MICRO BEHAVIOUR VALIDATION OF STRATEGIC AGENT BASED TRANSPORT MODELS

KASIA KOZLOWSKA AND FRED SHONE AND GERARD CASEY

ABSTRACT. Agent-Based Models (ABMs) are complex models which hold a lot of information about the individuals represented in the model. Compared with traditional transport models, ABMs give us the opportunity to simulate scenarios with higher fidelity. This means the inclusion of a lot of new complexity. Both in the complexity of the population of agents (like having lots of different types of agents with lots of different behaviours) and the complexity of their environment and interactions (for example agents navigating their world using scheduled public transit services). As we add complexity, it becomes more and more important to ensure that our models are behaving realistically. Even small inconsistencies could propagate errors and bias in the sensitivity of the model to new scenarios.

However, the higher fidelity of our ABMs allows us new mechanisms with which to validate and get assurance about our simulations. In this work we investigate the use of GPS traces in validating an ABM on an individual agent level (which we call micro-validation) as an addition to the more usual macro-level validation. We define both spatial and temporal metrics for benchmarking.

As far as we are aware the application of this micro-validation is unique in the transport field. The implications of our work are increased confidence in ABMs. More broadly, we show how the ABM framework allows for new innovative validation methodologies with new data. We expect this to pave the way for increasingly detailed simulations, capable of modelling increasingly complex interactions for transport.

1. Introduction

Using GPS in the creation of demand for an agent-based model is not novel. There has been plenty of work done within all sorts of different domains utilising the use of GPS or mobile phone data. In transport, for example, authors of [Gurram et al., 2019] have generated demand for the entirety of the United States of America using GPS data. [Yin et al., 2018] generated urban-based activities using call detail records, one of their validation methods was analysis of an agent-based simulation using the generated data—the inverse of what we are trying to achieve in this paper. The author also used the same software as we do—MATSim (Multi-Agent Transport Simulation https://www.matsim.org/).

We have also found evidence of GPS being used in validating models. For example, it has been widely used to validate ocean tide models which track tide-induced displacements of the Earth's crust (e.g. [Flohrer et al., 2005]). We have also found that GPS has been used in validating Geographical Information System (GIS) least-cost model predictions of gray squirrel movement in Britain [Stevenson et al., 2013]. GPS was used in validating the IRI-2012 model [Kumar et al., 2014] which models physical and chemical IRI parameters such as electron and ion densities, total electron content, electron, ion and neutral temperatures in the ionosphere. GPS-levelling (referring to physical height) and gravity data has been used to validate GOCE(Gravity field and Ocean Circulation Explorer Satellite)-only models [Szűcs, 2012].

Date: September 2020.

Key words and phrases. agent-based modelling; validation; GPS.

Presented at Transport Practitioners' Meeting (TPM) 2020 and European Transport Conference (ETC) 2020.

The subject of validating agent-based models is very broad. During our study, we have found a very interesting piece of work [Olsen and Raunak, 2016]. The approaches outlined in the paper are domain agnostic and use the idea of metamorphic testing—a property-based software testing technique. The actual tests are domain specific, but the method of testing applies to any ABM. [Ormerod and Rosewell, 2009] provides a general account of validation and verification of ABMs. It touches on model replicability and the challenges in validating complex agent-based models. [Rand and Rust, 2011] outlined guidelines for agent-based model development within the domain of marketing. They touch on the subject of micro validation and talk about rigorous agent-based modelling in the marketing domain, from model invention, set-up, verification and finally validation. They outline four major steps in validation:

(1) Micro-face validation: Elements of the implemented model correspond heuristically to the real world and the agents possess a realistic amount of information.
(2) Macro-face validation: The processes and patterns of the implemented model correspond heuristically to the real world. The theory of the model corresponds to our current understanding of the real world.
(3) Empirical Input Validation: The data used as inputs to the model corresponds to the real world. The authors suggest using training and test datasets to aid calibration of input data.
(4) Empirical Output Validation: The output of the model corresponds to the real world. For this validation the authors suggest using
    - Stylized facts based on domain expert knowledge
    - Validating against real-world data
    - Cross-validation—comparing against outputs of another validated model

Nb. There is no data directly compared to the model at the micro/macro-face validation stages.

With all models we must have some confidence that the model is behaving, and will behave, realistically. Benchmarking is one of the traditional approaches to building this confidence. Benchmarking involves building a scenario for a specific period, for which real data is available. Once the scenario is simulated, the outputs can then be validated against this real data.

If a model is working well, we might expect the outputs from this simulation to look like the real data. Generally the available data pertains to aggregate flows of people or vehicles across networks, such as:

- vehicle / passenger counts
- trip length distributions
- mode split information
- journey times along key routes
- transit boardings, alightings and interchanges

We know there are some limitations to even this 'real data'. In all cases the data is typically aggregated temporally or spatially. Walking is also often neglected. For traditional models, this data precision has been acceptable because outputs of transport models are similarly simplified and aggregated and able to inform the historic decisions made from them. But in the context of ABMs, we need more precise 'real data' to benchmark our outputs against. This is because agent-based simulations provide much more detail about what is going on in our model. For example, agents leave the house at precise times, walk to exact bus stop locations, catch specific bus services, or wait for exact amounts of time for the bus to arrive. Additionally, agents in our simulation are making decisions across the whole day, rather than isolated, independent decisions. The traditional validation approaches are very relevant but begin to struggle at this level of detail. In this paper we would like to propose a number of metrics that could be used to validate, describe and compare an ABM to GPS data.

