

Semi-automatic tool for bus route map matching

Book Chapter**Author(s):**

Ordóñez Medina, Sergio A.

Publication date:

2016-08

Permanent link:

<https://doi.org/10.3929/ethz-b-000119094>

Rights / license:

[Creative Commons Attribution 4.0 International](#)

Semi-Automatic Tool for Bus Route Map Matching

Sergio Arturo Ordóñez

18.1 Basic Information

Entry point to documentation:

<http://matsim.org/extensions> → `gtfs2matsimtransitschedule`

Invoking the module:

<http://matsim.org/javadoc> → `GTFS2TransitSchedule` → `GTFS2MATSimTransitSchedule` class

Selected publications:

Ordóñez Medina and Erath (2011)

Current public transport assignment models adapt network assignment models to work with public transport traffic. Many commercial software products like EMME/2 (Version 2 of EMME), VISUM and OmniTRANS offer sophisticated procedures that include timetable-based route search. However, these models do not include interaction between public transport services and private transport. As mentioned above, the MATSim implementation handles private car traffic and public transport traffic in an integrated way, but it needs accurate public transport line routing on the transport network. While this is usually straightforward for rail-based public transport modes, the routing problem for buses requires more attention; experience shows that assumption of a shortest-path between two consecutive stops leads to unsatisfactory results. To overcome this shortcoming, one can either draw the routes manually or employ map-matching algorithms dependent on tracking data. Due to the burden of manual procedures, and the increasing availability of GPS tracking data, map-matching is becoming increasingly relevant. However, common map matching algorithms are usually not designed to account for the peculiarities of public transport routing; the procedure is very sensitive to errors in network coding, inaccurate bus stop locations and the simplified link shapes in the model.

How to cite this book chapter:

Ordóñez, S A. 2016. Semi-Automatic Tool for Bus Route Map Matching. In: Horni, A, Nagel, K and Axhausen, K W. (eds.) *The Multi-Agent Transport Simulation MATSim*, Pp. 115–122. London: Ubiquity Press. DOI: <http://dx.doi.org/10.5334/baw.18>. License: CC-BY 4.0

This section presents a semi-automatic procedure combining public bus routes information (sequences of consecutive stop locations and sequences of geo-referenced points) with a high-resolution network (Ordóñez Medina and Erath, 2011). The objective is to obtain a sequence of links for every route of every line and to associate each bus stop with one single link in the network. The procedure was designed to prepare the Singapore scenario public transport extension, but the tools developed can be used to set up any other scenario with similar initial data (timetable and high resolution network).

18.2 Problem Definition

Generally, the problem can be defined as follows. Given:

- a set of stop locations (two-dimensional point coordinates),
- a set of route profiles (sequence of consecutive stops),
- a set of GPS points sequences (sequence of two-dimensional point coordinates), and
- a high resolution navigation network (two-dimensional directed graph with attributes),

the task is to associate each stop with a network link, and translate each route to a network path (connected sequence of links). Figure 18.1 illustrates the problem by providing an example of the available input information and correct output.

Input Information The GTFS (General Transit Feed Specification) is a recent, but already widely-used format for specifying public transport systems, created by Google for feeding its geographic information applications. As of April 2011, the Singapore public transport system featured

