

Sunsetting skim matrices: A trajectory-mining approach to derive travel time skim matrix in dynamic traffic assignment for activity-base model integration

Ye Tian

Tongji University
tianye@tongji.edu.cn

Yi-Chang Chiu

University of Arizona
chiu@email.arizona.edu

Jian Sun

Tongji University
sunjian@tongji.edu.cn

Chen Chai (corresponding author)

Tongji University
chaichen@tongji.edu.cn

Abstract: The travel impedance skim matrix is one of the most essential intermediate products within transportation forecasting models and is a fundamental input for activity-based transportation forecasting models. It reflects interzonal travel time, travel time reliability, travel costs, etc. by time of day. The traditional method to obtain skim matrices is to execute multiple times of time-dependent, shortest-path calculations. However, the computational and memory use burden can easily increase to an intractable level when dealing with mega-scale networks, such as those with thousands of traffic-analysis zones. This research proposes two new approaches to extract the interzonal travel impedance information from the already existing vehicle trajectory data. Vehicle trajectories store travel impedance information in a more compact format when compared to time-dependent link performance profiles. The numerical experiments highlight huge potential advantages of the proposed approaches in terms of saving both memory and CPU time.

Article history:

Received: April 14, 2019

Received in revised form:
November 4, 2019

Accepted: January 19, 2020

Available online: November 10,
2020

1 Introduction

The travel impedance or travel cost skim matrix is one of the most essential intermediate products within transportation forecasting models and is usually prepared after the route assignment is complete. It reflects the interzonal travel costs by time of day and is explicitly adopted by regional planning agents in trip distribution and mode choice steps within four-step forecasting models. The impedance is the summation of various terms such as travel time,

Copyright 2020 Ye Tian, Yi-Chang Chiu, Jian Sun & Chen Chai

<http://dx.doi.org/10.5198/jtl.2020.1551>

ISSN: 1938-7849 | Licensed under the [Creative Commons Attribution – Noncommercial License 4.0](https://creativecommons.org/licenses/by-nc/4.0/)

toll, travel time reliability, safety, etc. In trip distribution model, the amount of trips for each origin-destination-departure time slot (*o-d-t*) combination will be determined based on the corresponding travel impedance. Meanwhile, mode choice will be determined based on the travel impedances of different transportation modes. In gravity model (Isard, 1956; Levinson & Kumar, 1994), a widely known trip distribution model, the number of trips has a general proportional relationship with the reciprocal of the travel impedance of the corresponding *o-d-t*. The iterative four-step procedure aims to achieve consistency between the skim matrices generated at sequential iterations (Boyce, O'Neill, & Scherr, 2008). Such feedback mechanism found in four-step models is explicitly discussed in (Boyce, Zhang, & Lupa, 1994; Loudon, Parameswaran, & Gardner, 1997; Meyer, 2009).

Interzonal travel impedance data is also the fundamental input for another active stream of transportation planning models, Activity-Based transportation forecasting Models (ABM) (Mitchell & Rapkin, 1954). This is where travel behavior is better understood and represented, and where individual responses to the changes in travel environment and how the responses are temporally correlated is explored (Lam & Yin, 2001). Widely used ABMs include DaySim (Reinhart, 2006), SACSIM (Bradley, Bowman, & Griesenbeck, 2007) and CT-RAMP (Davidson, Vovsha, Freedman, & Donnelly, 2010). Throughout ABMs, network skim matrices are repeatedly called for interzonal travel impedance queries in order for ABMs to execute destination choice, departure time choice, etc.

The interzonal travel impedance data in skim matrix is conventionally captured by multiple calls of shortest-path algorithm, which is costly in both computational time and memory usage. In this paper, we aim to develop cost-effective approaches to obtain interzonal travel impedance so as to address computational and memory usage burden.

