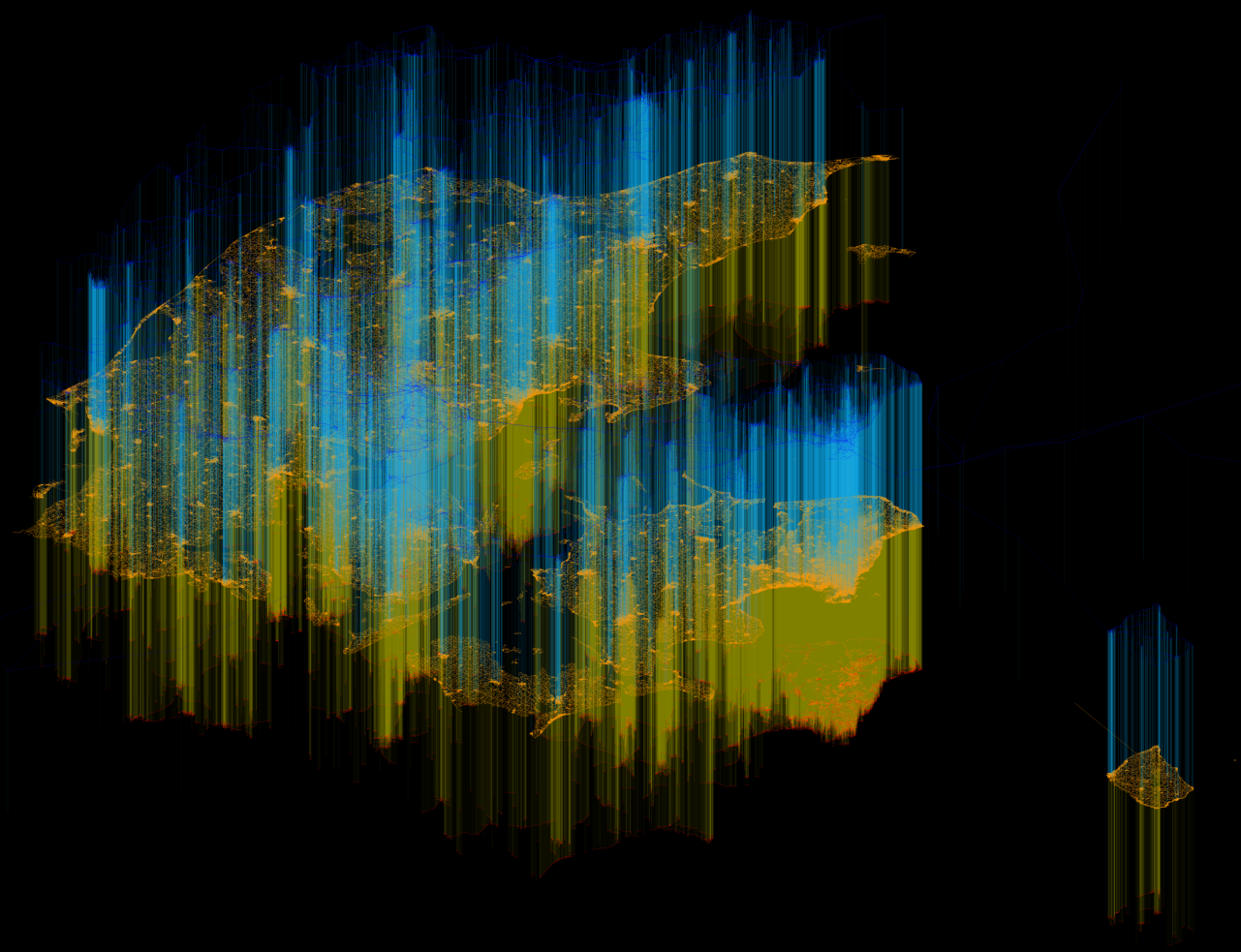


The N-Minute City

Evaluating multi-modal accessibility in Denmark and the United States

STADS: KISPECI1SE

Jan Leonard Schelhaas (jasch@itu.dk) & Peter Gregory Mehler (pmeh@itu.dk)

IT UNIVERSITY OF COPENHAGEN

The N-Minute City

Evaluating multi-modal accessibility in Denmark and the United States

Jan Leonard Schelhaas[1] **and Peter Gregory Mehler**[1]

[1] IT University of Copenhagen

Accurately simulating human mobility can open the door to higher quality and more robust research in city design, urban sociology and more. Towards this goal, many recent papers have included some combination of features to estimate real mobility patterns which consider walking, cycling, public transit, and high resolution routing in their analyses of human mobility. There is an emergent need for a comprehensive method that encompasses all of these features. In this thesis, we address the need for a comprehensive method by presenting a new, efficient open-source tool written in Rust and Python with a parallelized routing engine for fast mobility simulations. We show the potential of our method to highlight the individual contributions of walking, cycling, and public transit by evaluating the accessibility of various locations in Denmark and the US on a new metric, which we call the n-minute statistic. The n-minute statistic describes the median number of minutes needed to access the farthest category of essential point of interest from any origin, and can be used to measure any city's progress towards being an n-minute city. Our results show that in Denmark, bicycle infrastructure contributes to a 12.3% increase in accessibility and transit infrastructure contributes to a 21.5% increase on average. The increases in accessibility from bicycle infrastructure are larger for areas of high population density, while areas of low population density benefit more from the public transit network. Additionally, we estimate potentially large accessibility improvements from expanded bicycle infrastructure and increased transit frequencies across all population densities. The results of our case studies indicate that cycling and transit contribute to accessibility; and therefore suggest that including all modes of sustainable transport is important for comprehensively capturing accessibility. Our tool combines the advantages of previous approaches by including all modes of sustainable mobility and, when combined with our n-minute metric, provides a simple-to-understand and scalable means of evaluating urban accessibility.

1. Introduction

Despite widespread interest in modeling human mobility, there does not yet exist a comprehensive framework which incorporates essential multi-modal aspects of human mobility in high resolution at scale (1). In this study, we provide a new tool * which streamlines the accessibility analysis pipeline from building a multi-modal network to estimating accessibility with our simple metric. Our method for multi-layer network creation, which accounts for walking, public transit, and cycling, can be applied to a majority of cities worldwide, has high resolution for local infrastructure, and includes time sensitive transit routing. We show the potential and validity of our method for more accurate simulations of human mobility and provide an accessible, open-source software package for others to conduct further research.

We answer the following research questions using our multi-modal approach to human mobility:

1. To what extent does the presence of bicycle and transit infrastructure influence the accessibility of essential resources for people living in cities and rural areas in Denmark?
2. To what extent does the influence of bicycle and transit infrastructure on accessibility vary between people living in Denmark and the US?
3. Can we validate our methodology for simulating travel time without the use of proprietary data and closed-source tools?

We give a short overview of our definitions of accessibility and a high level description of our methodology below. To proxy accessibility, we create a metric modeled after the 15-Minute-City idea in urban design literature. The "15-Minute-City" concept is an idea coined by Carlos Moreno, in which all people living in cities should be able to reach all amenities necessary to live a healthy life within a 15-minute travel time (2). In the original formulation, the number 15 was chosen arbitrarily (2). This project takes an analytical approach and replaces the 15 minutes with a variable, n. Our metric, called the n-minute statistic, answers the following question: Given any city, in how many minutes can a resident reach essential services from their residence? More specifically, the n-minute statistic describes the median number of minutes needed to access the farthest essential point of interest (POI) from any origin. See section 4.2 for more details on the n-minute statistic calculation. For determining travel time, we first identify origins and destinations to simulate trips between. Origins are represented by residential buildings, and destinations are represented by buildings with user-defined labels like "supermarket" and "pharmacy." We simplify Moreno's definition of relevant destinations and choose the categories supermarkets, schools, sport facilities, parks, and pharmacies as POIs. See section 4 for details on the POI filters.

We then perform routing between origins and destinations on our multi-modal network which includes the possibility to walk, cycle and use transit. See section 4.1.1 for a complete overview of how we construct the network. Travel times resulting from the routing are recorded from every origin to each origin's closest POI for each category. These travel times are then used to compute the n-minute statistic.

Our data processing pipeline is set up to construct sustainable transit infrastructure and measure accessibility for any city that has the necessary data available. For this thesis, we focus on cities in Denmark and the United States and quantify the effect individual modes of transport contribute to increases in accessibility.

We choose not to include the road network and cars in our analysis because the original formulation of the 15-Minute-City specifically mentions the exclusion of cars in the ideal model (2). Moreno et al. justify the exclusion of cars because of the negative effects of non-active mobility and pollution on human health and the environment.

* https://github.com/hextransit/n-minute-city/

2. Background

There are many urban design paradigms which aim to capture the benefits of human-centered city design (2–5). One in particular has become increasingly popular in the public sphere – the “15-Minute-City”.

2.1. The 15 Minute City. The "15-Minute-City" is an urban design concept coined by Carlos Moreno, which lays out a roadmap for building cities designed for people (2). Moreno suggests building cities which center around distributed local communities, rather than central hubs, as a way to expand human relationships and culture in cities, while diminishing carbon footprints and enhancing public health (6, 7). Going beyond academic discussion, many decision-makers have chosen the 15-Minute-City paradigm as their preferred strategy to attain goals of improving life quality and sustainability (8). A study of 15-minute policies in 2022 by governments of major cities in North America and Australia found that 25 cities had planned or begun implementing policies towards the goal of the 15-Minute-City (8).

Interestingly, the definition of the 15-Minute-City varies greatly by city. Of the 25 cities, there were 11 unique names referencing the 15-Minute-City concept from "walkable neighborhoods" to the "20-minute city" in policy papers and speeches by leaders. This burst of popularity, as well as the variance in naming and measurement, is reflected in academic literature; Nearly every major study has their own method of measurement, often tailored to the design of a specific city (7, 9–13). In the following paragraphs, we will give a short overview of the major papers which have measured a version of the 15-minute statistic, and a short analysis of the pros and cons of each.

A variation of the 15-Minute-City, the 15-minute walkable neighborhood, has been outlined by Weng et al. (7) in which everything relevant to living a fulfilling life should be reachable through walking alone. The authors mention the health benefits of such a model and, of particular note to this project, the authors apply a Walk Score metric which computes the distance to various categories of amenities, weight each category, then add a gravity decay to penalize distance. Ferrer et al. propose a score based on walkability in Barcelona (10). Their 15-Minute-City score is based upon how many of 24 POI categories are accessible within 15 minutes. Gaglione et al. propose a binary approach by identifying areas of Naples, Italy which can and cannot reach all categories of POI within 15 minutes of walking (11). They also provide very fine-grained analysis where walking routes are scored by walk-ability using data such as sidewalk width and expected road traffic. Knap et al. use Open Street Map (OSM) to generate the bicycle network and use a two-step floating catchment area (2SFCA) metric to incorporate both competition for resources and gravity decay (12). Closest to our methodology, Birkenfeld et al. use cycling, walking, and transit in Montreal to identify residential origins which reach essential POIs using active mobility within 15 and 30 minute thresholds. They claim that the 15-Minute-City is not reachable for some North American cities, given that they show changes in the built environment predict limited increase in accessibility (13). Although metrics like 2SFCA likely more accurately capture accessibility, we prefer the n-minute metric for our analysis for its simplicity and interpretability. Keeping the measurement units in minutes, for example, was a deliberate choice for our metric design.