We use Python throughout this work. The simulation software is Java-based. There are two aims for this paper. First, we are looking to compute numeric values which will help us judge how close the outputs of the simulations are compared to the GPS data. These values provide us with signals of model performance. Second is a selection of visualisations which may help us understand the variations between ABM output and GPS data and thus help us debug any issues with the model.

2. Data Description and Preparation

To support this research we recruited Arup staff who agreed to share their data for the purpose of this work. We also collected a sample of journal information from Arup volunteers, which allowed us to verify how we processed raw-GPS data into useful journey information. In the future, the GPS records collected for use in validation could be anonymised, with the origin/destination points of trips approximated with small impact on the overall validation results.

Data was collected and processed in the winter of 2018/2019. Journey information is made up of trips and legs. Trips and legs are defined by origin and destination pairs and hold spatial and temporal information. A trip can include one or many legs. Modes are assigned to legs (walk, cycle, public transit, etc.). Legs are linked to trips via a unique index and contain times of departure and arrival between each leg segment. Each trip has a dominant mode which is derived from a leg in that trip with the longest distance.

We use the trips data to generate agents in the model, and the more detailed legs data to validate the model outputs. Each agent in our simulation is defined by two sets of data. The first—referred to as attributes—defines the agent's demographics; the second—referred to as plans—defines the agent's activities over a 24-hour period.

We subsetted the initial trips dataset significantly, reducing it to entire days satisfying the following criteria:

- The day is a weekday
- At least one trip on this day has an origin/destination within London (M25 orbital motorway boundary)
- For all trips within the day, if there is a subsequent trip, its origin is close to the destination of the preceding trip
- All trips on this day have corresponding legs data

In our case, the trips data did not include information about the purpose of each trip. We built a process to assign the purpose of each journey. We developed a simple land-use based activity classifier based on a majority vote on the type of buildings within a 30-metre radius of where a trip started and finished, using Open Street Map (OSM). Here is the mapping used between building types and activity types:

```
'work': ['office', 'offices', 'industrial', 'warehouse'],
'education': ['school', 'adult_education', 'university'],
'shop': ['shop', 'commercial', 'retail', 'department_store', 'kiosk', 'supermarket
    ', 'bakehouse'],
'personal': ['courthouse', 'townhall', 'roof', 'pharmacy', 'dentist', 'hospital',
    'bank', 'post_office', 'veterinary', 'register_office'],
'recreation': ['library', 'casino', 'cinema', 'restaurant', 'pub', 'cafe', 'bar',
    'community_centre', 'sports_centre', 'swimming_pool', 'fast_food', 'nightclub',
     'food_court', 'social_facility'],
'religious': ['place_of_worship', 'church', 'cathedral', 'chapel', 'church', '
    mosque', 'religious', 'shrine', 'synagogue', 'temple'],
'tourism': ['tourism'],
'other': ['parking', 'toilets', 'embassy', 'government', 'public', 'fuel'],
```

```
'escort': ['kindergarten']
```

Ideally, this process would not be required, with the relevant activity data being included in the dataset or acquired via interview for smaller samples.

In addition to this, the home activity was assumed to be the starting activity of the day and all subsequent activities within 200-metre radius of this location were also labelled as home. All test subjects work in the Arup London office, thus we used a spatial covering of the office to infer work activity in addition to the OSM buildings classifier.

All viable days were then used to create potential agents, investigated and picked based on validity of their GPS traces (no large discontinuities in the trace) and good distribution of corresponding leg modes for diverse representation in subsequent validation. The agents were created using an open source Python project which allows in-memory demand representation and provides functions to export the demand to MATSim format: https://github.com/arup-group/pam.

Some of the agents' data, though selected for the model, was altered further to remove trips which were deemed as errors in GPS or the trip inference algorithm. Figure 1 shows an example of two 'trips' which were removed from the agent's plan.

