Chapter 3

Algorithm development

3.1 Definition

In this section we define the algorithm and problem statement and how it was adjusted for this specific study. The first set of definitions explicitly define the common terms used in describing map-matching as used by Lou et al. (2009). The second part stipulates how these definitions relate to the Multi-Agent Transport Simulation (MATSim)-specific objects used for this study. MATSim is a collaborative open source project that can be used as a package in Java. This study use version 0.7.0 of MATSim and Java SE 1.7, for for more info visit `https://github.com/matsim-org` and `https://www.oracle.com/technetwork/java/javase/overview/index.html` respectively.

global positioning system (GPS) log: A collection of GPS points $\boldsymbol{Log} = \{p_1, p_2, \ldots, p_N\}$. Each GPS point $p_i \in \boldsymbol{Log}$ contains latitude p_i^{lat}, longitude p_i^{long} and timestamp p_i^{t}.

GPS Trajectory: A sequence of GPS points, $\mathbf{T}$, where the time interval between any consecutive GPS points do not exceed a certain threshold $\Delta T*$, i.e. $\boldsymbol{T} = \{p_1, p_2, \ldots, p_n\}$, where $p_i \in \boldsymbol{Log}$, and $0 < p_{i+1}^{\text{t}} - p_i^{\text{t}} < \Delta T|_{\forall 1 \leq i < n}$. ΔT is also referred to as the *sampling interval.*

The enumerated black dots in Figure 1.1 is an example of a GPS trajectory. For this sample dataset a GPS point was generated at the end of the true path when the vehicle was switched off, thus generating an additional point that has a ΔT which is less than the other points in the GPS log. Table 3.1 reflects the collection of the GPS points from Figure 1.1. The GPS coordinates use the standard World Geodetic System 84 (WGS84) as coordinate reference system.

Table 3.1: GPS trajectory from example data

Point	Longitude	Latitude	Time s
p_1	-33.962	18.473	0
p_2	-33.961	18.472	25
p_3	-33.960	18.469	50
p_4	-33.959	18.466	100
p_5	-33.958	18.465	150

Road network: A road network is a directed graph $\boldsymbol{G}(\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V}$ is a set of vertices representing the intersections and terminal points of the road segments, and $\boldsymbol{E}$ is a set of edges representing road segments. Figure 3.1 shows a portion of the MATSim network of the sample data depicted in Figure 1.1, the circles indicate the vertices and the arrows the edges of the road network. Edges e_1 and e_2 have also been annotated with their start and end points.

Road segment: A road segment, e, is a directed edge of a road network such that $e_i \in \boldsymbol{E}$. The segment e is associated with an id e^{id}, a typical travel speed e^{v}, a length value e^{l}, a starting point e^{start}, an ending point e^{end} and a list of intermediate points that describes the road using a polyline. Figure 3.1 shows several road segments from the example road network with the start and end points of edges e_1 and e_2 annotated.

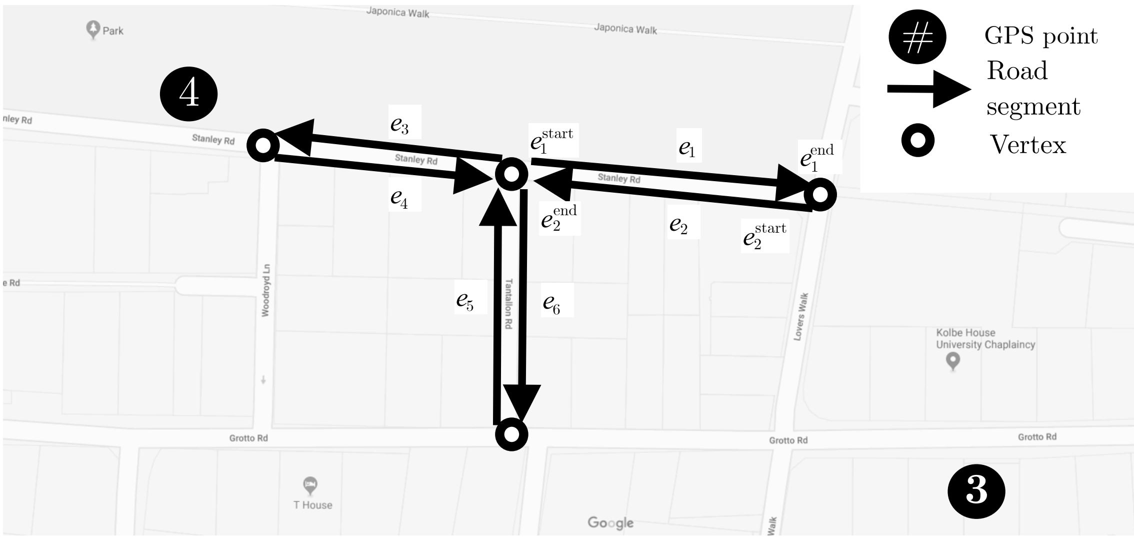


Figure 3.1: Edges on a road network

Path: Given two vertices v_i , v_j in a road network $\boldsymbol{G}$, a path $\boldsymbol{P}$ is a set of connected road segments that start at v_i and end at v_j , i.e. $\boldsymbol{P} = \{e_1, e_2, \ldots, e_n\}$, where $e_1^{\text{start}} = v_i$, $e_n^{\text{end}} = v_j$, $e_k^{\text{start}} = e_{k+1}^{\text{end}}$, $1 \leq k < n$.

Problem statement: Given a raw GPS trajectory, $\boldsymbol{T}$, generated by an agent travelling on a path in a road network $\boldsymbol{G}(\boldsymbol{V}, \boldsymbol{E})$, determine the most likely path $\boldsymbol{P}$ that the agent travelled.