The concept of the 15-Minute-City has gotten more attention recently, after it became the subject of a multitude of conspiracy theories, mostly propagated by conservative influencers and politicians in the United Kingdom and the US (14, 15). There is also criticism rooted in legitimate concern; 15-minute neighborhoods have been criticized as they could lead to an increase in socio-spatial inequalities (7).

In addition to the many methods for defining a 15-Minute-City metric, there also exists a multitude of methods for designing methods to model human mobility. We describe previous approaches to multi-modal routing in the following subsection.

2.2. Multi Modal Routing. In order to more accurately model human mobility, many have proposed routing on a multi-layer network of transport modes (1, 16–21). Gil et al. (19) proposed building multi-modal networks for human mobility using OSM. The authors provide an overview of OSM data quality and confirm the data source’s suitability for the task of multi-modal routing of human mobility. Aleta et al. (16) propose a method to model the various mobility layers. Of particular inspiration for this thesis, the authors describe a method to model transit layers, in which each transit route has its own layer in the network. Each transit route being on its own layer then allows for more accurate modeling of transfer and wait times. We implement this solution to our own network in section 4.1.1. Huang et al. (21) provide an overview of the different methods to incorporate time into the construction of the transit layers, and propose a "fuzzy" method to model non-static networks like those of ride-sharing services. All the previous methods generate the network directly from data of real world infrastructure, however, mobility networks can also be constructed directly from mobility traces.

We gave considerable thought into choosing between network generation from existing infrastructure, or network generation from measured mobile phone GPS signals. Zhang et al. 2023 (9) generate a mobility network of Nanjing, China from mobile phone GPS data. Their network architecture is defined by human mobility patterns, which has many benefits in accurately measuring the travel time taken by actual residents of an area. GPS data provides a data-driven approach to generating the network; however, it comes with biases and drawbacks which make it more difficult to adopt widely. Scalability and feasibility may become an issue when comparing cities globally, as GPS data is not widely available for public use. Additionally, biases exist in terms of who is using their phones, who travels at which speeds and more. Given unlimited data, this approach may be preferable to a network generated from transit infrastructure, however with current data availability, we argue that networks generated from transit infrastructure are preferred.

How does our method fit into and expand on previous research? Answering the call for a unified multi-modal transportation modeling method (1) we provide a comprehensive, accessible, and fast open-source tool to build the necessary multi-modal mobility network when conducting research in human mobility. Previous approaches have either not included bicycle infrastructure (16, 18, 20, 21), not used a precise measurement of street infrastructure (16, 17, 20), or not considered real time dependency in their transit routing (19, 20). We provide a method for creating a multi-layer human mobility network which encompasses each of these features, as well as provide an efficient routing engine, and present it as an open-source package for further use. Multi-modal transport provides robustness, reach (for destinations which are far), and speed to residents and visitors of the city. In order to capture the nuance of multi-modality, we implement a multi-layer network consisting of walk, bicycle, and public transport networks.

Data Type	Data Source	Main Tools
Routing Network Walk and Bicycle Nodes and Edges	Open Street Map	Custom Implementation
Routing Network Transit Nodes, Edges, and Timetables	Multiple Sources; Local Transport Authorities	Custom Implementation
Point of Interest Destinations	Open Street Map	Pyrosm, BBBike
Residential Origins	EU Global Human Settlement Layer	Rasterio, OSMnx
Population Density for Various Cities	Multiple Sources; Local Statistics Organizations	Pandas

Table 1. Data sources and main tools used for different data types.

3. Data

In this section we present the selected data sources, explain the rationale behind their selection, and evaluate their advantages and limitations.

3.1. Data Sources. We summarize our data types, sources, and main tools used in Table 1.

During processing, all of our data is converted to a hexagonal ID, using the H3 geospatial indexing system developed by Uber (22). It covers the globe in multiple, hierarchical sets of hexagons. A hexagonal index is an unsigned, 64-bit integer (u64). There are 16 resolutions available, covering the globe in 122 hexagons at resolution 0 and 569,707,381,193,162 hexagons at resolution 15. Each resolution also includes 12 pentagons, which are placed over the ocean and are not relevant for us. The hierarchical nature of the hexagons allows for easy mapping to higher and lower resolutions using fast bit shift operations. An example of the basic use case of H3 is as follows: Two points which are within the same hexagon will have the same ID, and will then be snapped to the center of the hexagon in further analysis.

We use the H3 system to convert all our data into one common, easily mergeable format that may account for some inaccuracies in the original data. When we process at resolution 12, where each hexagon has an edge length 10.8 meters and covers an area of about 300 m^2, we mitigate bias of points being snapped to the center of the hexagon while still gaining computational performance. There are 1,660,954,464,122 hexagons spanning the globe at this resolution. Another advantage is that nearest neighbor calculations are no longer necessary to map data points to each other, as we can instead merge directly on the H3 index. Additionally, we can also use the built-in k-ring functionality, which returns the indices of all neighboring hexagons up to a distance k, to find neighboring hexagons and search in the direct vicinity of a hexagon. This speeds up processing and simplifies the in- and outputs of our functions to u64 values or lists of those. The H3 functionality is especially useful in two cases: (1) The connection of two layers of the multi-modal network occurs where nodes of two layers share the same H3 index. (2) Any additional metadata added to the network, like origins and destinations for routing, are easily added by simply identifying the metadata's shared H3 indices with the network.

We rely on data from three sources to build the network and identify origins and destinations:

- Satellite raster data from the European Union's global human settlement layer (GHSL) to identify places of residency.
- Public transport timetable data in GTFS (general transit feed specification) format, sourced directly from local transit agencies
- Open Street Map (OSM) data in binary protocol buffer format (osm.pbf) downloaded from extract.bbbike.org for the extraction of points of interest and the network generation. We initially wanted to use pyrosm to automate destination extraction but encountered high RAM usage and high computation time when filtering with a bounding box (23).

To identify origins for our multi-modal routing, we use the EU Global Human Settlement Layer (GHSL) data for its scalability and comprehensiveness. Specifically, we use the GHS-BUILT-C-R2022A (24), a dataset which has human settlement boundaries as raster images covering most of the world at a resolution of 10 meters per pixel. The dataset has been split into 1000 by 1000 kilometer area sections. Each pixel has 25 possible labels with various characteristics, including whether the given pixel is a residential or commercial building. Given the constraint that our solution needs to be globally scalable to identify origins in any city, OSM would have required an exhaustive list of building type tags across many cities to accurately identify residential versus commercial areas. In this way, we are able to avoid manually building OSM tag filters and can better automate our origins pipeline. We use municipal boundaries extracted from OSM to crop the GHSL files, so that processing time is reduced when sampling origins.

A peer reviewed validation of the automatic classification of each area is currently unreleased, and as a result we manually validated the data on Copenhagen based upon a visual inspection of data quality. We verified that major commercial areas and residential areas in Copenhagen were correctly classified.

In line with previous literature (25, 26), the walking and cycling networks are created using Open Street Map (OSM). Adding the appropriate filters to OSM such that the network returned accurate structures was a lengthy and constant process. There are many unique tags in OSM which denote the type or intended use of a "way" or edge. This diversity causes issues in the cycling network in particular, as we had to build an extensive list of tags which should and should not be included. We use our experience and knowledge of Copenhagen to decide when to include and exclude certain roads for cycling and walking, and Google Street View for cities we are not familiar with. The walking network contains all ways that are not exclusive to cars or have restricted access. This means that motorways and trunk roads are excluded from this network. The bicycle network does not include all cyclable paths, but only the ones that feature dedicated cycling infrastructure, with a painted bicycle lane as a minimum. Both networks exclude ways that are marked to have access restrictions. We recognize that it is possible to cycle on roads without dedicated cycling infrastructure, but decided on not including, e.g., residential roads. Our network is comparable to the medium-lower bound network outlined by Reggiani et al. (27).

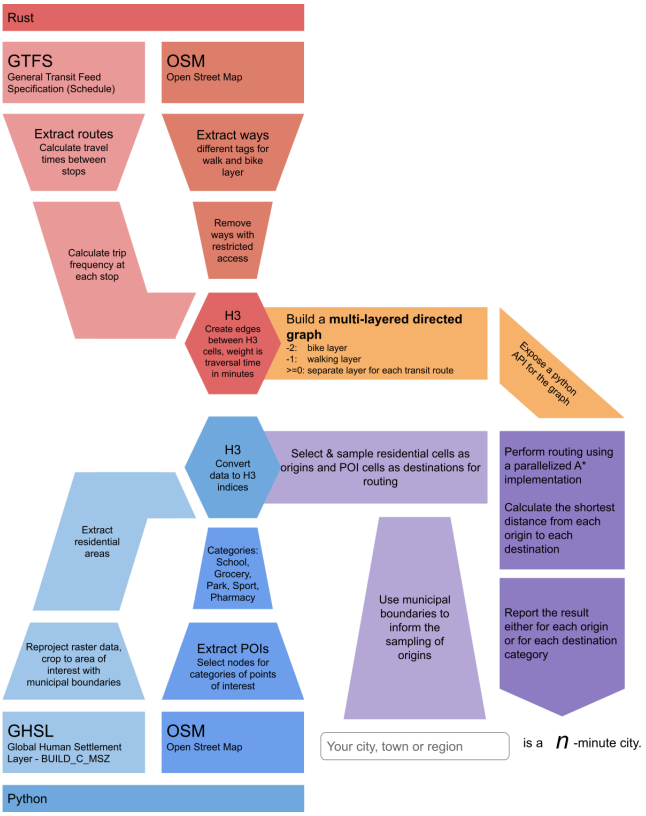


Fig. 1. Processing Pipeline: GTFS and OSM data gets processed in Rust (red) and converted into a Graph, GHSL and OSM data is processed in Python (blue) to determine origins and destinations for routing. The routing is called from Python, but the calculation is done in Rust. A central step in the pipeline is the mapping of all data to H3 indices. Note that once the necessary data sources are available, a potential user needs only to input the name of their city, town or region to generate results.

4. Methods

In this section we provide an overview of our processing pipeline for generating the graph, origins and destinations as well as describe how routing is conducted.

4.1. Data processing. This section covers the entire, custom-built data processing pipeline, as well as the data sources we used. We implemented parts of the infrastructure in Rust for performance reasons, while others remain in Python.

Data processing can be organized into two major components: The network creation and the identification of origins and destinations for the routing simulation. Find a summarized overview of the processing pipeline in Figure 1. The figure shows the two sides of our pipeline, one in Rust, colored red and one in Python, colored blue. All data gets normalized by mapping each point to an H3 index. This enables us to merge data from different sources with ease.