1.1 The usage of skim matrix

As a major category of route assignment models, Simulation-based Dynamic Traffic Assignment (SBDTA) models have drawn considerable attention and research efforts from the research community in the last three decades (Peeta & Ziliaskopoulos, 2001; Tian, Chiu, & Sun, 2019). With growing maturity in computational efficiency and solution quality, SBDTA is now considered as a promising approach to support various long- and short-term operational planning needs, such as pricing, provision of traffic information, integrated corridor management, evacuation, etc. SBDTA packages include DYNASMART (Mahmassani, 2000), DynaMIT (Ben-Akiva, Bierlaire, Koutsopoulos, & Mishalani, 1998), DynusT (Chiu, Nava, Zheng, & Bustillos, 2011; Li, Zhu, Sun, & Tian, 2019; Tian & Chiu, 2014), Dynameq (Florian, Mahut, & Tremblay, 2006), DTALite (Zhou, Taylor, & Pratico, 2014), and TransModeler (Caliper, 2011), to name a few.

Due to this potential, researchers have begun seeking out ways to connect ABM and SBDTA models. The integration models between SBDTA and ABMs have been explored by various researchers (Davidson et al., 2007; Hao, Hatzopoulou, & Miller, 2010; Lam & Yin, 2001; Lin, Eluru, Waller, & Bhat, 2008). As shown in Figure 1, ABM and SBDTA can be integrated in an iterative manner. From a four-step model perspective, ABM fully covers the roles of trip generation, trip distribution, and mode choice models. Travel demand is then produced and is fed into SBDTA in format of trip and tour roster. SBDTA then executes traffic assignment, simulates traffic dynamics, and produces network performance indices including the travel impedance skims. The network-wide Level-of-Service (LOS) measures and skims that result from SBDTA become inputs to ABM. SBDTA and ABM interplay in an iterative process until convergence is reached. As stated in Castiglione, Bradley, and Gliebe (2015), network skims are endogenous to the overall ABM system because they are produced by the network assignment model component that is linked to the ABM component. As such, the network skims should be carefully constructed to provide unbiased information required by ABM component.

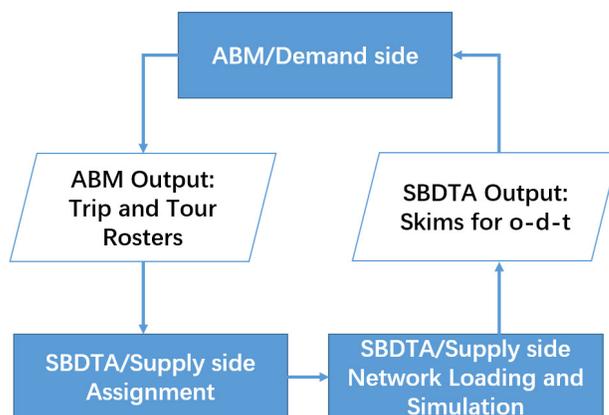


Figure 1. The illustrative cooperation between SBDTA and ABM. SBDTA and ABM could be integrated in an iterative manner to perform traffic forecasting. Skim is one of the main outputs of SBDTA that would be feedback to ABM for trip generations.

1.2 The size of skim matrix

In SBDTA packages, the traditional way to obtain a travel impedance skim matrix is to execute multiple instances of Time-Dependent Shortest-Path (TDSP) (Tian & Chiu, 2014) calculations based on time-dependent link travel time data, if only travel time is considered as a component within travel impedance. However, the computational and memory usage burden can easily increase to an intractable level when dealing with mega-scale networks with thousands of Traffic-Analysis Zones (TAZs). As stated in a FHWA report (Castiglione et al., 2015):

This network information is typically generated at more spatially aggregate levels such as TAZs, even in activity-based models where finer spatial resolutions, such as microzones, are used for other model inputs, in order to avoid having to produce and store very, very large skim matrices.

In that regard, practitioners attempted to aggregate network dynamic information from either the spatial dimension or the temporal dimension in order to reduce the size of skim matrices. However, populating and storing skims is still very difficult for modeling agents with limited computing resources, even in cases where spatial network information has already been highly aggregated into the TAZ level. Conventionally, a three-dimensional matrix is constructed and utilized as the container of travel impedance skims as shown in Figure 2. The three dimensions are the origin TAZ, destination TAZ and departure time. Departure time is usually discretized into uniformly divided time slots. Therefore, the memory usage for storing time-varying skim matrix is of memory size $|O|*|D|*|T|$, where O is the set of original TAZs, D is the set of destination TAZs and T is the set of departure time intervals.

