Extending Accessibility Analysis With True Multi-Modality

Master Thesis

Author: Moritz Gottschling (Student ID: 7350270)
Supervisor: Univ.-Prof. Dr. Wolfgang Ketter
Co-Supervisor: Philipp Peter

Department of Information Systems for Sustainable Society
Faculty of Management, Economics and Social Sciences
University of Cologne

November 14, 2023

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne die Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden. Ich versichere, dass die eingereichte elektronische Fassung der eingereichten Druckfassung vollständig entspricht.

Die Strafbarkeit einer falschen eidesstattlichen Versicherung ist mir bekannt, namentlich die Strafandrohung gemäß § 156 StGB bis zu drei Jahren Freiheitsstrafe oder Geldstrafe bei vorsätzlicher Begehung der Tat bzw. gemäß § 161 Abs. 1 StGB bis zu einem Jahr Freiheitsstrafe oder Geldstrafe bei fahrlässiger Begehung.

Moritz Gottschling

Köln, den xx.xx.20xx

Abstract

[Abstract goes here (max. 1 page)]

Contents

1 Introduction **1**

2 Related Work **3**
2.1 Accessibility Analysis . . . 3
2.2 Routing Algorithms . . . 6
2.2.1 Dijkstra . . . 7
2.2.2 MLC . . . 7
2.2.3 Graph-based Algorithms in Public Transport . . . 8
2.2.4 RAPTOR . . . 8
2.2.5 McRAPTOR . . . 12
2.2.6 MCR . . . 12
2.2.7 ULTRA . . . 13
2.2.8 McTB . . . 14
2.2.9 ULTRA-PHAST . . . 14
2.3 Public Transport Data . . . 14
2.4 Street Network Data . . . 14
2.5 Importance of Multimodality & Intermodality . . . 15

3 Method **17**
3.1 Metric . . . 17
3.2 Routing Algorithm . . . 19
3.2.1 Requirements . . . 19
3.2.2 Algorithm . . . 20
3.2.3 Modules . . . 21
3.2.4 Merging . . . 22
3.2.5 Enhanced MLC & McRAPTOR . . . 23
3.2.6 Example . . . 24
3.3 Accessibility Analysis Tool . . . 25

4 Experiment **29**
4.1 Scenarios . . . 29
4.2 Data . . . 30
4.2.1 Street Network & POIs . . . 31
4.2.2 Public Transport . . . 31
4.2.3 Bicycle Sharing . . . 32
4.2.4 Land Use . . . 33
4.3 Assumptions . . . 33
4.3.1 Pricing . . . 33

A Appendix - Overpass Query for Boundary of Cologne **35**

B Appendix - Pyrosm Network Filter **35**

C Appendix - Experiment Module Matrix Configuration **36**

References **37**

List of Figures

1 Iterating a route in RAPTOR 11
2 Modular Routing Algorithm 21
3 Example Repeating Module Matrix 24
4 Input Routine 25
5 Main Routine 26
6 Metric Routine 27
7 Metric Co-Routine 27

List of Tables

1 Categories of Accessibility-Based Planning Measures 3
2 Categories and their corresponding OSM tags 18
3 Scenarios for Urban Mobility Analysis 30
4 Speeds for different modes of transport 33
5 Driving Filter 35
6 Walking Filter 35
7 Cycling Filter 35

1 Introduction

Climate change poses a significant threat to our planet, and reducing emissions is crucial in addressing this global challenge. The Paris Agreement, with its ambitious goals to reduce global warming, serves as a call to action. However, current trends suggest that without immediate action to reduce emissions, achieving these goals is unlikely (Kriegler et al., 2018; Liu & Raftery, 2021). 61.8% of global emissions in 2015 came from cities, and predictions for 2100 estimate the share to exceed 80% by 2100 (Gurney et al., 2021). The connection between the large share of city emissions and climate change has led to a critical examination of urban planning and transportation. In response to the urgent need to reduce emissions, the concept of emission-free cities has emerged as a pivotal strategy. Emission-free cities aim to create sustainable environments that minimize the carbon footprint, promote the health of residents, and align with the global efforts to mitigate climate change. Transitioning to these cities is a proactive step toward sustainable living and securing a healthier future for our urban spaces. With vehicles on the roads accounting for 72% of all transport-related emissions (Sims et al., 2014), it is clear that urban transportation is a key area to reduce emissions.

In order to reduce the amount of car traffic, cities need to be planned in a way that allows people to access everything they need with alternative, more environmentally sustainable modes of transport. Therefore, a recent trend in urban planning is accessibility-based planning (Proffitt, Bartholomew, Ewing, & Miller, 2019) (Geurs & van Wee, 2004). In order for practitioners to be able to plan cities in an accessibility-based way, they need to be able to measure accessibility.

A modern way of measuring accessibility is to use the X-minute city metric, which is inspired by the 15-minute city concept, which recently gained traction during the COVID pandemic (Moreno, Allam, Chabaud, Gall, & Pratlong, 2021).

The concept of the 15-minute city is that all the things a person needs to live a good life should be accessible within 15 minutes of walking or cycling. Traditionally, the modes of transport don't include public transportation or vehicle sharing systems.

However, we argue that in order fully grasp the potential of sustainable transport, it's essential to incorporate all modes, not merely a subset, in the measurement of accessibility

We therefore develop a new tool for accessibility-based planning that incorporates all modes of travel (multi-modal, unrestricted inter-modal). In addition, this tool will use a metric based on the concept of the 15-minute city, extending

it to include all modes of transport and therefore presenting a more holistic view on accessibility via sustainable modes of transport.

We test our tool on the City of Cologne.

The remainder of this paper is structured as follows. In Section 2, we present related work on accessibility-based planning, the 15-minute city concept, and routing algorithms. Next, in Section 3, we describe the specifics of our tool, including the data collection process, the routing algorithm, and the accessibility metric. After that, we introduce our experiment in Section and it's results in Section. Finally, we conclude with a discussion and conclusion in Section and, respectively.

2 Related Work

...

2.1 Accessibility Analysis

Traditional mobility-based planning primarily focuses on reducing congestion and facilitating movement, often prioritizing automobile travel (Proffitt et al., 2019). However, this approach is becoming more and more out-of-date, as modern challenges like reducing emissions of greenhouse gases require a more holistic approach. This is where accessibility-based planning comes into play, which focuses planning cities in a way to provide residents with access to important services (Proffitt et al., 2019).

(Geurs & van Wee, 2004) describe four types of accessibility-based planning: (more info here...)

Table 1: Categories of Accessibility-Based Planning Measures

Category	Focus	Examples
Infrastructure-Based	- Traffic performance analysis - Service level of infrastructure	- Level of congestion - Average travel speed
Location-Based or Place-Based	- Level of accessibility to locations - w.r.t. time & cost	- number of jobs within 30 minutes
Person-Based	- Individual travel time	- Individual's travel time between activities
Utility-Based	- Economic benefits	- Transportation investments returns

The X-minute city metric is a location-based accessibility metric, which is derived from the concept of the 15-minute city.

The 15-minute city concept was first introduced by Carlos Moreno, a professor at the Sorbonne University in Paris (Moreno et al., 2021). It was popularized by the mayor of Paris, Anne Hidalgo, who made it a central part of her re-election campaign (Gongadze & Maassen, Wed, 01/25/2023 - 15:46).

The concept of the 15-minute city is that all the things a person needs to live a good life should be accessible within 15 minutes of walking or cycling.

cultivate stronger social relationsips, because people are more likely to meet (Allam, Moreno, Chabaud, & Pratlong, 2020) a kind of social distancing, because of primarily traveling by walking or cycling (Allam et al., 2020) a more environmentally sustainable mode of transport in cities, which contributes to SDG 11 &

13 (Allam et al., 2020) (Papas, Basbas, & Campisi, 2023)

reduced traffic resulting in economic benefits (Allam et al., 2020) (Papas et al., 2023)

developing cities around the 15-minute city concept reduces social inequalities, as walking is free (Weng et al., 2019) (Gustafson, 2022) therefore, when incorporating other fare-based modes of transport such as bicycle sharing, it is important to consider the fares and their impact (thats why we do multi-objective stuff)

Quantitative studies have developed various ways to measure how well cities match the idea of the 15-minute city concept, by developing metrics that encapsulate this principle. (Olivari, Cipriano, Napolitano, & Giovannini, 2023) contribute to this research by creating the NExt proXimity Index (NEXI), which has two components: the NEXI-Minutes and the NEXI-Global.

The NEXI-Minutes looks at the accessibility of various urban amenities, ranging from educational institutions to entertainment venues and grocery stores—by calculating the time needed to reach the closest facility within each category. Complementing this, the NEXI-Global, inspired by the Walk Score method (*Walk Score Methodology*, n.d.), combines these individual times through a weighted average into an overall score giving a holistic view of the accessibility of a city.

NEXI is unique because it is both global and local - globally applicable due to its reliance on OpenStreetMap data, yet sufficiently detailed to assess local conditions. This has been proven by applying it all over Italy, where it's showcased on an interactive map through a hexagonal grid that makes it easy to see which areas are doing well and need improvement. The significance of the NEXI, as underscored by (Olivari et al., 2023), is its role in enabling data-driven policy-making, to develop cities in the fashion of the 15-minute city framework. Therefore, their index enables accessibility-based planning.

Despite the potential benefits, (Olivari et al., 2023) acknowledge the challenges in realizing the 15-minute city model, notably the substantial investments and strategic planning required.

However, there are some limitations in the NEXI metrics. The NEXI-Minutes offers separate metrics for each category, which may lead to a fragmented understanding of urban accessibility. This multi-metric approach can make comprehensive evaluation challenging. In addition, the NEXI-Global, while aggregating these categories, introduces complexity through its weighted scoring system. The weights are hard for humans to evaluate and the score from 0 to 100 disconnects the metric from the intuitive meaning of minutes. This makes it more difficult for urban planners and policymakers to interpret and utilize the results effectively. These factors suggest a need for refinement in the NEXI methodology to enhance

its practical utility in urban development and planning.

Another study by (Nicoletti, Sirenko, & Verma, 2023) explores the connection between urban infrastructure and social inequality. The researchers developed an open, data-driven framework to analyze how different communities within cities access essential services. They discovered that access to urban amenities like healthcare, education, and transportation is not evenly distributed. In particular, communities with more minorities, lower incomes, and fewer university-educated individuals often have less access to these important services.

