
8	0.72	0.00	0.28	7.3%	Modify pavement material (MPM)
9	0.53	0.47	0.00	6.7%	Remove protrusions (RP)

Let J denote the set of intervention types listed in Table 4.11. Each intervention type $\iota \in J$ has an associated list L_ι containing all arcs in the Parma network that are eligible to receive an intervention of type ι. Since these candidate lists L_ι are very large, with some containing thousands of arcs, the sets K and J cannot be equivalent. In practice, $|A_k|$ for each intervention $k \in K$ is much smaller than $|L_\iota|$, where ι denotes the type associated with intervention k. For this reason, for each intervention type ι, we selected multiple arc subsets from L_ι. Each selected subset becomes an arc set $A_k \subseteq L_\iota$, and each A_k defines an intervention $k \in K$ of type ι. Consequently, K contains all such interventions generated across all intervention types. The full procedure for constructing the intervention set K and the corresponding arc sets A_k, as well as the computation of the cost reductions ϕ_{ijk}^h for $(i, j) \in A$, $k \in K$, and $h \in \{1, \dots, r\}$, is described in Appendix F. Regarding the building costs $\tau_{ij}^k, k \in K, (i, j) \in A_k$, they were estimated based on the regional price list of Emilia-Romagna, Italy [98], where the city of Parma is located.

To complete the generation of the real-data instances, we selected the center of Parma, the area most frequently traversed by cyclists according to census data [55], and divided it into five sections: (C)enter (Figure 4.8b), (N)orth, (S)outh, (W)est, and (E)ast. For each section, except for the center, we extended the border beyond the city center zone, resulting in the northern (Figure 4.8c), southern (Figure 4.8d), western (Figure 4.8e), and eastern sections (Figure 4.8f).

Each section, as well as combinations of these sections, was considered as a distinct graph. The motivation for also analyzing isolated portions of the city is that infrastructure planners often work by evaluating a specific zone and preparing a plan for it, rather than allocating a large budget to the entire city. Moreover, analyzing smaller sections reduces computational complexity and allows us to explore a broader range of scenarios instead of considering only full-city cases.

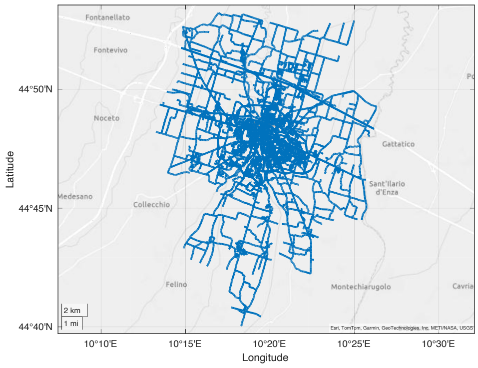
Note that the number of OD pairs and possible interventions was adapted to each region. The criterion for selecting combinations was that the number of possible interventions affecting the corresponding region should be greater than or equal to 10. Hence, graphs of different sizes were created, ranging from 2247 nodes and 4786 arcs to the full city graph, as described in Table 4.12.

For each city area in Table 4.12, we created three distinct instances, considering budgets B representing 30%, 55%, and 80% of the total building costs (4.6). Consequently, we obtained a final set of 36 real-world instances representing the city of Parma. The results of the computational experiments with CRM-BB (with BBS as search strategy), CRM-HKDP, and CRM-AM are reported in Table 4.13, in which each row corresponds to a single instance. Since the costs for distance, safety, and practicability were normalized in these instances, the gaps are calculated as

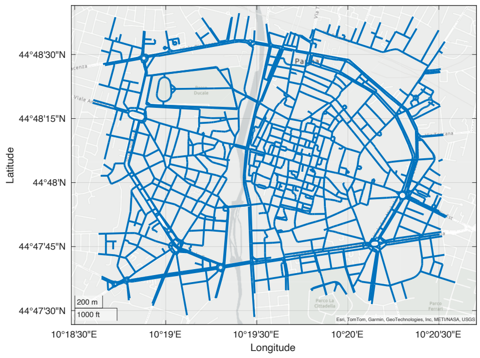
$$Gap_n(\%) = \frac{ub - lb}{F(\{\}, x_{\{\}}) - lb} \times 100 \quad \text{for CRM-BB, and}$$

$$Gap_n(\%) = \frac{F(K_m^*, x_m^*) - F(K_{best}, x_{best})}{F(\{\}, x_{\{\}}) - F(K_{best}, x_{best})} \times 100 \quad \text{for CRM-HKDP and CRM-AM.}$$

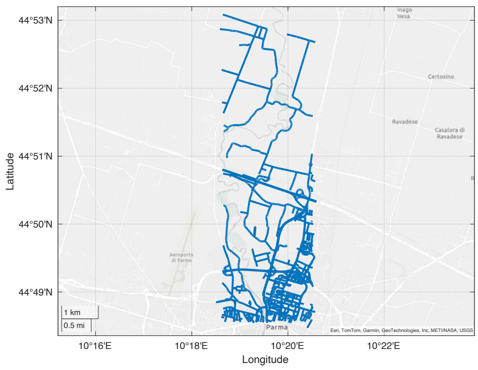
As shown in Table 4.13, CRM-BB found the optimal solutions for 13 out of 36 instances of Parma, although it generally required relatively extensive computational



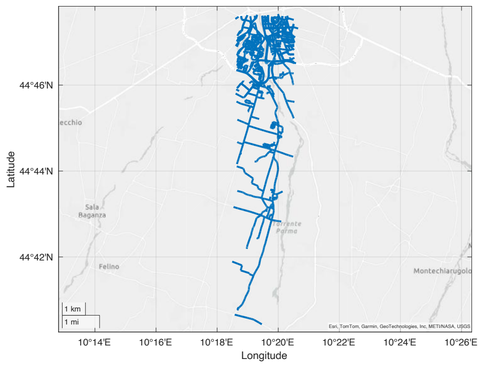
(a) Complete city.



(b) Central part of the city center.



(c) City center (north) + northern zone.



(d) City center (south) + southern zone.

(e) City center (west) + western zone.

(f) City center (east) + eastern zone.

Figure 4.8: Graphs corresponding to the city of Parma and its sections.

Table 4.12: Parma instances.

Instance		City sections
City area	$\|V\|/\|A\|/\|W\|/\|K\|$	
C(E)	2247/4786/72/14	Figure 4.8f
C(C)	3650/8542/331/28	Figure 4.8b
C(E+W)	4190/9072/167/20	Figures 4.8f and 4.8e
C(W+C)	5565/12800/647/30	Figures 4.8e and 4.8b
C(E+C)	5849/13276/601/40	Figures 4.8f and 4.8b
C(S)	5974/13442/215/17	Figure 4.8d
C(C+E+W)	7764/17534/958/42	Figures 4.8b, 4.8f, and 4.8e
C(W+S)	7917/17728/310/23	Figures 4.8e and 4.8d
C(E+S)	8221/18228/287/31	Figures 4.8f and 4.8d
C(S+C)	9567/21924/736/45	Figures 4.8d and 4.8b
City center	19359/43564/1844/59	Figures 4.8b, 4.8c, 4.8d, 4.8e, and 4.8f
Complete city	44820/98578/3806/59	Figure 4.8a

time in some cases. For the instances in which the optimal solution was not found within the time limit, the largest normalized gaps occurred for those with *B (%)* equal to 30%. When *B (%)* was 55%, most normalized gaps were below 5%, whereas for *B (%)* equal to 80%, the maximum normalized gaps were all below 0.001%. This behavior can be explained by the fact that smaller budgets allow only a few interventions, and thus feasible nodes in the BB tree tend to be located at the deeper levels, requiring more time to be reached.

Regarding the results attained by the heuristic procedures, both CRM-HKDP and CRM-AM were capable of finding the known optimal solutions for most Parma instances within highly efficient computational times. As before, when no feasible solution provided by CRM-BB is available, the normalized gaps for the heuristics are calculated using the global lower bound instead of the cost of the best feasible solution. This occurs in cases where, for CRM-BB gaps greater than 0.001%, at least one of the heuristics has the same gap as the exact method.

Although both methods are efficient in terms of time, CRM-AM demonstrates

Table 4.13: Summary of the results obtained for the instances of Parma.

Instances			CRM-BB		CRM-HKDP		CRM-AM	
City area	$\|V\|/\|A\|/\|W\|/\|K\|$	B (%)	$Gap_n(\%)$	$Time$ (s)	$Gap_n(\%)$	$Time$ (s)	$Gap_n(\%)$	$Time$ (s)
C (E)	2247/4786/72/14	30.000	0.000	231.496	0.000	1.582	0.000	0.209
		55.000	0.000	91.007	0.000	1.602	0.000	0.211
		80.000	0.000	90.959	0.000	1.581	0.000	0.213
C (C)	3650/8542/331/28	30.000	18.179	57600.000	18.179	15.621	18.179	1.110
		55.000	0.205	57600.000	0.205	15.652	0.205	1.125
		80.000	0.000	2654.111	0.000	15.623	0.000	1.148
C (E+W)	4190/9072/167/20	30.000	0.000	4005.257	0.000	4.872	0.000	0.469
		55.000	0.000	2006.588	0.000	4.896	0.000	0.475
		80.000	0.000	2003.039	0.000	4.873	0.000	0.479
C (W+C)	5565/12800/647/30	30.000	32.026	57600.000	32.026	52.277	32.026	3.430
		55.000	6.734	57600.000	6.734	52.569	6.734	3.436
		80.000	0.000	2309.710	0.000	52.301	0.000	3.458
C (E+C)	5849/13276/601/40	30.000	4.004	57600.000	4.004	74.245	4.004	3.726
		55.000	0.248	57600.000	0.248	74.043	0.248	3.759
		80.000	<0.001	57600.000	0.054	74.253	0.123	3.794
C (S)	5974/13442/215/17	30.000	0.000	12131.988	0.000	10.274	0.000	1.131
		55.000	0.000	3026.803	0.000	10.297	0.000	1.123
		80.000	0.000	1769.648	0.000	10.291	0.000	1.122
C (C+E+W)	7764/17534/958/42	30.000	20.573	57600.000	20.573	149.982	20.573	7.201
		55.000	3.010	57600.000	3.010	148.429	3.010	7.168
		80.000	<0.001	57600.000	0.000	149.234	0.000	7.219
C (W+S)	7917/17728/310/23	30.000	3.533	57600.000	3.533	17.903	3.533	1.492
		55.000	0.000	29858.791	0.000	17.884	0.000	1.501
		80.000	0.000	29853.677	0.000	17.930	0.000	1.516
C (E+S)	8221/18228/287/31	30.000	5.131	57600.000	5.131	23.377	5.131	1.501
		55.000	<0.001	57600.000	0.000	23.289	0.000	1.517
		80.000	<0.001	57600.000	0.000	23.210	0.000	1.526
C (S+C)	9567/21924/736/45	30.000	13.962	57600.000	13.962	145.447	13.962	6.604
		55.000	1.846	57600.000	1.846	145.483	1.846	9.893
		80.000	<0.001	57600.000	0.419	145.844	0.836	6.572
City center	19359/43564/1844/59	30.000	15.744	57600.000	15.744	921.538	17.944	47.230
		55.000	1.754	57600.000	1.754	917.529	1.754	31.831
		80.000	<0.001	57600.000	0.000	924.534	0.000	31.461
Complete city	44820/98578/3806/59	30.000	14.670	57600.000	14.670	4848.273	15.132	251.279
		55.000	1.523	57600.000	1.523	4850.964	1.523	164.579
		80.000	<0.001	57600.000	0.000	4868.558	0.079	166.515

even greater efficiency, particularly for the instances representing the complete city. In these cases, CRM-AM reaches cost convergence in a number of iterations much smaller than the number of possible interventions and is therefore more effective for very large instances than CRM-HKDP, even though the latter achieves slightly better normalized gaps in some cases, as shown in Table 4.13.

To complete the analysis of the results obtained for the Parma instances, Figure 4.9 shows the number of occurrences of each intervention, considering the experiments on all instances using the CRM-AM method. This figure indicates that, among all intervention types, only *adding bike crossings* (ABC), *building bike paths* (BBP), *installing markings* (IM) and *signage* (IS), and *removing protrusions* (RP) had all of their associated individual interventions applied (see Appendix F for more details on the generation of the Parma instances, including the number of interventions associated with each type).

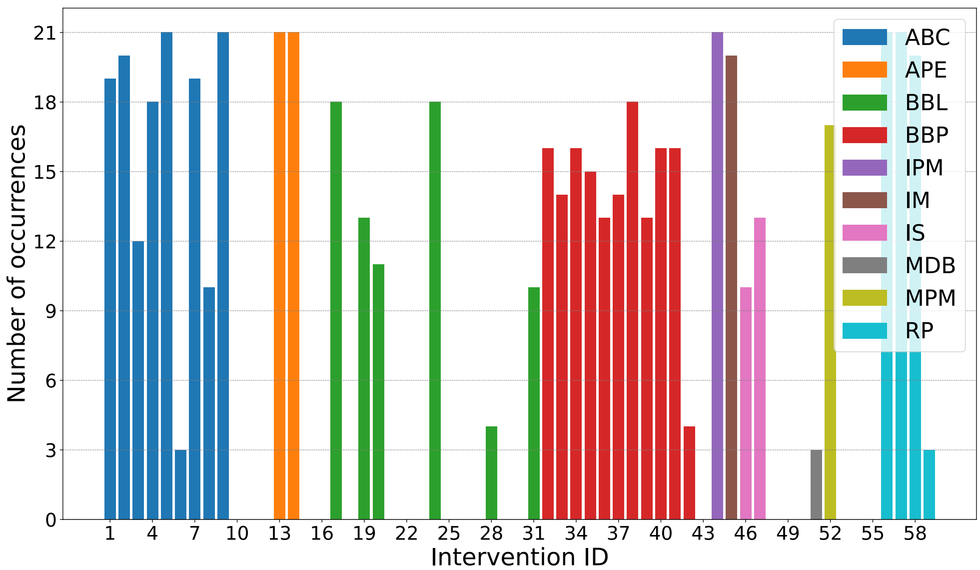


Figure 4.9: Frequency of application for each intervention.

An important fact to highlight is that, despite being more expensive, constructing bike paths is preferred over *building bike lanes* (BBL), demonstrating the importance that cyclists in Parma assign to the safety criterion, as bike paths are safer than bike lanes. Moreover, although not all interventions of the type *adding protective elements* (APE) were applied, the frequency of this intervention type at the locations covered by interventions 13 and 14 is relatively high, indicating its relevance depending on the scenario. Lastly, Figure 4.9 also shows that simple actions such as installing markings and removing protrusions can make a significant difference in minimizing cyclists' costs in Parma.

Figure 4.10 illustrates the frequency distribution of the intervention types across different portions of Parma. It shows that the most implemented intervention types, such as ABC and BBP, affect multiple portions of the city, with interventions selected in most instances.

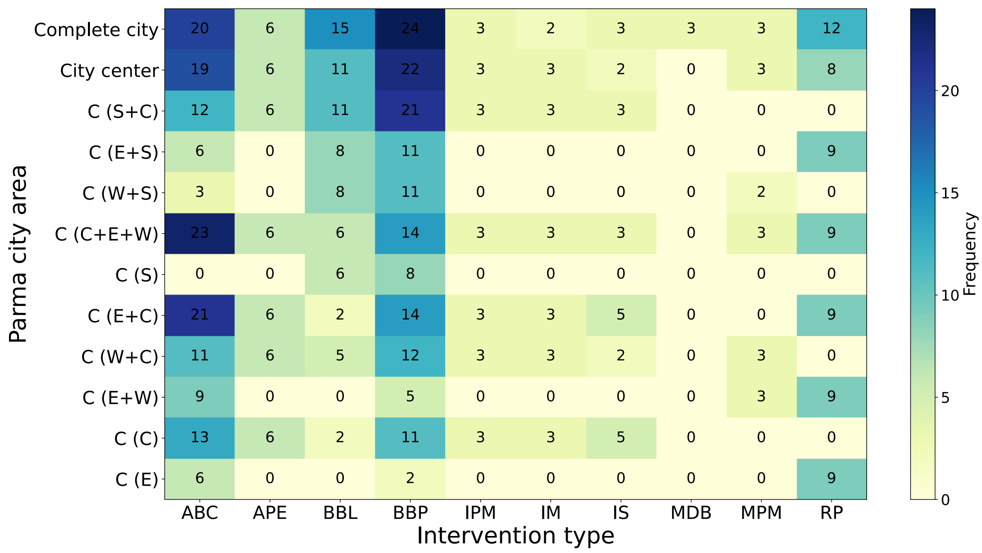


Figure 4.10: Application frequency of intervention types in Parma city areas.