In order to assess the best method for accomplishing multimodal routing, we reviewed available open source-tools in our preceding project (28). There, we used graph-tool (29) as the graph computation library to build our initial tool. As we further developed, it became clear that graph-tool has limitations which would prevent us from conducting our planned analyses. The graph-tool network required a lot of memory (around 60 GB for Denmark) and did not support many-to-many point routing in a fast manner. This means that we have a high number of origin-destination pairs that we need to perform routing for. We wanted to be able to calculate distances in parallel and therefore built a custom, simplified graph computation library that could do exactly that. Using the Rust programming language, we built a library with a Python interface, that stores compact graphs (the graph for Denmark uses around 10 GB of memory now) and allows for parallel read access to that data, so routing can be performed in parallel without copying any data. We go into more detail on that in section 4.3. With our new, custom tool at hand, we were now able to optimize for the exact workloads and tasks we were giving it and could leverage the speed and safety benefits that come from using pure Rust code.

4.1.1. Graph Setup. The multimodal graph is designed as a multi-layer network with a layer for walking, cycling, and a separate layer for each transit line. Transit routes are each on their own layer as in (16) to better model the reality of transfers between lines of public transit. All layers are connected to the walking layer, at points where they share an H3 index. See an illustration of our graph setup in Figure 2. All edges in the graph are directed and are weighted with their traversal time in minutes. Edges are connected from the centers of each hexagon. Using hexagon binning for graph approximation overestimates distance in the worst case by a factor of $\frac{2d}{\sqrt{3}} + \frac{4s}{\sqrt{3}}$. Where d is the distance of the path and s is the length of a side of the hexagon. The first term is explained in Figure 3. The second term represents the added distance in the case where start and end points are as close to each other as possible while still being in their respective hexagons. The extra error is then caused by the hex binning snapping the points to the hexagon centers.

The weighting for the walk and bicycle layer is dependent on the selected walking and cycling speed, while the transit edges are weighted using the difference between arrival and departure time according to the timetable data. The edges that connect the different layers model a mode switch penalty for the bicycle layer and the expected waiting time at transit stations. The penalty for switching to the bicycle layer exists to discourage the routing from switching between layers too

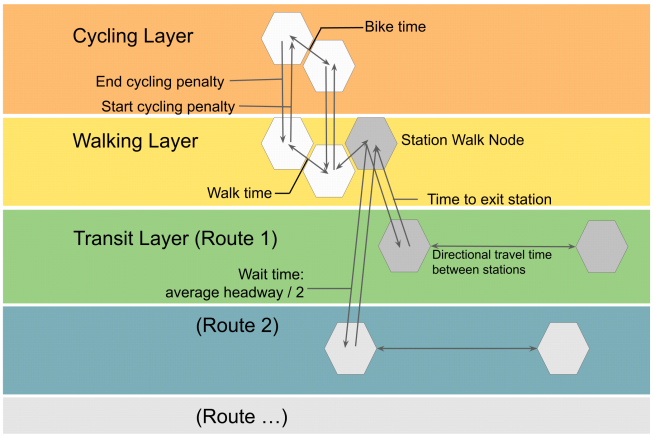


Fig. 2. An example of how we set up the multilayer graph. Every layer uses H3 hexagons as nodes. On the walk and bicycle layer, adjacent nodes are connected by an edge if they are also connected on the OSM network. The walk and bicycle layer are connected wherever they map to the same H3 index, connections to the transit layer are only at transit stops. There are connecting edges to each route, which model the expected weight time for the next departure. Every transit route is considered to be a separate layer. Adapted from (28).

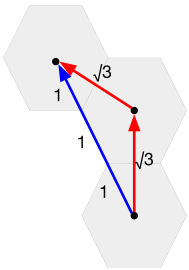


Fig. 3. Illustration of the hexagon worst case distance over-estimation without considering start and end point placement within start and end hexagons. Note that the distance from the center of the shown hexagons to a corner of the same hexagon is 1 unit. Therefore, the total length of the blue line is 3, while the total length of the red line is $2\sqrt{3}$

frequently. The expected wait time is either the best-case wait time for that station and route, or, if specified, the expected wait time at a given time of day. Wait time is defined as $t = \frac{freq}{2}$. We calculate this for each hour for every day of the week. During routing, we can select either the hour of the week, which is a number between 0 for Mondays at 00:00 and 167 for Sundays at 23:00, or just use the best case for that station. The default weighting is based on a walk speed of 1.5 m/s and a bicycle speed of 4.5 m/s which is supported by similar studies (10, 30, 31).

For some analyses, we modify the weights to simulate the impact of infrastructure changes. For example, to model a scenario where dedicated cycling infrastructure is present on all walking paths, we use the walking network but modify the edge weights to match cycling speed. We do this by changing the cycling speed parameter in the graph creation options, which are accessible from the Python interface. We can also simulate an increase in departure frequency for transit routes by changing the weights on the edges connecting to the transit layers.

Note that we define the bicycle layer to only include bicycle paths with dedicated bicycle infrastructure. We chose this to reflect a nice-to-cycle rather than a possible-to-cycle graph, given that the quality of bicycle infrastructure has been shown to induce demand for cycling (32).

4.1.2. GHSL processing. After graph creation, we identify residential areas as origins and relevant points of interest as destinations to use in our routing simulation. See the Python section of Figure 1 to place this methodology in the overall processing pipeline. We identify residential areas as origins in order to avoid distorting the n-minute calculation by simulating travel times for paths which are not traversed by residents in reality. Choosing random nodes in the network as origins could exaggerate the inaccessibility of non-residential areas. For example, if we randomly choose a network node in an industrial area with a population of 0, the likely low accessibility of this area will be included in the overall n-minute statistic. Thus, we argue that the identification of residential areas is essential for accurate measurements of the n-minute statistic.

Origins are extracted from GHSL data in 4 main steps.

1. Crop GHSL with bounding box from municipal boundaries
2. Reproject the cropped GHSL
3. Convert cropped reprojection to H3
4. Compute H3 set overlap with municipal boundaries

Each GHSL raster contains 10 billion pixels, and thus the first step to space and time efficiently process them is to crop the original grid to a bounding box just large enough to encompass the municipal boundaries of the city in question.

We generate the boundaries of the city by querying the function *geocode_to_gdf*() from the OSMnx Python package with a list of city names. City names have to be as they are listed in Open Street Map. For example, Copenhagen must be "København Kommune" to access the correct boundaries. All city name query, boundary pairs were plotted to ensure that the correct area was extracted. Additionally, functionality exists for multiple cities. For example, if one wanted to include Frederiksberg, which exists entirely within the boundaries of Copenhagen, both cities can be used as input in a list to return combined overall outer boundaries.

Using the extrema coordinates in all cardinal directions, we create our bounding box for cropping the GHSL raster. Before cropping, we reproject the bounding box to the GHSL Coordinate Reference System (CRS).

After cropping the GHSL raster, we then reproject the cropped section to $EPSG : 4326$ to be able to use standard latitude and longitude in further analysis. We then take all latitude and longitude coordinates for each pixel of the raster and convert them to their corresponding H3 index at a given resolution, keeping only those with residential labels.

To obtain only the H3 indices within the municipality boundary of interest, we create two sets of hexagons. (A) All possible hexagons contained within the municipal boundaries, which we convert from a Shapely (33) polygon after extracting them from OSMnx, and (B) The hexagons representing the residential origins. The final list of origins is the intersection between the two sets, $A \cap B$.

4.1.3. OSM processing. To generate our destinations, we extract an Open Street Map bounding box as Protocol Buffer Format (pbf) file from extract.bbbike.org, then query the resulting region with a filter to retrieve the relevant origins.

Edge effects have the potential to bias results in regions near borders by not taking into account interactions across boundaries (34, 35). Because we have access to data beyond the municipal border in question, accounting for edge effects can be done by expanding our destination boundaries well beyond our municipal boundaries for origins. As a result,

the destinations bounding box was chosen to be 5 kilometers beyond the extreme bounds of the residential coordinates for origins in all cardinal directions. The wider bound avoids many possible edge effects, as points of interest beyond the city limits are still accessible to residents and better reflect the reality of people living near municipal boundaries (34, 35). This means that, e.g., a supermarket in a neighboring municipality is considered in our routing. Within the bounding box we collect all points of interest with corresponding shapes, latitudes, and longitudes. To be able to do routing, destinations should be points with specific tags as destinations. The next steps are therefore to filter for relevant tags and convert all geometries (polygons, lines, and multilines) to single points. These single points are then mapped to respective H3 indices to be used as destinations.

Destination types are selected based on a filter applied to OSM data. We define our filter by simplifying the list of POIs designed for the original 15-Minute-City concept by Moreno et al. (2). Specifically, we do not include cultural, entertainment, or work related destinations. Our resulting category set is therefore pharmacy, park, supermarket, sport, and school. Although we chose a specific filter for this study, we leave the tag filter as an editable parameter in our software package, allowing future studies to modify which POIs are relevant for their purposes.

OSM data related to network structure is processed in the Rust side of the pipeline, see section 4.3.

4.2. The n-minute Calculation. In this section, we describe the method used for measuring the n-minute city. For a more detailed discussion of the pros and cons of aggregate statistics, see section 6. In Moreno's recent paper defining the 15-Minute-City, the ideal city is described as a place where people "are able to access all of their basic essentials at distances that would not take them more than 15 min by foot or by bicycle" (2). We interpret the use of "more than" to be taking a maximum. As a result, for a given residential origin, we define the n-minute statistic as the maximum of all shortest paths to each category of point of interest. See an illustration for a single origin in Figure 4. For example, if we take an apartment building to be an origin, we begin with a category, like supermarkets, and record the travel time to the nearest supermarket. We continue by computing the minimum travel time from the apartment building to the nearest pharmacy, park, etc. We then take the maximum of these, which could be, for example, the travel time to the nearest pharmacy. This is now the n-minute statistic for a single origin in our graph. For an entire city, we then aggregate all the n-minute statistics of each origin by taking the median of all origins. A main design decision of our n-minute statistic is that we determine n by looking at the point of interest that is furthest away from the origin out of all the nearest points from each category. This is a rather simple way of calculating accessibility, and other approaches have been implemented by others, such as weighting the different categories based on their importance (26). We discuss the impact of this design decision in section 5.2.2 and 6 and consider variations of the n-minute city measurement an area of future work. A more precise definition of our n-minute statistic follows below.

Let A be a $k \times l$ matrix where k is the number of origins, l is the number of categories, and $A_{i,j}$ is the length of the shortest path from origin i to destination category j.

If $\max(A_{i,1}, A_{i,2}, \ldots, A_{i,l})$ is the maximum distance of the i^{th} origin of A, then $\max_{\text{origin}}(A)$ is a column vector of length k containing the maximum shortest path of each origin of A.

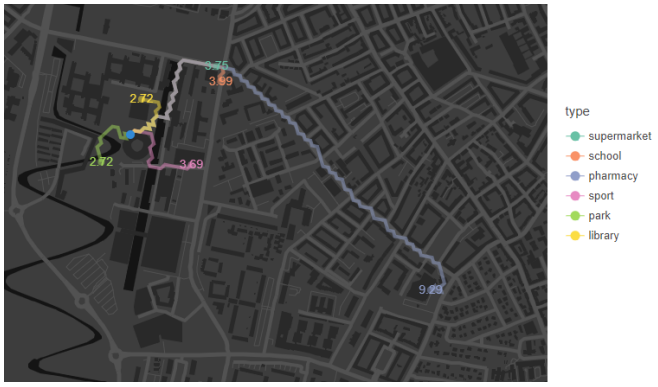


Fig. 4. Example routes from a single origin (blue) in Copenhagen to all nearest points of interest. The travel time in minutes is shown on the destination dots. Each path is a route on the hexagonal network from the origin to the nearest POI of each category. Only the path to the pharmacy utilizes the bicycle network. The distance to the nearest pharmacy is also what defines the origin's n-minute value, 9.29 minutes, as this is the essential POI that is furthest away.