The earliest ABM only used 4 or 5 time intervals across a 24 hour period, while nowadays most new ABM systems use time intervals as small as 15, 30, or 60 minutes (Castiglione et al., 2015). Now, considering a network with $m=2,500$ TAZs and a one day planning horizon, the total amount of entities within the skim matrix will be $2,500*2,500*\frac{1,440}{15}=600$ million if the interval is set at 15 minutes ($|T|=1,440/15=96$, which denotes the amount of departure time intervals). If travel time is stored as a float, the three-dimensional matrix alone would occupy around 2.34 GB of memory. Meanwhile, TDSP needs to be called for 600 million times if it is a one-to-one version (or 240 thousand times for all-to-one TDSP). It takes roughly 2 hours to generate a travel impedance skim matrix for the Sacramento, CA metropolitan network (1,532 zones, 2 hour skim time interval length, one-to-one TDSP)

on a 48-core 1.90 GHz server. Therefore, innovative approaches to obtain travel impedance that save both memory and CPU time, rather than just aggregating network dynamic information from either the spatial dimension (i.e., to aggregate to larger TAZs) or the temporal dimension (i.e., to aggregate it into longer departure time intervals), are in urgent need.

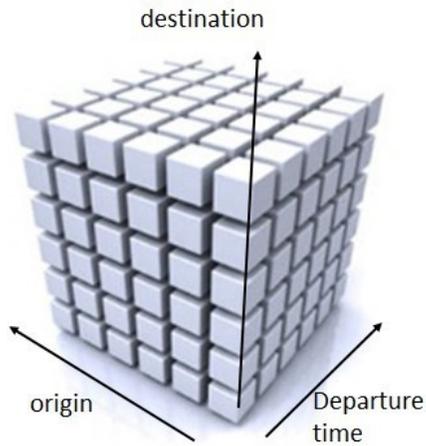


Figure 2. Structure of skim matrix. The matrix is of memory size $|O|*|D|*|T|$.

1.3 Research objectives

To make the best use of existing data generated by SBDTA tools and come out with new approaches to derive interzonal travel impedance, one may find vehicle trajectory data of great value. Vehicle trajectories are usually produced by SBDTA packages for animation and analysis purposes. It records the nodes along each trajectory in a sequence and the arrival time of each node as well. The interzonal travel time is actually buried in the massive amount of data in the form of trajectory segments. In an ideal user equilibrium state, each vehicle follows its corresponding shortest path to destination. In the final solution of SBDTA, the actual experienced travel time and the shortest-path travel time of vehicles are so close with each other (Chiu, Bottom, et al., 2011) that we could make the assumption that each vehicle does traverse on its corresponding shortest path. For vehicle i , suppose that n_i different zones have been traversed along its trajectory, totaling $\binom{n_i}{2}$ pairs of interzonal travel time data points that could be extracted from this certain trajectory, and each of them is the shortest-path travel time between the corresponding zone pair.

Therefore, this paper presents two new ideas of mining the interzonal travel impedance information from the already existing vehicle trajectories produced by SBDTA tools. The travel time skim matrix is maintained in memory in a much compact format to be queried by ABM. The approaches are designed such that ABM and SBDTA can communicate with each other in a more unimpeded way. It is not guaranteed that the interzonal travel time of every $o-d-t$ combination can be obtained by scanning through multiple trajectories, but it is certain that a large portion of the skim matrix can be computed by doing so. Another noteworthy point is that in ABM models, it is of higher possibility to evaluate again an $o-d-t$ that have been recorded in trajectories in previous iterations (mainly the $o-d-t$ with popular origin/destination and time of day with higher travel demand) in current iterations. Meanwhile it is of lower possibility to query an $o-d-t$ not covered by existing trajectories. In that regard, trajectory mining

approaches have the potential to capture most of the $o-d-t$ that will be queried in ABM even though it is not necessary to capture 100% of all possible $o-d-t$ combinations.