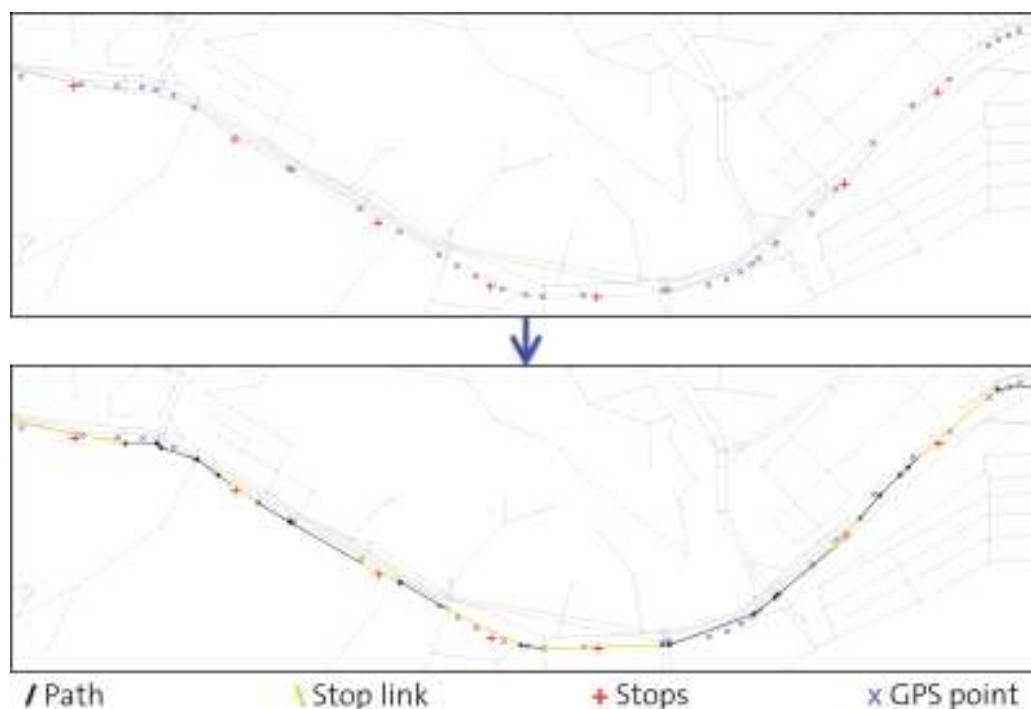


Figure 18.1: Input data and expected solution of the map-matching problem.

Source: Reprinted from Ordóñez Medina and Erath (2011, p.753), Copyright (2011), with permission from Hong Kong Society for Transportation Studies

4 584 bus stops serviced by 355 bus lines, all recorded on GTFS. Each line had several routes, i.e., different outward and return routes (due to one-way streets), as well as different coverage of serviced bus stops on weekdays and weekends. GTFS records the name and location of each bus stop; for bus lines, it records constituent bus routes as a sequence of stops, along with their shape (a sequence of GPS points) as additional information.

The GTFS data must be mapped to a high resolution network; for Singapore, this is a navigation network developed by NAVTEQ. The network is a directed graph where streets and intersections are represented as links and nodes. The links between nodes record attributes like street name, number of lanes, length, flow, free speed and capacity. Nodes are simply recorded as two-dimensional point coordinates. This network has a total of 79 835 links and 43 118 nodes.

Special Restrictions There are some intrinsic characteristics of the public transport system that should be considered serious restrictions. First, when a certain stop is assigned to a network link, this link should be a part of all paths belonging to this stop's routes. In other words: once established, stop-link relationships are fixed for resolving the missing routes. If the GPS points from a route including a specific stop suggest it should be associated with a different nearby link, then all other routes including that stop must be resolved again. Hence, the order in which the routes are resolved is important; it is preferable to resolve those routes first, when we completely trust supporting information quality (e.g., GPS trails).

Second, while many lines run in two directions, with most bus stops having a corresponding stop in the opposite direction (stop located on the other side of the street), this cannot be used to our advantage, because links defined by each return route are different, locations of stops are not necessarily exactly opposite to those in the opposite direction and return routes do not always use the same street.

However, some routes on the same line have an inclusion relationship; in peak hours, segments of bus routes with high demand are served by additional buses running on partial routes to meet demand. In these cases, if a full route is resolved, its partial routes solutions are included.

18.3 Solution Approach

It is not possible to automatically map-match the given GPS position with the network, as standard methods usually require at least 10 points for each link (Schüssler and Axhausen, 2009). In the Singapore GTFS, distance between consecutive points averages about 65 meters, and average link length is about 91 meters; thus, we have fewer than two points per link, on average. Furthermore, not all the routes have GPS points, which inhibits using a full automatic solution; in the Singapore GTFS, there are 38 bus routes without GPS points.