Formally,

$$\max_{\text{origin}}(A) = \begin{bmatrix} \max(A_{1,1}, A_{1,2}, \ldots, A_{1,l}) \\ \max(A_{2,1}, A_{2,2}, \ldots, A_{2,l}) \\ \vdots \\ \max(A_{k,1}, A_{k,2}, \ldots, A_{k,l}) \end{bmatrix}$$

A city's n-minute value is defined as the median of all $\max_{\text{origin}}(A)$.

Figure 4 shows an example of the n-minute calculation for a single origin on the walk and bicycle network. For this point, the routing utilizes the bicycle layer for the path to the pharmacy. Due to the bicycle penalty for beginning and ending cycling trips, all other routes are short enough and only use the walking layer. The bicycle penalty is a weight on the interlayer edges between the walk and bike network, used to discourage the routing from switching between layers too frequently.

4.3. Parallelized graph computations in Rust. To facilitate fast and memory efficient computations of the n-minute city, we implement a generic graph computation library in Rust. Rust is a low-level, strongly typed, memory and thread-safe programming language that offers comparable performance to C or C++ while avoiding many of the common errors that C and C++ code can produce, due to its unique ownership model for memory management. It also offers easy ways to integrate with Python through PyO3 (36). Our goals are to enable fast n-to-n point calculations and to reduce memory usage over our previous graph-tool implementation. To achieve that, we needed a graph data structure, that allows for multithreaded read access, so that multiple routing simulations can be performed at the same time. Additionally, all data should be stored in a compact format to reduce memory usage.

The graph is backed by an adjacency list, which is implemented as a HashMap with an optimized hash function for sequential unsigned integer indices. Access to the adjacency list is guarded by an Arc (atomic reference counter) and a RwLock (read/write lock). The Arc enables concurrent access to the data, while the RwLock controls the type of access. In our implementation, write access is granted for edge creation, while all other functions only require read access. Nodes are stored in a vector and are referenced using their index. The graph supports node and edge deletion, but will not shift indices when doing so. This means that memory usage will increase if a lot of node additions and deletions are performed

over time, but for our use case this is not relevant. We are not modifying the graph after creation. The graph is designed to be generic and supports any data type for nodes that fulfills a certain list of traits. In our case, a Node is of type H3Cell, with additional metadata being supported but not used. An H3Cell holds an H3 index and a layer. The memory footprint of one H3Cell is 10 bytes, with 8 holding the H3 index and 2 being allocated for the layer. An edge references two nodes by their index in the node list and holds additional, optional information, such as the weight and capacity. We use a bidirectional HashMap to allow for faster mappings between H3 values and node indices. We implemented both BFS and A* (A-star) as pathfinding algorithms. In addition to the basic implementation for both algorithms that returns a path, there is also a parallel version that returns the distance for all valid origin-distance pairs that are input. That version will use all available threads to calculate distances and scales by the number of origins provided. Our usage of A* is equivalent to using Dijkstra's algorithm, as we are using a constant heuristic of $h(x) = 0$ when searching for multiple destinations.

The worst-case time complexity of the parallel A* algorithm is $O(n \cdot |E|)$, with n being the number of origins. The underlying graph is not copied for parallel operations, but each routing still needs at worst $O(|V|)$ space. This makes the worst-case space complexity $O(n \cdot |V|)$.

In addition to the generic graph implementation, we added functions for processing GTFS and OSM data to the Rust crate as well. For those, we utilize third-party crates to ease reading in the files. The processing of OSM and GTFS is parallelized using rayon (37). The conversion into H3 indices is handled by the h3o (38) library. To combine the GTFS data with the OSM graph, we added a function to merge multiple graphs based on H3 indices. The merging function allows us to combine any number of OSM and GTFS files without creating duplicate nodes. The graph creation process for all of Denmark takes about 100 seconds on our test server. See the hardware specifications of that server in appendix B.To make sure that the graphs are consistent between runs, we implemented a function that hashes the nodes, and we print that hash when creating a new graph. To integrate the graph library as well as the OSM and GTFS parsing with the rest of our infrastructure, we utilize PyO3 to expose a Python interface for some of our functions. This allows us to create a graph and perform routing from Python, while the data and computations lie in Rust. Each time we update the graph library, we generate a pip-installable wheel using GitHub Actions for both Linux and Intel Mac. On the Python side, we support using the u64 representations of H3 indices as input to the routing functions.

The experience of using the graph library looks like this:

```
graph = PyH3Graph(weight_options={}, k_ring
    =2, layers="all")
graph.create(osm_path="denmark.osm.pbf",
    gtfs_paths=["rejseplanen.zip"])
```

We provide some debug output on graph creation as well as the hash of the finished graph:

```
processing osm pbf file: denmark.osm.pbf
converted OSM file into 14356812 edges
osm graph created with 13470012 nodes in
    87.322426 s
getting GTFS feed from rejseplanen.zip
routes: 1631
gtfs graph created with 103525 nodes in
    9.1905575 s
merged gtfs graph into osm graph, now has
    13546387 nodes, took 120 ms
hash: 1352011021897877533
```

The graph can be customized by selecting layers, the k-ring search radius for mapping an H3 input to nodes and options for weighting. A k-ring of 1 includes all six neighbors of the hexagon, at k=2 it contains the neighbors of the neighbors. We use the k-ring to map origins or destinations that are not in the graph to a node that is. Previous literature has used nearest neighbor search with a threshold of 100 m to establish interlayer connections (16). Using k-ring search with sufficient k-rings would produce an equivalent outcome to the previous method. Optimization of the k-ring parameter is an area of future work. By default, we use a search radius of 2, which translates to a ring with a diameter of about 40 meters. The weight options allow the customization of walking and bicycle speed, set a multiplier to the wait time at transit stations, and specify the penalty value for switching to and from the bicycle layer. The hash is used to ensure that re-processing the same data results in matching graphs. The hash is not stable across different versions of the graph library.

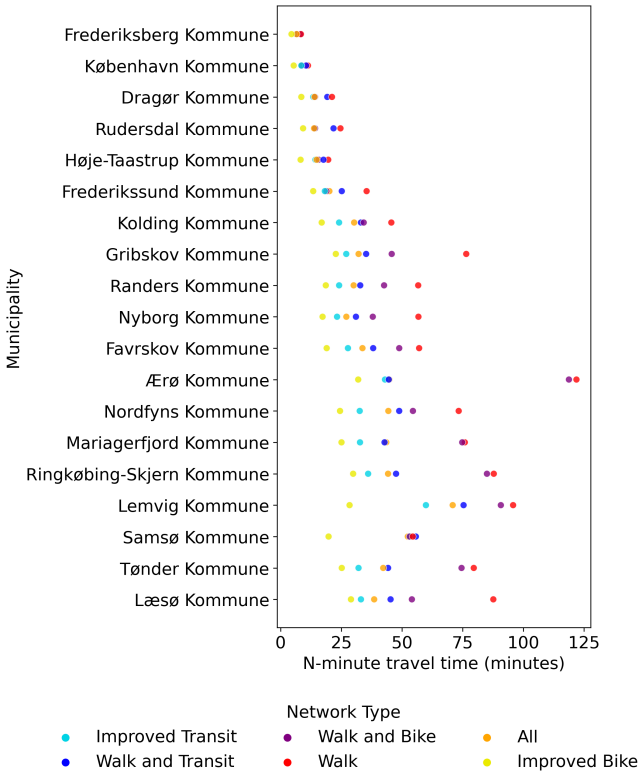
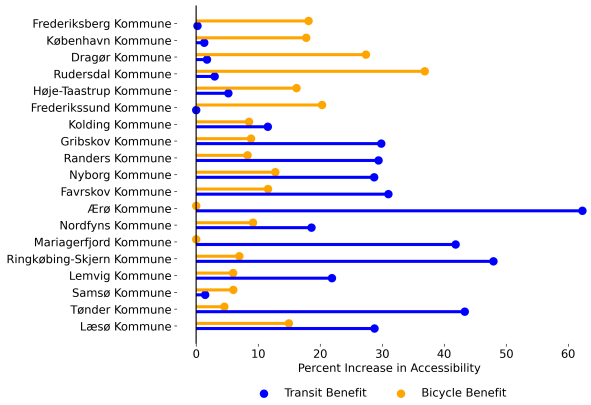
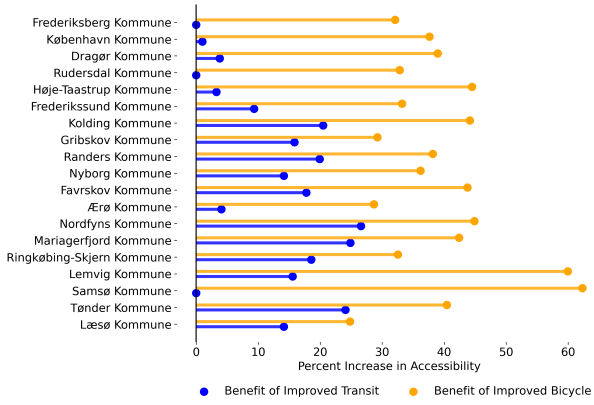


Fig. 5. The n-minute statistic is shown for various network types in various Danish municipalities. Municipalities are sorted by descending population density. Ærø Kommune, an island in southern Denmark, is clearly an outlier, and can be explained by the municipality's geography and POI spatial location. An inspection of the POI spatial distributions reveals that Ærø's single pharmacy on the island is on the east coast. The East/West span of the island is nearly 30 kilometers, and, with limited cycling infrastructure, Ærø becomes very difficult to traverse without public transit.



(a) Percent accessibility benefit contributed by current bicycle and transit infrastructure. The municipalities are ordered by population density in descending order.



(b) Percent accessibility benefit contributed by improved bicycle and transit infrastructure.

Fig. 6. Accessibility benefit plots for (a) current and (b) improved infrastructure.

5. Results

We measure the n-minute statistic for several Danish municipalities, Denver and Los Angeles with varying mobility infrastructure to show the potential effects multi-modality has on accessibility, and to highlight accessibility inequalities between countries, municipalities, and geography types.

The following section presents the results of our travel time estimates in Denmark and the United States with varying mobility infrastructure. We also identify potential explanatory factors contributing to the measured n-minute statistic.

5.1. Denmark: Municipality Comparison. We randomly sample 3 Danish municipalities for each of 5 population density quantiles in a population of 98 municipalities, then take the top 2 and bottom 2 municipalities for a resulting 19 in our n-minute measurement for Denmark. Densities range from 14.5 people per square kilometer for Læsø Kommune to 11909.0 for Frederiksberg Kommune. The mean population density is 600.1 while the median is a much lower 119.3. The majority of Danish municipalities are less dense, more rural areas, while the outlier dense urban areas of Copenhagen and Frederiksberg raise the mean.