2 Methodology

Vehicle trajectory profiles contain a vast amount of information, but the portions that will be utilized in this research are: 1) Node sequence along the trajectory, and 2) Arrival time of each node.

Note that if only the arrival time of each node is recorded, it can only represent the travel time information rather than the travel impedance or interzonal accessibility information. However, other parameters, such as toll, travel time reliability, etc. are also pulled out from SBDTA tools, making it possible to acquire general interzonal impedance data. At the same time, trip trajectories for other modes such as transit can also be generated. For the sake of simplicity, the travel time of auto trips will be used to represent travel impedance/accessibility throughout the rest of the paper.

In ABMs, the network skim matrix is usually called based on a query manner. For each individual traveler, the travel impedances pertaining to his/her possible $o-d-t$ options will be queried to assess the utility of each option. Therefore, a query-based model to address discrete $o-d-t$ combination queries and to output corresponding interzonal travel times is better suited to ABMs for the skim generation process. Otherwise computational time would be wasted to generate the travel impedances of $o-d-t$ that would never be queried by ABM. The following proposed approaches will be performed in this query-mode.

Two new approaches are proposed and they are: 1) single-trajectory mining approach (single TM), and 2) correlated-trajectory mining approach (correlated TM). The former one would only extract travel time information from single trajectory, whereas the latter one also considers $o-d-t$ that could only be captured by combining multiple trajectory segments together.

2.1 Single-trajectory mining approach (with no correlation)

This approach does not consider the correlation between different trajectories and thus is termed as single-trajectory approach. This approach is executed as follows. All trajectories are first scanned in pre-process to extract $o-d-t$ travel time information, which is then stored in a compact two-dimensional matrix. Once queried by ABM, the matrix is examined for corresponding $o-d-t$.

[Pre-process] Examine all the intermediate nodes within each vehicle trajectory. Each vehicle trajectory yields $\binom{n_i}{2}$ travel time data points, where n_i denotes the number of zones vehicle i visits. Since only a portion of $o-d-t$ combinations are captured, a compact data structure, rather than a traditional three-dimensional skim matrix, is maintained in memory. While processing vehicle trajectories, the travel time data is placed into a two-dimensional matrix with only the travel time of those captured $o-d-t$ combinations maintained. The two dimensions are the origin zone and the departure time interval. Each entity in the matrix contains the destination and travel time.

Take the trajectory of vehicle 1 shown in Figure 3 as an example. This vehicle visits zone A, B, C, and E along its trip in a sequence. The arrival times of those zones are timestamps 1, 2, 3, and 4, respectively. When single TM approach is applied, all the intermediate nodes along the trajectory will be examined, which means that the arrival times of each zone along its trip is collected. The interzonal travel time of $o-d-t$ combinations A-B-1, A-C-1, A-E-1, B-C-2, B-E-2 and C-E-3 could be obtained as a result (assuming departure time interval equals 1). There are 6 data points in total, which is in agreement with $\binom{4}{2}$.