Figure 4.10 also highlights that planning investments for combined areas of the city is not simply the sum of planning investments for those same areas considered separately. This is illustrated, among others, by the cases of center-south, C(S), and center-center, C(C), compared with the case in which these two regions are combined, C(S+C). For instance, when considering the combined area, it is preferable to decrease investments in ABC and IS interventions and invest more in BBL and BBP interventions. This reinforces that allocating a larger budget to combined areas can lead to different outcomes than implementing interventions with smaller budgets in isolated areas, whether simultaneously or at different times.

Finally, interventions of the types *improving pavement maintenance* (IPM) and *modifying pavement material* (MPM), although not relevant for the entire city, are frequently applied in the central and western portions, respectively, whereas interventions of the type *modifying lane dimensions and removing bottlenecks* (MDB) are included only in the investment plans for the entire city.

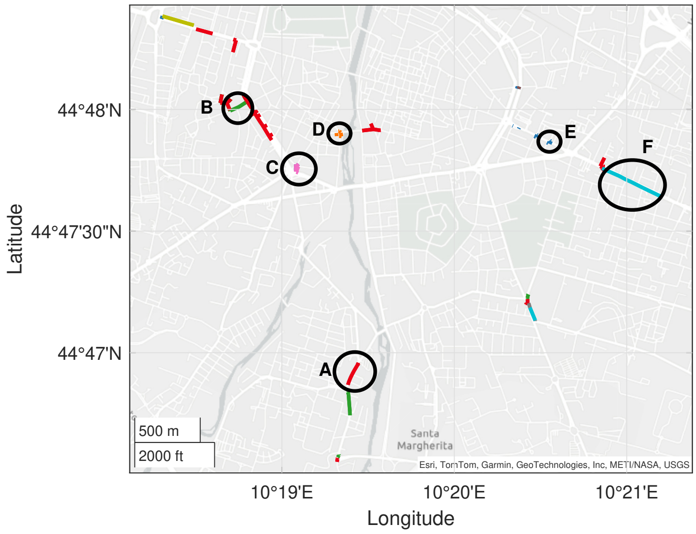


Figure 4.11: Identified interventions in Parma city. The circled interventions are shown in detail below.

The map in Figure 4.11 shows all the interventions identified in the city of Parma using the CRM-AM method (similar solutions were also obtained with the CRM-HKDP and CRM-BB methods), considering a budget equal to 80% of the total building costs. Six of these interventions are detailed in what follows.

Intervention A concerns the construction of a bike path along *Strada Langhirano*.

(a) BBP in Strada Langhirano.

(b) BBL in Via Rasori.

(c) IS in Piazzale Barbieri.

(d) APE in Piazzale Rondani.

(e) ABC in Via Emilia Est.

(f) RP in Via Emilia Est.

Figure 4.12: Selected locations in Parma where the proposed interventions will be implemented. Source: Google Street View (Google LLC).

As shown in Figure 4.12a, a bike lane already exists on this street, but it is particularly unsafe for cyclists. Strada Langhirano is a highly trafficked road, as it connects Langhirano, one of the largest municipalities in the province, with the city center of Parma. Moreover, this street is heavily used by cyclists, especially students commuting to the university campus. This intervention is therefore fully aligned with the city's mobility needs.

Intervention B involves the construction of a bike lane on *Via Rasori*, which provides secondary access to Parma's hospital (see Figure 4.12b). Although it is not the main entrance, this street is used by a significant number of cars and bicycles. The introduction of a bike lane would improve safety along this segment, where cyclists are currently forced to share the roadway with motor vehicles.

Intervention C focuses on the installation of signage in *Piazzale Barbieri* (see Figure 4.12c). This square is one of the most unsafe areas in Parma, with numerous accidents involving cyclists reported in the past. Improved signage would help regulate the interaction between cars, buses, and bicycles, enhancing overall traffic safety.

Intervention D consists of adding protective elements in *Piazzale Rondani* (see Figure 4.12d). This area is frequently used by students cycling to nearby high schools. As there is currently no dedicated cycling infrastructure, the addition of protective elements would significantly improve cyclist safety.

Intervention E concerns the introduction of cycle crossings along *Via Emilia Est*. This is one of the city's main arterial roads, running east–west across Parma. While a bike path exists on both sides of the street, there are currently no cycle crossings connecting them, limiting safe bicycle movement across the road (see Figure 4.12e).

Finally, as shown in Figure 4.12f, *Via Emilia Est* includes several segments where various obstacles are present along the bike path, making it difficult for cyclists to pass. *Intervention F* therefore proposes the removal of these protrusions in order to improve the practicability of the bike path. Given the six detailed interventions, this real-world application demonstrates that the proposed methods can serve as highly effective tools for decision support.

Chapter 5

Bicycle and motorized vehicles integration in urban networks

This chapter introduces an extension of the problem described in Chapter 4, which integrates bicycles and motorized vehicles into road networks. Section 5.1 defines the proposed extension, while Section 5.2 presents its mathematical formulation. It is worth noting that the research presented in this chapter is still ongoing. Therefore, a potential solution approach (see Section 5.3) is currently being implemented to address the problem, which can be viewed as a direct extension of the CRM-AM heuristic (see Section 4.3.4), adapted to solve the extended problem.

5.1 Problem definition

Consider the directed graph $G = (V, A)$, which represents the road network, including all definitions of cost vector $c_{0_{ij}}$; the number of road characteristics h; cyclist weight vectors $p^\ell, \ell \in \{1, \ldots, q\}$; the set of interventions K (including the subsets of arcs A_k, cost reductions ϕ^h_{ijk}, and building costs τ^k_{ij} associated with each intervention $k \in K$); and the budget B, all of which are presented in Chapter 4. Also consider the set of OD pairs W with cyclist demands u_w, as defined in that chapter.

We introduce a new element into the road network: the users of motorized vehicles. They are the main users of the network, which is generally designed to better accommodate this type of vehicle. Considering this, the proposed interventions applied to the bicycle network may negatively affect these users. Therefore, the selection of interventions to be implemented must also take these potential negative effects into account.

Given the above, let $\hat{u}_w$ denote the demand of motorized vehicle users associated with the OD pair $w = (o_w, d_w) \in W$. We assume that the travel time t_{ij} for motorized vehicles to traverse arc $(i, j) \in A$ is given by the Bureau of Public Roads (BPR) function [109], defined as

$$t_{ij} = t_{ij}^0 \left(1 + \gamma \left(\frac{f_{ij}}{Q_{ij}} \right)^{\varepsilon} \right),$$

where t_{ij}^0, f_{ij}, and Q_{ij} represent the uncongested travel time, vehicle volume (or flow), and vehicle capacity on arc $(i, j) \in A$, respectively. In addition, γ and ε are also parameters of the BPR function. According to Gore et al. [41], BPR functions are widely used in transportation problems due to their simplicity and good performance, and they have also been applied to bicycle network design problems [7].

An intervention $k \in K$, when applied to the bicycle network, can affect the travel time of motorized vehicle users. For instance, dedicating one of the road lanes exclusively to cyclists reduces the vehicles capacity of that road and, consequently, can increase the time required to traverse it, depending on the traffic conditions. Hence, let γ_k, ε_k, and θ_{ij}^k denote the effects of intervention k on arc $(i, j) \in A$ with respect to the parameters γ, ε, and Q_{ij}, respectively. Furthermore, let γ_0, ε_0, and Q_{ij}^0 be the values of these parameters for the scenario in which no intervention is applied (also referred to as the initial values).

Therefore, given the budget B, the objective of the problem is to select the best combination of interventions that minimizes the overall costs perceived by cyclists, as in the cost reduction problem (see Chapter 4), but considering that the total travel time of motorized vehicle users cannot exceed a certain limit. Specifically, let $A_0^w \subset A$ be a subset of arcs that compose the shortest path for motorized vehicle users traveling from o_w to d_w in the scenario where no intervention is applied, and let T_0^w denote the total travel time to traverse this shortest path, given by:

$$T_0^w = \sum_{(i,j)\in A_0^w} t_{ij}^0 \left(1+\gamma_0\left(\frac{f_{ij}}{Q_{ij}^0}\right)^{\varepsilon_0}\right).$$

Hence, the problem aims to select the best combination of interventions, given the available budget B, as well as to compute the cyclist and motorized vehicle user flows, such that the demands of all OD pairs for both modes of transport. Moreover, the solution of the problem must ensure that the total travel time for motorized vehicle users traveling from o_w to d_w does not exceed $\mu_w T_0^w$, where $\mu_w \geq 1$ is a scale factor associated with OD pair $w \in W$, representing the tolerance for travel time increases. These scale factors are not fixed, and instead their maximum value, μ_{max}, is minimized as part of the objective of the problem. Table 5.1 summarizes all the parameters of the problem addressed in this chapter, highlighting those that were added in this problem extension.

5.2 Mathematical formulation

As defined in Chapter 4, let $x_{ij}^{w\ell}$ denote the flow on arc $(i,j)\in A$ of cyclists of profile ℓ traveling from o_w to d_w, with $w \in W$; $y_k, k \in K$, the binary variables indicating whether intervention k is applied; and c_{ij}^h the continuous variables representing the cost of arc (i,j) with respect to road characteristic $h \in \{1,\dots,r\}$ after the implementation of the interventions.

Specifically for the problem addressed in this chapter, let $\hat{x}_{ij}$ and $\hat{t}_{ij}$ denote continuous variables representing the flow and travel time, respectively, of motorized vehicle users on arc $(i,j)\in A$. In addition, let $\hat{Q}_{ij}$ denote continuous variables representing the vehicle capacity of arc (i,j), and let $\hat{\gamma}$ and $\hat{\varepsilon}$ be also continuous variables associated with the parameters γ and ε of the BPR function, respectively. Lastly let μ_w be continuous variables indicating the tolerance for travel time increases associated with OD pair $w \in W$, with μ_{max} representing their maximum value, which is minimized in the objective function.

To complete the set of variables for the problem, we define $\hat{\chi}_{ij}^w$ as binary variables

Table 5.1: Summary of data and parameters for the problem of integrating bicycles and motorized vehicles.

Road network	
V	Set of vertices;
A	Set of arcs;
W	Set of OD pairs;
$o_w(d_w)$	Origin (destination) vertex associated with OD pair $w \in W$;
u_w	Demand of cyclists associated with OD pair $w \in W$;
$\hat{u}_w$	Demand of motorized vehicle users associated with OD pair $w \in W$;
Interventions	
K	Set of possible interventions;
A_k	Subset of arcs affected by intervention $k \in K$;
τ_{ij}^k	Cost of applying intervention $k \in K$ on arc $(i,j) \in A_k$ (building cost);
r	Number of road characteristics;
$c_{0_{ij}}^h$	Cost perceived by the cyclist related to road characteristic $h \in \{1,\ldots,r\}$ on arc $(i,j) \in A$;
ϕ_{ijk}^h	Cost reduction on arc $(i,j) \in A$ associated with road characteristic $h \in \{1,\ldots,r\}$, due to intervention $k \in K$;
B	Budget available;
Cyclist profiles	
q	Number of cyclist profiles;
p_h^ℓ	Weight assigned by a cyclist of profile $\ell \in \{1,\ldots,q\}$ to the road characteristic $h \in \{1,\ldots,r\}$;
α_ℓ	Fraction of the total cyclist population that belongs to profile $\ell \in \{1,\ldots,q\}$;
Travel time function for motorized vehicle users	
Q_{ij}^0	Initial capacity of arc $(i,j) \in A$ in terms of the number of vehicles;
γ_0, ε_0	Initial values of the parameters γ and ε, respectively;
T_0^w	Initial value of the travel time function for the motorized vehicle users (considering γ_0 and ε_0);
θ_{ij}^k	Reduction in the capacity of arc $(i,j) \in A$ due to intervention $k \in K$;
γ_k, ε_k	Contribution to the parameters γ and ε of the travel time function for motorized vehicle users, respectively, associated with intervention $k \in K$.

indicating whether arc (i, j) is part of the shortest path associated with OD pair w. Table 5.2 summarizes all the variables of the problem, highlighting those that were included in this extension.

Table 5.2: Summary of variables for the problem of integrating bicycles and motorized vehicles.

Continuous variables	
$x_{ij}^{w\ell}$	Cyclists' flow on arc $(i, j) \in A$, associated with OD pair $w \in W$ and cyclist profile $\ell \in \{1, \ldots, q\}$;
c_{ij}^{h}	Cost on arc $(i, j) \in A$ associated with road characteristic $h \in \{1, \ldots, r\}$ after the application of the interventions;
$\hat{x}_{ij}$	Motorized vehicle flow on arc $(i, j) \in A$;
$\hat{Q}_{ij}$	Capacity of arc $(i, j) \in A$ in terms of the number of vehicles;
$\hat{t}_{ij}$	Travel time on arc $(i, j) \in A$ for motorized vehicles;
$\hat{\gamma}$, $\hat{\varepsilon}$	Parameters of the travel time function for motorized vehicles that vary according to the applied interventions;
μ_w	Scale factor associated with OD pair $w \in W$ representing the allowed increase in users' motorized travel time relative to T_0^w;
μ_{max}	The maximum value among the variables $\mu_w, w \in W$, which is minimized.
Binary variables	
y_k	Indicates whether intervention $k \in K$ is applied;
$\hat{\chi}_{ij}^{w}$	Indicates whether arc $(i, j) \in A$ is part of the shortest path associated with OD pair $w \in W$.

