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 6, 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

List of Figures

List of Tables

1 Introduction

more needs to be done in order to reach the goals of the Paris Agreement (Mitchell et al., 2018). in order to reach the goals of the Paris Agreement emission reduction needs to start immediately (Kriegler et al., 2018). say that it is unlikely that countries will reach the goals of the paris agreement (Liu & Raftery, 2021). 72% of global transport emissions are from road vehicles (Sims et al., 2014).

Cities contribute massively to global emissions (citation) Therefore, transforming cities to be emission free is a key step in the fight against climate change.

In order to reduce the amount of car traffic, cities need to be planned in a way that makes allows people to access everything they need with alternative 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 to fully capture the potential of all sustainable modes of transport, they should be considered when measuring 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 the X-minute city metric, extending the concept of the 15-minute city to include all modes of transport and therefore presenting a more holistic view on accessibility via sustainable modes of transport.

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)

Some quantitative studies measured the degree to which cities are 15-minute cities, by introducing metrics that build upon the concept of the 15-minute city. For example, (Olivari, Cipriano, Napolitano, & Giovannini, 2023) the NExt proXimity Index (NEXI), which consists of two parts: the NEXI-Minutes and the NEXI-Global. The NEXI-minutes is defined over a set of categories, such as education, entertainment or groceries. For each category the NEXI-minutes is defined as the time required to reach the nearest facility of that category.

The NEXI-Global is insipred by the Walk Score methodology (*Walk Score Methodology*, n.d.) it is the weighted sum of scores assigned to each category, where the scores depend on the time required to reach the nearest facility of that category.

... explain more about NEXI paper.

"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)

... 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.

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.

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 $i \in \{1, \dots, k\}$ and $l'_i < l_i$ for at least one $i \in \{1, \dots, k\}$. 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.

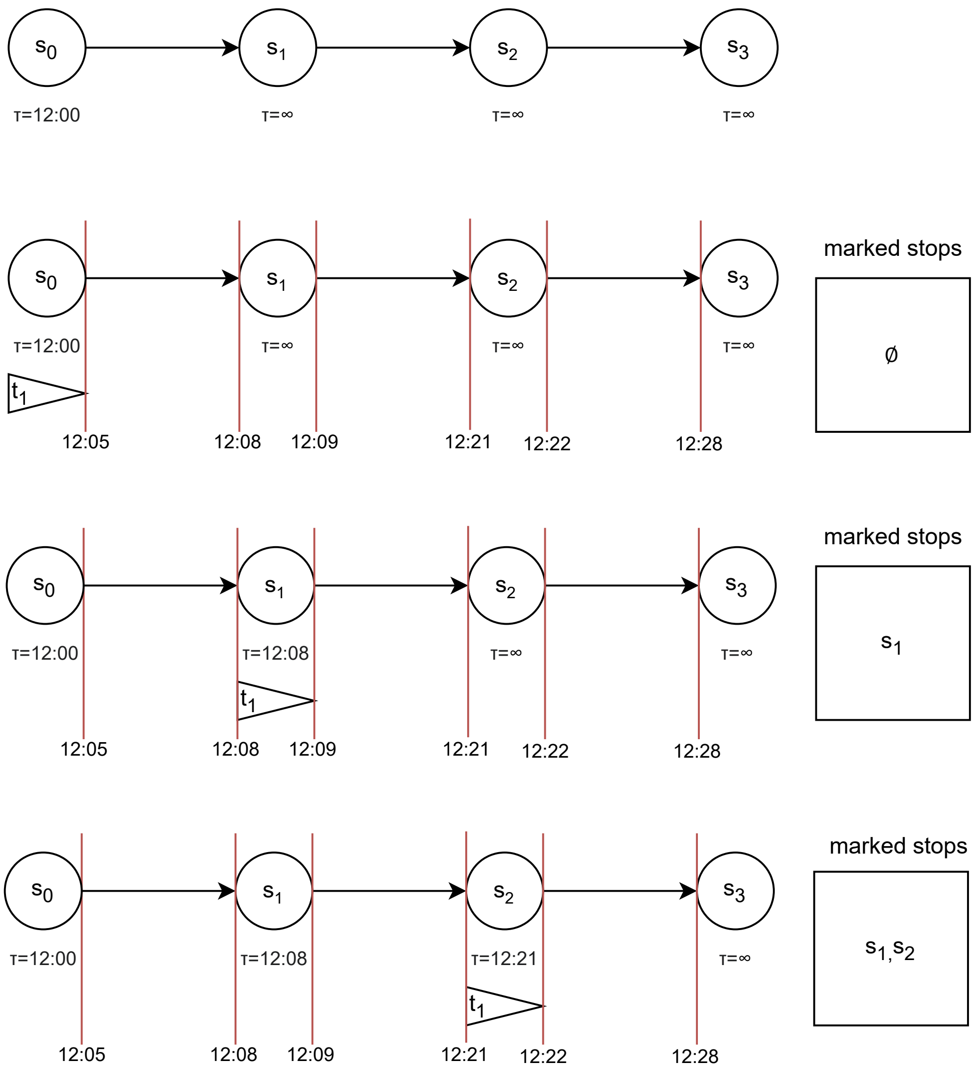

Figure 1: Iterating a route in RAPTOR

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.

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 modifies 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.

3 Method

Our method is split into to two parts, the routing algorithm and the accessibility analysis tool. Our routing algorithm is an applied version of MCR with minor variation to make it more suitable in terms of free-floating vehicle sharing.

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 intermodal 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.

3.1 Routing Algorithm

The input of our algorithm is a start time and a start node. The start node may be any node in the OSM network. The output of our algorithm are the bags for each node in the OSM network.