All the previous definitions are general terms used to describe the general format of road networks and GPS trajectories for most map-matching problems. Since the application of the algorithm developed was in MATSim, some specific terminologies need to be defined as well as their relationship to the previous definitions.

GPS point: In order to work with meters, the coordinate system used in MATSim is the SA-Albers projection which is adapted from the Albers equal-area conic projection. Table 3.2 shows the GPS trajectory after the conversion to the SA-Albers projection. A trajectory using the SA-Albers projection is defined as $\boldsymbol{T_{SAA}}$.

Link: Similar to the definitions of a road segment, a link L in MATSim represents an edge with only one direction of travel. In order to represent bidirectional traffic between

Table 3.2: GPS trajectory of example data with SA-Albers projection

Point	x	y	Time s
p_1	-513327.5134	-3716061.097	0
p_2	-513486.9662	-3715934.682	25
p_3	-513722.076	-3715759.626	50
p_4	-514033.4896	-3715661.143	100
p_5	-514115.2881	-3715473.827	150

two points, two links need to be defined, one for each direction of travel. A link has, inter alia, the following fields:

- link identifier, L^{id} - the name of the link;
- a from and to node L^{fn}, L^{tn}, the starting and ending points of a link respectively;
- free speed L^{fs} in m/s. This is the speed an object is allowed to travel on the link, and can vary depending on the time of day; and
- the length of the link L^{length} in meters.

In comparison to the road segment definition, a MATSim link does not contain intermediate points. To represent a curved road segment multiple links and nodes can be used to accurately represent the road shape in topological format. If the visual aspect of a link is not important a curved road segment can, for example, be represented by just one straight link with the correct length defined programmatically and not graphically. This presents a potential issue in using the spatial and topological analysis part of the map-matching algorithm as it relies on the network to be as topologically and spatially accurate as possible, and MATSim network can only contain straight lines. As can be seen in Figure 3.2, the MATSim network available for this study has relatively accurate topological structures. This is achieved by representing curved segments on the road with a number of small, straight links.

Node: A node in MATSim, N, is defined as a topological point where one or more links can be joined and an agent can move from one link to another. A node has, inter alia, the following fields that are used during calculations: node ID N^{id}, list of In links N^{inLinks} (links that end inside this node), list of Out links N^{outLinks} (links that start inside of this node). In Figure 3.2 the nodes are represented by small grey diamonds that connect the links.

Network: A network is a set of links connected to each other via a set of nodes to form and represent a transport network where upon agents can travel and interact with one another. A network is defined as $\boldsymbol{N_{MATSim}(L, N)}$, where $\boldsymbol{N}$ is a set of nodes representing the intersections and terminal points of the road segments, and $\boldsymbol{L}$ is a set of links representing road segments.

The MATSim representation of the entire example network is illustrated in Figure 3.2 and includes the true path travelled by the vehicle as well as the GPS trajectory in the applicable coordinate system.

Path: In MATSim a path is a list of connected links on which agent can travel on from one node to another, and is defined by the use of $\boldsymbol{P}_l$.

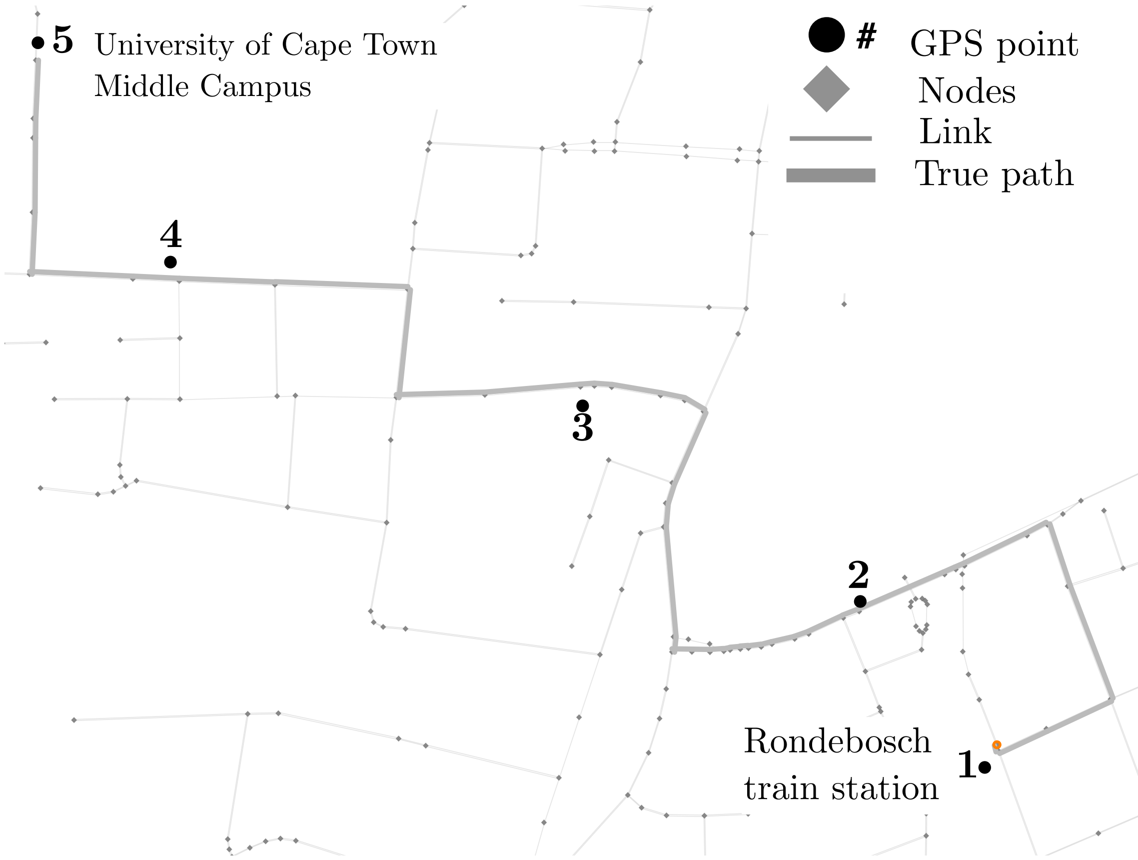


Figure 3.2: MATSim network of sample data

The map-matching problem can now be formulated as follows:

Updated problem statement: Given a raw GPS trajectory $\boldsymbol{T}_{SAA}$ generated by an agent travelling on a path on a road network $\boldsymbol{N_{MATSim}(L, N)}$, determine the most likely path $\boldsymbol{P}_l$ that the agent has travelled.

3.2 Candidate preparation

For each GPS point in a trajectory, there is a potential list of road segments or links on which the agent could have travelled on when generating the GPS point. These potential links are called *candidate links*. Given trajectory $\boldsymbol{T} = \{p_1, p_2, \ldots, p_n\}$, there exist a number of candidate links for each point $p_i \in \boldsymbol{T}$.

Lou et al. (2009) made use of a search radius to identify a number of road segments for each point in the GPS trajectory where the road segment is within the radius from the GPS point.

There is no native functionality in MATSim to efficiently collect all the links that are within a radius of a certain point. However, all the nodes within a radius can be transposed into a specific data structure referred to as a *quadtree* (more detail on the creation of the quadtree refer to section 4.3.2). When all the nodes within a radius have been identified, one can check whether or not the links to and from the nodes are within the specified radius from the GPS point by determining the projected distance from the GPS point to the nearest point on the link. However, there exists an issue with explicitly defining the radius for collecting all the applicable nodes without taking into consideration the characteristics of the links within the network. The issue is that some nodes might be connected to the link right next to the GPS point but the link's endpoint nodes are outside of the search radius and thus the link will not be evaluated, leading to an incorrect path being inferred for the GPS trace. Figure 3.3 illustrates such a possible situation, where GPS point 2 is connected to the highlighted line to its left, but the distance to the start and end nodes of this link is 1.7km and 1.5km away respectively. If the search radius for nodes in the quadtree does not include these outlying nodes, the GPS point will only have the short residential links to its right evaluated. Thus, the search radius needs to be a function of the longest link within the network to ensure the algorithm does not miss potential candidate links that might possibly be the correct links. Since the search for nodes within a distance is efficiently done using the quadtree the algorithm uses the longest link length within the network as the search radius, which is not an input into the algorithm as per Lou et al. (2009)'s implementation.

Once all the possible nodes have been identified and all the incoming and outgoing links from the nodes have been stored as potential candidate links for the GPS point being evaluated, a straight-line distance from the GPS point to each potential candidate link is calculated and saved. The list of potential candidate links identified is sorted according to proximity to the GPS point. The next step is to use only a certain number of links from the sorted list of potential candidate links for further analysis and result matching. The number of links used greatly affects the efficiency and effectiveness of the algorithm, and a number of experiments was run to determine the influence of these parameters on results.

The next step was to determine a *candidate point* on each candidate link using segment projection, and was defined as follows:

Candidate point: This is the projection of a point p to a link l and is defined as point c on l such that $c = \arg\min_{c_i \in l} dist(c_i, p)$, where $dist(c_i, p)$ returns the distance between

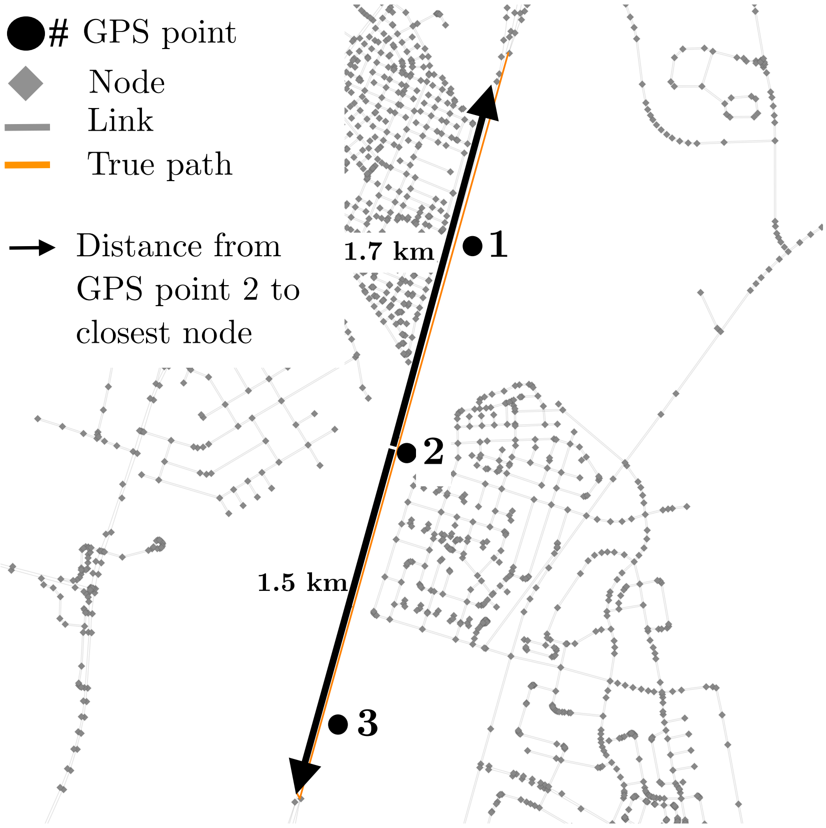


Figure 3.3: Search radius required when searching for candidate links

p and any point c_i on l. The jth candidate link and candidate point of p_i will then respectively be denoted as l_i^j and c_i^j. As shown in Figure 3.4, p_i's candidate points are c_i^1 c_i^2 and c_i^3. The candidate point represents the most likely point on the specific link at which the agent was when recording the GPS point being evaluated.