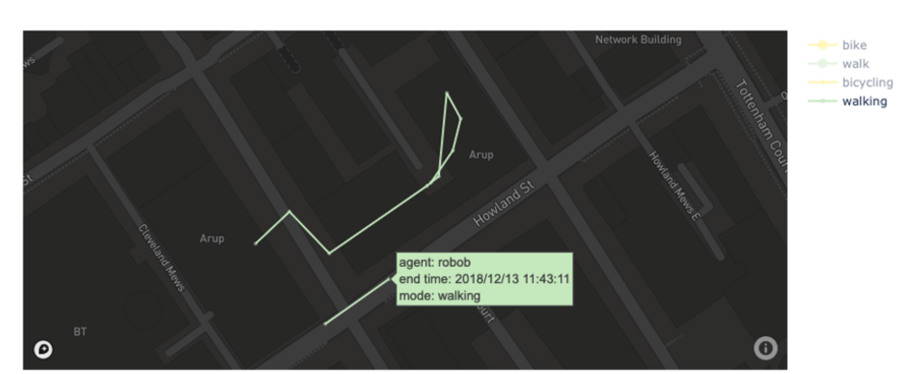

FIGURE 1. Removed trip

From 9 initial agents we generated 7 agents with single work activity from morning to afternoon. We have 3 agents using public transport (PT), 3 cycling to work and 1 using both PT and cycling. In general the agents can have any number of different types of activities in their activity chains, and indeed, the rest of the demand in this model consists of such complex demand data. The spatial plots of the selected agent plans can be seen in Figure 2 and their corresponding GPS traces in Figure 3.

3. Model description

For this work we used a transport model of London which has been developed in collaboration with Transport for London (TfL). For simulations we used the MATSim software version 11 with two main extensions:

- *com.github.SchweizerischeBundesbahnen.matsim-sbb-extensions* for simulating public transit, and
- *org.matsim.contrib.multimodal* version 11.0, which enables the agents to use the network for walking and cycling.

The model is described in detail in [Shone et al., 2019]. In summary, we distinguish two inputs for the model: demand and supply. Demand describes the population, it holds information about the agents, their plans for the 24 hour period and suggested modes of transport to get between their

FIGURE 2. Generated agent trips

FIGURE 3. GPS traces which formed the basis for the agent trips

activities, and demographic information. There are 3 different types of agents which model the demand for this London's transport network:

- London's population (based on household locations),
- Freight trips (entering or exiting London),
- Tours from outside of London (entering and exiting London).

In addition, this model also includes the 7 GPS-based agents with which we aim to validate the model outputs. Supply describes the transport network available to the agents. We synthesise it using Open Street Map (OSM) data—which provides information about the road and rail networks—and General Transit Feed Specification (GTFS)—which provides information about the public transport—appropriate for the model's spatial extent.

4. Metrics

This section defines the metrics we used to compare the journey legs data and the methods of computing them. We start by outlining a unified data structure which ensures a consistent shape in both the ABM outputs and the GPS data in order to meaningfully compare them. We grouped metrics into the following categories:

(1) Spatial
(2) Temporal

(3) Combined spatial and temporal
(4) Mode and activity

There are two aims in this section, and this paper in general. First, we are looking to compute values between 1 and 0 which judge how close the outputs of the simulations are compared to the GPS data, 0 being exactly the same. These values provide us with signals of model performance. Second is a selection of visualisations which may help us understand the variations between ABM output and GPS data and thus help us debug any issues with the model.

4.1. **Parsing ABM outputs.** MATSim simulation returns a number of outputs. For this work we are interested in agents' output plans. Each agent retains a number of best daily plans, each plan is scored and the scoring is relative. A plan score does not mean anything without knowing the difference in the numbers set for utilities in the simulation's configuration. When we run the simulations over many iterations, we don't look for the scores to reach a certain number, rather we look for the average of all the scores to stabilise. The most stable, final plan choice does not always appear rational. Agents don't simply choose the plan with the highest score. A higher score means the plan is more likely to be chosen but that is not guaranteed. We read all of the plans for our GPS agents, recording whether the plan was selected and how it scored. We also read all of the travel and routing in detail to be compared against the corresponding GPS traces.

4.2. **Unified data structure.** We create two tables (`pandas.DataFrame`s), one to hold the model outputs and another to hold the GPS data. We start by reading all activities and legs with their corresponding information and build a record of all trips, the schema for the tables and overlap between them is summarised in Table 1.

We expand both tables, first spatially, for all points along the network route or polyline, interpolating the time in seconds at each spatial point based on the distance travelled within the leg (assuming constant speed for each leg). We then interpolate the spatial latitude and longitude points for each second in travel. For comparison and metrics computations we extract information at every minute of the day. Nb. Initially, we use one second intervals (MATSim works on the same granularity) for capturing spatial elements because our network is not simplified (retains a lot of the geometry through node information). This allows us to capture the detailed routing for agents. For simplified networks, one minute intervals are sufficient and much less computationally expensive. We now have simulation and GPS data given in the same granularity. Minutes of the day without data are assumed as in-activity. To that end, we fill the rest of the minutes in the day of an agent by copying the last known spatial point, set the mode to 'Stationary' and, for the ABM results table, copy the last known activity type. We encode the modes:

```
0: 'Stationary',
1: 'Walk',
2: 'Bicycle',
3: 'Drive',
4: 'PT'
```

We merge the two tables on time and agent name, arriving at the following schema. The data below is driven by time, each agent is accounted for at each minute of the day, for each of their plans. The final schema is summarised in Table 2.

With the data set up, we can proceed to use simple operations on the table's columns to compute different metrics.