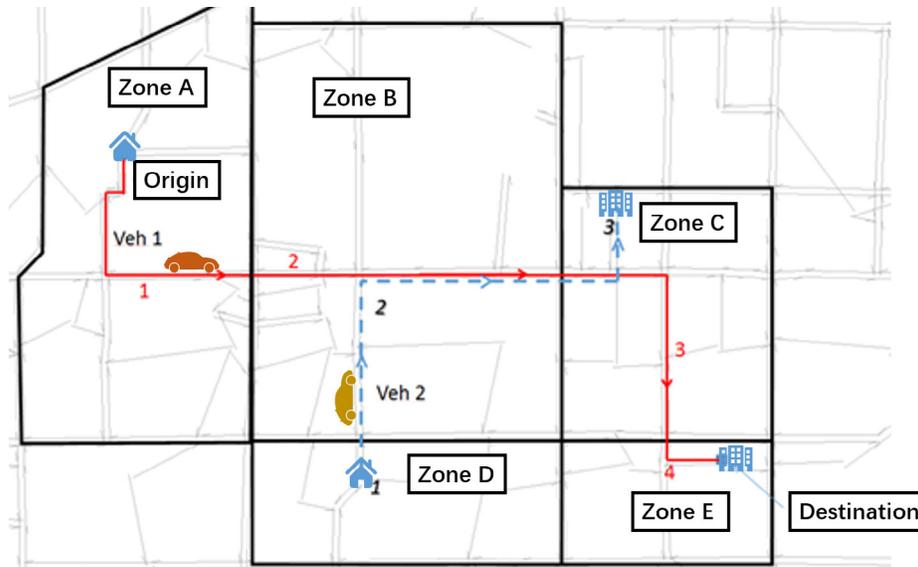


Figure 3. Illustration of vehicle trajectories. There are two vehicles in network. Vehicle 1 traverses from zone A to B, C, E, whereas vehicle 2 traverses from zone D to B, C.

Note that this approach doesn't consider correlation between trajectories. Therefore, the shortest paths that could only be constructed by trajectory segments from two or more trajectories cannot be captured.

[Query] When a query from ABM comes in, we first search the two dimensional matrix to see whether this specific *o-d-t* combination has already been captured after pre-processing. If so, the corresponding value in the matrix can be extracted for use directly. If not, a generation link within the origin zone is randomly selected as the origin. The beginning of the time interval is defined as the departure time. A one-to-one TDSP calculation is called, and the resulting travel time can be used to represent the time for that *o-d-t* combination. The resulting travel time from the TDSP calculation for this former uncaptured *o-d-t* combination is then inserted into the two-dimensional matrix for the use of later duplicate *o-d-t* queries.

2.2 Correlated-trajectory mining approach (with correlation)

This approach considers the correlation between multiple trajectories and thus is termed as correlated-trajectory mining approach. This approach is executed as follows.

[Pre-process] Note that there are tons of trajectories that simultaneously exist in the network and there must exist some *o-d-t* combinations whose corresponding shortest paths are composed of multiple trajectory segments from multiple vehicle trajectories. By building such a time-space diagram as shown in Figure 4, the correlation between different trajectories can be captured. The horizontal axis stands for aggregation intervals (interval length μ might be 1 min, 0.5 min, etc.), and the vertical axis stands for TAZs. Each solid arrow in the diagram represents a connection between the two connected zones at a certain time interval.

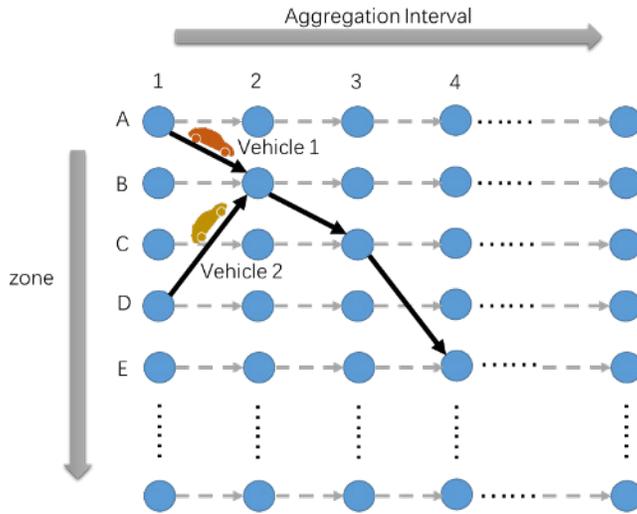


Figure 4. Compact time-space diagram for correlated TM. Solid arrows represent interzonal connections revealed in vehicle trajectories. Dashed arrows represent default intrazonal connections.

Suppose there are two vehicles in the network (shown in Figure 3), besides vehicle 1, which is already described above. Vehicle 2 visits zone D, B, and C in a sequence. The arrival times of those zones are timestamp 1, 2, and 3, respectively. The trajectories of vehicles 1 and 2 are decomposed and recorded in the time-space diagram in the form of the solid arrows. Let (z,t) denote zone z and time interval t (i.e., the unit at the z th row, t th column in Figure 4), and suppose the aggregation interval length μ is 1 min. The trajectory of vehicle 1 is represented in the form of $(A,1) \rightarrow (B,2) \rightarrow (C,3) \rightarrow (E,4)$. Meanwhile, the trajectory of vehicle 2 is represented as $(D,1) \rightarrow (B,2) \rightarrow (C,3)$. It is assumed that only the nodes within the neighboring columns can be connected. Note that the dashed arrows always exist as they represent the intrazonal connections.