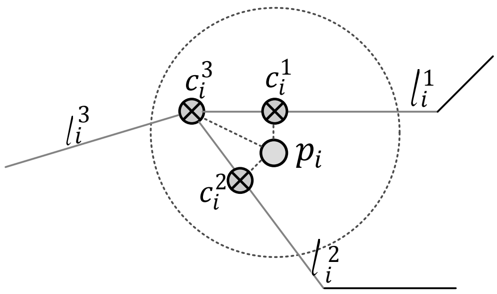


Figure 3.4: Point projection on candidate links

Updated problem statement Once the candidate link sets and their respective candidate points are retrieved for all the sampling points on the trajectory $\boldsymbol{T}_{SAA}$, the problem statement becomes how to choose one candidate from each set so that $\boldsymbol{P}_l = \{c_i^{j_1}, c_i^{j_2}, \ldots c_n^{j_n}\}$ best matches $\boldsymbol{T}_{SAA} = \{p_1, p_2, \ldots, p_n\}$

3.3 Spatial analysis

In the spatial analysis both the geometric and topological information of the road network is used to evaluate the candidate points found in the previous step. The geometric information is used in the *observation probability*, and the topological information in the *transmission probability*.

Observation probability: The observation probability is defined as the likelihood that a GPS sampling point p_i matches a candidate point c_i^j and is computed based on the distance between the two points, defined as $dist(c_i^j, p_i)$. The inherent error in a GPS measurement causes p_i to be a certain distance from c_i^j and can be approximated using a normal distribution $N(\mu, \sigma^2)$. This distribution can be used to indicate how likely a GPS observation p_i can be matched to a candidate point c_i^j on the real road without considering its neighbouring points. Formally the observation probability is defined as: $N(c_i^j)$, of c_i^j with regard to p_i as:

$$N(c_i^j) = \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{(x_i^j - \mu)^2}{2\sigma^2}} \tag{3.1}$$

where $x_i^j = dist(c_i^j, p_i)$, the distance between p_i and c_i^j. Similar to Lou et al. (2009) this study makes use of a zero-mean normal distribution with a standard deviation of 20 m for GPS error. Using these parameters the observation probability will always output very low probability values for any GPS point even if the point is within 0 m of the candidate point. During informal discussions with the original authors they indicated that the final spatial-temporal (ST) probability value should rather be viewed as a *normalised score* instead of a probability, per se, due to the low values. In this study the output of the ST function was still referred to as a probability.

For the map-matching example used in this section the 8 closest links were used to identify candidate links. The candidate links identified for GPS point 2 is shown in Table 3.3 and the candidate link names can be referenced from Figure 3.5.

Table 3.3: GPS point candidate links for GPS point 2

Candidate link	Distance to link	Observation probability
l_2^1	7.08	0.0187
l_2^2	7.08	0.0187
l_2^3	8.15	0.0184
l_2^4	8.15	0.0184
l_2^5	19.08	0.127
l_2^6	19.08	0.127
l_2^7	19.08	0.127
l_2^8	19.08	0.127

Since the observation probability calculation does not take into account the position context of a GPS point, it can sometimes lead to wrong matching results. Figure 3.6 shows such an example. The thick lines represent a highway, and the thin vertical line represents a local road. Although p_i is closer to c_i^1 on the local road, it should match p_i to c_i^2 on the highway if it is already known that p_i's neighbours p_{i-1} and p_{i+1} are on the highway. This is based on the assumption that a vehicle is unlikely to take a roundabout path Lou

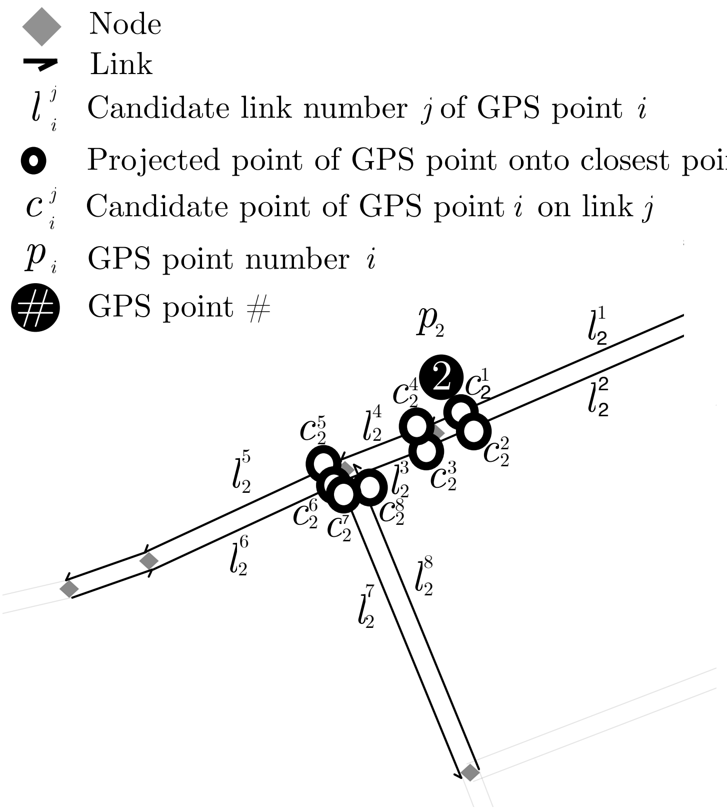


Figure 3.5: Example of candidate links

et al. (2009). To prevent this potential error the algorithm made use of the *transmission probability*.