For each of our 19 selected municipalities, we compute the n-minute statistic for varying combinations of infrastructure types. These infrastructure types and their descriptions are the following:

- *All* - Includes walking, biking and transit layers.
- *Walk* - Includes only walking layer
- *Walk and Cycle* - Includes only the walking and cycling layers.
- *Walk and Transit* - Includes only the walking and transit layers.
- *Improved Cycling* - All walking paths are modeled as also including dedicated cycling infrastructure.
- *Improved Transit* - All layers are included, and transit average wait times are reduced to 33% of their original values.

We provide networks with altered and or removed layers to be able to isolate the effect of certain layers. In Figure 5 we show the calculated n-minute statistic for all municipalities in Denmark for all network types. By comparing the network with all layers to *Walk and Cycle*, we can isolate the benefit that transit contributes to the improved accessibility. For example, for Favrskov Kommune, the network type *All* shows a decrease in the n-minute statistic of 4.4 minutes over the *Walk and Transit* layer. We can interpret this as current cycling infrastructure contributing a 4.4 minute decrease in the median maximum travel time across POI categories and a 4.4 minute increase in accessibility. Our results show current cycling and transit infrastructure contribute to reduced travel times in all areas. We can clearly see that the inclusion of various infrastructures alters the level of measured accessibility, suggesting that without inclusion of all layers, we may be underestimating accessibility.

Although we see increases in accessibility in all areas from the inclusion of bicycle and transit layers, the increase is not constant across all population densities. We can see in Figure 6a that areas of high population density receive the largest percentage decrease in travel time when including cycling infrastructure, and less dense areas receive the largest decrease in percentage travel time from public transportation. Contributions of cycling to accessibility decrease as population density falls, whereas contributions of transit to accessibility increase as population density falls. The preceding analysis estimates to what extent current infrastructure is contributing to accessibility. We also use our new methodology to simulate the effect of potential changes to cycling and transit infrastructure.

We consider our two cases of infrastructure improvement, using network types *Improved Cycling* and *Improved Transit*. Figure 6b shows that both network types show improvements across population densities, with the main exception of *Improved Transit* giving relatively minimal advantages in areas of high population density. This makes sense, given that transit wait times in areas like Copenhagen are already very low, so the benefit of their reduction is likely small. On the extreme end, *Improved Cycling* shows reductions in the n-minute statistic for rural areas as high as 57%. We hypothesize, that the large effects produced by our model are a result of a lower priority and therefore a lack of sufficient infrastructure for cycling and public transit as opposed to car transport in rural areas. A manual inspection of the network in rural areas shows that bike and transit infrastructure are very limited. The routing engine then chooses between walking to find the nearest bike path or waiting very long wait times for low frequency transit. When adding the *Improved Cycling* layer, all walking to transit is replaced with cycling. Because cycling speed in our model is 3 times that of walking, we expect this to be a large source of the simulated travel time reductions. Larger sample sizes in Denmark and beyond are necessary for more robust results in this area.

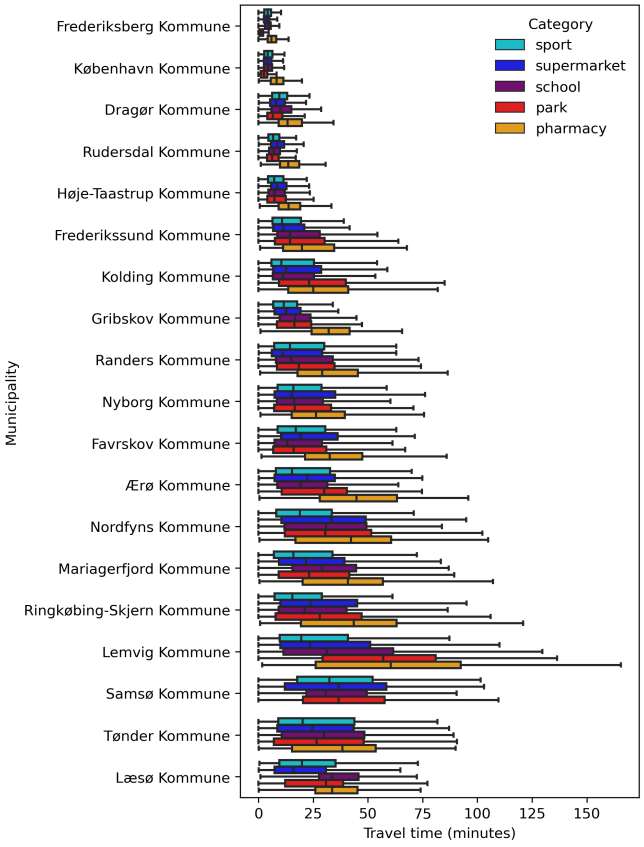


Fig. 7. Breakdown of Danish municipalities by travel time to each POI category using current infrastructure. Boxplots represent the median, first Inter-Quartile Range (IQR) and 1.5 times the IQR for the center lines, box ends, and whiskers respectively. Outliers are not shown. Note that at the time of plot generation, Samsø did not have a building tagged as a pharmacy, and so we leave out this category in the calculation.

The single aggregated n-minute statistic at the level of municipality hides the geographic distribution of travel times for each POI category. We therefore show box plots of travel times (note, these are not the n-minute statistic) to each category of POI for Danish municipalities in Figure 7. Variance becomes extremely large in the more rural areas, partially explained by the decrease in population density. Interestingly, the final 4 least dense municipalities see a slight inverse relationship between density and the distribution of n-minute statistics. Upon further inspection, Lemvig Kommune simply has very few pharmacies, low density, and many rural towns with limited public transport causing its very high pharmacy travel times. Samsø and Læsø are both small islands. We hypothesize that even though their densities are low, the island geography sets physical bounds on the furthest distance one must travel to reach POIs, and as a result they do not have high outliers. Finally, Tønder seems to have very well distributed town centers compared to Lemvig, thereby reducing the chance of having very remote residential areas. Further analysis of the factors which contribute to the n-minute statistic is an area of future work.

The n-minute statistic is sensitive to the category with the maximum travel time for each origin. We show the travel time distributions by category for Denmark overall in Figure 8. Clearly, pharmacies are the least accessible POI in Denmark. In fact, pharmacy ends up being the category which determines the n-minute statistic in 58.7% when taking the max across categories for each origin in Denmark as a whole. In the other extreme, even though sport POIs are on average the closest POI to our origins, supermarkets are least often the farthest

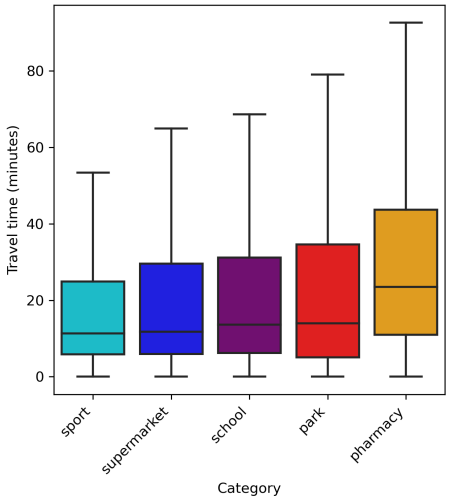


Fig. 8. Travel time box plots for Denmark Overall. Boxplots represent the median, first Inter-Quartile Range (IQR) and 1.5 times the IQR for the center lines, box ends, and whiskers respectively. Outliers are not shown.

category. Supermarkets are the farthest POI for only 4.5% of origins across Denmark.

Overall, our results suggest bicycle and transit layers contribute to the n-minute statistic for a given location. When comparing the n-minute statistic across municipalities, we see a clear negative trend between population density and the n-minute statistic. This begs the question, what is the quantified relationship between the n-minute statistic and population density? To address this, we fit a univariate regression and find a strong log-linear relationship between population density and the n-minute statistic. See Figure 13 in the appendix. More specifically, the log of population density explains 90% of the variance in the log of the n-minute travel time. Additionally, the Pearson correlation between the two variables is -0.95. These results point to a strong relationship between population density and the n-minute statistic. We hypothesize higher densities bring POIs and network features closer together, thereby increasing accessibility. Less dense areas likely benefit more from an increase in network density, e.g., when creating more safe cycling paths through a suburb.

5.2. Case studies. In this section, we highlight a few select cities in Denmark and the United States and look at how they differ in the n-minute statistic. We give explanations by breaking down the calculation by network type and categories of points of interest.

5.2.1. Copenhagen, Denmark. For the purpose of this case study, we include Frederiksberg Kommune in our analysis, as both Frederiksberg Kommune and København Kommune form the urban core of the Copenhagen area. Figure 14 in the appendix shows sampled origins, which are used for the n-minute calculation. Origins are sampled from the list of all residential H3 indices, which were collected by processing the GHSL data. Most calculations were done using 10,000 origins, except for Dragør Kommune, where we only used 1000 origins due to its small size.

We find that Copenhagen can be described as being around an 8-minute city if we round down with our definition of essential POIs. We break down the n-minute values by network type in Table 2. The table shows that even when including less common POIs like libraries in the list of destinations, the values remain low. We find that adding the option to cycle improves the n-minutes, but the impact of allowing public

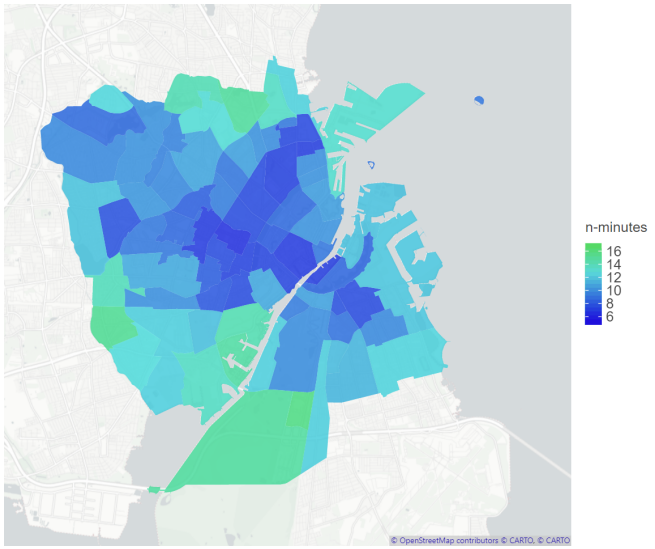
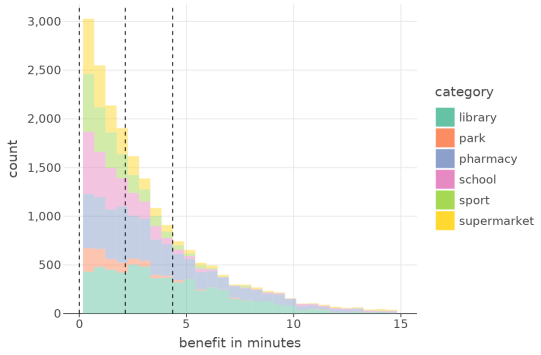


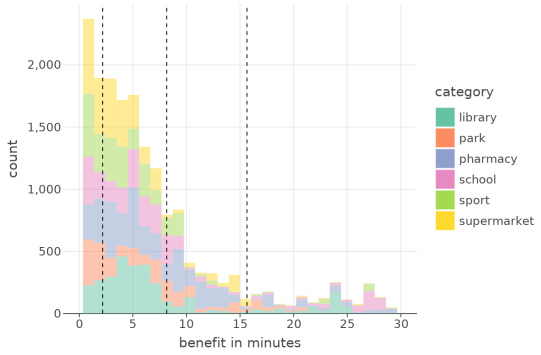
Fig. 9. N-minute values for Copenhagen and Frederiksberg, using the full network and a simulation of 10,000 origins. The map shows the median for each neighborhood.