4.3. **Spatial metrics.** We start with some basic spatial distance metrics. We compute:

Description GPS	Column name GPS	Column name ABM	Description ABM
Unique agent identifier	agent	agent	Unique agent identifier
		plan	Unique plan identifier, integer
		selected	Whether the plan was selected
		score	The score for the plan
Unique identifier for the trip	trip_id	trip	Id for the trip, sequence within the day
Unique identifier for the leg	uuid	leg	Id for the leg, sequence within the trip
Mode of transport	mode	mode	Mode of transport
Time and coordinates for origin of leg	origin	origin	Time and coordinates for origin of leg
Time and coordinates for origin of destination	destination	destination	Time and coordinates for origin of destination
		activity_at_destination	Purpose for the trip activity or MATSim's PT interaction activity
Distance in meters between origin and destination	distance	distance	Distance in meters between origin and destination
Time elapsed between origin and destination	duration	trav_time	Time elapsed between origin and destination
Encoded ordered (lat, lon) pairs defining the route for a leg	polyline	route	Network links used for the trip. Can be None if agent teleported

TABLE 1. Data overlap between ABM results and GPS

(1) Distance between spatial points in both ABM and GPS data and one arbitrary point (we opted for origin of each agent). Figure 4 and Figure 5 show distance from origin for GPS data and three different plans for a selected agent.
(2) Distance between spatial points in ABM and GPS data for each minute of the day (spatial proximity). Figure 6 shows a comparison of three different plans to GPS of the same agent, their scores and whether they have been selected.

The amalgamated spatial metric for each agent plan can be computed from normalised spatial proximity metric (2). To normalize we divide by the largest distance (in GPS data) from origin for

Column name	Description
`agent`	Unique agent identifier
`plan`	Unique plan identifier, integer
`selected`	Whether the plan was selected
`score`	The score for the plan
`trip`	Id for the trip, sequence within the day
`leg`	Id for the leg, sequence within the trip
`mode_abm`	Mode of transport in ABM outputs
`mode_gps`	Mode of transport in GPS outputs
`time`	Time, integer, minute of the day
`polyline_lon_abm`	Longitude of the point in space for agent in ABM at time t
`polyline_lat_abm`	Latitude of the point in space for agent in ABM at time t
`polyline_lon_gps`	Longitude of the point in space for the same agent's GPS data at time t
`polyline_lat_gps`	Latitude of the point in space for the same agent's GPS data at time t

TABLE 2. Merged ABM and GPS data schema

an agent for the day. To compute the metric,

$$\frac{1}{1440}\left(\sum_{t=0}^{1440} d_{n,t}\right) \tag{1}$$

where $d_{n,t}$ is the normalised distance metric from a particular plan n at time t. We get a dictionary of this sort:

```
'agent': {
    'Plan 1.0': 0.03354437654723302,
    'Plan 2.0': 0.033992265577307904,
    'Plan 3.0': 0.03354437654723302
    }
```

4.4. **Temporal metrics.** The idea behind a purely temporal metric would be to compute the temporal distance between the ABM output and GPS, while keeping spatial information fixed. This would probably mean computing the time difference to a record closest in space. However, an agent can be in the same place more than once, thus we would also need to consider a combination of proximity in time, as well as space, to find the correct point to compare and this raises questions of whether this is still a purely temporal metric.

An interesting metric to track agent's tardiness/punctuality or vehicle speed (agent's cycle speed and/or route in the case below) would be to use the spatial metric (2—distance from arbitrary point) to compute the difference in areas under each of the graphs, see Figure 7. This can be approximated by the following sum,

$$\frac{1}{1440}\left(\sum_{t=0}^{1440} (d_{n,t} - d_{g,t})\right) \tag{2}$$

Where $d_{n,t}$ is the distance metric from an arbitrary point for a particular plan n at time t and $d_{g,t}$ is the distance metric from an arbitrary point for the corresponding GPS point at time t. These can be normalised in a similar way as in the amalgamated distance metric above, if the arbitrary point is the origin. We again get a dictionary of scores:

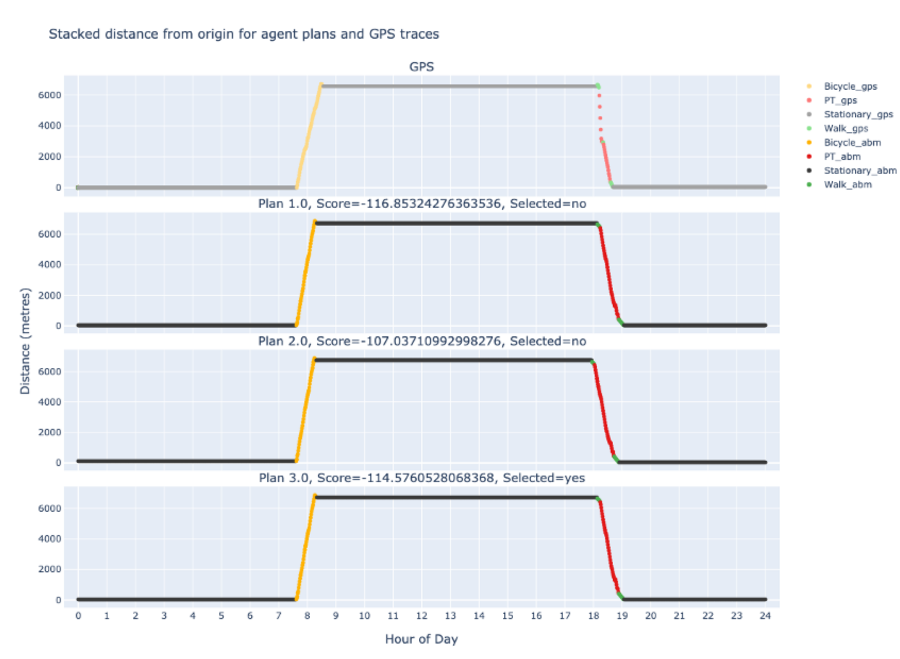

FIGURE 4. Distance from origin (home) location

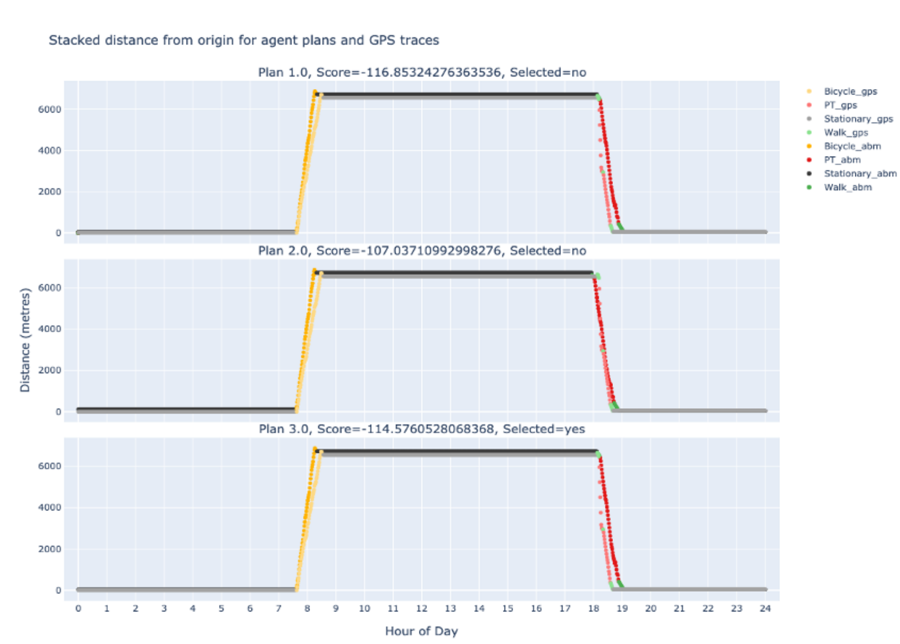

FIGURE 5. Distance from origin (home) location (overlayed with GPS traces)

```
'agent': {
    'Plan 1.0': 0.026035493624349846,
    'Plan 2.0': 0.022365580922375835,
    'Plan 3.0': 0.026035493624349846
    }
```