Consequently, the strategy for resolving each route consists of a semi-automatic procedure. Figure 18.2 illustrates the process. First, a simple map-matching algorithm is applied if the route is not part of a bigger route already solved (inclusion relationship described above). In this case, only a previous solution's partition is needed to obtain a first solution. Then, an automatic verification (described below) is performed. If the verification ends with a positive outcome, one can decide to finish the route and save the solution, or to continue editing. If one decides, or is forced, to modify the solution, there are two ways to proceed: changing parameters and running the automatic algorithm again, or editing the solution interactively with a graphical interface editing tool. In both cases, automatic verification must be executed again. If previously saved stop-link relationships are modified, prior routing solutions containing one of the involved stops are erased.

As long as more solutions are obtained, it becomes easier and faster to solve further routes, similar to a machine learning process. This happens for two reasons; first, because of the inclusion relationships that omit the algorithm and second because the increasing number of fixed stop-link relationships relaxes the algorithm (functioning explained in the following section).

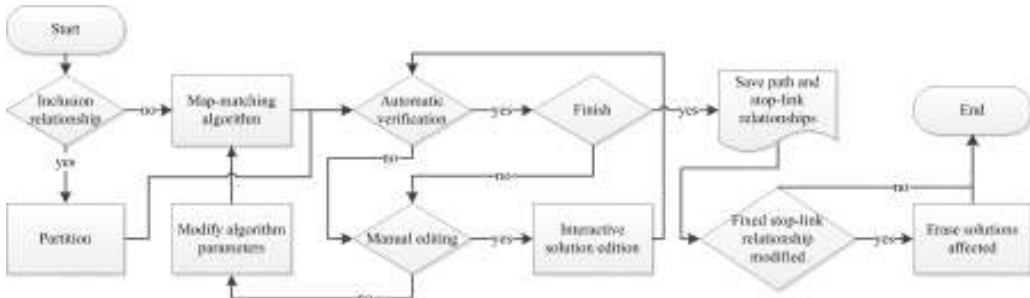


Figure 18.2: Semi-automatic process for one route.

Source: Reprinted from Ordóñez Medina and Erath (2011, p.754), Copyright (2011), with permission from Hong Kong Society for Transportation Studies

18.4 Map-Matching Automatic Algorithm

This algorithm's objective is to generate a solution (path or sequence of connected network links and a set of stop-link relationships) for one route, knowledge of its profile, a sequence of GPS points and a set of stop-link relationships. The algorithm is designed to deal with:

- low GPS point resolution,
- sporadic low network spatial resolution,
- long distances between two express routes stops, and
- understanding that the nearest link to a stop point is not always the correct one.

The route map-matching process is illustrated in Figure 18.3. Except for the first stop, the algorithm solves for each stop in the route profile, a portion of the links sequence (from previous to current) and, if this stop has no fixed link, a set of link candidates pooled from the one link selected.

Link candidates are defined as follows: the NL closest links to the stop point, within a distance D_{max} , define a set of candidates. Each set's element could be subjected to more restrictions; the closest point, between the stop point and the infinite line defined by the link, must be inside its line segment and the angle between the link direction and the nearest GPS points sequence direction must be lower than α_{max} .

The link's selection is performed as follows; from the previous stop link to each defined candidate, an A star search algorithm is applied for finding the shortest path. For running this algorithm, each link's cost depends on the link's travel time and distance to the GPS points. A product with flexible exponents was proposed as a first model:

$$C_{link} = \exp \frac{L_{link}}{S_{link}} w_1 \exp D_{GPS} w_2 \quad (18.1)$$

where L_{link} is its length, S_{link} is its free speed, D_{GPS} is its distance to the GPS points sequence and w_1 and w_2 are positive weights with a standard value of 1, but modifiable by the user, according to existence or quality of the GPS points sequence. The definition of D_{GPS} can also be modified; in the simplest approach, it is the minimum distance between the link and all GPS points (point-segment distance). From all calculated paths, the shortest is selected and added to the general route solution. The corresponding link candidate is also related to the stop.