We can express the problem as a bilevel problem, where the upper level focuses on optimizing network interventions for cyclists while considering the travel time increases for motorized vehicle users, and the lower level analyzes the impact of these interventions on motorized vehicle flows. The proposed mathematical formulation of this bilevel problem is presented as follows.

$$\min_{y_k, x_{ij}^{w\ell}, \mu_w, \mu_{max}} \psi_b \sum_{w \in W} \sum_{(i,j) \in A} \sum_{\ell=1}^{q} \sum_{h=1}^{r} c_{ij}^h p_h^\ell x_{ij}^{w\ell} + \psi_{mv} \mu_{max} \tag{5.1a}$$

s.t.

$$\sum_{\substack{j \in V: \\ (i,j) \in A}} x_{ij}^{w\ell} - \sum_{\substack{j \in V: \\ (j,i) \in A}} x_{ji}^{w\ell} = \alpha_\ell U_i^w \qquad i \in V, \ell \in \{1, \ldots, q\}, w \in W \tag{5.1b}$$

$$\sum_{k \in K} \sum_{(i,j) \in A_k} \tau_{ij}^k y_k \leq B \tag{5.1c}$$

$$c_{ij}^h = c_{0_{ij}}^h - \sum_{k \in K} \phi_{ijk}^h y_k \qquad (i,j) \in A, h \in \{1, ..., r\} \tag{5.1d}$$

$$\sum_{(i,j) \in A} \hat{t}_{ij} \hat{\chi}_{ij}^w \leq \mu_w T_0^w \qquad w \in W \tag{5.1e}$$

$$y_k \in \{0, 1\} \qquad k \in K \tag{5.1f}$$

$$c_{ij}^h \geq 0 \qquad (i,j) \in A, h \in \{1, ..., r\} \tag{5.1g}$$

$$x_{ij}^{w\ell} \geq 0 \qquad (i,j) \in A, \ell \in \{1, \ldots, q\}, w \in W \tag{5.1h}$$

$$1 \leq \mu_w \leq \mu_{max} \qquad w \in W \tag{5.1i}$$

$$\mu_{max} \geq 1 \tag{5.1j}$$

$$(\hat{t}_{ij}, \hat{\chi}_{ij}^w) \in S(\{y_k\}_{k \in K}) \tag{5.1k}$$

$$S(\{y_k\}_{k \in K}) = \arg \min_{\hat{t}_{ij}, \hat{\chi}_{ij}^w} \sum_{(i,j) \in A} \int_0^{\hat{x}_{ij}} t(\hat{\gamma}, \hat{\varepsilon}, \hat{Q}_{ij}, \xi) \, d\xi \tag{5.1l}$$

s.t.

$$\hat{t}_{ij} = t(\hat{\gamma}, \hat{\varepsilon}, \hat{Q}_{ij}, \hat{x}_{ij}) = t^0_{ij}\left(1+\hat{\gamma}\left(\frac{\hat{x}_{ij}}{\hat{Q}_{ij}}\right)^{\hat{\varepsilon}}\right) \qquad (i,j) \in A \qquad (5.1m)$$

$$\hat{\gamma} = \gamma_0 + \sum_{k \in K} \gamma_k y_k \qquad (5.1n)$$

$$\hat{\varepsilon} = \varepsilon_0 + \sum_{k \in K} \varepsilon_k y_k \qquad (5.1o)$$

$$\hat{Q}_{ij} = Q^0_{ij} - \sum_{k \in K} \theta^k_{ij} y_k \qquad (i,j) \in A \qquad (5.1p)$$

$$\sum_{\substack{j \in V: \\ (o_w,j) \in A}} \hat{\chi}^w_{o_w j} - \sum_{\substack{j \in V: \\ (j,o_w) \in A}} \hat{\chi}^w_{j o_w} = 1 \qquad w \in W \qquad (5.1q)$$

$$\sum_{\substack{j \in V: \\ (d_w,j) \in A}} \hat{\chi}^w_{d_w j} - \sum_{\substack{j \in V: \\ (j,d_w) \in A}} \hat{\chi}^w_{j d_w} = -1 \qquad w \in W \qquad (5.1r)$$

$$\sum_{\substack{j \in V: \\ (i,j) \in A}} \hat{\chi}^w_{ij} - \sum_{\substack{j \in V: \\ (j,i) \in A}} \hat{\chi}^w_{ji} = 0 \quad i \in V \setminus \{o_w, d_w\}, w \in W \qquad (5.1s)$$

$$\hat{x}_{ij} = \sum_{w \in W} \hat{u}_w \hat{\chi}^w_{ij} \qquad (i,j) \in A \qquad (5.1t)$$

$$\hat{\chi}^w_{ij} \in \{0,1\} \qquad w \in W, (i,j) \in A \qquad (5.1u)$$

$$\hat{x}_{ij}, \hat{Q}_{ij}, \hat{t}_{ij} \geq 0 \qquad (i,j) \in A \qquad (5.1v)$$

$$\hat{\gamma}, \hat{\varepsilon} \geq 0 \qquad (5.1w)$$

For the upper level, objective function (5.1a) minimizes the overall costs perceived by cyclists and the maximum travel time increase for motorized vehicle users, where ψ_b and ψ_{mv} are the weights associated with these two objectives, respectively. Constraints (5.1b)–(5.1d) are the same as those in the mathematical formulation of the cost reduction problem (see Section 4.2). Constraints (5.1e) impose that the total travel time along the shortest path for OD pair $w \in W$ cannot exceed $\mu_w T^w_0$. Constraints (5.1f)–(5.1h) define the domain of the decision variables also present in the cost-reduction formulation. Constraints (5.1i) require that μ_w, for each OD pair w, be greater than 1 and smaller than μ_{max}; while constraint (5.1j) imposes that $\mu_{max} \geq 1$. Finally, Constraint (5.1k) enforces that the shortest paths taken by motorized vehicle users and their corresponding travel times are optimal with respect to the lower level problem, parameterized by the interventions decision vector $\{y_k\}_{k \in K}$.

For the lower level, objective function (5.1l) minimizes the total travel time of motorized vehicle users. Constraints (5.1m) compute the travel time on each arc $(i, j) \in A$ using the BPR function, while constraints (5.1n)–(5.1p) define the parameters of the travel time function, $\hat{\gamma}$, $\hat{\varepsilon}$, and $\hat{Q_{ij}}$, respectively, according to the decisions made in the upper level problem regarding the selected interventions. Constraints (5.1q)–(5.1s) determine the shortest path associated with each OD pair $w \in W$, and constraints (5.1t) impose that the flows on the arcs composing these shortest paths must be equal to the demand u_w of the corresponding OD pair w. Finally, constraints (5.1u)–(5.1w) define the domain of the decision variables in the lower level problem.

5.3 Solution approach

The method being implemented to address the problem of integrating cyclists and motorized vehicle users in road networks is an extension of the CRM-AM method, which was originally proposed for the cost reduction problem (see Chapter 4). This extension, named *bicycles and motorized vehicles cost reduction model – alternating method* (BMVCRM-AM), starts by solving formulation (4.7a)–(4.7c), which is derived from (5.1a)–(5.1w) for a fixed set of interventions and a fixed solution of the lower level problem, the latter implying fixed values of μ_w, for all $w \in W$, and of μ_{max}. Specifically, in the first iteration of the method, the scenario with no interventions ($y_k = 0$, $k \in K$) is considered. This implies that motorized vehicle users follow paths whose arcs belong to the subsets $A_0^w, w \in W$ and, consequently, all μ_w variables and μ_{max} are equal to 1.

Fixing the cyclist flows obtained after solving formulation (4.7a)–(4.7c), and analyzing the intervention decisions with respect to cyclists only, formulation (4.9a)–(4.9c) is derived from (5.1a)–(5.1w) and solved using SDPKP, as in CRM-AM. The resulting set of interventions, obtained from solving formulation (4.9a)–(4.9c), leads to a new solution to the lower level problem and, consequently, to new values for the upper level variables μ_w and μ_{max}. Therefore, the lower level formulation (5.1l)–

(5.1w), when evaluated for the intervention decisions determined at the upper level, becomes equivalent to the following formulation:

$$\min \sum_{(i,j)\in A} \int_0^{\hat{x}_{ij}} t(\xi)\, d\xi \tag{5.2a}$$

s.t.

$$\hat{t}_{ij} = t(\hat{x}_{ij}) = t^0_{ij}\left(1+\hat{\gamma}\left(\frac{\hat{x}_{ij}}{\hat{Q}_{ij}}\right)^{\hat{\varepsilon}}\right) \qquad (i,j)\in A \tag{5.2b}$$

$$\sum_{\substack{j\in V:\\(o_w,j)\in A}} \hat{\chi}^w_{o_w j} - \sum_{\substack{j\in V:\\(j,o_w)\in A}} \hat{\chi}^w_{j o_w} = 1 \qquad w\in W \tag{5.2c}$$

$$\sum_{\substack{j\in V:\\(d_w,j)\in A}} \hat{\chi}^w_{d_w j} - \sum_{\substack{j\in V:\\(j,d_w)\in A}} \hat{\chi}^w_{j d_w} = -1 \qquad w\in W \tag{5.2d}$$

$$\sum_{\substack{j\in V:\\(i,j)\in A}} \hat{\chi}^w_{ij} - \sum_{\substack{j\in V:\\(j,i)\in A}} \hat{\chi}^w_{ji} = 0 \qquad i\in V\setminus\{o_w,d_w\}, w\in W \tag{5.2e}$$

$$\hat{x}_{ij} = \sum_{w\in W} \hat{u}_w \hat{\chi}^w_{ij} \qquad (i,j)\in A \tag{5.2f}$$

$$\hat{\chi}^w_{ij} \in \{0,1\} \qquad w\in W, (i,j)\in A \tag{5.2g}$$

$$\hat{x}_{ij} \geq 0 \qquad (i,j)\in A, \tag{5.2h}$$

which can be solved by applying Wardrop's equilibrium principle [117]. Once the new paths for motorized vehicle users, represented by the arc variables $\hat{\chi}^w_{ij}$, and their corresponding travel times $\hat{t}_{ij}$ have been obtained, the variables μ_w for all $w \in W$, and μ_{max} can be determined by solving the following linear programming formulation:

$$\min \ \mu_{max} \tag{5.3a}$$

s.t.

$$\sum_{(i,j)\in A} \hat{t}_{ij}\hat{\chi}^w_{ij} \leq \mu_w T^w_0 \qquad w\in W \tag{5.3b}$$

$$1 \leq \mu_w \leq \mu_{max} \qquad w\in W \tag{5.3c}$$

$$\mu_{max} \geq 1. \tag{5.3d}$$

After solving formulations (4.9a)–(4.9c), (5.2a)–(5.2h), and (5.3a)–(5.3d), the updated solution values for the variables y_k, μ_w, and μ_{max} are obtained. Consequently, the method is repeated iteratively until the solutions from the two phases (fixed intervention set and cyclist flows, respectively) converge to the same cost, the latter being expressed by the upper level objective function (5.1a). Figure 5.1 summarizes the steps of the proposed method, where $\mathscr{F}_1$ and $\mathscr{F}_2$ represents the solution costs of phases 1 and 2, respectively.

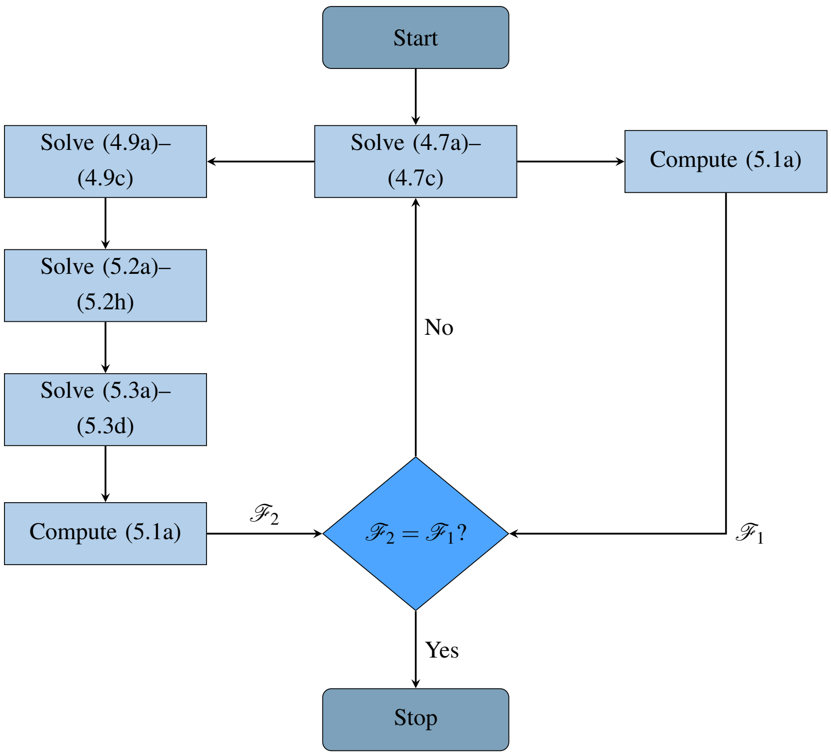


Figure 5.1: Problem of integrating bicycles and motorized vehicles in road networks solved using an alternating approach.

Chapter 6

Concluding remarks

This work presents optimization strategies for evaluating and designing bicycle networks in urban environments. Regarding network evaluation, we address the problem of identifying the criteria used by cyclists to choose their routes. We assume that cyclists travel along shortest paths (SP), where the costs of these paths are expressed as convex combinations of a few basic costs, which take into account different road features, including distance, safety, and practicability. For network design, we tackle the problem of selecting, given cyclists' preferences for a set of road features, the best combination of interventions to apply to the bicycle network in order to minimize the perceived costs for users and, consequently, improve its overall quality. The optimization is subject to budget and demand fulfillment constraints.

For the first problem, we propose two optimization models and corresponding solution algorithms, each based on a different type of data: aggregate user flows on some network arcs, and a set of paths followed by a sample of users. In the flow-based formulation, we first consider a simplified version of the problem in which the set of weights (i.e., the coefficients of the convex combinations defining the costs optimized by the users) is known in advance. This problem is solved through a polynomial-time algorithm based on the solution of multiple SP problems and a single convex quadratic programming problem. Next, an algorithm to identify an unknown set of weights is proposed. An advantage of this method is that it requires observations from only a subset of the network arcs, and such data can be collected, for example, via strategically placed cameras in the city. In the trajectory-based formulation, we define a k-median problem aimed at selecting, from all possible behaviors, up to k behaviors that are most representative of a given sample of user trajectories. A greedy algorithm is proposed to solve this problem. For both models we consider a deterministic formulation, in which the basic costs are fixed values, and a stochastic formulation, in which the basic costs are random variables following a normal distribution. The latter is motivated by the fact that different users may perceive road features differently.

For the second problem, we propose a mixed-integer programming formulation and four related solution approaches. Among these, two are exact methods, an enumeration mechanism and a branch-and-bound framework, and two are heuristic procedures based on solving knapsack problems with dynamic programming. These methods aim to improve the quality of cycling networks, while taking into account cyclists' preferences regarding road characteristics such as length, safety, and practicability. The developed approaches have proven to be useful tools for cycling infrastructure planning, as they allow planners to evaluate multiple types of interventions while considering their impact on different road characteristics. Furthermore, they are sufficiently general to be applied in different contexts, regardless of the set of interventions or road features considered. For example, the same model could be applied in a scenario where the objective is to minimize the costs perceived by car drivers in terms of travel distance, traffic intensity, and the number of unpaved roads traversed. In this case, possible interventions may include paving city roads and modifying the timing of traffic lights to reduce congestion.

Computational experiments for the two problems addressed in this research were conducted using both synthetic data and real-world data from the bicycle network of the city of Parma, Italy. Specifically, for the first problem, experiments for the flow-based formulation were performed on synthetic data, while the trajectory-based formulation was applied to the real-world case of the bicycle network in Parma. In both formulations, the results showed no significant differences between the solutions obtained from the deterministic and stochastic models, suggesting that the proposed algorithms are robust to perturbations in the basic costs. Regarding the second problem, the computational results showed that the proposed approaches are effective for tackling large instances, such as networks with more than 44,000 nodes and 98,000 arcs. While the heuristics, especially CRM-AM, were able to provide high-quality solutions in very efficient times, CRM-BB proved the optimality of most tested instances, including real-world cases with sizes of up to approximately 7,900 nodes and 17,700 arcs. These results were obtained within reasonable computational times, which demonstrates its applicability in practical scenarios. The experiments with the Parma instances also revealed how the choice of interventions is influenced by the specific regions of the city, offering valuable insights for planning large networks.

Despite the very promising results obtained, the proposed strategies for both problems present some limitations that could be addressed in future research. For the first problem, the assumption that users always choose the minimum-cost path is not necessarily realistic, as some users may simply take recreational tours and then return to their starting point. In such cases, the proposed trajectory-based formulation is unable to capture the decision-making criteria of these users. This suggests a potential direction for future work, in which the minimum-cost assumption could be relaxed or reconsidered. Furthermore, more diverse data could be incorporated into the experiments, including other types of cyclists beyond bike-sharing users, as well as a broader set of road features, since distance, safety, and practicability may not be the only factors influencing route choice. Another promising direction for future research on the first problem involves developing mechanisms to capture the individual preferences of cyclists belonging to the same user profile.

For the second problem, the proposed model could be extended to include decisions about which arcs compose the set A_k for at least some interventions $k \in K$, since the locations where each possible intervention is applied are not necessarily known in advance. Another possible extension involves the integration of budget minimization into the objective function. Furthermore, uncertainties in the demand for each OD pair could also be addressed in the formulation. Finally, the main (ongoing) research direction of this work is to analyze how the selected interventions for the cycling network of Parma affect motorized vehicle users.

Bibliography

[1] C. Aceves-González, K. Ekambaram, J. Rey-Galindo, and L. Rizo-Corona. The role of perceived pedestrian safety on designing safer built environments. *Traffic Injury Prevention*, 21(sup1):S84–S89, 2020.

[2] A. Agarwal, D. Ziemke, and K. Nagel. Bicycle superhighway: An environmentally sustainable policy for urban transport. *Transportation Research Part A: Policy and Practice*, 137:519–540, 2020.

[3] M. Akbarzadeh, S. S. Mohri, and E. Yazdian. Designing bike networks using the concept of network clusters. *Applied network science*, 3(1):1–21, 2018.