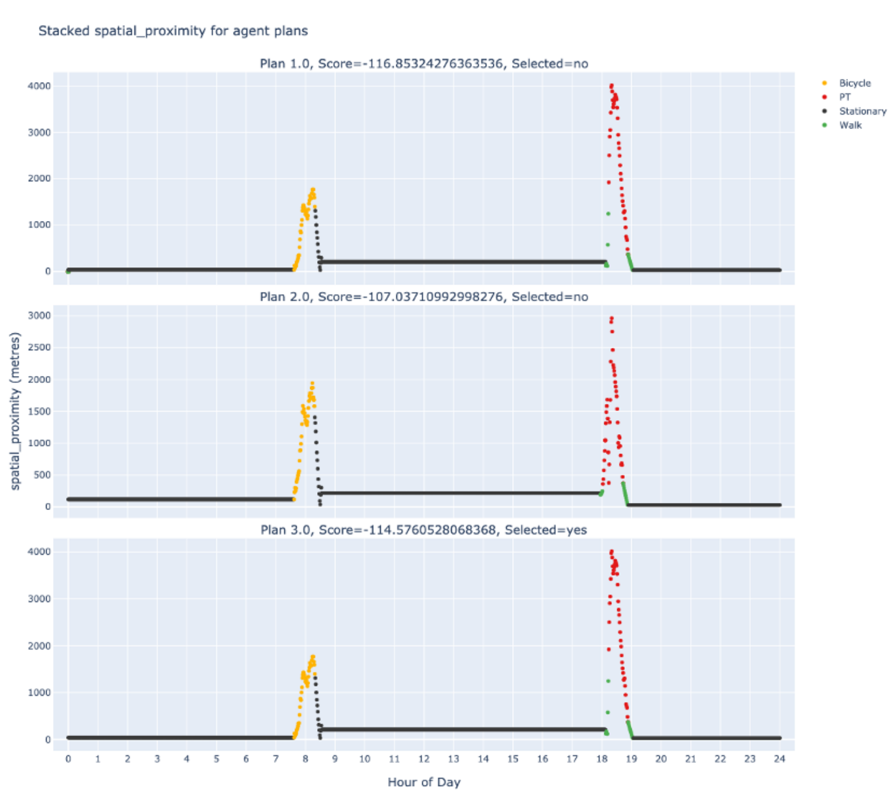

FIGURE 6. Distance between GPS and agent traces

4.5. **Combined spatial and temporal metrics.** We can compute a combined spatial and temporal metric using latitude, longitude and time for each record (GPS record and agent plan record at a point in time) and representing it as points in three-dimensional space (lat, lon, t). We construct a list, which is a closed loop composed of 3 dimensional points (lat,lon,t)

- a particular plan which is to be tested, ordered by time (red line in the plot above),
- GPS data points, ordered in reverse (black line in the plot above)
- and closing the loop, the first point of the ordered plan records.

We can then perform a volume calculation of a convex hull created by these points using Python's scipy library. We can compute a metric between 0 and 1, by normalising the (lat, lon, t) points. We used the min-max normalisation with minimum and maximum values of latitude, longitude and time for a specific agent plan and the agent's GPS data.

```
_df = df[(df['agent'] == agent) & (df['plan'] == 1)].sort_values('time', ascending=True)

lat=_df['polyline_lat_abm'].to_list() + _df['polyline_lat_gps'].to_list()
lon=_df['polyline_lon_abm'].to_list() + _df['polyline_lon_gps'].to_list()
t=_df['time'].to_list() + _df['time'].to_list()

_df['polyline_lat_abm_norm'] = (_df['polyline_lat_abm'] - min(lat)) / (max(lat) - min(lat))
_df['polyline_lon_abm_norm'] = (_df['polyline_lon_abm'] - min(lon)) / (max(lon) - min(lon))
```

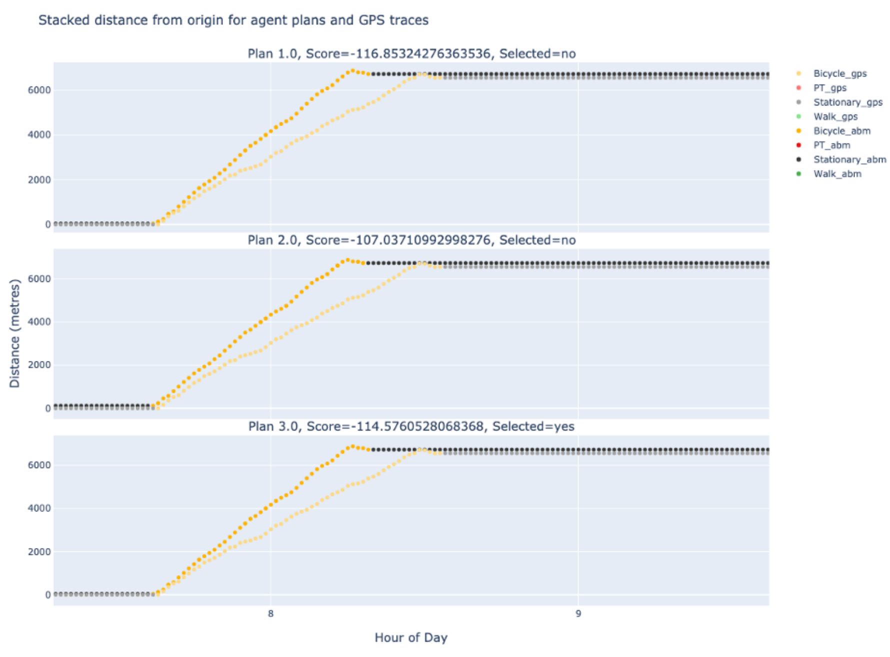

FIGURE 7. Close-up of distance from origin (home) location for GPS and agent traces

```
_df['time_norm'] = (_df['time'] - min(t)) / (max(t) - min(t))
_df['polyline_lat_gps_norm'] = (_df['polyline_lat_gps'] - min(lat)) / (max(lat) -
    min(lat))
_df['polyline_lon_gps_norm'] = (_df['polyline_lon_gps'] - min(lon)) / (max(lon) -
    min(lon))

abm_points = list(zip(_df['polyline_lat_abm_norm'], _df['polyline_lon_abm_norm'],
    _df['time_norm']))
_df = _df.sort_values('time', ascending=False)
gps_points = list(zip(_df['polyline_lat_gps_norm'], _df['polyline_lon_gps_norm'],
    _df['time_norm']))
polygon_pts = abm_points + gps_points + [abm_points[0]]

points = np.array(polygon_pts)
hull = ConvexHull(points)
volume = hull.volume
```

This extends the earlier metrics most significantly in the fact that it takes account of time and location rather than distances, directly incorporating routing into the calculation.