Transmission probability: Given two candidate points c_{i-1}^t and c_i^s for two neighbouring GPS sampling points p_{i-1} and p_i respectively, the transmission probability from c_{i-1}^t to c_i^s is defined as the likelihood that the true path from p_{i-1} to p_i follows the shortest path from c_{i-1}^t and c_i^s.

The transmission probability is defined as

$$V(c_{i-1}^t \rightarrow c_i^s) = \frac{d_{i-1 \rightarrow i}}{w_{(i-1,t) \rightarrow (i,s)}} t \tag{3.2}$$

where $d_{i-1 \rightarrow i} = dist(p_i, p_{i-1})$ is the Euclidean distance between p_i and p_{i-1}, and $w_{(i-1,t) \rightarrow (i,s)}$ is the length of shortest path from c_{i-1}^t to c_i^s.

Combining equations (3.1) and (3.2) the spatial analysis function can be defined as: $F_s(c_{i-1}^t \rightarrow c_i^s)$ as the product of observation probability and transmission probability:

$$F_s(c_{i-1}^t \rightarrow c_i^s) = N(c_i^s) * V(c_{i-1}^t \rightarrow c_i^s), 2 \leq i \leq n \tag{3.3}$$

where c_{i-1}^t and c_i^s are any two candidate points for two neighbouring GPS points p_{i-1} and p_i respectively.

Equation (3.3) computes the likelihood that an object moves from c_{i-1}^t to c_i^s using the product of two probability functions; thus, geometric and topological information are both taken into consideration. Note that in practice, it is unlikely for a moving object

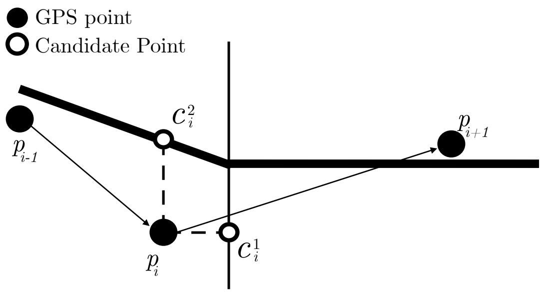


Figure 3.6: Transmission probability example

to always strictly follow the shortest path. Therefore the observation probability $N(c_i^s)$ cannot be omitted from Equation (3.3).

With spatial analysis, for any two neighbouring GPS points p_{i-1} and p_i, a set of candidate paths $c_{i-1}^t \rightarrow c_i^s$ are generated. Each path is assigned a spatial measurement value computed from Equation (3.3).

There exists a shortcoming inherent to low-sampling rate GPS data, because even though the map-matching algorithm determines the most likely link for every GPS point, the assumption is still that for two links found for subsequent GPS points not connected to each other, the links traversed between the two links is the shortest path. In real life very similar paths, in terms of distance, might exist between two links or a driver might take a non-shortest path route and still generate points at similar positions in the network. The temporal analysis aims to address part of this problem but without higher sampling rates there is no accurate way of identifying the correct route if many alternatives exist between two GPS points.

3.4 Temporal analysis

Although the spatial analysis can determine the actual path from the other candidate paths relatively accurately, there are situations where the spatial analysis might choose an unlikely path. When two paths that allow very different travel speeds are situated next to each other, spatial analysis might determine a path that spatially makes sense but when considering the speed at which the agent would travel on the chosen path, this is highly unlikely. The inferred speed might seem unlikely due to either the speed restrictions or the practical speed at which an agent would have to travel to reach the next candidate point within the given time between the subsequent samples. See the example shown in Figure 3.7. The two lines on the left is a highway, and the lines to the right are roads inside a residential area. The spatial analysis function may produce the same value whether two points p_{i-1} and p_i are matched to the highway or to the residential road. However, if one calculates the average speed from p_{i-1} to p_i on a straight line as $100\,km/h$, the algorithm would match the points to the highway considering the speed limits of the residential road.

More formally, given two candidate points c_{i-1}^t and c_i^s for two neighbouring GPS sampling points p_{i-1} and p_i respectively, the shortest path from c_{i-1}^t to c_i^s is denoted as a list of road segments $[l_1', l_2', \dots, l_k']$. The average speed $\overline{v}_{(i-1,t)\rightarrow(i,s)}$ of the shortest path is