Our algorithm is split into two phases, which are repeated iteratively. The algorithm is depicted in Figure 2. In the first phase the walking network is explored through the MLC algorithm. Walking is not considered a trip and there is no upper limit on the walking distance. In the initial iteration MLC starts with a single label containing the starting values in the bag of the start node. All other bags are emtpy. We run MLC until it converges and retrieve the final bags for each node.

In the second phase we explore the public transport network, as well as, all modes of unscheduled travel, except walking. We do so, because a public transport trip, as well as, a trip with any mode of unscheduled travel, except walking will be counted as a trip.

To explore the public transport network, we run one iteration of McRAPTOR. To retrieve the proper input bags for McRAPTOR we associate each stop in the public transport network with a node in the unscheduled walking network beforehand. Then we can use the bags of the nodes in the walking network that are associated with a stop in the public transport network as input for McRAPTOR.

At the same time, we run MLC again, for each unscheduled mode of transport except walking. For modes based on free-floating vehicle sharing, we use the bags of nodes in the walking network as an input, where a free-floating vehicle is present. The output is defined depending on where it is possible to drop off vehicles. If there are no restrictions the bags of all nodes are used.

The outputs of the McRAPTOR iteration and all MLC runs are merged. To do so first the output bags have to be translated into the common nodes of the walking network again. After that the bags are merged according to the merging rules explained in Section 2.2.5.

The bags that result the merge are the output of the first iteration and contain all optimal labels after exactly one trip. To obtain the optimal labels after X trips, both phases have to be repeated X times and the result bags of the second phase in iteration i are used as the input bags of the first phase in iteration $i - 1$.

3.2 Data

3.2.1 Data Collection

Our tool requires several datasets as an input: public transport schedules represented by GTFS files, street networks through OSM files, and data that represents free-floating vehicle sharing. Notably, both GTFS and OSM data can be accessed publicly and can be easily explored, downloaded and preprocessed with the help of our tool.

For GTFS data, 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 (?, ?).

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.

Using the combination of these resources, our tool ensures easy access to up-to-date GTFS and OSM data. This allows for easy reproducibility of our results, as well as, the possibility to use our tool for other cities.

3.2.2 Data Preperation

Our tool is able to trim GTFS data to a specific bounding box. This is especially useful for country-size GTFS feeds.

The GTFS data is also cleaned and converted into a format that is more suitable for RAPTOR.

Specifically, there are two major incompatibilities between the GTFS specification and RAPTOR's notion of routes and trips. Firstly, each trip belonging

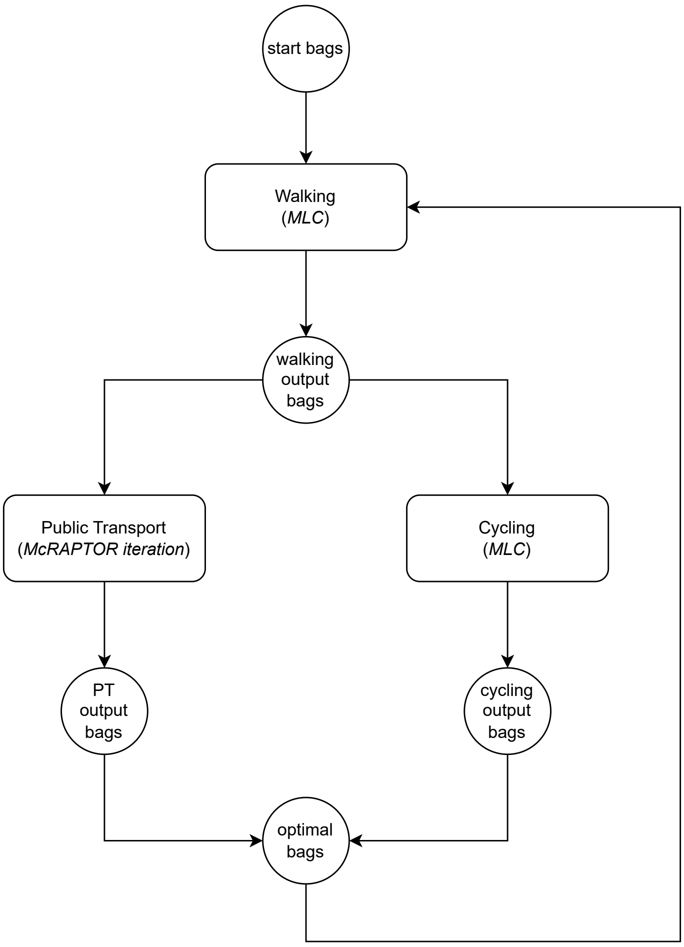

Figure 2: Routing Algorithm

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.

Similarly, our tool is also able to extract an actual graph from the OSM network. To do so it utilizes pyrosm. After extracting the graph from the OSM network, the graph is trimmed to the convex hull of the GTFS stop extended by a small buffer zone. 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.

3.3 Accessibility Analysis

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. (as described by) Each category is populated with Points of Interest (POIs) sourced from OSM, providing a comprehensive database of locations.

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 essential service category for any given area.

This metric is critical in assessing the performance of a neighborhood or a city at large against the 15-minute city ideal. It is not an average of accessibility across services but rather highlights the area of greatest need, providing a clear target for urban development and improvement.

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.

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).

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.

A Appendix

...

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). UnLimited TRAnsfers for Multi-Modal Route Planning: An Efficient Solution. , 16 pages. doi: 10.4230/LIPIcs.ESA.2019.14

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.

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

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

Mitchell, D., Allen, M. R., Hall, J. W., Muller, B., Rajamani, L., & Le Quéré, C. (2018, April). The myriad challenges of the Paris Agreement. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *376*(2119), 20180066. doi: 10.1098/rsta.2018.0066

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

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

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

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