4.6. **Mode and activity metrics.** In our models we allow agents to innovate with respect to route and mode of transport. This means we may be interested in quantifying whether, and to what degree, the agents are travelling with the right mode of transport. We remove those records for ABM and GPS from the table which are both of mode 'Stationary'—in other words, in activity—we compute the difference between encoded modes and find the percentage of matching data points.

$$\frac{1}{1440 - n}\left(\sum_{t=0}^{1440} \delta_t\right) \tag{3}$$

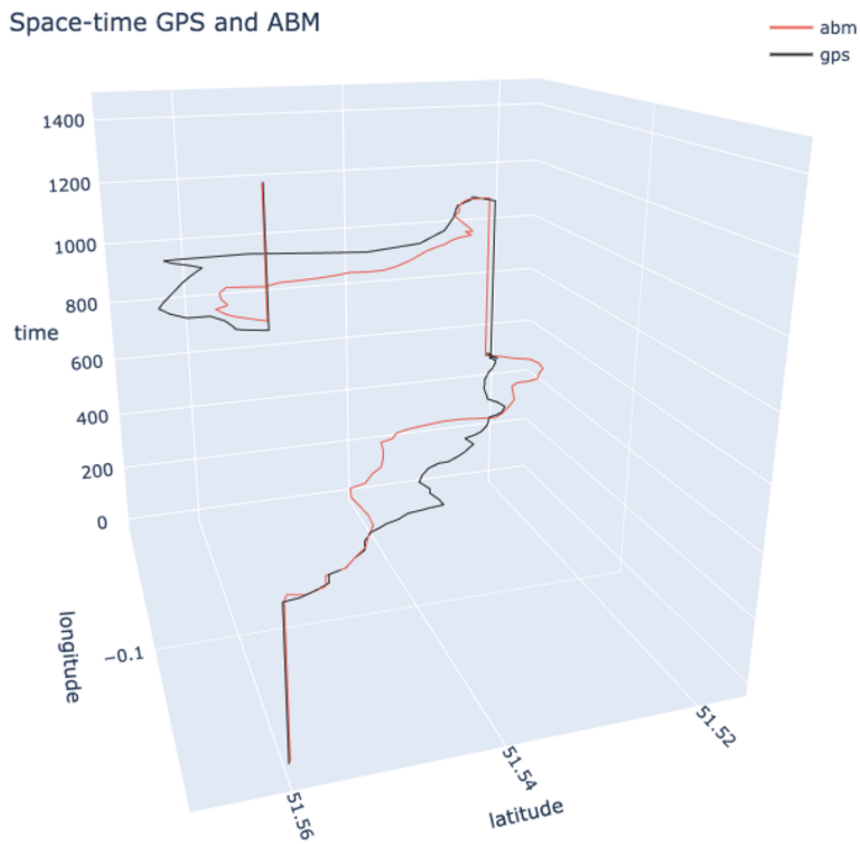

FIGURE 8. ABM and GPS traces in *latitude* × *longitude* × *time* space

Where n is the number of points in time where either the ABM or GPS mode is non-zero and δ_t is a Kronecker-like delta, in this case defined as

$$\delta_t = \begin{cases} 0 & \text{if } mode_{GPS} = mode_{ABM}, \\ 1 & \text{if } mode_{GPS} \neq mode_{ABM}. \end{cases} \tag{4}$$

In addition to above, we may be interested in overall agent activity; a combined metric which includes the mode of agent's travel activity. The computation of which is even simpler:

$$\frac{1}{1440}\left(\sum_{t=0}^{1440} \delta_t\right) \tag{5}$$

With δ_t defined in the same way as above.

5. Conclusions and further study

We have outlined our methods in selecting GPS data for agent creation with the view to later validate the behaviour of such an agent inside an agent-based simulation. We outlined a unified data schema which enables easy comparison of two different data formats and simplifies computations of several interesting numerical metrics. It also aids creation of visualisations, which proved helpful in debugging and developing a deeper understanding of our models. The most expensive of the computations outlined in this paper is the unpacking of the spatial polylines and network routes