[4] S. Ardizzoni, M. Laurini, R. Praxedes, L. Consolini, and M. Locatelli. Identification of cyclists' route choice criteria. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pages 6975–6980, 2024.

[5] S. Ardizzoni, B. Caselli, L. Consolini, M. Laurini, M. Locatelli, R. Praxedes, S. Rossetti, and F. Stabile. Optimization Approaches to Identify Cyclists' Route Choice Criteria. *Available at SSRN 5113533*, 2025. Submitted to Omega.

[6] N. Aydin, S. Seker, and B. Özkan. Planning location of mobility hub for sustainable urban mobility. *Sustainable Cities and Society*, 81:103843, 2022.

[7] S. A. Bagloee, M. Sarvi, and M. Wallace. Bicycle lane priority: Promoting bicycle as a green mode even in congested urban area. *Transportation Research Part A: Policy and Practice*, 87:102–121, 2016.

[8] J. Bao, T. He, S. Ruan, Y. Li, and Y. Zheng. Planning bike lanes based on sharing-bikes' trajectories. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume Part F129685, pages 1377–1386. Association for Computing Machinery, 8 2017.

[9] M. Ben-Akiva and M. Bierlaire. *Discrete Choice Methods and their Applications to Short Term Travel Decisions*, pages 5–33. Springer US, Boston, MA, 1999.

[10] M. Ben-Akiva, M. Bergman, A. J. Daly, and R. Ramaswamy. Modelling inter urban route choice behaviour. In *Papers presented during the Ninth International Symposium on Transportation and Traffic Theory held in Delft the Netherlands, 11-13 July 1984.*, 1984.

[11] M. E. Ben-Akiva and S. R. Lerman. *Discrete choice analysis: theory and application to travel demand*, volume 9. MIT press, 1985.

[12] P. C. Boxall and W. L. Adamowicz. Understanding heterogeneous preferences in random utility models: a latent class approach. *Environmental and resource economics*, 23:421–446, 2002.

[13] J. Broach, J. Dill, and J. Gliebe. Where do cyclists ride? a route choice model developed with revealed preference GPS data. *Transportation Research Part A: Policy and Practice*, 46(10):1730–1740, 2012.

[14] B. P. Bruck and A. Subramanian. Bike-sharing rebalancing problems. In *Encyclopedia of Optimization*, pages 1–9. Springer International Publishing, Cham, 2020.

[15] R. Buehler and J. Dill. Bikeway networks: A review of effects on cycling. *Transport Reviews*, 36(1):9–27, 2016.

[16] C. M. Burke and D. M. Scott. The space race: A framework to evaluate the potential travel-time impacts of reallocating road space to bicycle facilities. *Journal of Transport Geography*, 56:110–119, 2016.

[17] L. Caggiani, R. Camporeale, and M. Ottomanelli. Planning and design of equitable free-floating bike-sharing systems implementing a road pricing strategy. *Journal of Advanced Transportation*, 2017, 2017.

[18] Y. Cai, J. Chen, D. Lei, and J. Yu. The integration of multimodal networks: The generalized modal split and collaborative optimization of transportation hubs. *Journal of Advanced Transportation*, 2022, 2022.

[19] E. Cascetta, A. Nuzzolo, F. Russo, and A. Vitetta. A modified logit route choice model overcoming path overlapping problems. specification and some calibration results for interurban networks. In *Transportation and Traffic Theory. Proceedings of The 13th International Symposium On Transportation And Traffic Theory, Lyon, France*, pages 697–711, 1996.

[20] P. Chen, Q. Shen, and S. Childress. A gps data-based analysis of built environment influences on bicyclist route preferences. *International Journal of Sustainable Transportation*, 12:218–231, 3 2018.

[21] R. Cheng, S. Zhong, Z. Wang, O. Anker Nielsen, and Y. Jiang. A hyper-heuristic approach to the strategic planning of bike-sharing infrastructure. *Computers & Industrial Engineering*, 173:108704, 2022.

[22] J. Y. Chow and H. R. Sayarshad. Symbiotic network design strategies in the presence of coexisting transportation networks. *Transportation Research Part B: Methodological*, 62:13–34, 2014.

[23] R. L. Church and S. Wang. Solving the p-median problem on regular and lattice networks. *Computers & Operations Research*, 123:105057, 2020.

[24] S. Cobzaş, R. Miculescu, and A. Nicolae. Basic Facts Concerning Lipschitz Functions. In *Lipschitz Functions*, volume 2241, pages 99–142. Springer International Publishing, 2019. Series Title: Lecture Notes in Mathematics.

[25] F. Cueva, P. Shi, and P. Cedillo. Designing bike-sharing systems supported by data: A systematic literature review. *IEEE Access*, 12:162731–162754, 2024.

[26] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[27] R. Doorley, V. Pakrashi, W. Y. Szeto, and B. Ghosh. Designing cycle networks to maximize health, environmental, and travel time impacts: An optimization-based approach. *International Journal of Sustainable Transportation*, 14(5): 361–374, 2020.

[28] J. Duthie and A. Unnikrishnan. Optimization framework for bicycle network design. *Journal of Transportation Engineering*, 140(7):04014028, 2014.

[29] M. Ehrgott, J. Y. Wang, A. Raith, and C. van Houtte. A bi-objective cyclist route choice model. *Transportation Research Part A: Policy and Practice*, 46 (4):652–663, 2012.

[30] S. Elloumi. A tighter formulation of the p-median problem. *Journal of combinatorial optimization*, 19(1):69–83, 2010.

[31] B. Epperson. Evaluating suitability of roadways for bicycle use: toward a cycling level-of-service standard. *Transportation Research Record*, 1438:9–16, 1994.

[32] M. Eugénia Captivo. Fast primal and dual heuristics for the p-median location problem. *European Journal of Operational Research*, 52(1):65–74, 1991.

[33] Y. Fan, J. Ding, H. Liu, Y. Wang, and J. Long. Large-scale multimodal transportation network models and algorithms-part i: The combined mode split and traffic assignment problem. *Transportation Research Part E: Logistics and Transportation Review*, 164:102832, 2022.

[34] M. Fosgerau, E. Frejinger, and A. Karlstrom. A link based network route choice model with unrestricted choice set. *Transportation Research Part B: Methodological*, 56:70–80, 2013.

[35] L. Frank, N. Dirks, and G. Walther. Improving rural accessibility by locating multimodal mobility hubs. *Journal of Transport Geography*, 94:103111, 2021.

[36] E. Frejinger, M. Bierlaire, and M. Ben-Akiva. Sampling of alternatives for route choice modeling. *Transportation Research Part B: Methodological*, 43 (10):984–994, 2009.

[37] R. D. Galvão. A dual-bounded algorithm for the p-median problem. *Operations Research*, 28(5):1112–1121, 1980.

[38] D. A. Giménez-Gaydou, A. Cupido dos Santos, G. Mendes, I. Frade, and A. S. Ribeiro. Energy consumption and pollutant exposure estimation for cyclist routes in urban areas. *Transportation Research Part D: Transport and Environment*, 72:1–16, 2019.

[39] Global Designing Cities Initiative and National Association of City Transportation Officials. *Global street design guide*. Island Press, 2016.

[40] B. Goldengorin, D. Krushinsky, and P. M. Pardalos. *The p-Median Problem*, pages 25–73. Springer New York, 2013.

[41] N. Gore, S. Arkatkar, G. Joshi, and C. Antoniou. Modified bureau of public roads link function. *Transportation Research Record*, 2677(5):966–990, 2023.

[42] W. H. Greene and D. A. Hensher. A latent class model for discrete choice analysis: contrasts with mixed logit. *Transportation Research Part B: Methodological*, 37(8):681–698, 2003.

[43] A. Grisiute, N. Wiedemann, P. Herthogs, and M. Raubal. An ontology-based approach for harmonizing metrics in bike network evaluations. *Computers, Environment and Urban Systems*, 113, 10 2024.

[44] E. Güldü, Ç. Kuşçu Şimşek, and S. Selim. A study on the improvement of bicycle transportation in Sivas city using hybrid multi-criteria model based network analysis. *Environment, Development and Sustainability*, pages 1–20, 2024.

[45] N. Guo, R. Jiang, S. Wong, Q.-Y. Hao, S.-Q. Xue, and M.-B. Hu. Bicycle flow dynamics on wide roads: Experiments and simulation. *Transportation Research Part C: Emerging Technologies*, 125:103012, 2021.

[46] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024. URL `https://www.gurobi.com`.

[47] S. L. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12(3):450–459, 1964.

[48] S. L. Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13 (3):462–475, 1965.

[49] D. L. Harkey, D. W. Reinfurt, and A. Sorton. The bicycle compatibility index: A level of service concept, implementation manual. Technical report, United States. Federal Highway Administration, 1998.

[50] T. He, J. Bao, S. Ruan, R. Li, Y. Li, H. He, and Y. Zheng. Interactive bike lane planning using sharing bikes' trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1529–1542, 2020.

[51] J. Hood, E. Sall, and B. Charlton. A GPS-based bicycle route choice model for san francisco, california. *Transportation Letters*, 3(1):63–75, 2011.

[52] C. Howard and E. K. Burns. Cycling to work in phoenix: Route choice, travel behavior, and commuter characteristics. *Transportation Research Record*, 1773(1):39–46, 2001.

[53] J. Hrnčíř, P. Žilecký, Q. Song, and M. Jakob. Practical multicriteria urban bicycle routing. *IEEE Transactions on Intelligent Transportation Systems*, 18 (3):493–504, 2017.

[54] V. P. Il'ev. An approximation guarantee of the greedy descent algorithm for minimizing a supermodular set function. *Discrete Applied Mathematics*, 114 (1-3):131–146, 2001.

[55] Istat. Aspetti della vita quotidiana - anni 2013-2021. `https://www.istat.it/`, 2025. Accessed: June 27, 2025.

[56] J. G. Jin, H. Nieto, and L. Lu. Robust bike-sharing stations allocation and path network design: a two-stage stochastic programming model. *Transportation Letters*, 12(10):682–691, 2020.

[57] O. Kariv and S. L. Hakimi. An algorithmic approach to network location problems. I: The p-centers. *SIAM Journal on Applied Mathematics*, 37(3): 513–538, 1979.

[58] K. Khavarian, S. Vosough, and C. Roncoli. Bike users' route choice behaviour: Expectations from electric bikes versus reality in greater helsinki. *Journal of Cycling and Micromobility Research*, 2:100045, 12 2024.

[59] F. Kon, É. C. Ferreira, H. A. de Souza, F. Duarte, P. Santi, and C. Ratti. Abstracting mobility flows from bike-sharing systems. *Public Transport*, 14:545–581, 2022.

[60] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.

[61] G. Laporte, F. Meunier, and R. Wolfler Calvo. Shared mobility systems. *4OR*, 13:341–360, 2015.

[62] G. Laporte, F. Meunier, and R. Wolfler Calvo. Shared mobility systems: an updated survey. *Annals of Operations Research*, 271:105–126, 2018.

[63] X. Li, Y. Luo, T. Wang, P. Jia, and H. Kuang. An integrated approach for optimizing bi-modal transit networks fed by shared bikes. *Transportation Research Part E: Logistics and Transportation Review*, 141:102016, 2020.

[64] A. Liazos, C. Iliopoulou, K. Kepaptsoglou, and E. Bakogiannis. Geofence planning for electric scooters. *Transportation Research Part D: Transport and Environment*, 102:103149, 2022.

[65] J. Lim, K. Dalmeijer, S. Guhathakurta, and P. V. Hentenryck. The bicycle network improvement problem. *Journal of Transportation Engineering, Part A: Systems*, 148(11):04022095, 2022.

[66] J.-J. Lin and R.-Y. Liao. Sustainability SI: Bikeway network design model for recreational bicycling in scenic areas. *Networks and Spatial Economics*, 16: 9–31, 2016.

[67] J.-J. Lin and C.-J. Yu. A bikeway network design model for urban areas. *Transportation*, 40:45–68, 2013.

[68] J.-R. Lin and T.-H. Yang. Strategic design of public bicycle sharing systems with service level constraints. *Transportation Research Part E: Logistics and Transportation Review*, 47(2):284–294, 2011.

[69] J.-R. Lin, T.-H. Yang, and Y.-C. Chang. A hub location inventory model for bicycle sharing system design: Formulation and solution. *Computers & Industrial Engineering*, 65(1):77–86, 2013. Intelligent Manufacturing Systems.

[70] C. Liu, F. Johansson, I. Kristoffersson, and C. Rydergren. Estimating bicycle route choice model in value-of-time space, a case study for Gothenburg. *Journal of Cycling and Micromobility Research*, 4:100064, 6 2025.

[71] H. Liu, W. Szeto, and J. Long. Bike network design problem with a path-size logit-based equilibrium constraint: Formulation, global optimization, and matheuristic. *Transportation Research Part E: Logistics and Transportation Review*, 127:284–307, 2019.

[72] S. Liu, Z.-J. M. Shen, and X. Ji. Urban bike lane planning with bike trajectories: Models, algorithms, and a real-world case study. *Manufacturing & Service Operations Management*, 24(5):2500–2515, 2022.

[73] W. Lu, D. M. Scott, and R. Dalumpines. Understanding bike share cyclist route choice using gps data: Comparing dominant routes and shortest paths. *Journal of Transport Geography*, 71:172–181, 7 2018.

[74] Q. Luo, S. Li, and R. C. Hampshire. Optimal design of intermodal mobility networks under uncertainty: Connecting micromobility with mobility-on-demand transit. *EURO Journal on Transportation and Logistics*, 10:100045, 2021.

[75] H. Mahfouz, R. Lovelace, and E. Arcaute. A road segment prioritization approach for cycling infrastructure. *Journal of Transport Geography*, 113: 103715, 2023.

[76] T. Mai, M. Fosgerau, and E. Frejinger. A nested recursive logit model for route choice analysis. *Transportation Research Part B: Methodological*, 75: 100–112, 5 2015.

[77] F. Maronzana. On the location of supply points to minimize transportation costs. *Operations Research Quarterly*, 12:138–139, 1964.

[78] A. Mauttone, G. Mercadante, M. Rabaza, and F. Toledo. Bicycle network design: model and solution algorithm. *Transportation Research Procedia*, 27:969–976, 2017. 20th EURO Working Group on Transportation Meeting, EWGT 2017, 4-6 September 2017, Budapest, Hungary.

[79] A. Meister, M. Felder, B. Schmid, and K. W. Axhausen. Route choice modeling for cyclists on urban networks. *Transportation Research Part A: Policy and Practice*, 173, 7 2023.

[80] A. Meister, Z. Liang, M. Felder, and K. W. Axhausen. Comparative study of route choice models for cyclists. *Journal of Cycling and Micromobility Research*, 2:100018, 12 2024.

[81] G. Menghini, N. Carrasco, N. Schüssler, and K. W. Axhausen. Route choice of cyclists in zurich. *Transportation Research Part A: Policy and Practice*, 44: 754–765, 2010.

[82] M. Mesbah, R. Thompson, and S. Moridpour. Bilevel optimization approach to design of network of bike lanes. *Transportation research record*, 2284(1): 21–28, 2012.

[83] N. Mladenović, J. Brimberg, P. Hansen, and J. A. Moreno-Pérez. The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 179(3):927–939, 2007.

[84] R. D. Monteiro and I. Adler. Interior path following primal-dual algorithms. part ii: Convex quadratic programming. *Mathematical Programming*, 44(1-3): 43–66, 1989.

[85] A. T. Murray and R. L. Church. Applying simulated annealing to location-planning models. *Journal of Heuristics*, 2:31–53, 1996.

[86] R. Nair and E. Miller-Hooks. Equilibrium network design of shared-vehicle systems. *European Journal of Operational Research*, 235(1):47–61, 2014.

[87] E. Nash, A. Cope, P. James, and D. Parker. Cycle network planning: Towards a holistic approach using temporal topology. *Transportation Planning and Technology*, 28(4):251–271, 2005.

[88] L. E. Olmos, M. S. Tadeo, D. Vlachogiannis, F. Alhasoun, X. Espinet Alegre, C. Ochoa, F. Targa, and M. C. González. A data science framework for planning the growth of bicycle infrastructures. *Transportation Research Part C: Emerging Technologies*, 115:102640, 2020.