The study examined over 50 types of amenities across 54 cities worldwide and found a common pattern: in all cities, access to infrastructure followed a log-normal distribution, indicating that disparities in accessibility are linked to the city's growth. This pattern was consistent even when considering various socio-economic factors.

The framework introduced by (Nicoletti et al., 2023) is flexible and adaptable, allowing city planners to tailor it to local needs and priorities. It's a tool that can help identify which groups in a city are most affected by inequality in access to services.

Similar to (Olivari et al., 2023) they emphasize the role of open data to guide urban planning and policy. While (Olivari et al., 2023) simply provide a tool to measure accessibility and leave the analysis and interpretation to practitioners, (Nicoletti et al., 2023) directly reveal the disparities in accessibility and provide a framework to analyze them.

However, the computation of travel time by all previously named authors is based on walking simulations. This fails to capture the reality of urban mobility, which consists of various modes of transport, such as public transport, cycling, and driving. The potential benefits of infrastructure such as dedicated bicycle lanes, bicycle sharing systems, and highly available public transport are thus not reflected within the research of (Olivari et al., 2023) and (Nicoletti et al., 2023). To provide a more accurate measure, we propose to enhance the computation of accessibility by integrating a multimodal accessibility metric that accounts for these various transportation methods and their respective infrastructural elements.

In order to assess the accessibility from a given origin to one or multiple points of interest, a routing algorithm is required. The routing algorithm finds the shortest path from the origin to the destination. Therefore, we investigate different routing algorithms in the following section and find one that is suitable for our use case.

2.2 Routing Algorithms

The primary goal of routing algorithms is to identify the optimal path between a designated origin and a specific destination. Typically, this is captured using a graph representation:

$$G = (V, E)$$

where V represents a set of nodes (or locations) and E encapsulates the set of edges, which correspond to connections between these nodes.

For each edge $e \in E$, there's an associated weight $w(e) \in \mathbb{R}$ that characterizes the cost of traversing it. This cost might be determined by factors such as distance or travel time. Consequently, the shortest path can be expressed as:

$$\langle v_0, e_0, v_1, e_1, \ldots, v_n \rangle$$

Here, v_0 denotes the origin, v_n the destination, and the edges must connect the nodes in the sequence:

$$e_i = (v_i, v_{i+1}) \quad \text{for} \quad i \in \{0, \ldots, n-1\}$$

In accessibility contexts, the primary concern frequently revolves around determining the accumulated cost, $d(v_n)$, to reach the destination rather than the actual path.

In more complex real-world scenarios, the problem often encompasses multiple objectives, such as considering both time and monetary cost of travel. Under these circumstances, the edge weight is represented as a vector:

$$w(e) \in \mathbb{R}^k$$

where k stands for the total objectives count. Unlike the simpler single-objective case with a singular optimal path, the multi-objective scenario yields a Pareto set, constituting several optimal routes. A Pareto set refers to a set of solutions that are non-dominated by any other solution. This means that for each solution in the Pareto set, there is no other solution that is better in all objectives. The Pareto set represents an optimal trade-off among the different objectives, where improving one aspect would worsen another. For example, a Pareto set could contain multiple paths, where one path is faster, but more expensive, while another path is slower, but cheaper.

The value of these paths is depicted using a label:

$$l \in \mathbb{R}^k$$

where $l_i \in \mathbb{R}$ denotes the value for the i-th objective. This label can be thought of as a multidimensional extension of $d(v_n)$ from the single-objective scenario. The Pareto set associated with destination node v_n is often termed as a bag, expressed as $B(v_n)$, comprising labels that are not dominated by each other. Domination is defined as follows: l' dominates l if $l'_i \leq l_i$ for all $i \in \{1, \ldots, k\}$ and $l'_i < l_i$ for at least one $i \in \{1, \ldots, k\}$. Intuitively, this means that l' is at least as good as l in all objectives and strictly better in at least one objective.

The goal of routing algorithms used in accessibility analysis is finding the distance in the single objective case and the bag in the multi objective case. For accessibility analysis routing algorithms are often altered to not find the optimal path(s) between two nodes, referred to as one-to-one query, but the path from a single origin to all other nodes in the network, which we call one-to-all query.

2.2.1 Dijkstra

The most straightforward approach to compute the shortest paths in a graph is the Dijkstra algorithm (Dijkstra, 1959).

Dijkstra's algorithm initiates at a designated start node $s \in V$ and employs a priority queue to systematically determine the shortest path to each subsequent node $v \in V$. Initially, the distance to the start node s is set to zero, while the distances to all other nodes are set to infinity. In each iteration, the algorithm dequeues the node u with the smallest known distance from the priority queue. It then examines each outgoing edge $e = (u, v)$ from u, updating the distance to v if a shorter path through u is discovered. Specifically, if $\text{dist}(u) + w(e) < \text{dist}(v)$, then $\text{dist}(v)$ is updated to $\text{dist}(u) + w(e)$, and v is enqueued into the priority queue for future exploration. The node u is marked as visited by adding it to the set V_{visited}. Depending on the goal, the algorithm terminates either when the destination node is dequeued (one-to-one) or when the priority queue is empty (one-to-all).

However, this simple approach has multiple problems. Firstly, the Dijkstra algorithm is not able to handle multiple criteria. Secondly, the runtime of Dijkstra's algorithm is $O(|E| + |V| \log |V|)$, which is too slow for large graphs.

2.2.2 MLC

The Multi-Label-Correcting (MLC) (Hansen, 1980) algorithm is an extension of Dijkstra's algorithm to handle multi-objective scenarios. As mentioned in Section 2.2 in the multi-objective case we try to find the bag of the destination node. Specifically, for k criteria, each node v retains a bag of k-dimensional labels. Such a list encapsulates a set of Pareto-optimal paths from the starting node to

v. Similarly to Dijkstra's algorithm, MLC initializes all nodes with an empty bag, except for the start node, which is initialized with a label of $(0, \ldots, 0) \in \mathbb{R}^k$. Each iteration extracts the lexicographically smallest label, as opposed to selecting the node with the minimum distance. When a label is extracted and v is its corresponding node, updates are made for all connected edges (v, w). The update process consists of comparing a newly generated tentative label against all labels within the bag of w. This new label is only inserted into the bag if it isn't dominated by any existing label. Conversely, any label now dominated by the new entry is removed. Each time a label is inserted into a bag, it is also inserted into the priority queue. The algorithm terminates when the priority queue is empty.

The major drawback of the MLC algorithm is its runtime, which is even slower than Dijkstra's algorithm, because each node can be visited multiple times.

2.2.3 Graph-based Algorithms in Public Transport

In the context of accessibility analysis the previously mentioned algorithms can be used directly for walking, cycling and driving networks. However, public transport networks pose a challenge, since they contain time-dependent information, such as the departure time of a trip. To overcome this challenge two different approaches are commonly used, the time-expanded and the time-dependent approach, as explained by (Müller-Hannemann, Schulz, Wagner, & Zaroliagis, 2007). While enabling the use of graph-based algorithms, both approaches still suffer from the previously mentioned runtime problems Dijkstra's algorithm and MLC have.

2.2.4 RAPTOR

To overcome the runtime problems of graph-based approaches, (Delling, Pajor, & Werneck, 2015) introduce one of the most prominent routing algorithms for public transport, called Round based Public Transit Optimized Router algorithm (RAPTOR). Unlike traditional Dijkstra-based algorithms, RAPTOR operates in rounds, looking at each route (such as a bus line) in the network at most once per round.

As RAPTOR does not operate on a graph, we first introduce the problem statement. Raptor operates on a scheduled network consisting of routes r, trips t, stops p, and stop times that associate trips with stops. A route is associated with a sequence of stops $stops(r) = \langle p_1, \ldots, p_n \rangle$. A route has multiple trips ordered by their departure time $trips(r) = \langle t_1, \ldots, t_m \rangle$. One trip associates arrival and departure times with each stop of the route, denoted by $arrivalTime(t, s) \in \mathbb{N}$ and $departureTime(t, s) \in \mathbb{N}$ respectively. Trips of the same must not overtake

each other, formally:

$$departureTime(t_i, p_j) \leq arrivalTime(t_{i+1}, p_j)$$

for all $i \in \{1, \ldots, m-1\}$ and $j \in \{1, \ldots, n\}$. Each stop p has a minimal exchange time $\tau_{ch}(p) \in N$ associated with it. Often, the exchange time is set to a fixed time $\tau_{ch}(p) = \tau_{ch}$ for all stops p. When transferring from a trip t to another trip t' within at a stop p, the exchange time has to be smaller than the difference in arrival and departure time of the two trips, formally:

$$arrivalTime(t, p) + \tau_{ch}(p) \leq departureTime(t', p)$$

In addition to transfer within stops, RAPTOR also allows footpaths. Footpaths allow transferring from one stop to another without using public transport, therefore, they are time-independent. Each footpath is associated with a travel time $l(p, p')$. The input of the RAPTOR algorithm, in addition to the previously described scheduled network, are source stop p_s, and, in the case of a one-to-one query, target stop p_t, as well as, the departure time at the source stop τ.

RAPTOR operates in rounds. Before the first round, some variables are initialized. We denote the earliest possible arrival time at iteration i with $\tau_i(p)$ and the best earliest possible arrival time over the course of all iterations with $\tau^\star(p)$. For the source stop, τ_p, we set $\tau_0(p) = \tau$ and $\tau^\star(p) = \tau$. For all other stops, we set $\tau_0 = \infty$ and $\tau^\star = \infty$. In addition, we initialize a set of marked nodes M to only contain the source stop p_s and a set of marked route-stop pairs, denoted by Q, to the empty set. A route-stop pair is simply a tuple that contains a route and one of its stops. The set of marked stops will contain all stops whose earliest possible arrival time has been updated in the current round. Similarly, the set of marked route-stop pairs contains the routes of the marked stops, together with the earliest stop of that route that has been marked.

Each round consists of three major steps. In the first step, the routes that have to be iterated are collected. In the second step, the routes are iterated by "hopping" on their trips. And in the third stage, potential footpaths are explored.

First, we clear the set of marked route-stop pairs Q. Then we check the routes that are connected to each marked stop. For each of these routes, we store the route-stop pair in Q. However, the routes in Q should be unique. If there are two marked stops that are connected to the same route, we choose the stop that is earlier in the sequence of stops of that route. Now, we clear the set of marked stops.

We iterate the route-stop pairs in Q. The following step can be regarded as hopping on the earliest possible trip that we can catch of that route at that stop.