By performing pre-process, the time-space diagram is maintained in memory in a much compact format and the correlation between trajectories is maintained in this unweighted diagram. The diagram is of dimension $|M| \times |T'|$, where $|M|$ is the number of zones and $|T'|$ denotes the number of aggregation intervals.

Note that μ should be wisely chosen, as a short μ would lead to larger $|T'|$ with a given planning time horizon, and further lead to exceeded computational complexity of searching and memory storage as they are both highly dependent on $|T'|$. Besides, if μ is too short, it's likely that vehicles would stay in one single zone for multiple aggregation intervals, which brings in induced estimation error. Whereas a long μ , for instance, 5 min or 10 min, would scarify travel impedance accuracy to some extent. In situation where $\mu=5$ min, the resulting travel time has to be multiples of 5 min. Besides, if μ is too long, it's likely that vehicles would traverse multiple zones within one single aggregation interval, which makes the information of all those intermediate visited zones deadweight losses.

At the same time, a two dimensional matrix, which is the same with the one described in the single-trajectory mining approach, is initialized after the pre-process and before querying. This matrix is used to store the travel times of those $o-d-t$ combinations that are not captured by trajectory mining but later captured by the TDSP calculation.

[Query] When a query comes in, a Breath-First Search (BFS) (Park, Lee, & Yoon, 2013) is first performed in the time-space diagram to see if there are any connections between o and d , departing at t .

Note that during the BFS, an eligible criterion should be checked every time a zone is to be put into the scan eligible set. Let z be this zone, if $\frac{disz}{t'-t} < v_{min}$, where dis stands for the distance between origin and zone z , t' stands for the current time, and v_{min} is the pre-defined minimal speed threshold, then zone z is not an eligible zone, since the found path is not likely to be a shortest path between the origin and zone z . This non-optimal situation is brought in due to the fact that the correlation between trajectories is taken into consideration. For instance, there is a path in Figure 4 such that $(D,1) \rightarrow (B,2) \rightarrow (C,3) \rightarrow (E,4)$, that is generated by combining the sub-segments of trajectories of both vehicle 1 and vehicle 2. However, it doesn't appear to be a shortest path according to the network topology as has been shown in Figure 3. One may need to exclude such a path by adopting the eligible criterion mentioned above. Generally speaking, the higher v_{min} is, the less connections will be found in the time-space diagram shown in Figure 4 and thus less $o-d-t$ combinations will be captured. The BFS terminates whenever the destination is reached, since the path obtained by BFS is guaranteed to be the shortest one among all those paths that can be found in the time-space diagram. By adopting unweighted static BFS, the worst-case time complexity to find the shortest path is now $O(|M|+|T'|)$.

If an eligible path for the queried $o-d-t$ is found, the travel time can be directly detected by subtracting the departure time from the arrival time. If not, we look into the two dimensional matrix to see if the travel time of that $o-d-t$ has been calculated by previous TDSP calculations. If this $o-d-t$ has never been handled before, then a generation link within the origin zone is randomly selected as the origin, the beginning of the time interval is defined as the departure time, and a one-to-one TDSP is called. The resulting travel time is used as the representing time for this certain $o-d-t$ combination. The result is then placed into the two dimensional matrix for future use.

The following Figure 5 presents the general workflow of both proposed trajectory mining approaches.