[89] J. P. Ospina, J. C. Duque, V. Botero-Fernández, and A. Montoya. The maximal covering bicycle network design problem. *Transportation Research Part A: Policy and Practice*, 159:222–236, 2022.

[90] T. J. Pikora, F. C. Bull, K. Jamrozik, M. Knuiman, B. Giles-Corti, and R. J. Donovan. Developing a reliable audit instrument to measure the physical environment for physical activity. *American Journal of Preventive Medicine*, 23 (3):187–194, 2002.

[91] L. Pirolo and R. Moscarelli. Integrating cycling tourism, e-mobility and local development: A spatial model to plan slow tourist paths for e-bikes. *Regional Science Policy & Practice*, page 100237, 2025.

[92] C. G. Prato. Route choice modeling: past, present and future research directions. *Journal of choice modelling*, 2(1):65–100, 2009.

[93] C. G. Prato, K. Halldórsdóttir, and O. A. Nielsen. Evaluation of land-use and transport network effects on cyclists' route choices in the copenhagen region in value-of-distance space. *International Journal of Sustainable Transportation*, 12:770–781, 11 2018.

[94] R. Praxedes, A. Subramanian, S. Ardizzoni, L. Consolini, M. Laurini, and M. Locatelli. Optimization methods to improve the quality of a cycling network under budget constraints. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, pages 4982–4987, 2025.

[95] R. Praxedes, A. Subramanian, S. Ardizzoni, L. Consolini, M. Laurini, and M. Locatelli. Optimizing cycling network design in the city of Parma. Submitted to Transportation Research Part A: Policy and Practice, 2025.

[96] J. Pucher and R. Buehler. Making cycling irresistible: Lessons from the netherlands, denmark and germany. *Transport Reviews*, 28(4):495–528, 2008.

[97] H. Reckermann, M. Gutjar, and M. Kowald. Studying shared bike route choice behavior using a bike-sharing system in Germany. *Journal of Cycling and Micromobility Research*, 2:100017, 12 2024.

[98] Regione Emilia-Romagna. Elenco regionale dei prezzi delle opere pubbliche. `https://territorio.regione.emilia-romagna.it/osservatorio/elenco_regionale_prezzi`, 2025. Accessed: May 11, 2025.

[99] M. G. Resende and R. F. Werneck. A hybrid heuristic for the p-median problem. *Journal of heuristics*, 10:59–88, 2004.

[100] C. S. ReVelle and R. W. Swain. Central facilities location. *Geographical Analysis*, 2(1):30–42, 1970.

[101] N. Rieser-Schüssler, M. Balmer, and K. W. Axhausen. Route choice sets for very high-resolution data. *Transportmetrica A: Transport Science*, 9(9):825–845, 2013.

[102] S. Rossetti, B. Caselli, F. Stabile, and M. Carra. How do SUMPs Consider Factors Influencing Walkability and Cyclability? A Review of Literature and Planning Tools. *EUROPEAN TRANSPORT/TRASPORTI EUROPEI*, 97:1–11, 2024.

[103] S. Ryu, A. Chen, J. Su, and K. Choi. A multi-class, multi-criteria bicycle traffic assignment model. *International Journal of Sustainable Transportation*, 15: 524–540, 2021.

[104] M. Rönnqvist, G. Svenson, P. Flisberg, and L. E. Jönsson. Calibrated route finder: Improving the safety, environmental consciousness, and cost effectiveness of truck routing in sweden. *Interfaces*, 47:372–395, 9 2017.

[105] J. Shen. Latent class model or mixed logit model? a comparison by transport mode choice data. *Applied Economics*, 41(22):2915–2924, 2009.

[106] C. Shui and W. Szeto. A review of bicycle-sharing service planning problems. *Transportation Research Part C: Emerging Technologies*, 117:102648, 2020.

[107] A. Sobhani, H. A. Aliabadi, and B. Farooq. Metropolis-hasting based expanded path size logit model for cyclists' route choice using gps data. *International Journal of Transportation Science and Technology*, 8:161–175, 6 2019.

[108] K. Sohn. Multi-objective optimization of a road diet network design. *Transportation Research Part A: Policy and Practice*, 45(6):499–511, 2011.

[109] H. Spiess. Conical volume-delay functions. *Transportation Science*, 24(2): 153–158, 1990.

[110] D. Stauffer and A. Aharony. *Introduction to Percolation Theory*. Taylor & Francis, London, 2 edition, 1992.

[111] C. Steinacker, D.-M. Storch, M. Timme, and M. Schröder. Demand-driven design of bicycle infrastructure networks for improved urban bikeability. *Nature Computational Science*, 2(10):655–664, 2022.

[112] M. Tira and R. Busi. *Safety for pedestrians and two-wheelers/Sicurezza dei pedoni e dei conducenti dei mezzi a due ruote*, volume 3. Bios, 2001.

[113] D. Ton, D. Duives, O. Cats, and S. Hoogendoorn. Evaluating a data-driven approach for choice set identification using gps bicycle route choice data from amsterdam. *Travel Behaviour and Society*, 13:105–117, 10 2018.

[114] K. E. Train. *Discrete choice methods with simulation*. Cambridge university press, 2009.

[115] Y. Wang, H. Liu, Y. Fan, J. Ding, and J. Long. Large-scale multimodal transportation network models and algorithms-part ii: Network capacity and network design problem. *Transportation Research Part E: Logistics and Transportation Review*, 167:102918, 2022.

[116] Y. Wang, Y. Wu, Z. Li, K. Liao, C. Li, and G. Song. Route planning for active travel considering air pollution exposure. *Transportation Research Part D: Transport and Environment*, 103:103176, 2022.

[117] J. G. Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers, Part II*, 1(36):325–378, 1952.

[118] S. Weikl and P. Mayer. Data-driven quality assessment of cycling networks. *Frontiers in Future Transportation*, 4, 3 2023.

[119] N. Wiedemann, C. Nöbel, L. Ballo, H. Martin, and M. Raubal. Bike network planning in limited urban space. *Transportation Research Part B: Methodological*, 192:103135, 2 2025.

[120] M. Winters, M. Brauer, E. M. Setton, and K. Teschke. Mapping bikeability: A spatial tool to support sustainable travel. *Environment and Planning B: Planning and Design*, 40(5):865–883, 2013.

[121] J. Ye, Y. Jiang, J. Chen, Z. Liu, and R. Guo. Joint optimisation of transfer location and capacity for a capacitated multimodal transport network with elastic demand: a bi-level programming model and paradoxes. *Transportation Research Part E: Logistics and Transportation Review*, 156:102540, 2021.

[122] P. Zattoni Scroccaro, P. van Beek, P. Mohajerin Esfahani, and B. Atasoy. Inverse optimization for routing problems. *Transportation Science*, 2025.

[123] F. Zhang and W. Liu. An economic analysis of integrating bike sharing service with metro systems. *Transportation Research Part D: Transport and Environment*, 99:103008, 2021.

[124] W. Zhang, R. Buehler, A. Broaddus, and T. Sweeney. What type of infrastructures do e-scooter riders prefer? a route choice model. *Transportation Research Part D: Transport and Environment*, 94, 5 2021.

[125] Z. Zhao and Y. Liang. A deep inverse reinforcement learning approach to route choice modeling with context-dependent rewards. *Transportation Research Part C: Emerging Technologies*, 149, 4 2023.

[126] S. Zhu and F. Zhu. Multi-objective bike-way network design problem with space–time accessibility constraint. *Transportation*, 47(5):2479–2503, 2020.

[127] M. Zimmermann, T. Mai, and E. Frejinger. Bike route choice modeling using gps data without choice sets of paths. *Transportation Research Part C: Emerging Technologies*, 75:183–196, 2 2017.

[128] T. Zuo and H. Wei. Bikeway prioritization to increase bicycle network connectivity and bicycle-transit connection: A multi-criteria decision analysis approach. *Transportation Research Part A: Policy and Practice*, 129:52–71, 2019.

Appendix A

Choosing basic costs

The need to explore and interact with the urban environment and society through more sustainable mobility options is crucial for reducing car dependence and improving the quality of life in cities. Many destinations can be reached by bicycle, a sustainable travel mode and accessible to the entire community. The debate on developing bicycle-friendly urban environments has been, and continues to be, intense, and the research on how to properly plan and design cycling paths is extensive. Consequently, the choice of basic costs is part of a broader research in which inspection-based cycling indicators were identified and developed [102], assessing aspects such as accessibility, safety and security, inclusive design, and enjoyment. For the purpose of this research, some of these Key Performance Indicators (KPIs) were selected, specifically road safety and practicability KPIs. Road safety assesses conflicts with motor vehicles and exposure to potential risks, evaluating the level of safety for cyclists in the urban environment and ensuring a safe experience for users. The practicability indicator seeks to guarantee that infrastructures and spaces are accessible and suitable for all users, including those who are most vulnerable.

Each indicator is influenced by groups of factors. Regarding the improvement of road safety, central elements include reducing speed limits and safeguarding dedicated infrastructure from vehicular traffic. Another important aspect in ensuring road safety is the presence of dedicated vertical signage and horizontal markings [1, 112].

For the practicability, the characteristics of the infrastructure are crucial. Good design of cycle paths depends on their geometry and construction quality, including the width of the paths, the paving materials used, and the level of maintenance. In addition, other factors should be considered, like the presence of permanent and temporary external elements that reduce the effective width of the cycle lane, creating bottlenecks [39, 90] (see also below for a list of such elements).

The methodology for calculating the considered KPIs involves acquiring and managing georeferenced cycling infrastructure datasets within a geographic information system environment. A value for evaluation is assigned to each analyzed factor. The overall assessment value for each infrastructure is derived from the normalized average of the factors comprising each indicator. This average is calculated by summing the values of the factors and dividing by the total number of factors. The calculation concludes with normalization to the minimum and maximum possible values for each KPI. This normalization process standardizes the data, allowing for fair comparisons between different indicators and enabling a more accurate assessment of the overall performance. Table A.1 lists the selected factors and the value associated with each one, regarding the road safety and practicability KPIs (the higher the value, the better).

In Table A.1, for road safety, protection from vehicular traffic is classified as *high or no interaction* when continuous physical protection elements are present, such as trees, shrubs, barriers, kerbs, or bollards of varying heights with separating kerbs. This protection is rated *medium* when discontinuous physical elements at the same height, are present. A *low* level indicates only pavement with a separating gap, while an *absent* level signifies no physical separation between the cycle path and the carriageway. Additionally, *signage* and *markings* refer to presence of vertical signs and horizontal markings indicating the bicycle path. Lastly, the term *bike box* refers to a designated stop-and-wait area for bicycles at intersections controlled by traffic lights.

For practicability, *dimension* refers to the width of the cycle path, while *bottlenecks* indicate the presence of a restriction, which can be absent, caused by permanent elements (such as public lighting, road signs, advertising signs, traffic lights, bus

Table A.1: Considered KPIs – Selected attributes and assigned values.

Indicator	Factors	Evaluation	Values
Road safety	Protection from vehicular traffic	High or no interaction, Medium, Low, Absent	1, 0 -1, -2
	Speed limit	30 km/h, 50 km/h, ≥ 70 km/h	2, 1, -1
	Signage	Yes, No	0, -1
	Markings	Yes, No	0, -1
	Bike-box	Yes, No	1, 0
Practicability	Dimension and Bottlenecks	≥ 2.50 m + Absent or Temporary	1
		≥ 2.50 m + Permanent	0
		≥ 1.50 m and ≤ 2.50 + Absent or Temporary	0
		≥ 1.50 m and ≤ 2.50 + Permanent	-1
		≤ 1.50 m + Absent or Temporary	-1
		≤ 1.50 m + Permanent	-2
	Pavement maintenance	Excellent, Good, Sufficient, Poor	2, 1, 0, -1
	Pavement material	Marble, Cobblestones, Cubes	-1
		Stone slabs, Concrete slabs, Concrete, Asphalt, Blocks, Self-binding, Gravel Grass pavers, Aggregates, Other	0
	Protrusions	Yes, No	-1, 0

stops, trees, etc.), or by temporary elements (such as street furniture, recycling bins, and speed cameras). Moreover, *pavement material* and *pavement maintenance* factors refers to the material type and quality of maintenance, respectively, describing the condition and material of the pavement. Lastly, the *protrusion* factor identifies the presence of elements protruding from the pavement along the cycle path, with a height exceeding 2.5 cm.

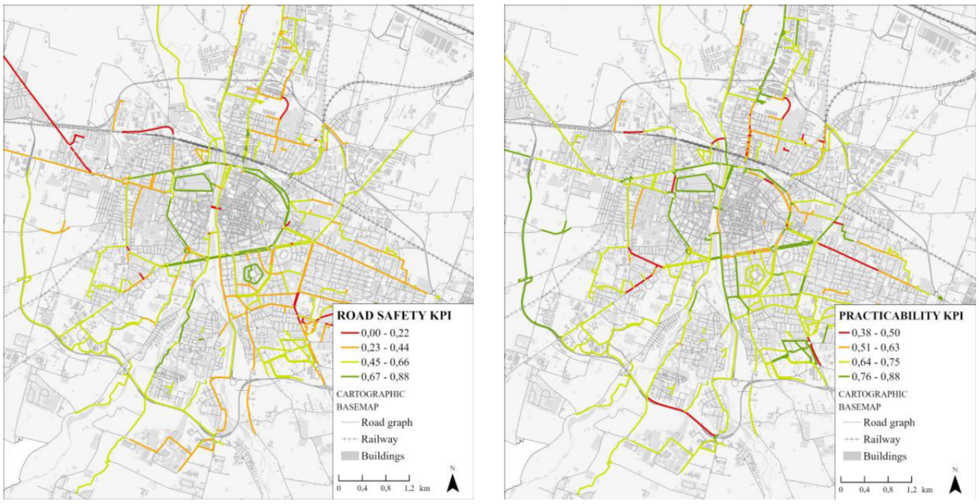
We applied these KPIs to the bicycle infrastructure system of the city of Parma, Italy, to estimate the cost associated with each road segment based on these indicators. For cycle roads, all factors described in Table A.1 were considered. However, their corresponding values were mapped through an affine transformation in order to fit within an interval between 0 and 1, as described in what follows: for each road criterion (safety and practicability), we calculate the maximum and minimum possible KPI values associated with it: for $h \in \{1,\ldots,r\}$, let $C^h_{\max} := \max C^h_{ij}$, and $C^h_{\min} := \min C^h_{ij}$, respectively. For example, for road safety, the maximum possible KPI value is 4, obtained by summing the maximum values of each contributing factor, whilst the minimum one is -5 (see Table A.1). After determining these values, the KPI of an arc (i, j) for a given criterion h is calculated by summing the values of the factors associated with that arc and influencing the analyzed criterion, obtaining its cumulative value C^h_{ij}. Finally, we compute costs c^h_{ij} for any criterion $h \in \{1, \ldots, r\}$ and arc of the network as follows:

$$c^h_{ij} := 1 - \frac{C^h_{ij}}{C^h_{\max} - C^h_{\min}}.$$

Note that value $\frac{C^h_{ij}}{C^h_{\max} - C^h_{\min}}$ is between $[0, 1]$ and c^h_{ij} corresponds to its complement within the same interval. This is due to the fact that a low KPI for a certain criterion, in our approach, is perceived as a cost which we wish to minimize. For instance, an extremely unsafe road segment will be associated with a basic cost close to 1, which is considered a high cost to be minimized.

Additionally, due to lack of information regarding the non-cycle roads, we considered, for these cases, the speed limit and protection from vehicular traffic as factors for the road safety indicator, and the pavement materials (cobblestones for the historical center, gravels for parks, and asphalt for the others) as a factor for the practicability. Note that including non-cycle roads is necessary, both because the bike-network of Parma is not a connected graph, and because some users favor non-cycle roads when they allow shortening the path. It is important to mention that the values for the safety and practicability indicators of a specific road were multiplied by its length as part of the normalization process.

Figure A.1 illustrates the results obtained from applying these KPIs to the cycle paths of Parma, without the multiplication by the roads length. These results are represented visually using graphical elaborations with an interval scale and a graduated color scale, ranging from red (highlighting critical segments) to green (indicating segments with positive characteristics). Note that in the figure, only the segments of the cycle roads are colored, even though KPI values were assigned to all streets. This choice was made to better highlight the values associated with the cycling infrastructure. Including the KPI values of all streets would have made the visualization too uniform and would have visually overwhelmed the cycle paths, making the interpretation less clear.