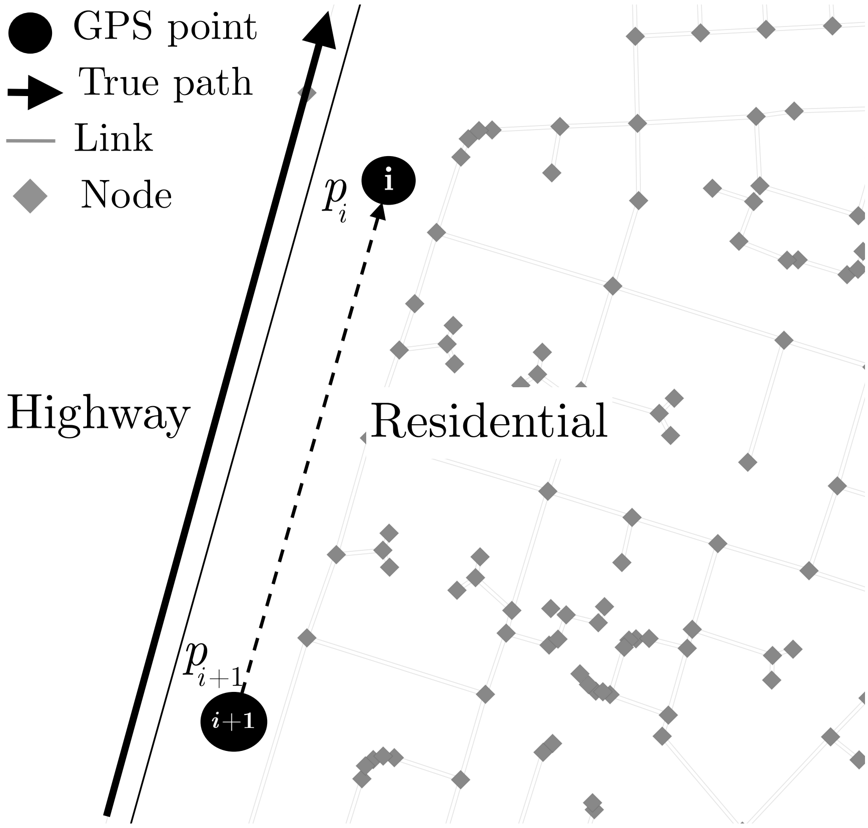


Figure 3.7: Temporal analysis example

computed as follows:

$$\overline{v}_{(i-1,t)\to(i,s)} = \frac{\sum_{u=1}^{k} \ell_u}{\Delta t_{i-1\to i}} \tag{3.4}$$

where $\ell_u = l_u^{'length}$, the length of $l_u^{'}$, and $\Delta t_{i-1\to i} = p_i^t - p_{i-1}^t$, the time interval between two sampling points p_i and p_{i-1}.

Note that each road link $l_u^{'}$ is also associated with a typical speed value $l_u^{'fs}$, referred to as the free speed of a link. One of the benefits of using a MATSim network was that the free speed can be a function of the time and as such one is able to incorporate more accurate free speed data based on time of day to evaluate the temporal probability of the algorithm.

The cosine distance is used to test the similarity between the actual average speed from c_{i-1}^t to c_i^s and the speed constraints of the path between the two points. Consider the vector that contains k elements of the same value $\overline{v}_{(i-1,t)\to(i,s)}$ and the vector $(l_1^{'fs}, l_{21}^{'fs}, \ldots, l_k^{'fs})^T$

Temporal analysis: The temporal analysis function is defined as follows.

$$F_t(c_{i-1}^t \to c_i^s) = \frac{\sum_{u=1}^{k}(l_u^{'fs} \times \overline{v}_{(i-1,t)\to(i,s)})}{\sqrt{\sum_{u=1}^{k}(l_u^{'fs})^2} \times \sqrt{\sum_{u=1}^{k} \overline{v}^2_{(i-1,t)\to(i,s))}}} \tag{3.5}$$

As in the spatial analysis function, c_{i-1}^t and c_i^s are any two candidate points for p_{i-1} and p_i respectively.

3.5 Result matching

Once the spatial and temporal analyses have been completed, a candidate graph $\boldsymbol{G}_T'$ is generated with every candidate point for every GPS point representing a node in the graph and every link in the graph representing the movement between the neighbouring candidate points.

Formally the candidate graph for trajectory $\boldsymbol{T} = \{p_1, p_2, \ldots, p_n\}$ is $\boldsymbol{G}_T'(\boldsymbol{L}_T', \boldsymbol{N}_T')$, where $\boldsymbol{N}_T'$ is a set of candidate points for each GPS sampling point, and $\boldsymbol{L}_T'$ is a set of links representing the possibility of moving from one candidate link to another, as depicted in Figure 3.8. Each link also contains attributes for the connection between the two nodes,

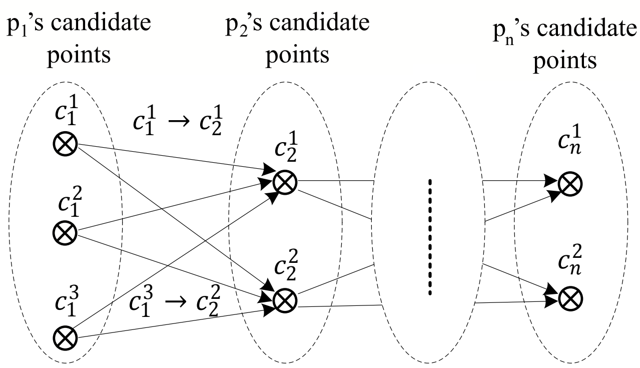


Figure 3.8: Schematic view of candidate graph $\boldsymbol{G}_T'(\boldsymbol{L}_T', \boldsymbol{N}_T')$

namely the value of the ST algorithm's probability that the object traversed from the one candidate point to the other, stored as the length of the link. The link also stores some metadata such as the shortest path on the original road network between the candidate links if they are not connected by a common node. The shortest path is used to reconstruct the inferred route once the most probable candidate points have been identified. Each node in $\boldsymbol{G}'$ also stores the observation probability $N(c_i^s)$ and each link the average speed, $V(c_{i-1}^t \to c_i^s)$ and temporal analysis $F_t(c_{i-1}^t \to c_i^s)$ for detailed analysis afterwards.