If the current stop has a stop-link relationship, only the shortest path to this stop defines the solution. Thus, the process continues with the next stop in the route profile. If the first stop of the

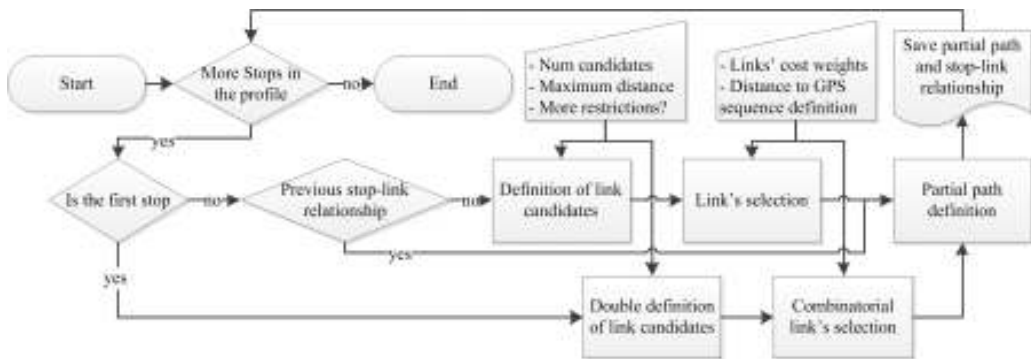


Figure 18.3: Map-matching algorithm.

Source: Reprinted from Ordóñez Medina and Erath (2011, p.755), Copyright (2011), with permission from Hong Kong Society for Transportation Studies

profile has no fixed link, a similar algorithm between the first and the second stop is performed. The definition of candidates' procedure is applied to the first and the second stops. Then, the candidates' selection procedure consists of obtaining the shortest path of all combinations between the two sets of candidates, then selecting the shortest one. This path defines links for both stops.

18.5 Automatic Verification

In this step, accuracy of the routing solution is automatically checked by performing the following ordered verification:

1. Is the path joined?
2. Is the path without U turns?
3. Is the path without repeated links?
4. Does every stop of the route have a stop-link relationship?
5. Is every link related to a stop inside the path?
6. Is the related links' order in the path the same as the corresponding stops' order in the route profile?
7. Is the nearest point between the stop point and the infinite line defined by the link inside its line segment in every stop-link relationship?
8. Are the first and last links of the path related to the first and last stops of the route profile?

Verifications (2), (3) and (7) are not mandatory and can be deactivated through the user interface. User interaction is necessary to (i) cover possible errors, and (ii) include actual route characteristics: some bus routes do include U turns, some repeat exactly the same street, in the same direction, during their travel and the geometric restriction presented in (7) is not always valid in big stop facilities, like bus interchanges.

18.6 Manual Editing Functionalities and Implemented Software

The edit functions' objective is to allow the user to modify the automatically generated routing solution. Even if the automatic algorithm generates a correct solution based on input data,

problems like recent changes in routes, differences in release dates between GPS points and network data, erroneous GPS points, or lack of network element all require manual changes. Although one also could modify and correct the input data, or the generated solution, with direct data modifications, two-dimensional visualization and keyboard-mouse user interaction are two quality attributes that help reduce time and effort. Developed functional requirements and quality attributes are:

1. **Visualization:** A navigation network is displayed, including all relevant information for working with a single route. This includes the route's profile, given sequence of GPS points, and its current solution (path and stop-link relationships). Selected elements are drawn in a different color. Everything is displayed in a two-dimensional and interactive way, including the cursor location in working coordinates, panning, zoom and view-all options.
2. **Selection:** Different options for selecting solution elements, or elements from the network, are provided. It is possible to select the nearest link from the solution or from the network, the nearest node from the network, or the nearest stop from the solution, to a point indicated by the user. When a stop that already has a stop-link relationship is displayed, its corresponding link is highlighted as well. If a solution path link is selected and does not have a subsequent link connected, a new one from the network is selected with one click; the selected link is that with the angle most similar to the line defined by the end node of the initial link and a point indicated by the user.
3. **Path modification:** The first link of the sequence can be added by selecting any network link. If a solution path link does not have a subsequent link connected, it is possible to add one, according to the selection function described in (2). If there are two unconnected sequential links in the solution (a gap), a sub-sequence connecting these links is added, using the shortest path algorithm, with the current parameters. Further, selecting one solution path link, it is possible to delete it, or to delete all links before or after it. Finally, stop-link relationships can be modified by selecting either elements. If the modified relationship was fixed, the user is prevented from modifying the relationship, because the tool will erase the solutions of the routes to which the selected stop belongs.
4. **Network modification:** New nodes to the road network can be added. In addition, with any node selected, it is possible to add a new link selecting the end node.

These functions were implemented in a software package developed from scratch in Java and using the Java2D library for graphics. The package reproduces the described solution approach, looking for non-solved routes, and running the map-matching algorithm and the automatic verification for each one. Figure 18.4 shows the user interface and a demo video can be accessed at <http://www.vimeo.com/27137889>.

18.7 Conclusion and Outlook

The semi-automatic procedure designed for map-matching bus lines with a high resolution navigation in Singapore was successful, allowing the solving of all bus routes and stops in only ten days, even taking into account the quality of the input information offered, highlighting the low spatial and temporal resolution of the GPS points given for each route. Analysis indicates that reducing manual modification time is the best way to improve the procedure, which can be done by modifying the automatic algorithm to obtain more accurate results for the initial routes to be solved, or in other words, for routes not affected by the learning process.

As GTFS is becoming so popular for defining public transport systems and the code in which this process is implemented is open source, it can be used for matching routes with high

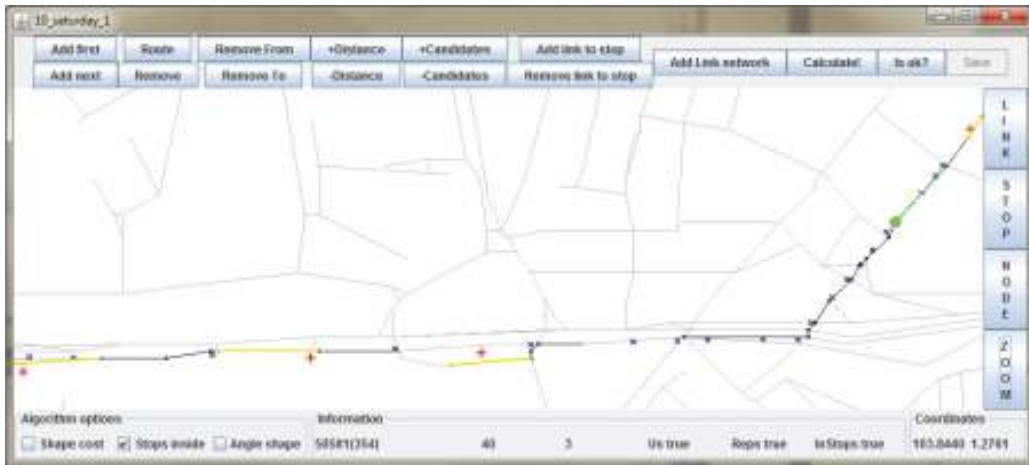


Figure 18.4: User interface of the application to edit automatic solutions.

Source: Reprinted from Ordóñez Medina and Erath (2011, p.757), Copyright (2011), with permission from Hong Kong Society for Transportation Studies

resolution networks of any GTFS-specified place. The tools are available as a MATSim contribution (GTFS2TransitSchedule). For generating MATSim simulation scenarios, the procedures have been used by research teams in the province of Gauteng, South Africa, on the Toronto scenario and on a different public transport simulation model developed by SMART-MIT in Singapore.