(a) Road safety indicator.　　(b) Practicability indicator.

Figure A.1: Results of the calculation of the indicators for the case studies of Parma.

The analysis of the Road Safety KPI reveals an overall good safety level for cycle paths, with some exceptions in isolated cases where interaction with vehicular traffic or the combination of protection level and speed limit has a high negative impact. The presence of buffer elements, such as hedges, trees, or continuous artificial barriers between the cycle infrastructure and the roadway, would further enhance user safety. For example, paths isolated by greenery or located in reduced-speed zones report a high safety rating. Regarding the Practicability KPI, the outcome is largely influenced by the characteristics of the infrastructure, including pavement width, the presence of bottlenecks, the type of materials used, and pavement quality.

Appendix B

Proofs

Section 3.1.2

Proposition 2. *Let $P' \supset P$. Then, $g(P') \leq g(P)$.*

Proof. Let us denote by $M_{\bar{A},P}$ the restriction of matrix M with rows in $\bar{A}$ and columns in P. Moreover, let $\Delta_{|P|} = \left\{ \alpha \in \mathbb{R}_+^{|P|} \mid \sum_{i=1}^{|P|} \alpha_i = 1 \right\}$. Then,

$$
\begin{aligned}
g(P) &= \min_{\alpha_P \in \Delta_{|P|}} \|M_{\bar{A},P}\alpha_P - \bar{x}\|^2 \\
g(P') &= \min_{\alpha_{P'} \in \Delta_{|P'|}} \|M_{\bar{A},P'}\alpha_{P'} - \bar{x}\|^2.
\end{aligned}
$$

Since $P \subset P'$, then we have $\alpha_{P'} = [\alpha_P, \alpha_{P' \setminus P}]$ and $M_{\bar{A},P'} = [M_{\bar{A},P} | M_{\bar{A},P' \setminus P'}]$. Let $\bar{\alpha}_P$ be a feasible solution of the optimization problem (3.10) with set of weights P. If we set $\bar{\alpha}_{P' \setminus P} = 0$, then $\bar{\alpha}_{P'} = [\bar{\alpha}_P, \bar{\alpha}_{P' \setminus P}]$ is a feasible solution of the optimization problem (3.10) with set of weights P', with the same objective function value as $\bar{\alpha}_P$. Therefore, to each feasible solution of the first problem with set P, we can associate a feasible solution of the second problem with set P', and the two solutions have the same objective function value. Then the inequality $g(P') \leq g(P)$ immediately follows. □

Corollary 3.1.1. *Let P be a set of weights and α be an optimal solution of the* *optimization problem* (3.10)*. If $\bar{P} \subset P$ is such that, for all $\mathbf{p}^\ell \in P \setminus \bar{P}$, $\alpha_\ell = 0$, then $g(P) = g(\bar{P})$.*

Proof. The optimal solution α can be written as $[\alpha_{\bar{P}}, \alpha_{P\setminus\bar{P}}]$, where, by assumption, $\alpha_{P\setminus\bar{P}} = 0$. Then, $\alpha_{\bar{P}}$ is a feasible solution of (3.10) with set of weights $\bar{P}$ and its objective function value is equal to $g(P) \geq g(\bar{P})$. Since by Proposition 2, it must hold that $g(P) \leq g(\bar{P})$, it follows that $g(P) = g(\bar{P})$, and $\alpha_{\bar{P}}$ is also an optimal solution of (3.10) with set of weights $\bar{P}$. □

Lemma 3.1.3. *For each $w \in W$ and $\mathbf{p}^\ell \in \Delta_r$, let $S^{w,\ell}$ denote the set of SPs when the cost of each arc (i, j) is $\mathbf{c}_{ij}^\top \mathbf{p}^\ell$. Under Assumption 3.1.2, the set*

$$\Delta_r(w,\ell) = \left\{ \mathbf{p}^\ell \in \Delta_r \;\mid\; |S^{w,\ell}| > 1 \right\},$$

has null measure in Δ_r.

Proof. Let $\mathbf{p}^\ell \in \Delta_r(w,\ell)$. Then there exist $\pi_1, \pi_2 \in S^{w,\ell}$. It follows that these two paths have the same cost

$$\sum_{h=1}^{r} p_h^\ell \lambda_h(\pi_1) = \sum_{h=1}^{r} p_h^\ell \lambda_h(\pi_2),$$

and so $\sum_{h=1}^{r} p_h^\ell (\lambda_h(\pi_1) - \lambda_h(\pi_2)) = 0$. We want to prove that $\dim(\Delta_r(w,\ell)) < \dim(\Delta_r) = r-1$. Note that the space $\Delta_r(w,\ell)$ is the set of weights that satisfy the system of two linear equations

$$R\mathbf{p}^\ell = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

where

$$R = \begin{bmatrix} 1 & \dots & 1 \\ \lambda_1(\pi_1) - \lambda_1(\pi_2) & \dots & \lambda_r(\pi_1) - \lambda_r(\pi_2) \end{bmatrix}.$$

We show that $\rho(R) = 2$, where $\rho(R)$ denotes the rank of matrix R. If this is the case, then $\dim(\Delta_r(w,\ell)) = r - \rho(R) = r-2$. By contradiction, if $\rho(R) = 1$, there are two possibilities:

- the second row is a multiple of the first: in this case the system has no solution;

- the second row consists only of zeros, that is, $\lambda_h(\pi) = \lambda_h(\pi')$ for all $h \in \{1,\dots,r\}$: this is not allowed by Assumption 3.1.2.

Since $\dim(\Delta_r(w,\ell)) < \dim(\Delta_r)$, the subspace of weights such that there exist at least two paths with same cost has null measure in Δ_r.

$\square$

Proposition 3. *For all $\bar{\mathscr{Q}} = (\bar{\mathbf{p}}^1,\dots,\bar{\mathbf{p}}^q) \in \Delta_r^q$ except those over a set of null measure over Δ_r^q, it holds that there exists $\delta > 0$ such that*

$$(\forall \mathscr{Q} \in I_\delta(\bar{\mathscr{Q}}))\ \bar{g}(\mathscr{Q}) = \bar{g}(\bar{\mathscr{Q}}),$$

where

$$I_\delta(\bar{\mathscr{Q}}) = \{\mathscr{Q} = (\mathbf{p}^1,\dots,\mathbf{p}^q) \in \Delta_r^q \ \mid\ \|\mathbf{p}^\ell - \bar{\mathbf{p}}^\ell\| < \delta,\ \ell \in Q\},$$

is a neighborhood of $\bar{\mathscr{Q}}$.

Proof. Let $\bar{\Delta}_r = \Delta_r \setminus \left[\cup_{w\in W} \cup_{\ell=1}^q \Delta_r(w,\ell)\right]$. For each O-D pair $w \in W$ and each $\ell \in Q$, let $\bar{\mathbf{p}}^\ell \notin \Delta_r(w,\ell)$, that is, $\bar{\mathscr{Q}} = (\bar{\mathbf{p}}^1,\dots,\bar{\mathbf{p}}^q) \in \bar{\Delta}_r^q$. Note that, in view of Lemma 3.1.3, $\cup_{w\in W} \cup_{\ell=1}^q \Delta_r(w,\ell)$ is a set of null measure in Δ_r^q. For each $w \in W$, there exists a unique SP $\pi^{w,\ell} \in S^{w,\ell}$ for all $\ell \in Q$. This means that there exist $a_{ij}^{w,\ell}, b_{ij}^{w,\ell}$ with:

$$(\forall (i,j) \in A)\ (\forall \ell \in Q)\ a_{ij}^{w,\ell} < \mathbf{c}_{ij}^\top \bar{\mathbf{p}}^\ell < b_{ij}^{w,\ell},$$

such that if all arcs $(i,j) \in A$ have costs lying in $[a_{ij}^{w,\ell}, b_{ij}^{w,\ell}]$, then the SP $\pi^{w,\ell}$ remains the same as the one with costs of the arcs equal to $\mathbf{c}_{ij}^\top \bar{\mathbf{p}}^\ell$. Now, let

$$[a_{ij}^\ell, b_{ij}^\ell] = \bigcap_{w\in W} [a_{ij}^{w,\ell}, b_{ij}^{w,\ell}].$$

Note that $a_{ij}^\ell < \mathbf{c}_{ij}^\top \bar{\mathbf{p}}^\ell < b_{ij}^\ell$ for all $(i,j) \in A$ and $\ell \in Q$, must also hold. Then, if for some $\mathscr{Q} = (\mathbf{p}^1,\dots,\mathbf{p}^q)$ it holds that:

$$(\forall (i,j) \in A)\ (\forall \ell \in Q)\ a_{ij}^\ell \le \mathbf{c}_{ij}^\top \mathbf{p}^\ell \le b_{ij}^\ell, \tag{B.1}$$

then $\bar{g}(\mathscr{Q}) = \bar{g}(\bar{\mathscr{Q}})$. Now, let $\beta_{ij}^\ell = \min\{\mathbf{c}_{ij}^\top \bar{\mathbf{p}}^\ell - a_{ij}^\ell, b_{ij}^\ell - \mathbf{c}_{ij}^\top \bar{\mathbf{p}}^\ell\} > 0$ for each arc $(i,j) \in A$, and $\beta^\ell = \min_{(i,j)\in A}\{\beta_{ij}^\ell\} > 0$. Moreover, after setting $c = \max_{(i,j)\in A}\{\|\mathbf{c}_{ij}\|\} >$

0, for all $\ell \in Q$ we define $\delta_\ell = \frac{\beta^\ell}{c} > 0$. It follows that if $\|\mathbf{p}^\ell - \bar{\mathbf{p}}^\ell\| < \delta_\ell$, then for all $(i,j) \in A$:

$$|c_{ij}^\top(\mathbf{p}^\ell - \bar{\mathbf{p}}^\ell)| \le \|\mathbf{p}^\ell - \bar{\mathbf{p}}^\ell\| c \le \beta^\ell,$$

from which the inequalities (B.1) hold. Therefore, the thesis is proved with $\delta := \min_{\ell \in Q}\{\delta_\ell\}$.

□

Section 3.2

Proposition 5. *Function f is a supermodular function, i.e., it satisfies the following condition:*

$$f(P_1) - f(P_1 \cup \{\mathbf{p}^{\bar{\ell}}\}) \ge f(P_2) - f(P_2 \cup \{\mathbf{p}^{\bar{\ell}}\}). \tag{B.2}$$

for any $P_1 \subseteq P_2 \subseteq P$ and $\mathbf{p}^{\bar{\ell}} \in P \setminus P_2$.

Proof. For each $\pi_i \in \bar{\Pi}$ define $f_i(\hat{P}) = \min_{\mathbf{p}^\ell \in \hat{P}} e(i,\ell)$. Note that $f(\hat{P}) = \sum_{\pi_i \in \bar{\Pi}} f_i(\hat{P})$. By Proposition 4, if $P_1 \subseteq P_2$, then $f(P_2) \le f(P_1)$ and also $f_i(P_2) \le f_i(P_1)$ for each $\pi_i \in \Pi$. To prove (B.2), it is enough to prove that for each $\pi_i \in \Pi$:

$$f_i(P_1) - f_i(P_1 \cup \{\mathbf{p}^{\bar{\ell}}\}) \ge f_i(P_2) - f_i(P_2 \cup \{\mathbf{p}^{\bar{\ell}}\}). \tag{B.3}$$

Note that

$$f_i(P_2 \cup \{\mathbf{p}^{\bar{\ell}}\}) = \min_{\mathbf{p}^\ell \in P_2 \cup \{\mathbf{p}^{\bar{\ell}}\}} e(i,\ell) = \min(\min_{\mathbf{p}^\ell \in P_2} e(i,\ell),\ e(i,\bar{\ell})).$$

We distinguish two cases:

- if $\min_{\mathbf{p}^\ell \in P_2} e(i,\ell) \le e(i,\bar{\ell})$, then $f_i(P_2 \cup \{\mathbf{p}^{\bar{\ell}}\}) = f_i(P_2)$. Therefore, $f_i(P_2) - f_i(P_2 \cup \{\mathbf{p}^{\bar{\ell}}\}) = 0$. Moreover, by Proposition 4, $f_i(P_1) - f_i(P_1 \cup \{\mathbf{p}^{\bar{\ell}}\}) \ge 0$ from which (B.3) follows.

- if $\min_{\mathbf{p}^\ell \in P_2} e(i,\ell) > e(i,\bar{\ell})$, then $f_i(P_2 \cup \{\mathbf{p}^{\bar{\ell}}\}) = e(i,\bar{\ell})$. Moreover, since $\min_{\mathbf{p}^\ell \in P_1} e(i,\ell) \ge \min_{\mathbf{p}^\ell \in P_2} e(i,\ell)$, it follows that $\min_{\mathbf{p}^\ell \in P_1} e(i,\ell) > e(i,\bar{\ell})$ and, consequently,

  $$f_i(P_1 \cup \{\mathbf{p}^{\bar{\ell}}\}) = \min(\min_{\mathbf{p}^\ell \in P_1} e(i,\ell),\ e(i,\bar{\ell})) = e(i,\bar{\ell}) = f_i(P_2 \cup \{\mathbf{p}^{\bar{\ell}}\}).$$

  As a result, since $f_i(P_2) \le f_i(P_1)$, condition (B.3) is satisfied.

□

Proposition 6. *Let us assume that* $c_{ij} > 0$ *for all arcs* $(i,j) \in A$. *Let* $P^\star$ *be an optimal solution of problem* (3.17) *and* $\bar{P} \subset \bar{\Delta}_r$ *be an optimal solution of problem* (3.19). *Let us denote with* δ *the discretization step of* $\bar{\Delta}_r$. *Then:*

$$f(\bar{P}) - f(P^\star) \leq |\bar{\Pi}| M \, \delta \sqrt{rk}, \tag{B.4}$$

where M is defined in (3.23).