Network type	Essential POI's	Essential POI's and libraries
Walk	10.28	13.59
Walk and Cycle	8.41	10.41
Walk and Transit	9.90	11.84
All	8.34	10.08

Table 2. n-minute values for Copenhagen, for only essential POI's and including libraries



(a) Travel time benefit simulation of 10,000 origins for **Copenhagen** and **Frederiksberg**. 30,000 trips with no benefit are not displayed in this plot.



(b) Travel time benefit simulation of 1000 origins for **Dragør Kommune**. Trips with no benefit are not displayed in this plot.

Fig. 10. Travel time benefit of the *All* network over just the *Walk* network for (a) Copenhagen and (b) Dragør. The three dashed lines mark the 0.5, 0.8 and 0.9 quantile.

transit only becomes clear when adding less frequent points of interest. After removing outliers, which are values greater than the 95th percentile, the worst-case travel times we see are about 17 minutes on the full network.

The map in Figure 9 shows the n-minute values over all origins in Copenhagen. We can see that in most parts of the city, the n-minute values are below 10 minutes, with some areas that are further away from the city center reaching a value of 17 minutes. Especially the inner city as well as parts of Nørrebro and Østerbro produce very low values, with the neighborhood median being around 7 minutes.

When investigating the difference between the trip lengths for each origin on the four network types, we can see that for most origins, there is no or almost no benefit in travel time when given the option to bicycle or use transit. For 80% of trips, the option to use the full network results in either no travel time benefit or a benefit of two minutes or less. For 90% of trips, the benefit is less than 4.5 minutes. The histogram in Figure 10a shows the travel time benefit of using the *All* network over the *Walk* network. It also shows that almost all trips with a travel time benefit over five minutes are to libraries or pharmacies.

Dragør Kommune, which lies just outside the city of Copenhagen on the island of Amager, displays a slightly different situation. For Dragør, there is a benefit of 2 minutes or more for half of the trips. 20% of trips see a benefit of over 8 minutes, while 10% see a benefit of over 15 minutes. Figure 10b shows the histogram of travel times for Dragør. It leads us to the conclusion that using multiple transportation modes such as public transit and cycling provide the most benefit in under-served areas when it comes to access to essential services. In dense urban areas such as the city of Copenhagen, most points of interest are reached by walking, and public transit would only gain relevance for longer trips across the city.

For Dragør, we investigate the effect of the time of day on the n-minute statistic. We look at the median and the 95th percentile at two different times of a week. The median is what we report as the n-minute statistic, but to see the effects of reduced transit service, we also report the 95th percentile of travel times. We investigate Monday morning at 9:00, which is a time of high service where many are using transit for commuting, and on Sunday morning at 1:00, where most bus lines are no longer running. We acknowledge that this is not a time of day when the points of interest we selected are relevant or open, this analysis is more focused on the difference in our statistic caused by reduced transit service. The median travel time at both points in time, Monday morning at 9:00 and Sunday morning at 1:00, is very similar with 13.1 and 13.2 minutes respectively. But when looking at the 95th percentile, we can see a clear difference; the values are 29.8 minutes and 41.9 minutes. This shows again that transit plays an important role for under-served origins that are further away from points of interest.

5.2.2. United States. We choose to look at Denver, Colorado, because it has a similar population to Copenhagen. Notably, Denver covers roughly double the land area as Copenhagen does. Table 3 shows the values for Denver. The option to use the bicycle network results in a significant travel time benefit when comparing to the walk network. When breaking down the results by category, we see striking differences between Denver and Copenhagen. While sport facilities and parks are in close

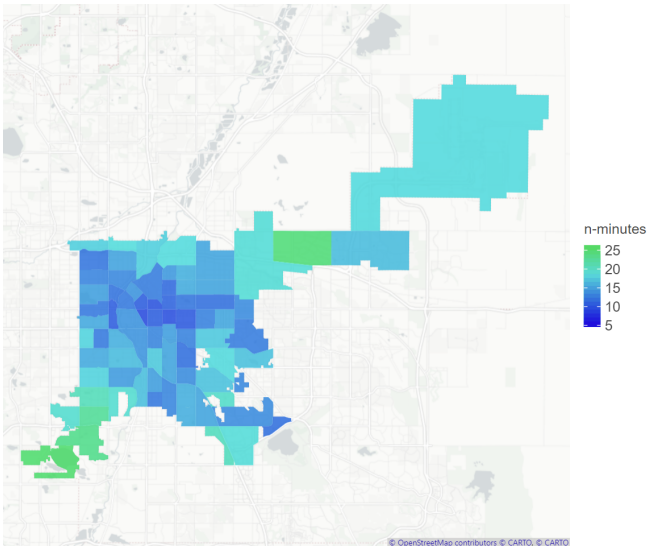


Fig. 11. n-minute values for neighborhoods in Denver, CL. The simulation used 10,000 origins and the full network. Note that the top-right neighborhood is the Denver Airport. We investigated origins identified by the GHSL data in this region, and some seem to be inaccurate. The Airport only appears in simulations that include the transit network, as it is not accessible by bicycle or foot and includes none of the selected amenities.

reach for almost all origins in both cities, the travel times to supermarkets are much longer in Denver than in Copenhagen. In Denver, the median travel time to a supermarket is 10 minutes on the full network, as opposed to just 4 minutes for Copenhagen. The map in Figure 11 shows that the city center of Denver is accessible, with values as low as 8.7 minutes. The more suburban parts of the city are doing noticeably worse on this metric.

We also compare Denver to Los Angeles, known for being one of the most car-centric cities in the US (39). See the exact values for Los Angeles in 4 in the appendix. The n-minute values for Los Angeles are about 1.5 minutes more than Denver's. Much of that difference can be explained by a higher value for parks and sport facilities. On the full network, travel time to sports and parks for Los Angeles have a median value of 9 and 7 minutes respectively, while in Denver the median values are 4.3 minutes for parks and 6.7 minutes for sports.

The n-minute statistic is dominated by the category of POI's with the worst travel time. We find that for Los Angeles, 56% of origins report the distance to the nearest pharmacy to be the highest. For 16% of origins, the distance to the nearest supermarket is the dominating metric. For Denver, the distribution is 65% for pharmacies and 19.6% for supermarkets. Given that most people need to visit a supermarket more frequently than a pharmacy, this statistic is relevant. One can argue that the perceived n-minutes for an origin would divert from our statistic, both when a rare POI is the dominating factor, but also when a frequently needed POI is far away. Creating more individualized versions of the n-minute statistic is as an area of future work.

This in-depth look into the n-minute of specific neighborhoods reveals the disparity in accessibility hidden by city-wide n-minute statistic. Further analysis into the inequalities revealed by the n-minute statistic is an area of future work.

5.3. Rejseplanen routing comparison. To ensure that our model produces sensible travel times, we compare our outputs to the routing service provided by Rejseplanen. Rejseplanen is the organization that manages transit timetables and routing

Network type	Essential POI's	Essential POI's and libraries
Walk	17.47	21.16
Walk and Cycle	13.38	15.42
Walk and Transit	16.23	18.83
All	13.20	15.15

Table 3. n-minute values for Denver, for only essential POI's and including libraries

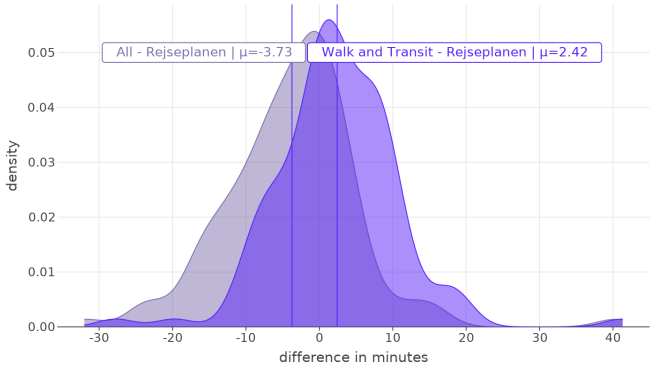


Fig. 12. Distribution of routing time differences between our full network (All) and Rejseplanen. The plot shows the two distributions of routing time differences between our full network (All) and Rejseplanen, as well as our network with only walking and transit and Rejseplanen.

services for all Danish transit providers. We find that our model produces comparable travel times to Rejseplanen. Our All network creates slightly faster travel times on average, while the network without the option to cycle, *Walk and Transit*, is on average 4 minutes slower than Rejseplanen. The evaluation is performed by sending 110 random routing requests within the bounds of Copenhagen to the Rejseplanen API, via a Cloudflare worker proxy. The origins and destinations were random nodes from the graph. The Cloudflare worker we built converts the H3 indices into latitude and longitude and sends the request to Rejseplanen in the correct format. It returns the original Rejseplanen response in JSON format. Using the worker makes requests to the API more intuitive but provides no performance benefit. For comparability reasons, we set the time of day parameter in our routing to be the current time, which is the one Rejseplanen uses for its routing. The comparison was performed on a Monday morning. Routes where Rejseplanen returned no result were omitted. We only selected the first result that was returned by the API.

Overall, we find that our full network is on average 3.7 minutes faster, with a standard deviation of 9 minutes, while the network without the option to cycle is on average 2.4 minutes slower, with a standard deviation of 8.4 minutes (Figure 12). This could be explained by the fact that we, as opposed to Rejseplanen, do not perform timetable-based routing but instead model the expected wait time. Figure 15 in the appendix shows that for most routes, the calculated travel time is similar, however there are a few outliers. Rejseplanen will not recommend cycling routes by default, so we assume that some routes are faster when allowing for cycling. Although this does not suggest anything conclusive about our routing compared to real travel times, the analysis does show that our routing system predictions are comparable to an authoritative routing source. A more rigorous analysis with a larger sample size would be necessary to provide further validity to our model.

6. Discussion

Ultimately, the effects of travel time reductions are likely to be felt on a much smaller scale than entire cities. Adding or removing infrastructure such as single bicycle lanes or transit lines happens on a per-neighborhood basis, and thus the n-minute city single statistic (one number for an entire city) is likely too coarse a measurement to be used for these purposes. A benefit of our n-minute statistic is that it can be calculated on nearly every scale; from individual buildings to entire countries. As a result, we could provide this statistic for any geographic area, including neighborhoods, where the impact of new infrastructure could be estimated.