For each route-stop (r, p) pair, we iterate over the stops in r in the sequence that is associated with r, beginning with p. We check for the earliest possible trip that we can catch regarding the last arrival time at the current stop $\tau_{k-1}(p)$ and the minimum exchange time $\tau_{ch}(p)$. If there is a trip that is possible to catch, we save it as the current trip t_{curr} and continue to iterate the stops of the route r. Now that we are on a trip, we have to check whether we need to update the earliest possible arrival time of the current stop $\tau_k(p)$ and $\tau^\star(p)$ by comparing the stop time of the current trip with the best earliest arrival time of that stop $\tau^\star(p)$, formally:

$$\tau_k(p) = \min\{\tau_k(p), arrivalTime(t_{curr}, p)\}$$

Here one optimization comes into play. In the case an update is necessary, we also add the current stop p to the marked stops.

Lastly, we check all marked stops for potential footpaths. Remember: the marked stops are those for which the earliest possible arrival time was updated in this iteration. For each footpath that is connected to a marked stop, we check whether the earliest possible arrival time of the other stop could be improved by the footpath. If that is the case, we update the earliest arrival times and also mark that stop.

If no stops are marked, then there are no new routes to iterate, and the algorithm stops.

After termination

$$\tau_k(p)$$

contains the earliest possible arrival time at stop p with at most k transfers.

One limitation of RAPTOR is the transfer graph, which is used to represent footpaths. The transfer graph has to be transitively closed, which means that each node has to be connected with an edge to all other nodes that can be reached from that node. This has the advantage that in the algorithm we only have to check for direct neighbors of a stop, which is very fast. In practice, there are many possibilities how the transfer graph could look. First, we should note that a realistic transfer graph should be derived from a street network, as passengers should be able to walk from one stop to another using sidewalks. To keep the transfer graph small, one could limit the maximum walking distance. However, this may remove optimal journeys from the search space. In general, creating the transfer graph requires some amount of preprocessing. Therefore, finding a fitting transfer graph is challenging.

Through its round-based nature, RAPTOR is able to optimize for two criteria at the same time. However, RAPTOR cannot incorporate more criteria and one of the criteria will always be the number of transfers.

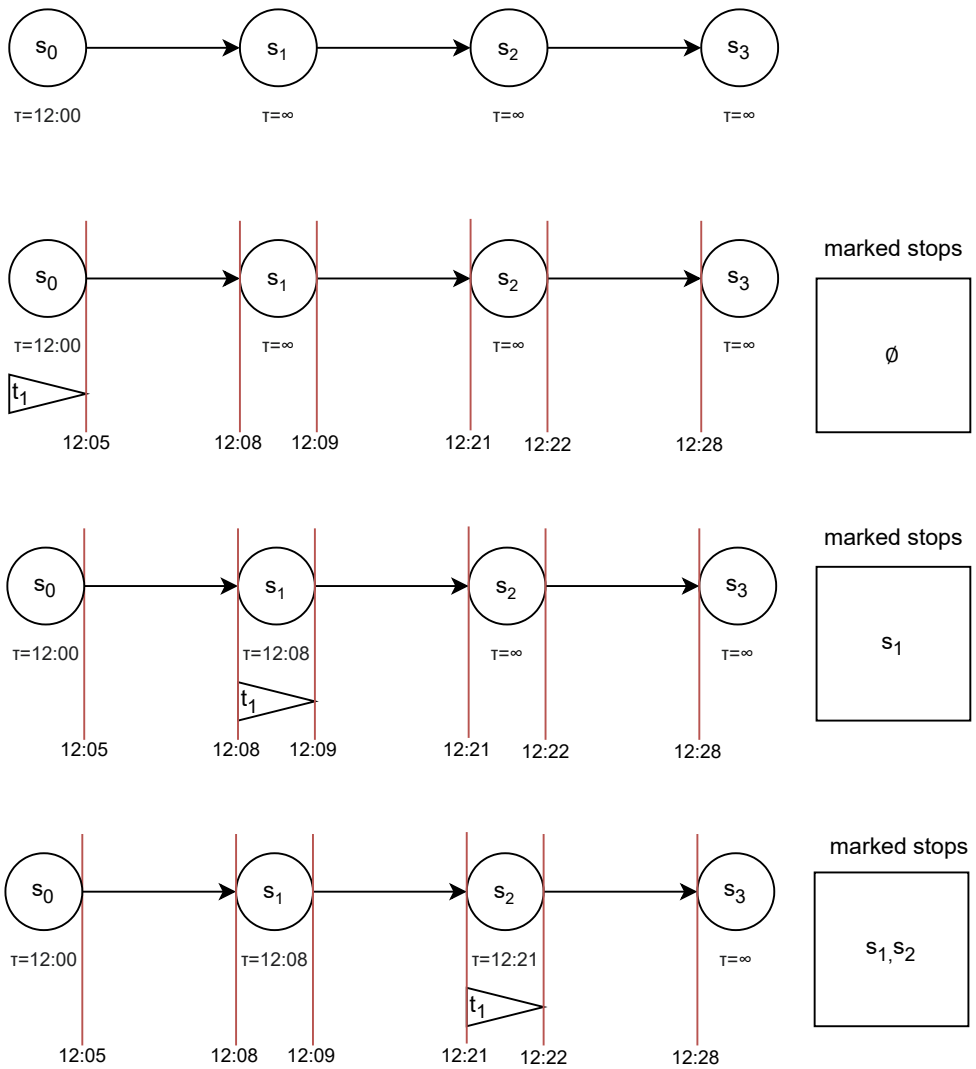

Figure 1: Iterating a route in RAPTOR

2.2.5 McRAPTOR

McRAPTOR (Delling et al., 2015) is an extension of RAPTOR that allows an arbitrary number of criteria. Like MLC, McRAPTOR also uses the notion of bags containing non-dominating labels. McRAPTOR does not pose any restrictions on how the objectives are updated during the algorithm.

The algorithm of McRAPTOR only requires slight modifications to the algorithm of RAPTOR. In the initialization step, each stop p is assigned an empty bag, except the source stop p_s, which is assigned a bag containing a starting label. The starting label can be defined as an input, but is usually $(\tau, 0, 0, \ldots, 0)$, where τ is the departure time at the source stop.

When iterating over the route-stop pairs $(r, p$, McRAPTOR creates a route bag that contains all labels that are in the current bag of p. In addition labels in the route bag are associated with a trip. During creation of the route bag, each label in the route bag is associated with the first trip that is possible to catch according to the labels earliest arrival time at the current stop p. Then the route is processed, stop by stop, just like in RAPTOR. At each stop the labels in the route bag are updated according to the current trip. This update must include updating the earliest arrival time, but can also include updates to other criteria. After the route has been processed, the route bag is merged into the bag of the current stop. Merging a bag B_1 into a bag B_2 means that all labels in B_1 that are not dominated by any label in B_2 are added to B_2 and all labels in B_2 that are dominated by a label in B_1 are removed from B_2. After the route bag has been merged into the bag of the current stop, the bag of the current stop is merged into the route bag. Lastly, the trips that are associated with the labels in the route bag are updated according to the labels earliest arrival time at the current stop.

Each time a label is added to a stop bag, this stop is marked. If no stop is marked after a round, the algorithm terminates.

Note that McRAPTOR allows updates to the route bags at any time during processing. When and how the route bag should be updated fully depends on the objective and what it represents.

While McRAPTOR has a slower runtime than RAPTOR it is still magnitudes faster than MLC. However, McRAPTOR still suffers from the same problem as RAPTOR, namely that the transfer graph is hard to compute.

2.2.6 MCR

To overcome problem of RAPTOR (Delling, Dibbelt, Pajor, Wagner, & Werneck, 2013) introduce Multimodal Multicriteria RAPTOR (MCR). MCR modi-

fies McRAPTOR so that the transfer graph must not be transitively closed. This enables MCR to directly use the street network as an input and therefore no preprocessing is necessary.

This allows us to use the street network as the transfer graph, which has the benefit that during the traversal of the tranfer graph the objectives can be updated. This is important if we want multiple modes of transfer, that contain free-floating vehicle sharing systems. For example, consider the following case. For an optimal journey a passenger has to first walk five minutes to a free-floating bicycle, with which the passenger then travels to the next stop. There is no way to represent this in RAPTOR, because the specifics of the transfer depend on the current label, which is unknown before running the algorithm. Therefore, it is not possible to precompute the transfer graph.

MCR is very similar to RAPTOR. It only replaces the footpath traversal step through MLC.

The authors find that the bottleneck of MCR is the MLC step. Therefore, they employ a technique called contraction (Geisberger et al., 2012) to speed up MLC. Contraction is a preprocessing technique that reduces the size of the graph by removing nodes and adding shortcut edges.

As previously mentioned, MCR is able to use the street network as the transfer graph and requires no preprocessing. However, when comparing the runtime of a simple query of MCR and McRAPTOR, MCR is slower, as MLC on the street network takes much more time than just checking the neighbors of a stop in the transfer graph. Generally using MCR or McRAPTOR is a trade-off of runtime and preprocessing time.

2.2.7 ULTRA

(Baum, Buchhold, Sauer, Wagner, & Zündorf, 2019) propose another algorithm building on MCR, called UnLimited TRAnsfers for Multi-Modal Route Planning (ULTRA). ULTRA capitalizes on the observation that extensive exploration of the transfer graph is often unnecessary for transfers between public transport trips, but is more crucial for initial and final transfers. Therefore, they propose a preprocessing step that computes intermediate transfers that contribute to optimal journeys on the transfer graph. They then use these precomputed transfers as the transfer graph for RAPTOR. To account for initial and final transfers, ULTRA employs the Bucket Contraction-Hierarchies (Bucket-CH) algorithm (Geisberger, Sanders, Schultes, & Delling, 2008), an efficient one-to-many approach, together with RAPTOR.

While ULTRA demonstrates a runtime improvement over MCR, it is limited to optimizing only time and the number of transfers. Using ULTRA in an

accessibility analysis setting is also unsuitable, because ULTRA runs a reverse Bucket-CH query from the end node to all stops, to compute potential final transfers. This means that ULTRA, unlike MCR, is incompatible with one-to-many queries.

2.2.8 McTB

(Potthoff & Sauer, 2021)

2.2.9 ULTRA-PHAST

2.3 Public Transport Data

In order to run routing algorithms for public transport networks in practice, a common data format is needed.

The General Transit Feed Specification (GTFS) (?, ?) serves as a standardized format for public transportation schedules and associated geographic details. It is divided into two main components: GTFS Schedule and GTFS Realtime. GTFS Realtime provides live transit updates. On the other hand, GTFS Schedule offers information about routes, schedules, fares, and geographic transit details.