A candidate path sequence for the entire trajectory $\boldsymbol{T}_{saa}$ is a path in the candidate graph, denoted as $\boldsymbol{P}_c = \{c_1^{s1}, c_2^{s2}, \ldots, c_n^{sn}\}$. The overall score for such a candidate sequence is $F(\boldsymbol{P}_c) = \sum_{i=2}^{n} F(c_{i-1}^{s_{i-1}}, c_1^{s_i})$. From all the candidate sequences the aim is to find the one with the highest overall score as the best matching path for the trajectory. More formally, the best matching path $\boldsymbol{P}$ for a trajectory $\boldsymbol{T}$ is selected as:

$$\boldsymbol{P} = \underset{\boldsymbol{P}_c \in \boldsymbol{G}_T'(\boldsymbol{L}_T', \boldsymbol{N}_T')}{\arg\max} F(\boldsymbol{P}_c) \tag{3.6}$$

To get the candidate sequence with the highest overall score the entire graph needs to be traversed for all the possible combination of routes from the start to the end point. The score of a route is the the sum of the link lengths in a route where the link length is the ST algorithm's score for the link being the correct link which the agent traversed while generating the GPS points. This process can be executed using existing graphing algorithms like Dijkstra's algorithm. But since Dijkstra's algorithm uses the shortest path as an objective function, instead of using the length of the links as the travel distance utility this implementation uses $1-l^{\text{length}}$. Since the length of the link is the ST algorithm's probability that the object traversed from the one candidate point to the other, i.e. the probability that it is the right route, Dijkstra's algorithm was actually calculating the routes that is least likely to *not* be the wrong route. Changing Dijkstra's algorithm to essentially calculate the longest route is only possible because the graph is a directed acyclic graph, that is, it is a finite directed graph with no directed cycles, thus no infinite loops will occur when looking for the longest path.

The Dijkstra method requires a single starting and ending node for the algorithm to start at and navigate to. Since the graph has multiple starting points, for point p_1, and multiple end points, for point p_n, a dummy starting and ending node was created. The starting dummy node was connected to the first tier of nodes with a length equal to the observation probability of the node it was connected to. The ending dummy node was connected to the last tier of nodes with length 1.

Figure 3.9 shows the graphical representation of the map-matching example given in Chapter 1. Every tier in the graph, excluding the starting and ending node, represents the eight possible candidate points for each GPS point, P1 $\rightarrow$ P5. Once the longest path has been calculated from the graph, the original links from the road network can be inferred from the graph link and node data.

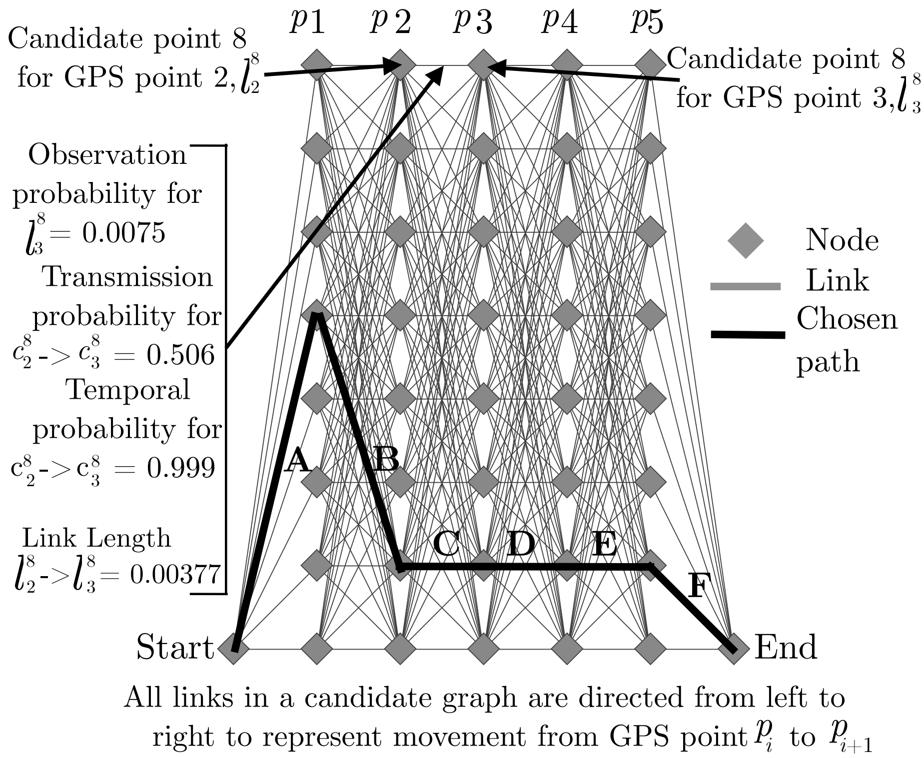


Figure 3.9: Candidate graph for sample data

In Table 3.4, the probability for each chosen link can be seen as well as the average link probability, which was used to calculate the overall probability of the inferred path (IP).

Graph link F did not form part of the calculation as it was only used to create an end point for the Dijkstra algorithm to use and the length of the link is 1.

Table 3.4: Graph link ST probability data

Graph link	**ST probability**
A	0.0159
B	0.0083
C	0.0119
D	0.0153
E	0.0079
Average	0.0119
Min	0.0079
Max	0.0159
Std Dev	0.0033