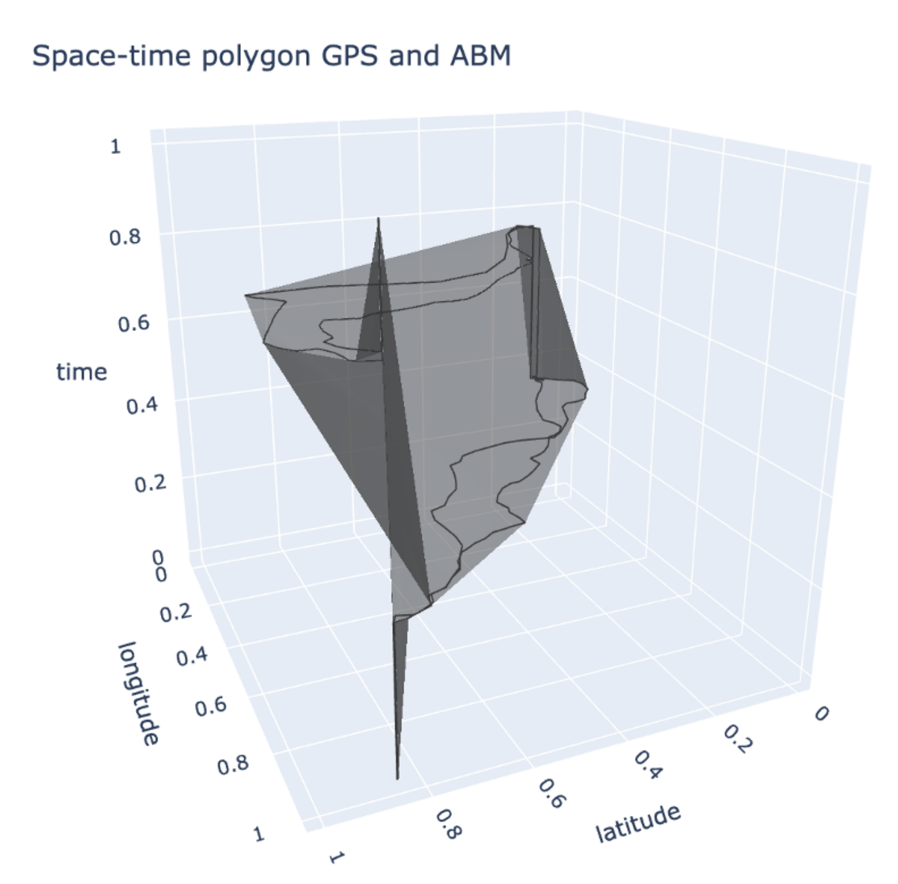

Figure 9. Convex Hull of the points contained in ABM and GPS traces for a selected agent plan

(which require network information to be loaded into memory for reference). A number of these processes can be split to be processed in parallel, each agent is independent of the others. The metrics themselves, when the minute time granularity is chosen, are fast to compute, but again could be processed for each agent independently if the size of the GPS agent pool is large. Finally, this work raises more interesting questions we could consider.

(1) We could perform a study on the amount of GPS data needed to sufficiently validate the model without skewing the overall model demand.
(2) Our models rely on choices of numeric values for various monetary utilities. We can choose the values such that the model simulates behaviours closest to validation or benchmarking data. It would be interesting to take above GPS-based metrics and perform sensitivity analysis to ascertain which metric carries the most impact in driving model configuration.
(3) It would be interesting to initiate the agents with a completely different mode of transport and find how many iterations are required to achieve sufficient matching to GPS data.
(4) Given our GPS data, we could also add validation of the specific public transport service the agents use to travel.

References

[Flohrer et al., 2005] Flohrer, C., Dach, R., Hugentobler, U., Schaer, S., and Beutler, G. (2005). Validating ocean tide loading models using gps. *Journal of Geodesy*, 78:616–625.

[Gurram et al., 2019] Gurram, S., Sivaraman, V., Apple, J. T., and Pinjari, A. R. (2019). Agent-based modeling to simulate road travel using big data from smartphone gps: An application to the continental united states. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 3553–3562.

[Kumar et al., 2014] Kumar, S., Tan, E., and Razul, S. e. a. (2014). Validation of the iri-2012 model with gps-based ground observation over a low-latitude singapore station. *Earth, Planets and Space*, 66:17.

[Olsen and Raunak, 2016] Olsen, M. and Raunak, M. (2016). Metamorphic validation for agent-based simulation models.

[Ormerod and Rosewell, 2009] Ormerod, P. and Rosewell, B. (2009). Validation and verification of agent-based models in the social sciences. In Squazzoni, F., editor, *Epistemological Aspects of Computer Simulation in the Social Sciences*, pages 130–140, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Rand and Rust, 2011] Rand, W. and Rust, R. T. (2011). Agent-based modeling in marketing: Guidelines for rigor. *International Journal of Research in Marketing*, 28(3):181 – 193.

[Shone et al., 2019] Shone, F., Casey, G., Kozlowska, K., Bruce, C., and Karim-Khaku, F. (2019). An agent-based model for london: Exploratory case study collaboration with transport for london.

[Stevenson et al., 2013] Stevenson, C. D., Ferryman, M., Nevin, O. T., Ramsey, A. D., Bailey, S., and Watts, K. (2013). Using gps telemetry to validate least-cost modeling of gray squirrel (sciurus carolinensis) movement within a fragmented landscape. *Ecology and Evolution*, 3(7):2350–2361.

[Szűcs, 2012] Szűcs, E. (2012). Validation of goce time-wise gravity field models using gps-levelling, gravity, vertical deflections and gravity gradient measurements in hungary. *Periodica Polytechnica Civil Engineering*, 56(1):3–11.

[Yin et al., 2018] Yin, M., Sheehan, M., Feygin, S., Paiement, J., and Pozdnoukhov, A. (2018). A generative model of urban activities from cellular data. *IEEE Transactions on Intelligent Transportation Systems*, 19(6):1682–1696.

Arup, London, UK
Email address: kasia.kozlowska@arup.com
Email address: fred.shone@arup.com
Email address: gerard.casey@arup.com