Single summary statistics inevitably mask a large amount of variance; however, they serve to be easy to interpret scores, widening the potential audience and facilitating comparability across cities. So, while we present each city as a single value, we also display distributions of the n-minute score and more fine-grained case studies to include more information about the equality of accessibility. Additionally, the n-minute statistic's simplicity comes at the cost of leaving some relevant factors of accessibility uncaptured. For example, there could be a case where two locations have the same n-minute score, but for one location all other POIs, besides the farthest one, are all much closer than the other location's POIs.

A simpler method to impact the n-minute statistic could be to provide POIs in areas where they do not yet exist. This has probably the biggest possible impact on the metric; building a new pharmacy in a neighborhood that did not have one before will drastically reduce travel times for all people in the area and thus lower the n-minute value. This shows that our tool has potential value not only for urban planners in municipalities, but also for businesses that want to determine the ideal location for a new store. The tool can be adjusted to include the specific points of interest a business is interested in, such as their own and their competitors' locations, to find areas that are under-served.

6.1. Applying the N-Minute calculation to areas of low population Density. The initial intention of the 15-Minute-City was meant to be applied to exclusively urban areas. The lack of mention for rural areas in Moreno's paper on the 15-Minute-City is reflected in the previous research. None of the previous literature included an analysis of the 15-Minute-City outside of large cities. Despite this precedent, we decided to extend our analysis to areas of lower population density. We argue that accessibility to essential POIs should also apply to people living in more rural areas, with the caveat that comparisons should be between areas of similar population density. Given the strong relationship between population density and the n-minute statistic within Denmark, comparisons of cities using the n-minute statistic may make the most sense across areas of similar population density. Accessibility comparisons between rural and urban areas alone would likely not yield interesting results, given huge contrasts between geographies, populations, and residential layouts. Identifying actionable items as a result of differences in the n-minute statistic therefore may be most fruitful comparing areas of similar population density. For example, asking why a rural town is less accessible than a big city does not provide reasonable recommendations into how the rural town can improve. This also does not suggest that the city can be labeled accessible when the comparison is clearly unfair. Comparing two towns of similar densities, however, may provide more reasonable assessments of individual town accessibility.

Our results have show building dedicated bicycle infrastructure or improving transit may increase accessibility for under-served areas. In reality, infrastructure additions like these are likely costly. A more thorough cost-benefit analysis is a topic of future work, but could provide powerful insights for decision makers to be able to pay for increased accessibility in their area. Some previous work has shown, for example, that Copenhagen's extensive bicycle network provides an annual societal benefit of €400,000 per kilometer of bicycle lanes (32). For rural areas, we can see a bigger benefit from transit infrastructure and suggest that increasing coverage and frequency of bus or train services would significantly impact the n-minute statistic. It might, however, not be financially feasible to increase bus service in every area, which is why car and bicycle infrastructure may continue to be important in rural areas. If essential points of interest are reachable comfortably and safely by bicycle or bus, people may use these services more often as a result of induced demand. Induced demand suggests that the easier using active modes of mobility or transit is, the more those modes will be used (40).

6.2. Denver and Los Angeles. One explanation for why cities in the US are less accessible could be the well documented, car-centric development cities in the US have undergone over the past century. Car-focused development has been suggested to have created a lock-in effect, where participation in society often requires a car (41). Simple to understand statistics such as our n-minute measurement may make the concept of human-centered neighborhoods more approachable, giving a current state and a setting a clear goal to work towards. A neighborhood that currently achieves a 21-minute score in our statistic can set the goal to reduce that to 15 and measure their progress. Overall, there is no universal n which is reasonable for all cities. As such, setting n-minute goals can, and perhaps should, be set according to the needs and wants of a community.

6.3. Limitations. Our approach has limitations including data quality issues, absence of exact timetable routing, and an impersonalized definition of essential POI. As mentioned in Moreno's paper, due to the high variability in the needs of individuals, there can be no general list of points of interest which suit all people. Our calculation excludes measures of proximity for social life, culture and work, which may be essential for many. The model also lacks certain important details, like, for example, the type of school one lives near. Including more fine-grained details would better tailor the model to the needs of specific individuals, however OSM often does not include such details, as is the case with schools. For this information, we would have to rely on municipal data, which is not standardized and would therefore hinder scalability. Especially for cities in the United States, we would have liked to incorporate information about pedestrian infrastructure in the construction of the walking network. However, OSM tags on the presence of sidewalks are not consistent enough to be a reliable source. Therefore, we assume that all roads we selected have a sidewalk or are otherwise walkable, but that might not be the case in reality. Although OSM data has been shown to be reliable in previous studies, any use of crowdsourced data comes with caveats in reliability. Because the n-minute statistic takes the maximum distance to POIs, the metric is very sensitive to the presence of accurately tagged POIs. Finally, although we include time-sensitive frequency based routing, we do not calculate exact wait times from public transport timetables. This is a planned feature for a future version of the software package.

6.4. Expansions. A major design decision when building our analysis tool was to make it as versatile and expandable as possible. Any city which has publicly available transit data and good OSM coverage can be investigated in the same way we showed here. Although we only look at several Danish municipalities and two US cities, scaling to hundreds of cities may only take one week of work, instead of months, using our tool. Additionally, any data that can be mapped to an H3 index can be added to the graph and included in routing. This would allow for analyzing the effects of new bus lines, bike infrastructure and more. In addition to that, the impact of infrastructure closures can also be analyzed by deleting edges from the graph or changing their weights. While not all of these features are exposed to the Python API, they are present on the Rust side of the project. Overall, the system has been designed from the beginning to scale well, as the memory requirements grow linearly with respect to the input and n-to-n point routing scales with the number of available CPU cores. The codebase we provide is versatile, maintainable and high performance, and we hope that by making it available under a permissive open-source license, it will help other researchers in the area.

7. Conclusion

In this thesis, we introduce a new multi-modal network building tool which encompasses all sustainable modes of transportation, then apply the tool to a case study comparing the accessibility of cities within Denmark and the US with a proposed new, simple accessibility metric.

The n-minute metric is designed to be applicable to cities, towns, and areas around the globe and provides a simple method of examining any area's progress towards being a livable, equitable 15-Minute-City or better. Our tool can perform the necessary calculations fast, is memory efficient, and combines the advantages of previous approaches. More specifically, our implementation is a comprehensive multi-modal network building tool which encompasses all sustainable modes of transportation, uses high resolution network features and allows for time-sensitive routing. We combine data sources through an efficient pipeline, relying on H3 hexagons to simplify the setup while minimizing inaccuracies. Our system is built to be extended and adapted easily; many scenarios can be simulated by adjusting origins and destinations, as well as routing and graph setup parameters.

Applying our methodology to a case study of the US and Denmark, we find that current bicycle infrastructure contributes to reductions in travel time, with the greatest benefit for areas of higher population density. We find current transit infrastructure contributes to increases in accessibility as well, however the largest benefit is in areas of lower population density. We further show the potential of cycling and transit by simulating improved versions of each. Our estimates show that more widespread cycling infrastructure and more frequent transit service both contribute significantly to accessibility. Compared to the US, cities in Denmark score better on the n-minute statistic, however differences are not so stark.

Finally, we show that our estimated travel times are comparable to estimates by the official transit provider in Denmark. Further validation is needed to be able to claim that our model accurately models travel time.

7.1. Future Work. Our project just scratches the surface of what is possible with a true multi-modal network analysis. The data processing pipeline and infrastructure allows for the analysis of changes to the network and can provide insights into broad, neighborhood-wide effects of infrastructure upgrades. For example, a new metro line could be assessed by adding another layer to the graph and re-running the analyses. Although we applied a very general POI filter, the n-minute calculation could easily be adapted to account for different living situations by including more categories of points of interest. More specific definitions of accessibility requiring more than one instance of, e.g., a supermarket to be nearby is another possible extension of the project. In terms of improving performance, the network could be expanded to include timetable-based routing to be more accurate. Additionally, GPS trace data could be used for more robust validation of the model, and potentially be used to tune the network to represent expected, data driven travel times.