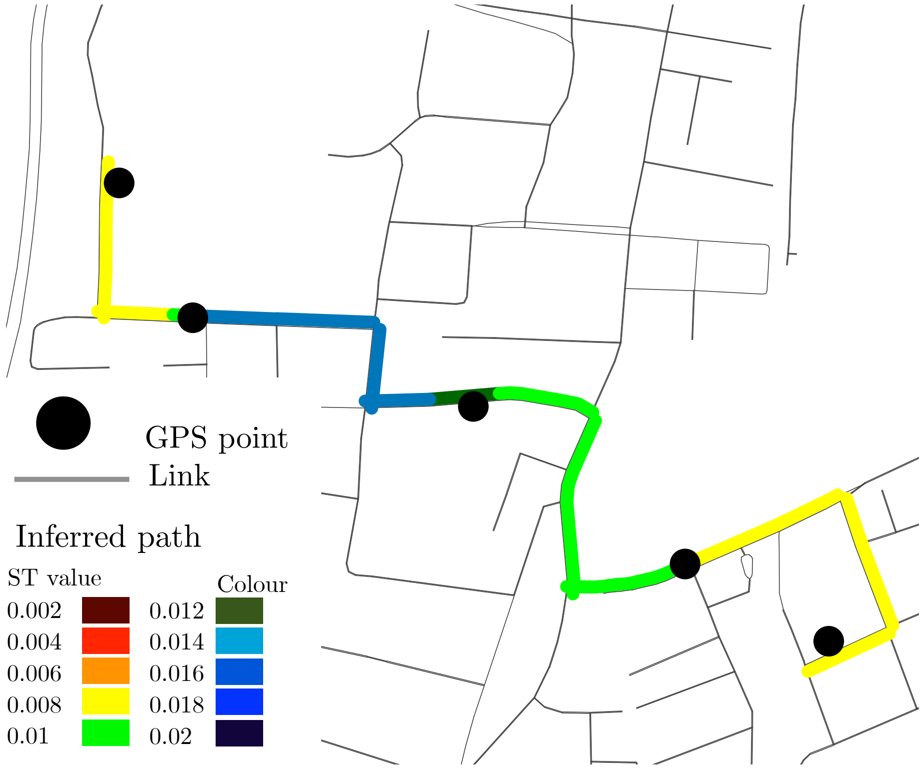
3.6 Analysing results

Since a true path (TP) is available in this example, one can analyse the results of the algorithm using the criteria defined in section 2.2 as per the Table 3.5.

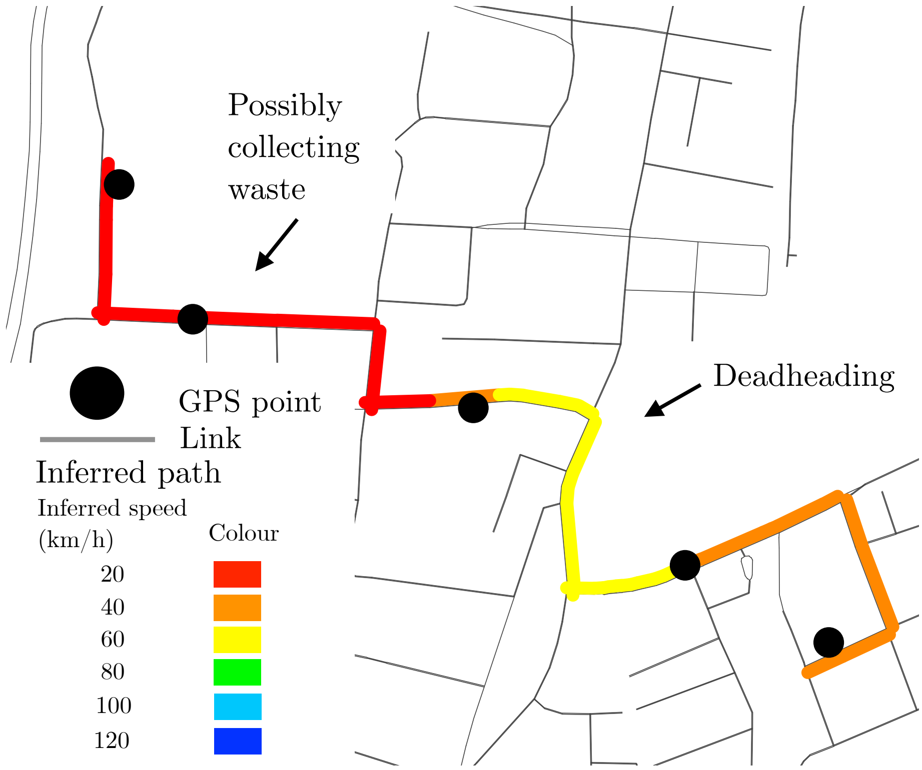
Table 3.5: Example results analyses against TP

Measurement	**Value**
ARR	0.997
IARR	0
ARRn	0.977
Al	0.997
Average ST value	0.0119

To analyse the results of the algorithm in the absence of a TP one can review the inferred speed as well as the ST probability value. The inferred speed can be calculated from the IP using the time stamps of the GPS points while the probabilities can be transferred from the graph links to the associated network links. Since subsequent GPS points can be allocated to candidate links that are not connected, some graph links contain a list of multiple network links representing the shortest path between the one candidate link and the subsequent candidate link. An original network link can also be associated with more than one graph link if the link is a candidate link for more than one GPS point. In order to assign a probability to the original network links, the probability of each graph link can be assigned to the candidate link and intermediate links with which it was associated. If there are shared original network links between graph links, an average between the probabilities and calculated speed was assigned to these links. The result of this analysis can be seen in Figure 3.10. From Figure 3.10b, we can infer what the waste collection vehicle was doing on each by part of its journey by using the speed as a proxy. It appears from the red lines that the vehicle might have been collecting waste, while the yellow lines imply it was most likely deadheading, i.e. driving to a new service area and not collecting any waste. This simple analysis of inferred speed answers the specific question posed by the use case of this study.



(a) Probability value of ST algorithm



(b) Inferred speed of vehicle

Figure 3.10: Analyses of example map-matching results

To gauge the *confidence level* of the inferred route one can analyse the assigned probabilities from the ST algorithm, as per Figure 3.10a, but without further experimentation to create a benchmark for probabilities versus accuracy, it is difficult to conclude a confidence level in the results purely based on the probabilities of different route sections. It should be noted that low speeds travelled by the vehicle would lead to lower probabilities because the temporal part of theSTalgorithm analyses the calculated speed against the free speed of the network and the more it differs, the lower the value. This poses a potentially significant impact on the analysis of waste collection vehicles since they regularly travel at very low speeds while servicing an area.