For our study and tool, we focus solely on GTFS Schedule and omit considerations related to GTFS Realtime.

Central to the GTFS format are several core concepts. A *route* defines the overall path taken by a particular public transport service, identified by attributes like name, ID, and the mode of transport such as bus or subway. A *trip* refers to a specific run of a vehicle along a route, distinguishing between different timings or sequences of service on the same route. A *stop* is a specific point along a route where passengers embark or disembark. Stops have unique IDs, names, and geographical coordinates. *Stop times* specify when a vehicle is expected to be at a particular stop during its trip. This data pinpoints both the arrival and departure timings at each stop.

GTFS data is often distributed in plain text (.txt) files which are bundled together and compressed into a .zip file. This packaging makes it both compact for distribution and straightforward for developers to parse and use.

In essence, GTFS provides a comprehensive overview of a transit agency's service, covering both the spatial aspects of transit and the temporal aspects.

2.4 Street Network Data

Just as GTFS provides a standardized format for public transport schedules, the need for a consistent data format for street network information is addressed by

OpenStreetMap (OSM) (?, ?).

OSM is a collaborative initiative that offers freely available geographic data. This data captures various features on the Earth's surface, including roads, trails, establishments, railway stations, and more.

In the context of street networks, potentially used by routing algorithms, OSM represents roads and paths using interconnected nodes and ways. Nodes specify distinct geographical coordinates, defined by latitude and longitude, while ways connect these nodes to delineate linear structures or area boundaries. Importantly, these ways have meta-data assigned to them containing information about what vehicles can travel along them and how long they are.

In addition, OSM offers vast amounts of data about points of interest, potentially useful in accessibility analysis.

OSM is extensive, regularly updated, and most importantly freely available, which makes it indispensable for projects seeking reproducibility and generalizability.

2.5 Importance of Multimodality & Intermodality

"Results show that bicycles significantly reduce the average transfer times, the average path length of passengers' trips and the Gini coefficient of an urban public transport network" (Yang et al., 2018).

Public transport frequency is significantly positively correlated with the number of bicycle trips, especially short and medium distance trips up to 3 km (Radzimski & Dzięcielski, 2021).

(Murphy & Usher, 2015) conduct a questionnaire, which shows that 39% of bicycle sharing users (in Dublin) use bicycle sharing in conjunction with another mode of transport. Of those, 91.5% use public transport, which indicates that bicycle sharing is synergetic with public transport.

(Fishman, Washington, & Haworth, 2013) perform a literature review on bicycle sharing in general and find that bicycle sharing is synergetic with public transport.

(Ma, Liu, & Erdoğan, 2015) run a linear regression with data ... in which the number of passengers of public transport is regressed on the number of bicycle sharing trips. They find a positive correlation between the two and conclude that bicycle sharing and public transport are complementary. As a possible reason for this, they state that bicycle sharing can be used to solve the first and last mile problem.

"Through an extensive experimental study, it's demonstrated that allowing unrestricted walking considerably reduces travel times compared to scenarios

where walking is limited." (Wagner & Zündorf, 2017)

Fact: Computing with unrestricted walking takes way longer to compute. (Wagner & Zündorf, 2017)

3 Method

Our method is split into to two parts, the routing algorithm and the accessibility analysis tool.

3.1 Metric

To evaluate the accessibility in cities, we employ a metric that is an implementation of the 15-minute city concept. The concept measures how fast the access to a variety of important amenities is. To measure this, we categorize amenities into seven essential services: grocery, education, health, banks, parks, sustenance, and shops. Each category is populated with Points of Interest (POIs) sourced from OSM, providing a comprehensive database of locations. Our categorization is based on the work of (Olivari et al., 2023) with slight modification and can be seen in Table 2.

Each service category encapsulates several POIs. For instance, the "Parks" category may include multiple locations tagged in OSM as "leisure: park" or "leisure: dog park".

The core of our metric is the determination of temporal proximity to these amenities. For each category, we calculate the minimum travel time required to reach at least one POI of that category. The metric is then defined as the maximum value among these minimal times across all categories. This approach yields a singular measure that reflects the most significant time distance barrier within an urban area, which effectively captures the least accessible category for any given area. We think that it is beneficial to focus on the least accessible category, as measuring accessibility in cities by averaging accessibility across all categories can mask disparities categories. This ensures that the metric is targeted to areas of greatest need. By leveraging this metric, we aim to help city planners to create urban environments that prioritize sustainability, enhance the well-being of residents, and reduce dependency on vehicular transport, thus contributing to the broader goals of efficient urban planning and improved quality of urban life.

Our metric presents several advantages compared to the NEXI-minutes and NEXI-global, as outlined by (Olivari et al., 2023). Firstly, unlike the NEXI-minutes which calculates separate metrics for each of seven categories, our metric evaluates all categories together. This unified approach makes it more straightforward and easier to understand. In contrast, while NEXI-global also considers all categories in one assessment, it converts the results into a 0-100 score. This percentage system can obscure the real value of the data, making it more difficult to interpret.

Moreover, the NEXI-global's practice of assigning different weights to each

Table 2: Categories and their corresponding OSM tags

Category	**OSM Key**	**OSM Value**
Grocery	shop	alcohol, bakery, beverages, brewing supplies, butcher, cheese, chocolate, coffee, confectionery, convenience, deli, dairy, farm, frozen food, greengrocer, health food, ice-cream, pasta, pastry, seafood, spices, tea, water, supermarket, department store, general, kiosk, mall
Education	amenity	college, driving school, kindergarten, language school, music school, school, university
Health	amenity	clinic, dentist, doctors, hospital, nursing home, pharmacy, social facility
Banks	amenity	atm, bank, bureau de change, post office
Parks	leisure	park, dog park
Sustenance	amenity	restaurant, pub, bar, cafe, fast-food, food court, ice-cream, biergarten
Shops	shop	department store, general, kiosk, mall, wholesale, baby goods, bag, boutique, clothes, fabric, fashion accessories, jewelry, leather, watches, wool, charity, secondhand, variety store, beauty, chemist, cosmetics, erotic, hairdresser, hairdresser supply, hearing aids, herbalist, massage, medical supply, nutrition supplements, optician, perfumery, tattoo, agrarian, appliance, bathroom furnishing, do-it-yourself, electrical, energy, fireplace, florist, garden centre, garden furniture, fuel, glaziery, groundskeeping, hardware, houseware, locksmith, paint, security, trade, antiques, bed, candles, carpet, curtain, flooring, furniture, household linen, interior decoration, kitchen, lighting, tiles, window blind, computer, electronics, hifi, mobile phone, radio-technics, vacuum cleaner, bicycle, boat, car, car repair, car parts, caravan, fishing, golf, hunting, jet ski, military surplus, motorcycle, outdoor, scuba diving, ski, snowmobile, swimming pool, trailer, tyres, art, collector, craft, frame, games, model, music, musical instrument, photo, camera, trophy, video, videogames, anime, books, gift, lottery, newsagent, stationery, ticket, bookmaker, cannabis, copy shop, dry cleaning, e-cigarette, funeral directors, laundry, moneylender, party, pawnbroker, pet, pet grooming, pest control, pyrotechnics, religion, storage rental, tobacco, toys, travel agency, vacant, weapons, outpost

category complicates its analysis. Our metric, by focusing on the lowest-performing category across all areas, simplifies the understanding and highlights where improvement is most needed. This method prevents the dominance of stronger areas over weaker ones, ensuring a more balanced and fair evaluation of urban development.

Traditionally, the 15-minute city concept is applied to walking and or cycling and ignores other modes of transport. Some researchers, in the context of location-based metrics, even go as far to only calculate the bee-line distance to the nearest amenity and ignore the street network altogether (Gastner & Newman, 2006), while most only consider walking (Olivari et al., 2023; Nicoletti et al., 2023). We, however, believe that to accurately determine the accessibility of a city, all modes of transport must be considered, and the routing needs to be

as realistic as possible. We will therefore calculate our metric for various combinations of modes of transport, namely driving with a personal car+walking, free-floating bicycle sharing+walking, public transport+walking, free-floating bicycle sharing+public transport+walking, and walking. The car mode will serve as a baseline metric and show how competitive more sustainable modes of transport are.

Adding potentially fare-based modes of transport poses a challenge, as we need to consider the cost of the trip. Ignoring the cost of the trip could potentially lead to a misrepresentation of accesibility, as a trip with a high cost might be inaccessible to some people. To account for this, we will calculate Pareto sets that balance the minimum travel time and the cost of the trip.

3.2 Routing Algorithm

Our routing algorithm is an applied version of MCR with minor variation to make it more suitable in terms of free-floating vehicle sharing.

3.2.1 Requirements

In order to fully grasp the potential of the combination of the sustainable modes of transport, we require our routing algorithm to be **multi-modal**, **multi-objective**, and **unrestricted inter-modal**, and run in a reasonable time.

Multi-modal means that our routing algorithms allows multiple modes of transport, including scheduled transport systems, like public transfer and an arbitrary number of unscheduled transport systems, like walking, cycling and driving. In addition, we require that free-floating vehicle sharing systems are incorporated realistically. That means, that our routing algorithm must consider that switching to a free-floating vehicle is possible at any location, where a free-floating vehicle is available and parking a free-floating vehicle is possible anywhere where it's allowed.

Multi-objective means that our algorithm must find all pareto optimal journeys according to an arbitrary amount of objectives. The algorithm must provide the possibility to update the values of any objective whenever a *movement* occurs. We define a movement either as an edge traversal in an unscheduled network or a step in the route traversal during McRAPTOR. In the case of an edge traversal the new objective must be a function of the old objective and the edge weights, formally: $l' = f(l, w(e))$, where l and l' are the old and new labels, respectively, and $w(e)$ are the weights of the edge that is traversed. In the case of an update during a step of the route traversal, the new objective must be a function of the old objective (to be continued).

Inter-modal means that the different transport modes may be sequenced in any order. For example, when considering walking, cycling through a bicycle sharing system and public transport, the algorithm needs to consider journeys with bicycle rides between two consecutive public transport trips. **Unrestricted** means that the algorithm fully searches the unscheduled network graphs, and does not pose restrictions like a maximum of 10 minutes walking distance.