Proof. We calculate the partial derivative of $d(\pi, \cdot)$ with respect to p_h^ℓ

$$\frac{\partial}{\partial p_h^\ell} \, d(\pi, \mathbf{p}^r) = \frac{\partial}{\partial p_h^\ell} \, \frac{c(\pi, \mathbf{p}^r) - c^*(w_\pi, \mathbf{p}^r)}{c^*(w_\pi, \mathbf{p}^r)} = \frac{\partial}{\partial p_h^\ell} \, \frac{c(\pi, \mathbf{p}^r)}{c^*(w_\pi, \mathbf{p}^r)}.$$

If $r \neq \ell$ the partial derivative is null, otherwise, denoting with π^* the optimal path for the (O-D) pair w_π,

$$\frac{\partial}{\partial p_h^\ell} \, d(\pi, \mathbf{p}^\ell) = \frac{\partial}{\partial p_h^\ell} \, \frac{\sum_{(i,j)\in\pi} \mathbf{c}_{ij}^\top \mathbf{p}^\ell}{\sum_{(i,j)\in\pi^*} \mathbf{c}_{ij}^\top \mathbf{p}^\ell} =$$

$$= \frac{\left(\sum_{(i,j)\in\pi} c_{ij}^h\right)\left(\sum_{(i,j)\in\pi^*} \mathbf{c}_{ij}^\top \mathbf{p}^\ell\right) - \left(\sum_{(i,j)\in\pi} \mathbf{c}_{ij}^\top \mathbf{p}^\ell\right)\left(\sum_{(i,j)\in\pi^*} c_{ij}^h\right)}{\left(\sum_{(i,j)\in\pi^*} \mathbf{c}_{ij}^\top \mathbf{p}^\ell\right)^2},$$

which, in view of $\mathbf{c}_{ij} > 0$ and $\mathbf{p}^\ell \geq 0$, can be overestimated by

$$\frac{\left(\sum_{(i,j)\in\pi} c_{ij}^h\right)\left(\sum_{(i,j)\in\pi^*} \mathbf{c}_{ij}^\top \mathbf{p}^\ell\right)}{\left(\sum_{(i,j)\in\pi^*} \mathbf{c}_{ij}^\top \mathbf{p}^\ell\right)^2} = \frac{\left(\sum_{(i,j)\in\pi} c_{ij}^h\right)}{\left(\sum_{(i,j)\in\pi^*} \mathbf{c}_{ij}^\top \mathbf{p}^\ell\right)} \leq M,$$

where the inequality follows from Lemma 3.2.3 and from the definition (3.23) of M. Therefore, function $d(\pi, \cdot)$ is Lipschitz with constant (bounded from above by) M. Since the minimum of Lipschitz functions is a Lipschitz function with constant given by the maximum of the constants (see, e.g., the work by Cobzaş et al. [24]), then $\min d(\pi_i, \cdot)$ is also a Lipschitz function with constant (bounded from above by) M. Furthermore, function f is Lipschitz with constant $|\bar{\Pi}| \cdot M$ because it is a sum, over the set of all the paths in $\bar{\Pi}$, of Lipschitz functions (see, again, the work by Cobzaş et al. [24]). It follows from Lemma 3.2.2 that there exists $\tilde{P} \subset \bar{\Delta}_r$, $|\tilde{P}| = k$, such that, if $\tilde{Q}, Q^\star$ are vectorizations of $\tilde{P}$ and $P^\star$,

$$f(\tilde{P}) - f(P^\star) = \tilde{f}(\tilde{Q}) - f(Q^\star) \leq |\bar{\Pi}| M \cdot \|\tilde{Q} - Q^\star\| \leq |\bar{\Pi}| M \, \delta \sqrt{rk}.$$

Since $\bar{P} \subset \bar{\Delta}_r$ is an optimal solution of Problem (3.19), then $f(P^\star) \leq f(\bar{P}) \leq f(\tilde{P})$ and we can conclude that

$$f(\bar{P}) - f(P^\star) \leq |\bar{\Pi}| M \, \delta \sqrt{rk},$$

as we wanted to prove. □

Appendix C

Sensitivity analysis with respect to the discretization step

In this section we evaluate the impact of the cardinality of the finite discretization $\bar{\Delta}_3$ of Δ_3. In particular we consider the sets $\bar{\Delta}_3^1 \subset \bar{\Delta}_3^2 \subset \bar{\Delta}_3^3 \subset \bar{\Delta}_3^4$ with $\bar{\Delta}_3^i$ defined as in (3.18) obtained with the discretization step $\delta^1 = 0.2$, $\delta^2 = 0.1$, $\delta^3 = 0.05$ and $\delta^4 = 0.025$, respectively. We solved the problem of Section 3.4.2 through Algorithm 4 (with ascent greedy) using the four different discretization steps defined above. In Figure C.1 we show the final values of the objective function by varying the cardinality of the approximation set $\bar{\Delta}_3$. Note that by increasing $|\bar{\Delta}_3|$ (i.e., decreasing the discretization step δ), the optimal value of the approximated problem decrease. However, the decrease is not particularly significant, and the value becomes almost constant for $\delta^2, \delta^3, \delta^4$, i.e., as the cardinality of $\bar{\Delta}_3$ increases.

Moreover, as we show below in Tables C.1–C.4, the solutions obtained in the four cases are not very different from one another. There are some differences for the solutions with $\delta^1 = 0.2$ and $\delta^2 = 0.1$, but then the solutions become quite stable for $\delta^3 = 0.05$ and $\delta^4 = 0.025$.

In order to quantitatively evaluate the differences between them we used the Earth Mover's Distance (EMD), which takes into account not only the weights but also the percentages associated with them. Indeed, the EMD measures how different two

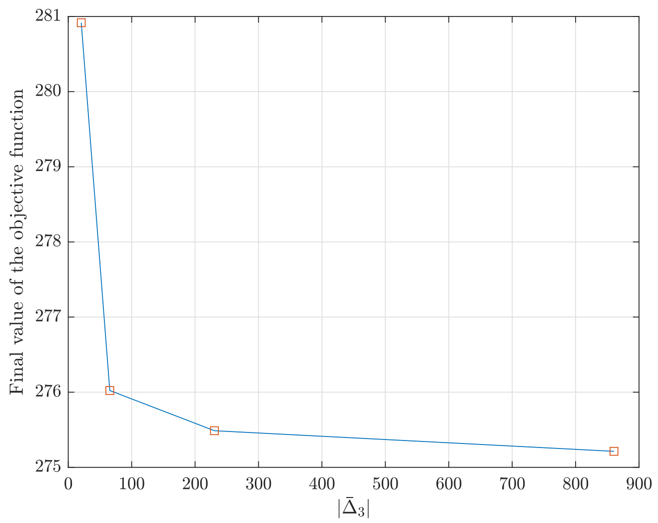


Figure C.1: Optimal values of the approximated problem by increasing the cardinality of $\bar{\Delta}_3$.

Table C.1: Solution (S_1) obtained by solving Problem (3.19) with discretization step $\delta^1 = 0.2$.

Choice Criteria	Percentage of users	Weights		
		distance	safety	practicability
$\mathbf{p}^1$	22.32%	0	0	1
$\mathbf{p}^2$	16.43%	0	1	0
$\mathbf{p}^3$	14.07%	1	0	0
$\mathbf{p}^4$	7.16%	0.2	0.6	0.2
$\mathbf{p}^5$	7.03%	0.6	0	0.4
$\mathbf{p}^6$	7.35%	0.2	0.8	0
$\mathbf{p}^7$	8.82%	0	0.4	0.6
$\mathbf{p}^8$	5.88%	0.4	0	0.6
$\mathbf{p}^9$	10.93%	0.4	0.6	0

Table C.2: Solution (S_2) obtained by solving Problem (3.19) with discretization step $\delta^2 = 0.1$

Choice Criteria	Percentage of users	Weights		
		distance	safety	practicability
$\mathbf{p}^1$	20.24%	0	0	1
$\mathbf{p}^2$	16.02%	0	1	0
$\mathbf{p}^3$	11.21%	1	0	0
$\mathbf{p}^4$	11.40%	0	0.7	0.3
$\mathbf{p}^5$	8.46%	0.7	0	0.3
$\mathbf{p}^6$	12.11%	0.3	0.7	0
$\mathbf{p}^7$	7.88%	0.1	0.1	0.8
$\mathbf{p}^8$	7.37%	0	0.4	0.6
$\mathbf{p}^9$	5.32%	0.8	0.2	0

Table C.3: Solution (S_3) obtained by solving Problem (3.19) with discretization step $\delta^3 = 0.05$

Choice Criteria	Percentage of users	Weights		
		distance	safety	practicability
$\mathbf{p}^1$	20.80%	0	0	1
$\mathbf{p}^2$	14.83%	0	1	0
$\mathbf{p}^3$	12.84%	1	0	0
$\mathbf{p}^4$	10.46%	0	0.7	0.3
$\mathbf{p}^5$	7.96%	0.7	0	0.3
$\mathbf{p}^6$	11.49%	0.25	0.75	0
$\mathbf{p}^7$	7.77%	0.15	0.1	0.75
$\mathbf{p}^8$	7.57%	0	0.4	0.6
$\mathbf{p}^9$	6.29%	0.55	0.45	0

Table C.4: Solution (S_4) obtained by solving Problem (3.19) with discretization step $\delta^4 = 0.025$

Choice Criteria	Percentage of users	Weights		
		distance	safety	practicability
$\mathbf{p}^1$	20.30%	0	0	1
$\mathbf{p}^2$	14.90%	0	1	0
$\mathbf{p}^3$	12.85%	1	0	0
$\mathbf{p}^4$	10.34%	0	0.7	0.3
$\mathbf{p}^5$	7.90%	0.7	0	0.3
$\mathbf{p}^6$	11.37%	0.25	0.75	0
$\mathbf{p}^7$	7.90%	0.175	0.075	0.75
$\mathbf{p}^8$	7.84%	0	0.4	0.6
$\mathbf{p}^9$	6.62%	0.525	0.475	0

solutions are by calculating the minimum “cost” of transforming one into the other, where the cost is the product between the fraction of users that needs to be moved and the distance by which they are moved.

As can be seen in Table C.5, the largest distance is between the first and the last solution, and even then it remains below 0.1. The last two solutions are almost identical. These values allow us to conclude that, when varying the discretization step from coarse to fine, the solutions exhibit good stability, as they do not change in any significant way.

Table C.5: Lower-triangular display of Earth Mover's Distances between solutions S_1, S_2, S_3, and S_4. Upper-triangle and diagonal cells are masked in black.

$\mathrm{EMD}(\cdot,\cdot)$	S_1	S_2	S_3
S_1			
S_2	0.1156		
S_3	0.0988	0.0507	
S_4	0.0952	0.0523	0.0096

Appendix D

Choosing the best branching strategy for CRM-BB

This appendix compares all branching strategies for CRM-BB. Table D.1 reports the average number of nodes, computed lower bounds, and computational time for each strategy, tested with all node selection criteria and using the randomly generated instances with grid sizes of up to 16. The table also provides scores for each strategy, which are used to identify the best one.

The scores are calculated as follows. Let $\Upsilon = \{\text{BCB, CRB, CRTB, IBB, U}\}$ denote the set of all branching strategies. For each strategy $\upsilon \in \Upsilon$, let $\overline{\overline{\mu}}_{\text{node}}(\upsilon)$, $\overline{\overline{\mu}}_{\text{lbs}}(\upsilon)$, and $\overline{\overline{\mu}}_{\text{time}}(\upsilon)$ represent the average of the averages for the number of nodes, computed lower bounds, and computational times, respectively. These values are reported in the *Averages* row of Table D.1.

The score function we use takes into account $\overline{\overline{\mu}}_{\text{lbs}}(\upsilon)$ and $\overline{\overline{\mu}}_{\text{time}}(\upsilon)$. Note that $\overline{\overline{\mu}}_{\text{node}}(\upsilon)$ is not considered in the score computation, since the number of computed lower bounds is approximately half of the number of nodes for binary branchings (BCB, CRB, CRTB, and IBB) and equal to the number of nodes for non-binary branching (U). Hence, the number of nodes is already implicitly accounted for through the number of computed lower bounds. Because these quantities are on different

Table D.1: Comparison of the branching strategies.

Instances		**CRM-BB-BCB**			**CRM-BB-CRB**			**CRM-BB-CRTB**		
Grid size	Node selection	Nodes	LBs	Time (s)	Nodes	LBs	Time (s)	Nodes	LBs	Time (s)
4	BFS	106782.481	53391.741	2.730	106937.370	53469.185	2.720	107561.593	53781.296	2.736
	DFS	103747.667	51874.333	2.548	104549.000	52275.000	2.558	106073.593	53037.296	2.612
	BBS	102895.222	51448.111	2.832	102945.222	51473.111	2.861	104160.333	52080.667	2.895
8	BFS	195035.148	97518.074	115.675	194016.333	97008.667	116.054	194620.407	97310.704	116.024
	DFS	193132.778	96566.889	112.853	192356.852	96178.926	112.492	193049.667	96525.333	112.612
	BBS	191841.296	95921.148	113.093	190858.481	95429.741	113.944	191445.296	95723.148	114.217
16	BFS	183232.111	91616.556	1645.864	183335.741	91668.370	1647.404	183558.333	91779.667	1646.892
	DFS	182942.333	91471.667	1627.567	183101.148	91551.074	1638.179	182980.556	91490.778	1629.655
	BBS	181165.000	90583.000	1592.114	181113.148	90557.074	1618.065	181194.704	90597.852	1615.358
Averages		**160086.004**	**80043.502**	**579.475**	**159912.588**	**79956.794**	**583.809**	**160516.054**	**80258.527**	**582.556**
Scores			**0.1862**			**0.1867**			**0.1869**	

Instances		**CRM-BB-IIB**			**CRM-BB-U**		–//–	–//–	–//–	–//–
Grid size	Node selection	Nodes	LBs	Time (s)	Nodes/LBs	Time (s)	–//–	–//–	–//–	–//–
4	BFS	105091.963	52546.481	3.249	68136.963	3.575	–//–	–//–	–//–	–//–
	DFS	104226.407	52113.704	3.080	66943.074	3.430	–//–	–//–	–//–	–//–
	BBS	100963.444	50482.222	3.336	65754.778	3.610	–//–	–//–	–//–	–//–
8	BFS	193879.000	96940.000	139.923	130691.370	152.273	–//–	–//–	–//–	–//–
	DFS	193076.704	96538.852	137.081	130092.259	153.196	–//–	–//–	–//–	–//–
	BBS	190455.593	95228.296	138.027	128910.296	150.563	–//–	–//–	–//–	–//–
16	BFS	183327.593	91664.296	1992.497	112634.741	2009.367	–//–	–//–	–//–	–//–
	DFS	183147.963	91574.481	1974.033	111669.074	1988.312	–//–	–//–	–//–	–//–
	BBS	181002.037	90501.519	1955.713	110767.370	1974.200	–//–	–//–	–//–	–//–
Averages		**159463.412**	**79732.206**	**705.215**	**102844.436**	**715.392**	–//–	–//–	–//–	–//–
Scores			**0.2056**		**0.2346**		–//–	–//–	–//–	–//–

scales, we normalize them such that

$$\sum_{\upsilon \in \Upsilon} \overline{\overline{\mu}}_{\text{lbs}}(\upsilon) = \sum_{\upsilon \in \Upsilon} \overline{\overline{\mu}}_{\text{time}}(\upsilon).$$

After normalization, the score for each $\upsilon \in \Upsilon$ is calculated as

$$\text{score}(\upsilon) = 0.5\overline{\overline{\mu}}_{\text{lbs}}(\upsilon) + 0.5\overline{\overline{\mu}}_{\text{time}}(\upsilon).$$

The resulting scores are reported in the *Scores* row of Table D.1. Since BCB has the lowest score, it is selected as the preferred branching strategy.

Appendix E

Warm start for CRM-BB

This appendix discusses the influence of the initial upper bounds provided by the heuristics on the performance of the proposed branch-and-bound algorithm (CRM-BB). Experiments with and without these bounds were conducted using all randomly generated instances and considering all node selection strategies (BFS, DFS, and BBS). As previously shown, BCB outperformed the other branching strategies and was therefore the only one adopted in these tests. In addition, the time limit of 16 hours (57600 seconds) was also considered. The results of these experiments are presented in Table E.1, which reports the average number of nodes and the average computational time for each instance group.

From Table E.1, we observe that the differences in the number of nodes and computational time, with or without the use of initial upper bounds, are not significant. This suggests that CRM-BB is not particularly sensitive to warm starts provided by the heuristics, indicating that the method is capable of performing well even when no promising initial upper bound is available. In particular, for BBS, all instances solved to optimality with a warm start (those with grid sizes up to 16, as well as those with grid sizes of 32 or 40 and up to 15 interventions) were also solved with the same number of nodes when the initial upper bounds were not provided.

Table E.1: Comparison of the node selection strategies with and without (*) warm start, considering BCB as branching strategy.