8. References

1. L Alessandretti, LG Natera Orozco, M Saberi, M Szell, F Battiston, Multimodal urban mobility and multilayer transport networks. p. 23998083221211081 (2022).
2. C Moreno, Z Allam, D Chabaud, C Gall, F Pratlong, Introducing the "15-minute city": Sustainability, resilience and place identity in future post-pandemic cities. **4**, 93–111 (2021).
3. R Cervero, Transit-oriented development in the united states: Experiences, challenges, and prospects. (2004).
4. J LaPlante, B McCann, Complete streets: We can get there from here. *ITE journal* **78**, 24 (2008).
5. KM Leyden, Social capital and the built environment: the importance of walkable neighborhoods. *Am. journal public health* **93**, 1546–1551 (2003).
6. D Capasso Da Silva, DA King, S Lemar, Accessibility in practice: 20-minute city as a sustainability planning goal. *Sustainability* **12**, 129 (2019).
7. M Weng, et al., The 15-minute walkable neighborhoods: Measurement, social inequalities and implications for building healthy communities in urban china. *J. Transp. & Heal.* **13**, 259–273 (2019).
8. M Lu, E Diab, Understanding the determinants of x-minute city policies: A review of the north american and australian cities' planning documents. *J. Urban Mobil.* **3**, 100040 (2023).
9. S Zhang, F Zhen, Y Kong, T Lobsang, S Zou, Towards a 15-minute city: A network-based evaluation framework. *Environ. Plan. B: Urban Anal. City Sci.* **50**, 500–514 (2023).
10. C Ferrer-Ortiz, O Marquet, L Mojica, G Vich, Barcelona under the 15-minute city lens: Mapping the accessibility and proximity potential based on pedestrian travel times. *Smart Cities* **5**, 146–161 (2022).
11. F Gaglione, C Gargiulo, F Zucaro, C Cottrill, Urban accessibility in a 15-minute city: a measure in the city of naples, italy. *Transp. research procedia* **60**, 378–385 (2022).
12. E Knap, MB Ulak, KT Geurs, A Mulders, S van der Drift, A composite x-minute city cycling accessibility metric and its role in assessing spatial and socioeconomic inequalities–a case study in utrecht, the netherlands. *J. Urban Mobil.* **3**, 100043 (2023).
13. C Birkenfeld, R Victoriano-Habit, M Alousi-Jones, A Soliz, A El-Geneidy, Who is living a local lifestyle? towards a better understanding of the 15-minute-city and 30-minute-city concepts from a behavioural perspective in montréal, canada. *J. Urban Mobil.* **3**, 100048 (2023).
14. O Wainwright, In praise of the '15-minute city' – the mundane planning theory terrifying conspiracists (2023).
15. T Hsu, He wanted to unclog cities. now he's "public enemy no. 1." (2023).
16. A Aleta, S Meloni, Y Moreno, A multilayer perspective for the analysis of urban transportation systems. *Sci. reports* **7**, 1–9 (2017).
17. C Fink, W Klumpenhouwer, M Saraiva, R Pereira, H Tenkanen, r5py: Rapid realistic routing with r5 in python (2022) This record refers to version 0.0.4 of r5py (up- to-date as of 2022-09-09). Up-to-date versions are available from PyPi, conda-forge and the project's web page https://r5py.readthedocs.io/.
18. SD Blanchard, P Waddell, Urbanaccess: generalized methodology for measuring regional accessibility with an integrated pedestrian and transit network. *Transp. research record* **2653**, 35–44 (2017).
19. J Gil, Building a multimodal urban network model using openstreetmap data for the analysis of sustainable accessibility. *OpenStreetMap GIScience: Exp. Res. Appl.* pp. 229–251 (2015).
20. WB Du, XL Zhou, M Jusup, Z Wang, Physics of transportation: Towards optimal capacity using the multilayer network framework. *Sci. reports* **6**, 19059 (2016).
21. H Huang, D Bucher, J Kissling, R Weibel, M Raubal, Multimodal route planning with public transport and carpooling. *IEEE Transactions on Intell. Transp. Syst.* **20**, 3513–3525 (2018).
22. Uber, H3 (2022).
23. Henrikki Tenkanen, pyrosm (2022).
24. M Schiavina, et al., Ghsl data package 2022. (2022).
25. J Gil, Analyzing the configuration of multimodal urban networks. *Geogr. analysis* **46**, 368–391 (2014).
26. L Nicoletti, M Sirenko, T Verma, Disadvantaged communities have lower access to urban infrastructure. *Environ. Plan. B: Urban Anal. City Sci.* **50**, 831–849 (2023).
27. G Reggiani, T Verma, W Daamen, S Hoogendoorn, A multi-city study on structural characteristics of bicycle networks. *Environ. Plan. B: Urban Anal. City Sci.* p. 23998083231170637 (2023).
28. J Schelhaas, P Mehler, The n-minute city: A review of available tools for modeling multi-modal human mobility. (2022).
29. TP Peixoto, The graph-tool python library. *figshare* (2014).
30. NFS Gastelú, MAS Albán, DGM Yanez, TVP Poveda, GCA Vizcarra, Ciudad en pandemia. una aproximación desde la escala humana a las prioridades urbanas. *E@ACUTEACCENT@ıdos* pp. 71–86 (2020).
31. FAG Burgos, JCP D'Otero, Variables microscópicas en la velocidad de caminata. *Estudios de Transp.* **19** (2015).
32. M Fosgerau, M Łukawska, M Paulsen, TK Rasmussen, Bikeability and the induced demand for cycling. *Proc. Natl. Acad. Sci.* **120**, e2220515120 (2023).
33. S Gillies, et al., Shapely (2023).
34. F Gao, W Kihal, N Le Meur, M Souris, S Deguen, Does the edge effect impact on the measure of spatial accessibility to healthcare providers? *Int. journal health geographics* **16**, 1–16 (2017).
35. EM Van Meter, et al., An evaluation of edge effects in nutritional accessibility and availability measures: a simulation study. *Int. J. Heal. Geogr.* **9**, 1–12 (2010).
36. PyO3 Project and Contributors, PyO3 (2023).
37. Rayon-Rs, Rayon-rs/rayon: Rayon: A data parallelism library for rust (2023).
38. HydroniumLabs, Hydroniumlabs/h3o: Rust implementation of the h3 geospatial indexing system. (2022).
39. J Schuetz, G Giuliano, EJ Shin, Can a car-centric city become transit oriented? evidence from los angeles. *Cityscape* **20**, 167–190 (2018).
40. KM Hymel, KA Small, KV Dender, Induced demand and rebound effects in road transport. *Transp. Res. Part B: Methodol.* **44**, 1220–1241 (2010).
41. G Mattioli, C Roberts, JK Steinberger, A Brown, The political economy of car dependence: A systems of provision approach. *Energy Res. & Soc. Sci.* **66**, 101486 (2020).

A. Usage instructions for the n-minute city Python tool

Our tool is available on GitHub: https://github.com/hextransit/n-minute-city/

To use the Python interface, install the wheel using pip:

```
# linux
pip install wheels/graph_ds-0.1.0-cp37-cp37m-manylinux2010_x86_64.whl

# mac (intel)
pip install wheels/graph_ds-0.1.0-cp37-cp37m-macosx_10_9_x86_64.whl
```

The wheels are generated automatically by GitHub Actions.

Graphs can be created from OSM and GTFS data using the create function. The graph will be multi-layered, with a base layer of hexagon cells for the walking network, a layer for the bike network and one additional layer for every route in the GTFS data. The edge weights represent time in minutes. The chosen H3 hexagon resolution is 12. Please ensure that the OSM file contains coordinates on ways and that the GTFS feed is valid beforehand.

Create a new graph object:

```
graph = PyH3Graph(weight_options={
    bike_penalty: 1.0,
    wait_time_multiplier: 1.0,
    walk_speed: 1.4,
    bike_speed: 4.5,
} | {}>, k_ring=2, layers="all")
graph.create(osm_path="<path>", gtfs_paths=["<path>"])
```

The `layers` keyword argument allows specifying the layers the graph should contain after processing. The walk network is always included. Supported layer tags are: `all` (default), `walk`, `walk+bike`, `walk+transit`.

PyH3Graph exposes two functions for pathfinding:

- `matrix_distance` – returns the distance between all hexagon cells
- `dijkstra_path` – returns the path between two hexagon cells

H3 cells need to be input in their u64 integer representation. Only cells on the base layer are valid start and end points.

```
# get the distance matrix
distances = graph.matrix_distance(origins=[u64], destinations=[u64], hour_of_week=int,
    infinity=Optional[float], dynamic_infinity=bool)

path = graph.dijkstra_path(start=u64, end=u64, hour_of_week=Optional[int])
```

For testing purposes, you can obtain a random node from the graph by calling `graph.get_random_node()`

The optional `hour_of_week` parameter allows the transit layers to model expected wait time based on the time of day. The input expects an integer representing the hour of the week, starting at 0 for Monday 00:00 and ending at 167 for Sunday 23:00.

The parameters `infinity` and `dynamic_infinity` are used to set the maximum distance between two cells. If `dynamic_infinity` is set to `True`, the pathfinding will lower the `infinity` value during calculation. This is only useful when searching for minimum distances.

If a given index is not present in the graph, the pathfinding will attempt to map it to an index nearby, with a maximum permitted distance of `k_ring` cells. If no nearby index is found, an empty list will be returned for that origin.

B. Hardware

We used the following hardware configuration to run the simulations for this project:

- CPU: Intel Core i7-12700F with 12 cores and 20 threads.
- RAM: 64 GB
- Network: 1 GB/s internet connection

C. Additional Figures

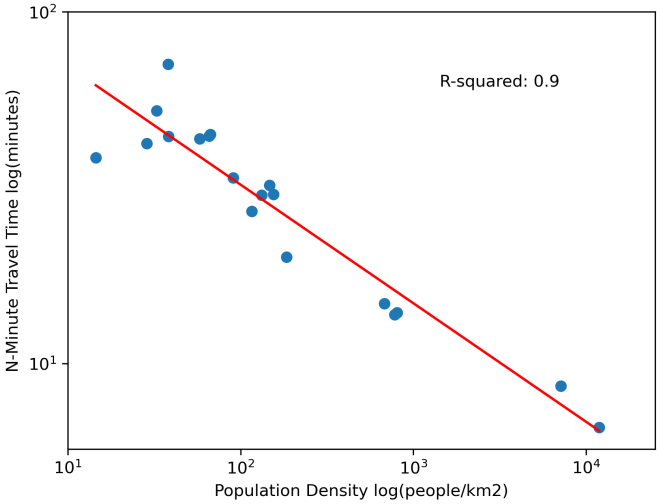


Fig. 13. Univariate Linear Regression between population density and the n-minute statistic for selected Danish Municipalities. Both axes are log scaled.

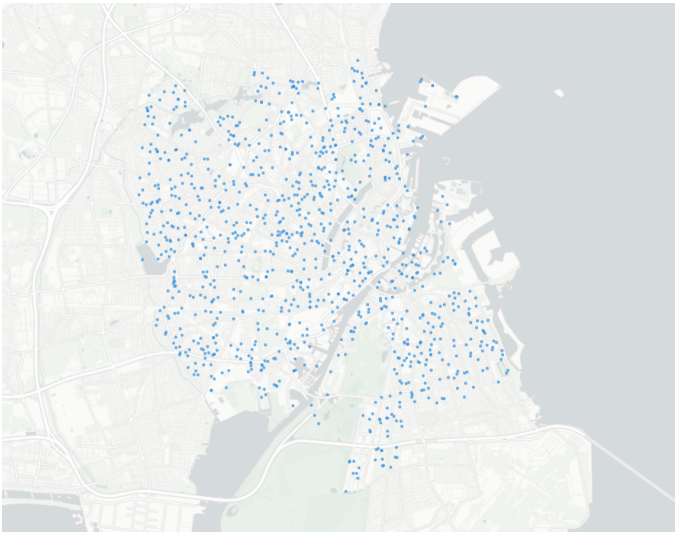


Fig. 14. 1000 random origins in Copenhagen and Frederiksberg. Origin sampling is informed by the data collected from GHSL, which differentiates between residential and commercial buildings. We only select residential points of origin

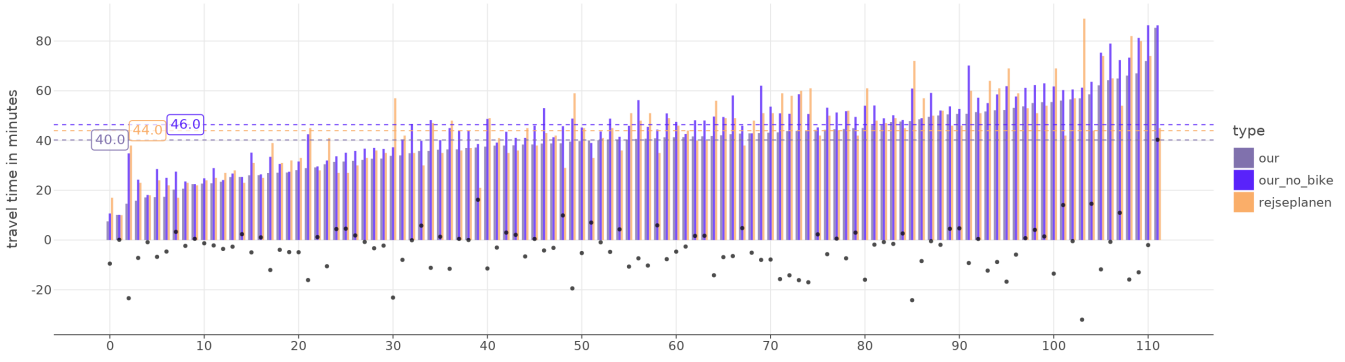


Fig. 15. Comparison of routing travel times between our networks and Rejseplanen. Individual trips were sorted by the travel time on our full network. Each trip length was calculated using our network, our network with no bicycle layer and the Rejseplanen API. The black dots show the difference between our routing and Rejseplanen.

D. Additional tables

Network type	Essential POI's	Essential POI's and libraries
Walk	20.57	24.07
Walk and Cycle	14.97	16.82
Walk and Transit	19.22	21.55
All	14.95	16.69

Table 4. n-minute values for Los Angeles, for only essential POI's and including libraries