Both Dijkstra and MLC are not considered due to their impractical runtime. Furthermore, the need for multi-objective solutions excludes Dijkstra, RAPTOR, and ULTRA. The requirement for unrestricted inter-modal travel makes RAPTOR and McRAPTOR unsuitable in practical scenarios. To explain this, let's examine a straightforward example.

Consider the OSM graph of the key regions in Cologne, which comprises 125,176 nodes and 142,074 edges. For RAPTOR to compute a transitively closed graph, it requires calculating the walking distance between each node. This computation would yield $125,176^2 = 15,669,030,976$ edges, a number vastly greater than the original 142,074 edges.

While MCR does support multi-objective solutions with unrestricted inter-modal transfers, it doesn't fully encapsulate the multi-modal concept we require. Although it theoretically permits various modes of unscheduled transport, it is primarily tailored for station-based vehicle sharing systems. Our focus, however, is on the increasingly prevalent free-floating systems. In MCR, unscheduled networks are contracted, leading to the removal of certain nodes. If an optimal route requires a mode change at a deleted node, MCR will be unable to identify that path. As a result, MCR is not a viable option for our needs.

In the following section, we detail the modifications made to MCR to tailor it to our requirements.

3.2.2 Algorithm

As our algorithm should be easily adaptable to different modes of transport and the combination of different modes of transport, we formulate it in a modular fashion. The algorithm described next presents a scaffolding that needs to be augmented by different modules, where a module represents a specific mode of transport and running a module may be seen as fully exploring the network through this mode of transport. One module, for example, would be walking, and running the walking module would mean to traverse the whole walking graph given the current state of bags.

A module always takes bags as an input and returns bags as an output. As explained in Section 2 a bag is a set of labels, that are Pareto optimal with respect to the objectives. In addition, in our work, a bag is associated with a node some

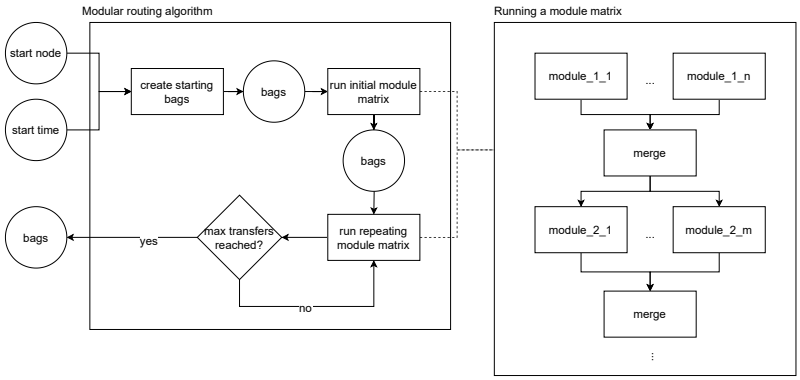

Figure 2: Modular Routing Algorithm

network. For the bags that are the input and output of the modules, we further require that they are associated with the same network. This is necessary in order to merge the bags of different modules. In our work, the common network will be the walking graph. This also has a real-world interpretation, as walking in between different modes of transport is very common.

To further explain how running a module looks like, consider the example of calling the walking module on the starting bags, which are a set of bags, where every bag is empty, except for the one of the starting node, where exactly one label is present that contains the starting time and zero costs. The output would be a set of bags, where each node's bag contains exactly one label where the time would be equal to the time it takes to walk to the node and the cost would be zero, as walking never costs anything.

The scaffolding of the algorithms is shown in Figure 2. The algorithm takes a start node and time as input and returns bags. First it creates the starting bags described above from the start node and the start time. Next, it runs the initial modules given by the initial module matrix.

A module matrix is an irregular matrix of modules. To run the module matrix, first the modules of the first row are run in parallel. Their respective outputs are then merged into a single set of bags. After that the second row is run and so on and so forth. This process is also described in Figure 2.

After the initial modules are run, the same is done for the repeating modules for the specified number of times. By convention, we count each iteration of the repeating modules as one trip.

3.2.3 Modules

In our experiments, we categorize four modules into two types: unscheduled and scheduled. The unscheduled modules consist of walking, free-floating vehicle sharing, and personal vehicle use and are based on MLC, while the scheduled

module is public transport, which is based on McRAPTOR.

Walking is the simplest unscheduled module, it simply consists of running MLC on the walking graph. The edges of the walking graph should contain the time it takes to traverse them by foot and should not have any monetary cost associated with them.

The free-floating vehicle sharing module is a bit more complex. Before running MLC on the respective vehicle graph, the module filters our all bags that are located at a node where no free-floating vehicle is available. In addition, the vehicle graph is augmented with a walking graph. This means that at nodes in the vehicle graph, where it is allowed to park the vehicle, there is an edge to the closest node in the walking graph. This augmentation is necessary, as the output bags of the free-floating vehicle sharing module need to be associated with nodes in the walking graph, as it is the common network. The module may also define any form of monetary cost.

For the personal vehicle module, we assume that there is only one personal vehicle and that it is located at the starting node. Therefore, the module filters out all bags that are not located at the starting node. Obviously, this module is only useful for the first trip, or more precisely, the module should only be used in the first row of the initial module matrix. After that MLC is run on the vehicle graph again augmented with the walking graph. TODO: monetary cost

In comparison to the unscheduled modules, the public transport module does not use MLC, but McRAPTOR. As a first step the module filters out all bags that are associated with a node that is not near a stop in the public transport system. Next, the module performs a single iteration of McRAPTOR, which represents a single trip within the public transport system. Last, the resulting bags have to be associated with nodes in the walking network again. To do so, the modules simply use the node that is closest to the coordinates of the public transport stop. Just like the free-floating vehicle module, the public transport module may define any form of monetary cost.

3.2.4 Merging

The merging of bags after running multiple modules in parallel is quite simply. Each bag represents a set of Pareto optimal labels and each bag is associated with a node. We therefore merge bag-wise. We differentiate between two cases:

1. There is only one bag associated with a node.
2. There are multiple bags associated with the same node.

In the first case, we simply put the bag into the output bags. In the second case, we create a new bag that contains all labels of all bags associated with the

node, which might break the Pareto optimality of the labels. Therefore, to restore the Pareto optimality of the labels, we remove all labels that are dominated by another label.

3.2.5 Enhanced MLC & McRAPTOR

To address the multi-objective optimization involving both time and monetary cost, we introduce enhancements to MLC and McRAPTOR. The standard versions of MLC and McRAPTOR do not adequately capture dynamic pricing models, which is necessary to realistically represent monetary costs.

The original MLC associates a fixed cost with an edge, which cannot represent variable pricing, such as a bike-sharing tariff that costs 1€ per 15-minute increment. Labels are only updated by adding the cost of a given edge to the label's values. Similarly, McRAPTOR updates the labels at each stop during route traversal only based on the information of the current trip and stop. With this McRAPTOR is unable to represent a pricing scheme that varies with the number of stops, like the one used by the Cologne Transport Authority.

Our proposed modifications involve the use of 'hidden values' within the labels that are used by these algorithms. These hidden values carry additional information which is not considered when comparing labels, but that may be used to update cost dynamically.

In the case of MLC, the hidden values may be updated along any edge, just like the regular values of the label. A hidden value, may carry information on how long the current trip with the shared vehicle is. We then additionally allow defining a function that updates while traversing an edge and may use the values and hidden values both before and after the traversal to do the update. With this functionality it is easy to increment the cost by 1€ every time the time spent on the trip exceeds the 15-minute interval.

Similarly, the hidden values may be updated during McRAPTOR after every iteration of a stop. We can, therefore, store how many stops the current trip already traversed and if that number exceeds four, we can easily increase the price from 2.20€ to 3.20€.

Additionally, as the concept of hidden values isn't specific to MLC or McRAPTOR, the hidden values can be used across iterations. To understand the benefit, again consider the example of the pricing of the Cologne Transport Authority. The ticket that costs 3.20€ allows traveling any number of trips within Cologne, no matter if it is necessary to change to a different trip. Therefore, if we were to first travel two stations with one trip and then get out to catch another trip that consists of five stops, we would still have the information that we already commuted two stops. Therefore, we can charge 3.20€, instead of charging 2.20€

two times, which is more realistic.

These enhanced versions of MLC and McRAPTOR are used in our modules, so that we can use the more realistic dynamic pricing schemes.

3.2.6 Example

To illustrate our algorithm, we will now go through an example step by step. The module configuration we use in our example represents travelling by free-floating vehicle sharing, public transport and walking. The initial module matrix just contains the walking module, in order to initially reach the free-floating vehicles, as well as, the public transport stops. The repeating module matrix consists of first the free-floating vehicle sharing module and the public transport module in parallel and then the walking module. It can be seen in Figure 3.

$$\begin{pmatrix} \text{free-floating vehicle sharing module} & \text{public transport module} \\ \\ \text{walking} & \end{pmatrix}$$

Figure 3: Example Repeating Module Matrix

In our example, we assume that both public transport and vehicle sharing has some form of cost associated with it. The objective is to minimize arrival time and cost. We also only consider a maximum of two trips.

First we run the initial modules, which in our case is just the walking module on the starting bags. As the starting bags only consist of one non-empty bag at the starting node with exactly one label, running MLC on the walking graph is equivalent to running Dijkstra's algorithm. In the real-world this represents walking to all nodes in the walking network from the start node. Note, that after the initial walking module all bags only contain exactly one label, as the cost to go anywhere by foot is zero.