Instances		CRM-BB (BFS)		CRM-BB (BFS)*		CRM-BB (DFS)		CRM-BB (DFS)*		CRM-BB (BBS)		CRM-BB (BBS)*	
$\|V\|/\|A\|/\|W\|/\|K\|$	r	Nodes	Time (s)	Nodes	Time (s)	Nodes	Time (s)	Nodes	Time (s)	Nodes	Time (s)	Nodes	Time (s)
16/48/10/10	3	477.000	0.014	518.333	0.016	474.333	0.014	539.000	0.015	468.333	0.014	468.333	0.013
	4	477.667	0.014	543.667	0.015	474.333	0.013	537.667	0.015	471.667	0.014	471.667	0.013
	5	277.667	0.008	340.333	0.010	265.000	0.007	397.000	0.011	251.000	0.008	251.000	0.007
16/48/10/15	3	26130.333	0.700	31093.000	0.823	26197.667	0.677	28306.333	0.716	25861.000	0.736	25861.000	0.735
	4	8751.667	0.218	10652.333	0.259	8753.000	0.209	9472.333	0.227	8751.000	0.223	8751.000	0.224
	5	17867.000	0.464	19832.333	0.506	18209.667	0.460	22884.333	0.574	17217.000	0.462	17217.000	0.459
16/48/10/20	3	432913.667	9.926	486790.333	11.005	416671.000	9.236	423613.667	9.286	411047.667	10.256	411047.667	10.278
	4	36392.333	0.952	45663.667	1.170	34210.333	0.861	44615.000	1.114	34106.333	0.946	34106.333	0.948
	5	437755.000	12.276	462354.333	12.725	428473.667	11.452	431802.333	11.302	427883.000	12.826	427883.000	12.818
64/224/39/10	3	587.667	0.378	629.667	0.396	589.667	0.378	639.000	0.401	582.333	0.369	582.333	0.370
	4	585.667	0.307	613.000	0.314	572.333	0.298	671.000	0.345	564.333	0.291	564.333	0.290
	5	707.000	0.404	743.000	0.419	707.000	0.406	789.667	0.446	707.000	0.400	707.000	0.400
64/224/39/15	3	23759.000	12.640	25396.333	13.367	23795.667	12.453	24904.333	12.866	23660.333	12.299	23660.333	12.267
	4	18774.333	10.856	19261.000	10.998	18581.667	10.639	18829.000	10.560	18179.000	10.261	18179.000	10.278
	5	27325.667	15.255	27905.667	15.360	26977.000	14.811	27444.333	14.818	26493.667	14.537	26493.667	14.497
64/224/39/20	3	414265.000	247.859	431601.667	255.239	412936.333	243.312	420835.000	244.576	412525.667	246.247	412525.667	244.882
	4	916245.667	551.597	941450.333	562.627	909981.667	539.624	917907.000	540.172	902769.000	539.468	902769.000	539.386
	5	353066.333	201.775	373401.000	210.961	344053.667	193.754	350041.000	195.773	341090.333	193.964	341090.333	193.835
256/960/154/10	3	711.000	6.539	749.667	6.778	708.333	6.514	773.000	6.998	673.000	6.094	673.000	6.074
	4	545.000	4.959	565.000	5.041	548.333	5.016	600.333	5.343	531.667	4.744	531.667	4.761
	5	1334.333	12.396	1417.000	12.896	1335.000	12.361	1421.000	12.900	1325.667	12.058	1325.667	12.044
256/960/154/15	3	32658.333	307.211	33057.000	306.608	32449.000	303.524	33231.667	304.706	32056.333	294.217	32056.333	294.161
	4	7574.333	69.221	7949.000	71.323	7577.667	69.039	8725.000	78.072	7572.333	67.973	7572.333	67.955
	5	23395.667	215.737	24673.667	224.258	23195.000	213.483	23697.000	214.258	23141.667	209.351	23141.667	209.497
256/960/154/20	3	656032.333	5944.437	659810.333	5894.824	657719.000	5893.551	687070.333	6113.262	653032.333	5770.820	653032.333	5762.226
	4	77534.333	702.579	81555.000	729.967	75995.000	686.385	77920.333	691.400	75165.667	670.239	75165.667	669.212
	5	849303.667	7549.701	859338.333	7528.233	846953.667	7458.227	873575.667	7559.832	836986.333	7293.532	836986.333	7277.394
1024/3968/615/10	3	1564.333	245.835	1607.667	247.128	1564.333	245.169	1618.333	248.882	1564.333	240.400	1564.333	240.888
	4	780.333	120.072	814.333	122.558	779.667	119.602	852.333	127.968	779.667	117.170	779.667	117.224
	5	1096.333	163.848	1139.667	167.163	1096.333	164.270	1191.000	174.402	1096.333	160.838	1096.333	160.475
1024/3968/615/15	3	29936.333	4581.731	30198.333	4529.294	29787.000	4544.019	30232.333	4528.930	29629.000	4433.812	29629.000	4431.184
	4	18331.000	2793.548	18954.333	2835.536	18323.667	2784.835	19535.667	2912.890	18316.333	2732.203	18316.333	2734.728
	5	50276.333	7491.121	50403.667	7395.504	50165.000	7453.703	50694.333	7442.606	50034.333	7290.507	50034.333	7295.645
1024/3968/615/20	3	374317.000	57600.000	380824.333	57600.000	376726.333	57600.000	383251.667	57600.000	381083.000	57600.000	381210.333	57600.000
	4	280094.333	43606.623	286988.333	43773.444	282121.667	43589.087	290606.333	44012.792	285074.333	43387.748	284861.667	43400.644
	5	296301.000	43702.481	300598.333	43627.218	298189.000	43749.261	302702.333	43745.951	301625.667	43544.516	300953.667	43536.500
1600/6240/960/10	3	886.333	349.107	901.000	350.831	887.000	349.710	947.000	368.287	881.667	339.983	881.667	340.251
	4	1305.000	500.781	1339.000	505.820	1304.333	500.359	1375.000	522.649	1304.333	490.576	1304.333	491.160
	5	730.333	282.558	782.333	303.181	730.333	282.835	813.667	307.789	730.333	276.314	730.333	276.230
1600/6240/960/15	3	48735.667	18630.853	48941.667	18330.793	48951.667	18654.874	49919.667	18631.621	48346.333	18046.156	48346.333	18042.109
	4	32162.333	12325.981	32604.333	12222.586	32125.667	12293.713	33572.333	12559.033	32008.333	11992.578	32008.333	11971.408
	5	20498.333	7810.910	21311.000	7928.243	20497.667	7783.660	21514.333	7984.909	20410.333	7583.038	20410.333	7583.223
1600/6240/960/20	3	148269.000	57600.000	151631.667	57600.000	149234.333	57600.000	152225.000	57600.000	151676.333	57600.000	151451.667	57600.000
	4	150953.667	57600.000	153872.333	57600.000	151409.000	57600.000	154535.667	57600.000	154339.667	57600.000	153861.667	57600.000
	5	151700.333	57600.000	155280.333	57600.000	152698.333	57600.000	151409.667	57600.000	154983.667	57600.000	154928.333	57600.000
Averages		**132741.904**	**8641.864**	**137468.837**	**8635.899**	**131888.807**	**8635.596**	**135071.978**	**8666.638**	**131488.837**	**8586.849**	**131455.163**	**8585.927**

Appendix F

Parma dataset: Intervention set and cost reductions

This appendix describes how the set of interventions was determined, as well as how the cost reductions were calculated. For the intervention set, recall that we considered the set J of possible intervention types (see Table 4.11). Each intervention type $\iota \in J$ is associated with a list L_ι of candidate arcs, from which subsets of arcs affected by each intervention are selected. Since, in practice, interventions are not applied to all candidate arcs, we follow the procedure described in Algorithm 9, which generates the intervention set K and the corresponding arc sets A_k. The resulting set K comprises different intervention types and may include multiple interventions of the same type.

For each OD pair $w = (o_w, d_w) \in W$, we compute the shortest path from o_w to d_w using distance as the minimization criterion (lines 3–4). Based on these paths, we calculate the frequency of each arc $a \in A$ (lines 5–6). The motivation for using only distance as a criterion is that, if all three criteria (distance, safety, and practicability) were considered, unsafe arcs and arcs with low practicability would hardly be selected due to their low frequencies. However, such arcs are precisely those that should be prioritized to improve the overall quality of the cycling network. In a hypothetical scenario in which the safety and practicability scores are uniformly at their

Algorithm 9: Generation of the intervention set K and the corresponding arc sets A_k.

Initialize arc frequency $\nu(a) \leftarrow 0$ for all $a \in A$;
$K \leftarrow \{\}$;
for *each OD pair* $w = (o_w, d_w) \in W$ **do**
  Compute the shortest path from o_w to d_w ;
  for *each arc a on the path* **do**
    $\nu(a) \leftarrow \nu(a) + 1$;
Compute $\nu_{\max} \leftarrow \max_{a \in A} \nu(a)$;
for *each intervention type* $\iota \in J$ **do**
  $\tilde{L}_\iota \leftarrow \{a \in L_\iota \mid \nu(a) \geq \lceil 0.4\, \nu_{\max} \rceil\}$;
  for *each arc* $\tilde{a} = (i, j) \in \tilde{L}_\iota$ **do**
    Create a new intervention k, $K \leftarrow K \cup \{k\}$;
    Initialize $A_k \leftarrow \{\tilde{a}\}$;
    $\mathscr{S} \leftarrow \{i\} \cup \mathscr{N}(i) \cup \mathscr{N}_2(i)$;
    for *each arc* $a = (u, v) \in L_\iota$ *with* $u \in \mathscr{S}$ *and not in any* A_k *of type* ι **do**
      $A_k \leftarrow A_k \cup \{a\}$;
  Let $M_\iota \leftarrow \max\{|A_k| : k$ associated with type $\iota\}$;
  for *each intervention k associated with type* ι **do**
    if $|A_k| < \lceil 0.4 M_\iota \rceil$ **then**
      Remove intervention k, $K \leftarrow K \setminus \{k\}$;

highest possible values, route choice would be based only on distance. Hence, we assume this hypothetical scenario in order to improve the safety and practicability scores of arcs that are close to those identified as relevant when considering only distance minimization, as described in what follows.

For each intervention type $\iota \in J$, we consider only the candidate arcs $a \in L_\iota$ whose frequency is greater than or equal to 40% of the maximum arc frequency among all arcs, forming the set $\tilde{L}_\iota$ (lines 7–9). For each $\tilde{a} = (i, j) \in \tilde{L}_\iota$, an intervention $k \in K$ of type ι is created, and the corresponding arc set A_k is initialized with $\tilde{a}$ (lines 10–12). Let $\mathcal{N}(i) = \{j \in V \mid (i, j) \in A\}$ denote the set of nodes directly connected to node i, referred to as first-order neighbors, and let $\mathcal{N}_2(i) = \cup_{j \in \mathcal{N}(i)} \mathcal{N}(j)$ denote the set of second-order neighbors of node i, that is, the nodes reachable from i through a path of length two. To select additional arcs for each set A_k, the set associated with the arc $\tilde{a} = (i, j) \in \tilde{L}_\iota$ is augmented with each arc $(u, v) \in L_\iota$ such that $u \in \{i\} \cup \mathcal{N}(i) \cup \mathcal{N}_2(i)$ and that does not belong to any other arc set A_k associated with an intervention of type ι (lines 13–15). Finally, we exclude any subset A_k, and consequently the corresponding intervention k, whose size is less than 40% of the size of the largest subset associated with the same intervention type (lines 16–19).

All interventions are summarized in Table F.1, which lists the considered intervention types, the indices k of interventions associated with each type ι, and the maximum size M_ι among all subsets A_k corresponding to ι.

Table F.1: Description of the considered interventions.

Intervention type		**Intervention** (k)	**Max size** (M_ι)
ι	Description		
1	Add a bike crossing (ABC)	[1,...,9]	12
2	Add protective elements (APE)	[10,...,16]	10
3	Build a bike lane (BBL)	[17,...,31]	18
4	Build a bike path (BBP)	[32,...,42]	18
5	Improvement pavement maintenance (IPM)	[43,...,44]	9
6	Install markings (IM)	[45]	17
7	Install signage (IS)	[46,...,47]	17
8	Modify lane dimensions and remove bottlenecks (MDB)	[48,...,51]	10
9	Modify pavement material (MPM)	[52,...,55]	9
10	Remove protrusions (RP)	[56,...,59]	10

For the cost reductions, Table F.2 presents the factors associated with the safety and practicability criteria that are affected by each intervention type ι. Note that a key performance indicator (KPI) value is associated with each factor. These values are integers ranging between -2 and 2. The effect of an intervention k of type ι, when applied, is to increase the corresponding KPI for all arcs $a \in A_k$ if the intervention affects that factor.

Table F.2: Factors associated with the selected road features (safety and practicability) affected by each intervention type.

Safety factors	affected by **Intervention types** (ι)
Protection from vehicular traffic	{1, 2, 3, 4}
Signage	{1, 7}
Markings	{1, 6}
Practicability factors	affected by **Intervention types** (ι)
Dimension and Bottlenecks ($\geq$ 2.50 m)	{8}
Dimension and Bottlenecks ($\geq$ 1.50 m and $<$ 2.50 m)	{8}
Dimension and Bottlenecks ($<$ 1.50 m)	{8}
Pavement maintenance	{5}
Pavement material	{9}
Protrusions	{10}

Let $\mathscr{C}_{ij}^f$ denote the original (i.e., before the application of any intervention) KPI value for factor f on arc (i, j). The total KPI, C_{ij}^h, on arc (i, j) associated with road feature h is calculated as

$$C_{ij}^h = \sum_{f \in \mathscr{F}_h} \mathscr{C}_{ij}^f,$$

where $\mathbb{F}_h$ represents the set of factors associated with road feature $h \in \{1, \ldots, r\}$. The cost c_{ij}^h is therefore defined as:

$$c_{ij}^h = 1 - \frac{C_{ij}^h}{C_{\max}^h - C_{\min}^h},$$

where $C^h_{\max} = \max\{C^h_{ij}\}$ and $C^h_{\min} = \min\{C^h_{ij}\}$ are the maximum and minimum possible KPI values that can be assigned to road feature h, respectively (see Chapter 3 for more details).

As described in Table F.2, a intervention k of type ι affects one or more factors $f \in F_h$ associated with road feature h, increasing the corresponding KPI values. Hence, let $\mathscr{C}^{fk}_{ij} \geq \mathscr{C}^{f}_{ij}$ denote the KPI value after the application of intervention $k \in K$ for factor $f \in \mathbb{F}_h$ on arc (i, j) (equality only holds if intervention k does not affect factor f). The KPI variation on arc (i, j) associated with intervention k for road feature h is calculated as

$$\bar{\phi}^h_{ijk} = \sum_{f \in \mathbb{F}_h} \left(\mathscr{C}^{fk}_{ij} - \mathscr{C}^{f}_{ij} \right).$$

However, as the cost reductions ϕ^h_{ijk}, are related to the cost values, not directly to KPI values, they must be computed as

$$\phi^h_{ijk} = \sum_{f \in \mathbb{F}_h} \left(\frac{\mathscr{C}^{fk}_{ij} - \mathscr{C}^{f}_{ij}}{C^h_{\max} - C^h_{\min}} \right).$$

It is worth mentioning that for each arc (i, j) and factor f, only the intervention leading to the maximum KPI improvement is considered. This choice not only contributes to guarantee non-negativity of the resulting basic costs, but also reflects practical implementation constraints in the scenario corresponding to the Parma instances, as multiple interventions affecting the same factor on the same arc are not cumulatively effective.

Acknowledgments

I start my acknowledgments by thanking God for granting me the opportunity to pursue this PhD, a goal I had long awaited and deeply desired.

I would like to thank my wife, Ana Luiza, for her love, support, and understanding, and above all, for being such an incredible partner, even when we were geographically far apart.

My heartfelt thanks go to my parents, Marcelo and Christiane, for their unconditional support. I also include my brothers, Matheus and Miguel, my sisters-in-law, grandparents, uncles, and all members of my family. I extend this gratitude to my wife's relatives, who are also part of my family.

I would like to express my deepest gratitude to my supervisors, Professors Marco Locatelli and Anand Subramanian, for their invaluable guidance, continuous support, and encouragement throughout my PhD journey.

I am also grateful to Professors Manuel Iori and Luca Consolini for their collaboration and assistance in my research.

My thanks extend to my colleagues and friends from the Department of Engineering and Architecture at UNIPR and from DISM at Unimore, who created a friendly and collaborative environment. Among them, special thanks to Stefano and Mattia for their valuable collaboration in my research.

My sincere thanks also go to my friends from the Logistics and Optimization Group at UFPB; especially Anand, Teobaldo, Bruno, Luciano, Igor, Vitor, Yure, and Eduardo; for encouraging me to pursue my PhD from the very beginning. Thank you so much.

I am also grateful to all my Brazilian friends who have accompanied me through every step of my life. Among them, I would like to mention my dear friend Matheus Amorim, who visited me and shared many wonderful moments with me in Italy, as well as Victor and Heloisa, whom I had the pleasure of meeting in Italy and sharing very special moments with.

I would also like to thank Tiago Silveira, a former Brazilian PhD student from UNIPR, for his invaluable help during my PhD.

A special note of gratitude goes to Tatiana Ramalho, my friend and English teacher, who prepared me for this journey abroad.

Finally, I would like to thank everyone who supported me during this intense and challenging period of my life. I am deeply grateful to all of you.