Next the modules of the first row of the repeating matrix are run. In the real-world this means that after an initial walk the traveler would either drive with a free-floating vehicle starting from a location where one is available or commute by public transport starting from one stop. For public transport one could imagine that we commute with all possible trips from all stops and update the bags of the stops along each route accordingly. After running these modules and merging their result bags, each bag may contain more than one label, as public transport and driving with a vehicle may be faster than walking, but also cost money. It may even be that some bags contain three different labels, if, for example, driving

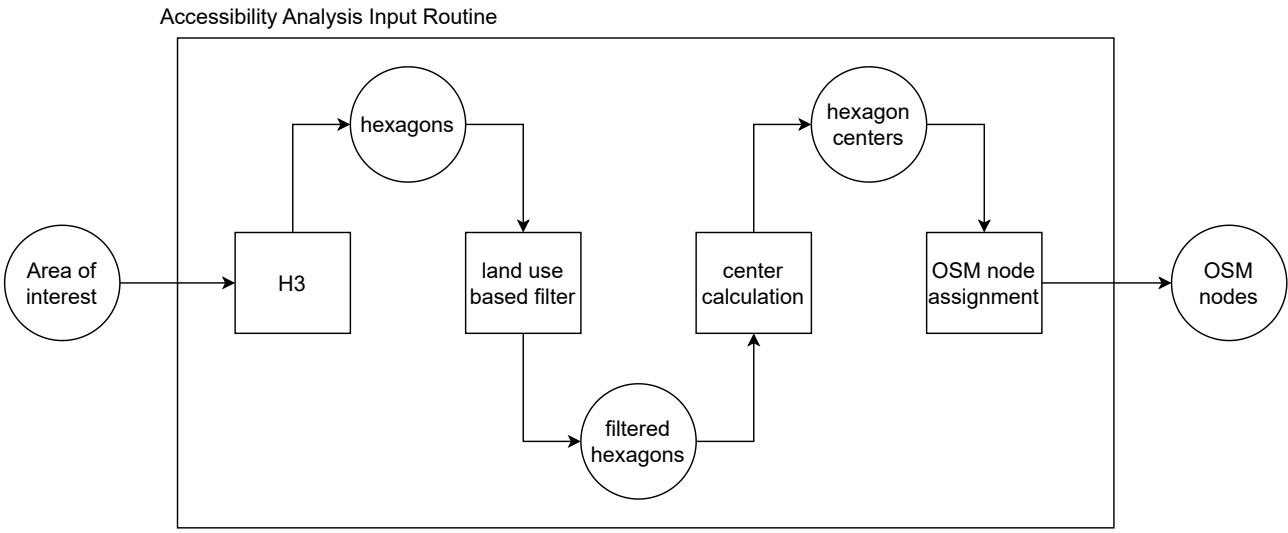

Figure 4: Input Routine

with a vehicle is the fastest but also costs the most money and commuting by public transport is faster than walking. The next step consists of running the second row of the repeating module matrix, which in our case is the walking module again. Running the walking module in the repeating module matrix is important in order to reach nearby POIs after commuting through the public transport system. After that the repeating module matrix is run again, as we consider a maximum of two trips. The result of the second run of the repeating module matrix is our final result.

3.3 Accessibility Analysis Tool

We embed the routing algorithm described in Section 2.2 in our accessibility analysis routine to compute the metric described in Section 3.1.

Our accessibility analysis routine consists of three parts: the input routine, the main routine and the metrics routine.

In the input routine, depicted in Figure 4, we first create an even grid that covers the whole area of interest, for example a city. To create such a grid, we use H3 (*H3 | H3*, n.d.), which uses hexagons to evenly discretize an area. Our goal will be to calculate our metric for each hexagon, so that we get detailed spatial information about the accessibility in the area of interest. The chosen H3 resolution determines the size of these hexagons: a higher resolution means smaller hexagons, enhancing the granularity of our analysis. As such, selecting an appropriate H3 resolution is pivotal as it allows us to calculate our metrics for each hexagon with increased spatial accuracy, yielding a detailed spatial dataset that reflects the accessibility variations within the area of interest. We recommend a resolution of nine, which corresponds to a hexagon edge length of roughly 200

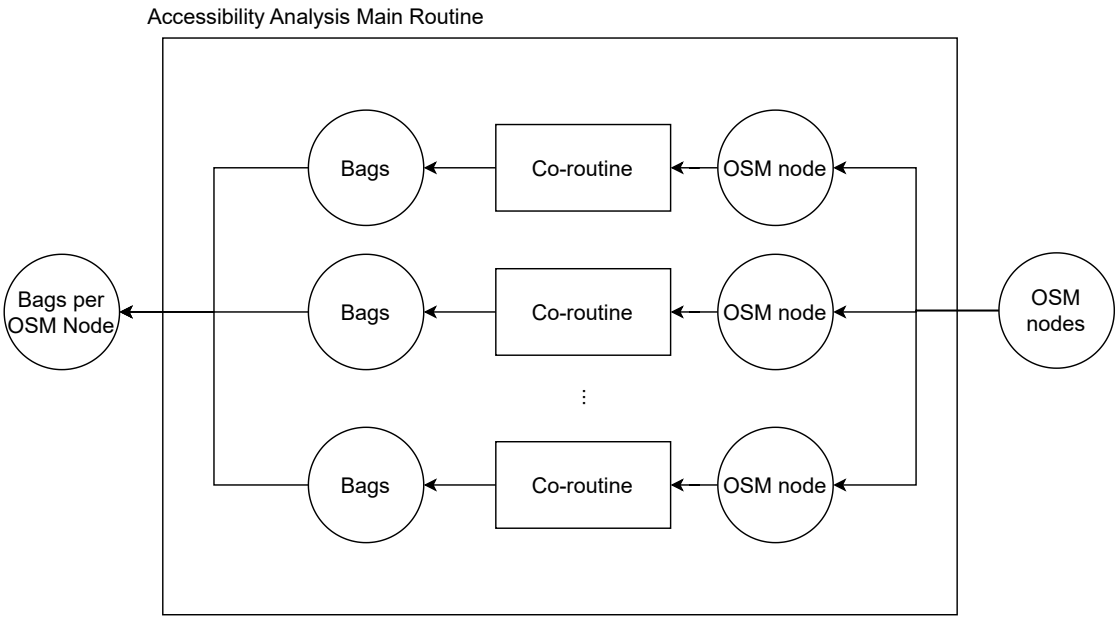

Figure 5: Main Routine

meters, as it is a good compromise between accuracy and computation time.

The underlying street network needs to be larger than the area of interest. Otherwise, border regions will have a lower accessibility than they should, as the street network is incomplete.

The input routine also filters out uninteresting hexagons. For, example we filter out hexagons that don't contain any residential areas, as there are no people living there is no need to access any amenities.

Next the input routine retrieves the centroid of each hexagon and then calculates the Euclidean distance between the centroids and the OSM nodes in order to assign the closest OSM node to each centroid.

The result of the input routine is a set of OSM nodes, for which we want to compute the accessibility.

The main routine, depicted in Figure 5, calls our routing algorithm described in Section 2.2 on each OSM node provided by the input routine. This results in a set of Bags for each node.

The metrics routine, depicted in Figure 6, processes the bags into Pareto sets, where one entry in the Pareto set is a tuple of the X-minute city metric and the related cost. To do so, we process each collection of bags separately - one co-routine for each OSM node/collection of bags.

The co-routine is depicted in Figure 7 and works as follows. We start by collecting all unique cost values in each label in each bag and then sort them in ascending order. For each cost value we then determine the associated X-minute city metric. We do so by checking whether every category is reachable given the

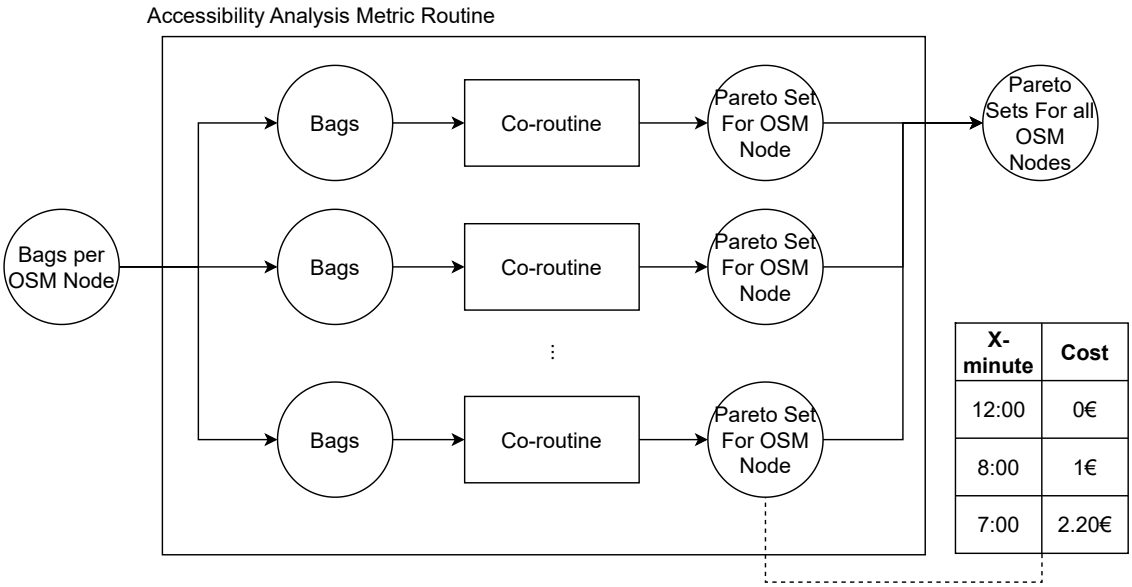

Figure 6: Metric Routine

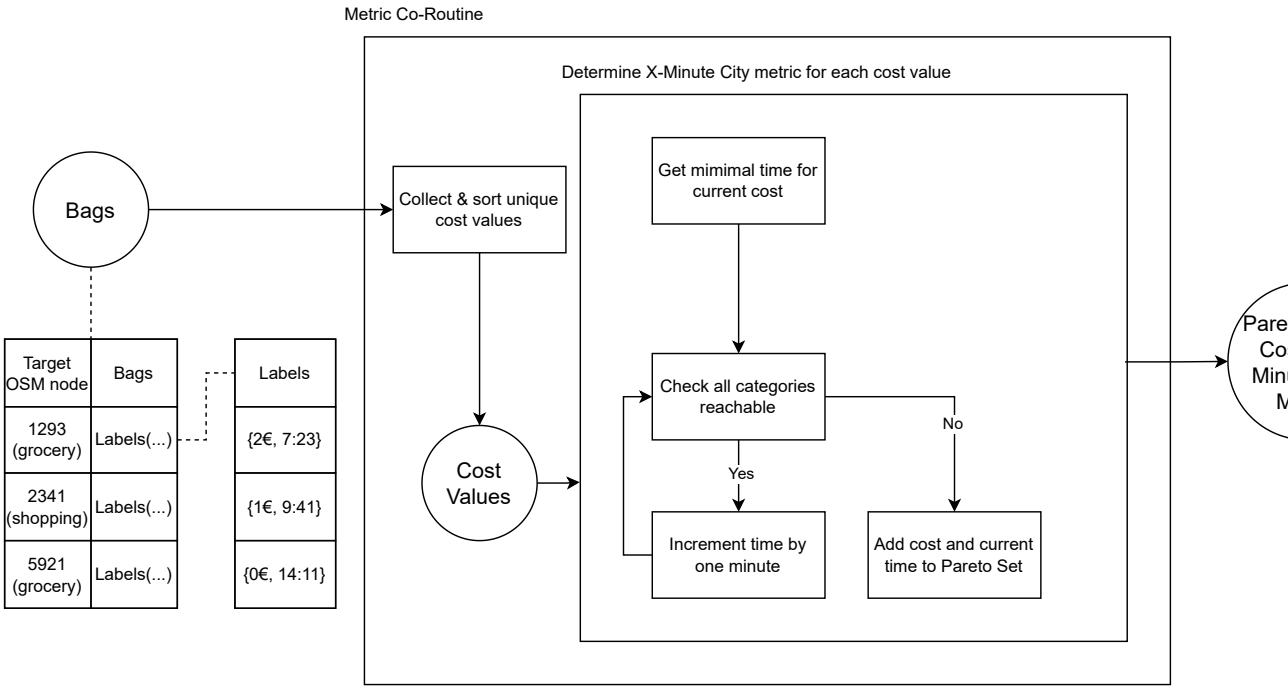

Figure 7: Metric Co-Routine

cost value and an iteratively increasing time value. We start with a time value that is equal to the minimum time across all labels. If all categories are reachable, we've found the X-minute city metric, which together with the cost value is added to the Pareto set. If not, we increase the time value by one minute and check again. We repeat this until all cost values are processed. The result is a Pareto set for each OSM node/collection of bags.

4 Experiment

To answer the question of how competitive sustainable modes of transport are in comparison to the traditional mode of travel by car, we run an experiment in the city of Cologne. We do so by calculating the X-minute city metric for hexagons all over Cologne.

As we want to compare different modes of transport, we will calculate the metric multiple times for different scenarios, each with a different combination of modes of transport.

4.1 Scenarios

No matter, the mode of transport, we always allow for walking, for two reasons. First, walking is the most accessible mode of transport, available to almost everyone and without any additional costs. Second, most other modes of transport require walking at some point, be it to the next bus stop or to the next available bicycle.

The first scenario we consider is our baseline scenario, which only considers walking. This scenario measures what is possible without any additional infrastructure. Distinct from other scenarios, it does not require any additional cost, thus presenting the most basic form of urban mobility.

Building on this, the second scenario we consider is the scenario of only walking. We consider this scenario as the benchmark scenario, as we hope to achieve similar (or even better) results with more sustainable modes of transport. Therefore, we use it to answer the question of how competitive sustainable modes of transport are in comparison to the traditional mode of travel by car.

Transitioning from the simplest form of mobility, the third scenario, focused on public transport, becomes essential to understand the effectiveness and accessibility of urban transit systems. This scenario evaluates how well-connected and time-efficient public transportation networks are, and their role in reducing reliance on personal vehicles. It also investigates the impact of public transport on urban mobility and its potential in contributing to a more sustainable urban environment. Specifically, it assesses whether public transport is a viable alternative to the personal car and whether it actually offers significant advantages over walking, considering the X-minute city metric.

Next, in the fourth scenario, we shift our focus to the dynamics of bicycle sharing systems. This scenario is important for assessing the feasibility and attractiveness of cycling as a primary mode of transportation in urban areas. We will directly compare it to the public transport scenario, to understand which sustainable mode of transport is superior.

Finally, the fifth scenario combines public transport and bicycle sharing, offering insights into the synergy between these two modes of transport. This integrated approach mirrors a growing trend in urban mobility solutions, where multi-modal transport options are increasingly favored. It underscores how this combination can bridge the gaps in accessibility and efficiency found when each mode is used independently. This scenario is expected to be the most competitive against cars, offering a comprehensive and sustainable urban transit model that could reshape the landscape of city mobility.

We summarize the scenarios in Table 3.

Scenario	**Modules**	**Key Points**
Walking	Walking	Baseline scenario
Personal Car	Personal Vehicle, Walking	Benchmark scenario
Public Transport	Public Transport, Walking	Evaluate the effectiveness public transport systems
Bicycle Sharing	Vehicle Sharing, Walking	Evaluate the effectiveness of bicycle sharing systems
Public Transport and Bicycle Sharing	Public Transport, Bicycle Sharing, Walking	Evaluate the effectiveness of sustainable multi-modal transport systems

Table 3: Scenarios for Urban Mobility Analysis

The specific configuration of the module matrices for each scenario can be found in Appendix C.

4.2 Data

To calculate the Pareto sets of the X-minute city metric for the different scenarios, we use four different datasets. First, we require data that depicts the street network of the city of Cologne. Second, we need to know the locations of the POIs, which we want to reach. Third, we need to know the locations of the public transport stops, as well as, the schedules of the public transport. For the bicycle sharing scenario, we also need to know the locations of the bicycles. Lastly, we also use land use data to identify where residential areas are located, so that we can calculate the X-minute city metric only for these areas.

As we will query spatial datasets of various formats, from different sources, the area covered by the datasets will not be the same. Therefore, we first define an area of interest and later trim the datasets to this area. In our case, this area is defined as the area of the administrative district of Cologne, specifically the

"Stadtkreis Köln". We retrieve the specific boundary of this area with the help of the Overpass API (*Overpass API/Overpass QL – OpenStreetMap Wiki*, n.d.). The specific query can be found in Appendix A.

4.2.1 Street Network & POIs

For the street network and the POIs, we use data from OpenStreetMap (OSM) (?, ?). To use OSM data in practice various tools and services have been developed. Among these we use, pyrosm (?, ?) which is a Python library designed specifically for reading OSM data in different formats and conducting data processing operations. Through pyrosm, we can automatically fetch data from sources like Geofabrik (?, ?) and BBBike (?, ?), which are two of the most popular OSM data providers. In our case, we use the data for the city of Cologne from BBBike. However, due to the flexibility of pyrosm, it is easily possible to use data from other sources as well and expand our analysis to other cities.

After retrieving the data, our tool automatically retrieves a graph representation of the street network trimmed to the area of interest plus a buffer of 5km. This buffer is important, because without it calculating the X-minute city metric at the border of the area of interest would result in a higher value than the actual value. As a last cleaning step, we remove all nodes, that are not part of the largest weakly connected component. A weakly connected component is a subgraph in which, if all directed edges were treated as undirected, any two vertices from the subgraph would be connected. Multiple weakly connected components in graphs derived from OSM data, mostly happen at the border of the considered area and can be neglected.

Because we consider multiple different modes of transport on the network, it is important to filter out all edges that are not accessible by the respective mode of transport. To do so, we use pyrosm's built-in filtering functionality. For reproducibility, we list the filters that pyrosm uses in Appendix B.

To retrieve the POIs, we use the Overpass API (*Overpass API/Overpass QL – OpenStreetMap Wiki*, n.d.). Our tool will automatically retrieve all POIs that fall into one of our predefined categories specified in Section 3.1 inside of the area of interest plus the buffer mentioned before.

4.2.2 Public Transport

To handle public transport data, we use the General Transit Feed Specification (GTFS) (?, ?). To retrieve it we rely on the Mobility Database (?, ?). This database serves as an open-source repository containing links to publicly available GTFS feeds globally, standing as the subsequent version of TransitFeeds (?, ?).

Similarly to the OSM data, we trim the GTFS data to the area of interest plus the buffer.

The GTFS data is also cleaned and converted into a format that is more suitable for our algorithm, or more specifically McRAPTOR, which is part of our algorithm.

Specifically, there are two major incompatibilities between the GTFS specification and RAPTOR's notion of routes and trips. Firstly, each trip belonging to a single route in RAPTOR visits the same stops in the same order. It is not possible that a trip skips some stops that another trip of the same route visits, much less use a completely different sequence of routes. In GTFS routes do allow that, as they are much more a group of trips that is presented to the rider under the same name or identifier. Secondly, GTFS trips allow visiting the same stop multiple times, which is not allowed in RAPTOR.

To overcome these difference our tool splits up routes into smaller routes, that follow the same sequence of stops. Additionally, it also removes circular trips, altogether.

4.2.3 Bicycle Sharing

Our bicycle sharing data was retrieved from the NextBike API over a time period of one year. TODO: MORE INFORMATION TO BE ADDED HERE

The data consists of all trips that were made with the NextBike system in the city of Cologne from the 15th of January 2022 to the 31st of August 2023. To get representative samples of the locations of all bicycles we employ the following strategy.

We first discretize the data spatially and temporally. For the temporal discretization, we derive the location of each bicycle every hour. For the spatial discretization, we use H3 hexagons with a resolution of 9. The resulting data is the information how many bicycles were at each hexagon at each hour. This data is then used as an input for k-medoids clustering (Rdusseeun & Kaufman, 1987) with a k of 4. K-medoids, also known as PAM (Partitioning Around Medoids) algorithm, is a clustering technique that partitions a dataset into K clusters, where each is assigned a medoid, that is the most centrally located object in a cluster. Unlike K-means, which uses mean values as cluster centers, K-medoids an actual data point as the center of a cluster. This has the advantage that the centers are part of the dataset and therefore are realistic samples. We use the resulting medoids as different configurations of the locations of the bicycles.

4.2.4 Land Use

To identify the residential areas, we use the land use data from the CORINE Land Cover (CLC) project (*CORINE Land Cover 2018 (Vector), Europe, 6-Yearly - Version 2020_20u1, May 2020*, n.d.). The data covers the whole of Europe and is publicly available, which again makes it possible to expand our analysis to other cities in Europe. We trimmed the data to the area of interest and then filtered for the land use types "Continuous Urban Fabric" and "Discontinuous Urban Fabric". These two land use types represent the residential areas of the city.

4.3 Assumptions

To calculate the X-minute city metric we have to abstract from reality to some degree. We do so by making the following plausible assumptions.

Firstly, we assume that travelling along an edge on the street network, by walking, cycling or driving, is always proportional to the length of the edge. To obtain the time it takes to travel along an edge, we divide the length of the edge by the speed of the mode of transport. The different speeds for the different modes of transport are listed in Table 4.

Mode	**Speed (m/s)**
Walking	1.4
Cycling	4.0
Driving	11.0

Table 4: Speeds for different modes of transport

We also pose some assumption on the transitioning between different modes of transport, as well as, in the case of public transport, the transfer time at the stops. For the transfer time at stops we assume a fixed time of one minute. To transition from any OSM network-based mode of transport to public transport, we assume that the stop is precisely at the location the closest node of the OSM network. As OSM networks contain public transport stops, there should be no difference between the two. Similarly, we assume that the bicycles are located at the closest node of the OSM network. As the OSM network, especially in city is very dense, this assumption is reasonable. We also assume that bicycles and cars can be parked anywhere on their respective network for the sake of simplicity.

4.3.1 Pricing

We try to implement a pricing scheme in our scenarios that represents the real-world circumstances as closely as possible.

For bicycle sharing, we use the pricing scheme of NextBike, which is 1€ every 15 minutes. To depict this we add a hidden value to the labels processed in MLC that depicts how long the current bicycle trip is. As two consecutive bicycle trips are considered separately we nullify this hidden value after each run of MLC.

For public transport, we use the pricing scheme of the Cologne Transport Authority (KVB), which is 2.20€ for trips that span four stops or less. For any trip that spans more than four stops or any multitude of trips the KVB charges 3.20€. To depict this we add a hidden values to the labels processed in McRAPTOR that depicts how many stops the traveler already traversed. Because two consecutive trips are considered together, we don't need to nullify the hidden value.

TODO: personal car

A Appendix - Overpass Query for Boundary of Cologne

```
[out:json][timeout:50];
area["name"="Köln"]->.searchArea;
relation["boundary"="administrative"]["admin_level"="6"](area.searchArea);
out body;
>;
out skel qt;
```

B Appendix - Pyrosm Network Filter

Pyrosm filters out all ways that have the following tags:

Table 5: Driving Filter

Key	Values
area	yes
highway	cycleway, footway, path, pedestrian, steps, track, corridor, elevator, escalator, proposed, construction, bridleway, abandoned, platform, raceway
motor_vehicle	no
motorcar	no
service	parking, parking_aisle, private, emergency_access

Table 6: Walking Filter

Key	Values
area	yes
highway	cycleway, motor, proposed, construction, abandoned, platform, raceway, motorway, motorway_link
foot	no
service	private

Table 7: Cycling Filter

Key	Values
area	yes
highway	footway, steps, corridor, elevator, escalator, motor, proposed, construction, abandoned, platform, raceway, motorway, motorway_link
bicycle	no
service	private

C Appendix - Experiment Module Matrix Configuration

TODO: add module matrix configuration for each scenario here

References

Allam, Z., Moreno, C., Chabaud, D., & Pratlong, F. (2020). Proximity-Based Planning and the "15-Minute City": A Sustainable Model for the City of the Future. In *The Palgrave Handbook of Global Sustainability* (pp. 1–20). Cham: Springer International Publishing. doi: 10.1007/978-3-030-38948-2_178-1

Baum, M., Buchhold, V., Sauer, J., Wagner, D., & Zündorf, T. (2019). Un-Limited TRAnsfers for Multi-Modal Route Planning: An Efficient Solution. , 16 pages. doi: 10.4230/LIPIcs.ESA.2019.14

CORINE Land Cover 2018 (vector), Europe, 6-yearly - version 2020_20u1, May 2020. (n.d.). https://sdi.eea.europa.eu/catalogue/copernicus/api/records/71c95a07-e296-44fc-b22b-415f42acfdf0?language=all.

Delling, D., Dibbelt, J., Pajor, T., Wagner, D., & Werneck, R. F. (2013). Computing Multimodal Journeys in Practice. In D. Hutchison et al. (Eds.), *Experimental Algorithms* (Vol. 7933, pp. 260–271). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-38527-8_24

Delling, D., Pajor, T., & Werneck, R. F. (2015, August). Round-Based Public Transit Routing. *Transportation Science*, *49*(3), 591–604. doi: 10.1287/trsc.2014.0534

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, *1*(1), 269–271.

Fishman, E., Washington, S., & Haworth, N. (2013, March). Bike Share: A Synthesis of the Literature. *Transport Reviews*, *33*(2), 148–165. doi: 10.1080/01441647.2013.775612

Gastner, M. T., & Newman, M. E. J. (2006, July). Optimal design of spatial distribution networks. *Physical Review E*, *74*(1), 016117. doi: 10.1103/PhysRevE.74.016117

Geisberger, R., Sanders, P., Schultes, D., & Delling, D. (2008). Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks. In C. C. McGeoch (Ed.), *Experimental Algorithms* (pp. 319–333). Berlin, Heidelberg: Springer. doi: 10.1007/978-3-540-68552-4_24

Geisberger, R., Sanders, P., Schultes, D., & Vetter, C. (2012, August). Exact Routing in Large Road Networks Using Contraction Hierarchies. *Transportation Science*, *46*(3), 388–404. doi: 10.1287/trsc.1110.0401

Geurs, K. T., & van Wee, B. (2004, June). Accessibility evaluation of land-use and transport strategies: Review and research directions. *Journal of Transport Geography*, *12*(2), 127–140. doi: 10.1016/j.jtrangeo.2003.10.005

Gongadze, S., & Maassen, A. (Wed, 01/25/2023 - 15:46). Paris' Vision for a '15-Minute City' Sparks a Global Movement.

Gurney, K. R., Kilkis, S., Seto, K., Lwasa, S., Moran, D., Riahi, K., ... Luqman, M. (2021, October). Greenhouse Gas Emissions from Global Cities Under SSP/RCP Scenarios, 1990 to 2100.

Gustafson, D. (2022). *Examining Spatial Change in the Form of the 15-Minute City and Its Capability to Address Social Inequalities in Stockholm, Sweden.*

H3 | H3. (n.d.). https://h3geo.org/.

Hansen, P. (1980). Bicriterion Path Problems. In G. Fandel & T. Gal (Eds.), *Multiple Criteria Decision Making Theory and Application* (pp. 109–127). Berlin, Heidelberg: Springer. doi: 10.1007/978-3-642-48782-8_9

Kriegler, E., Luderer, G., Bauer, N., Baumstark, L., Fujimori, S., Popp, A., ... van Vuuren, D. P. (2018, April). Pathways limiting warming to 1.5°C: A tale of turning around in no time? *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *376*(2119), 20160457. doi: 10.1098/rsta.2016.0457

Liu, P. R., & Raftery, A. E. (2021, February). Country-based rate of emissions reductions should increase by 80% beyond nationally determined contributions to meet the 2 °C target. *Communications Earth & Environment*, *2*(1), 1–10. doi: 10.1038/s43247-021-00097-8

Ma, T., Liu, C., & Erdoğan, S. (2015, January). Bicycle Sharing and Public Transit: Does Capital Bikeshare Affect Metrorail Ridership in Washington, D.C.? *Transportation Research Record*, *2534*(1), 1–9. doi: 10.3141/2534-01

Moreno, C., Allam, Z., Chabaud, D., Gall, C., & Pratlong, F. (2021, March). Introducing the "15-Minute City": Sustainability, Resilience and Place Identity in Future Post-Pandemic Cities. *Smart Cities*, *4*(1), 93–111. doi: 10.3390/smartcities4010006

Müller-Hannemann, M., Schulz, F., Wagner, D., & Zaroliagis, C. (2007). Timetable Information: Models and Algorithms. In F. Geraets, L. Kroon, A. Schoebel, D. Wagner, & C. D. Zaroliagis (Eds.), *Algorithmic Methods for Railway Optimization* (pp. 67–90). Berlin, Heidelberg: Springer. doi: 10.1007/978-3-540-74247-0_3

Murphy, E., & Usher, J. (2015, February). The Role of Bicycle-sharing in the City: Analysis of the Irish Experience. *International Journal of Sustainable Transportation*, *9*(2), 116–125. doi: 10.1080/15568318.2012.748855

Nicoletti, L., Sirenko, M., & Verma, T. (2023, March). Disadvantaged communities have lower access to urban infrastructure. *Environment and Planning B: Urban Analytics and City Science*, *50*(3), 831–849. doi: 10.1177/23998083221131044

Olivari, B., Cipriano, P., Napolitano, M., & Giovannini, L. (2023, December). Are Italian cities already 15-minute? Presenting the Next Proximity Index: A novel and scalable way to measure it, based on open data. *Journal of Urban Mobility*, *4*, 100057. doi: 10.1016/j.urbmob.2023.100057

Overpass API/Overpass QL – OpenStreetMap Wiki. (n.d.). https://wiki.openstreetmap.org/wiki/Overpass_API/Overpass_QL.

Papas, T., Basbas, S., & Campisi, T. (2023, January). Urban mobility evolution and the 15-minute city model: From holistic to bottom-up approach. *Transportation Research Procedia*, *69*, 544–551. doi: 10.1016/j.trpro.2023.02.206

Potthoff, M., & Sauer, J. (2021, October). *Fast Multimodal Journey Planning for Three Criteria* (No. arXiv:2110.12954). arXiv. doi: 10.48550/arXiv.2110.12954

Proffitt, D. G., Bartholomew, K., Ewing, R., & Miller, H. J. (2019). Accessibility planning in American metropolitan areas: Are we there yet? *Urban Studies*, *56*(1), 167–192.

Radzimski, A., & Dzięcielski, M. (2021, March). Exploring the relationship between bike-sharing and public transport in Poznań, Poland. *Transportation Research Part A: Policy and Practice*, *145*, 189–202. doi: 10.1016/j.tra.2021.01.003

Rdusseeun, LKPJ., & Kaufman, P. (1987). Clustering by means of medoids. In *Proceedings of the statistical data analysis based on the L1 norm conference, neuchatel, switzerland* (Vol. 31).

Sims, R., Schaeffer, R., Creutzig, F., Cruz-Núñez, X., D'Agosto, M., Dimitriu, D., ... Tiwari, G. (2014). Transport [Journal Article]. In *Climate change 2014: Mitigation of climate change* (chap. 8). Institute of Transportation Studies, University of California, Davis.

Wagner, D., & Zündorf, T. (2017). Public Transit Routing with Unrestricted Walking. In G. D'Angelo & T. Dollevoet (Eds.), *17th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2017)* (Vol. 59, pp. 7:1–7:14). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi: 10.4230/OASIcs.ATMOS.2017.7

Walk Score Methodology. (n.d.). https://www.walkscore.com/methodology.shtml.

Weng, M., Ding, N., Li, J., Jin, X., Xiao, H., He, Z., & Su, S. (2019, June). The 15-minute walkable neighborhoods: Measurement, social inequalities and implications for building healthy communities in urban China. *Journal of Transport & Health*, *13*, 259–273. doi: 10.1016/j.jth.2019.05.005

Yang, X.-H., Cheng, Z., Chen, G., Wang, L., Ruan, Z.-Y., & Zheng, Y.-J. (2018, January). The impact of a public bicycle-sharing system on urban public transport networks. *Transportation Research Part A: Policy and Practice*, *107*, 246–256. doi: 10.1016/j.tra.2017.10.017