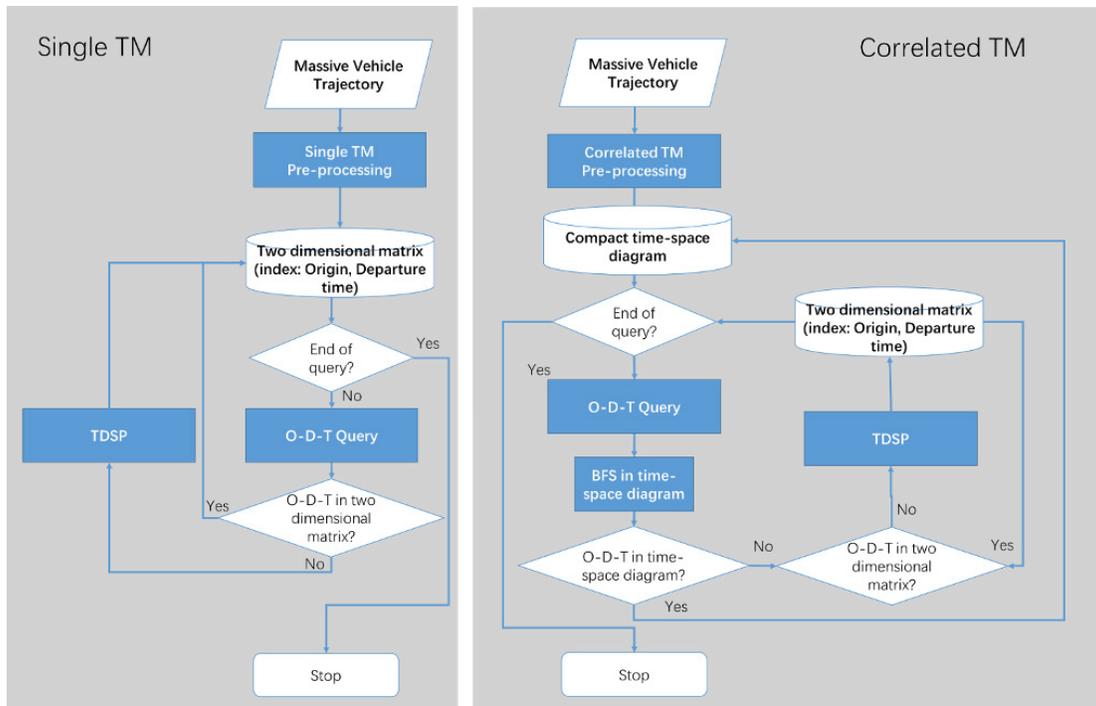


Figure 5. The flowcharts of single TM and correlated TM

2.3 TDSP approach

The TDSP approach, which is one of the traditional methods to generate skim matrices in SBDTA tools, is also listed here for comparison.

[Pre-process] For every $o-d-t$ combination, call one-to-one TDSP and use the resulting travel time as the representing time for that $o-d-t$ combination. A randomly selected generation link within the origin zone is regarded as the origin for TDSP, a randomly selected destination node within the destination zone is regarded as the destination, and the beginning of the departure time interval is regarded as the departure time. The skims are maintained in memory as a three-dimensional matrix as discussed above.

[Query] Directly extract the corresponding travel time from the three-dimensional matrix based on the queried $o-d-t$.

3 Numerical test

3.1 Experiment configuration

The aforementioned three approaches were examined on two networks. The first one is the Shanghai Expressway network, which consists of 226 zones, 2,610 nodes and 3,158 links. A total of 240,000 vehicles were loaded for a 1-hour simulation. The skimming interval is 15 minutes. The Shanghai Expressway SBDTA model was newly built to facilitate the promotion of SBDTA in China. The second network is the Sacramento, CA (SACOG) network, which consists of 1,532 zones, 9,956 nodes and 21,698 links. A total of roughly 475,000 single-occupant vehicles were loaded for a 2-hour simulation. The skimming interval for SACOG is 2 hours. Figure 6 presents the snapshots of both networks, along with the TAZ configuration for SACOG network. The SBDTA package DynusT was used to generate the vehicle trajectory files. DynusT is based on a mesoscopic simulation engine, therefore, it is capable of recording trajectory data and printing it out to files for animation, visualization, and analysis purposes. The trajectory data (i.e., the node sequence data) is maintained along with the arrival time of each node and the distance of the corresponding link. A threshold n_{min} for the length of trajectory was also introduced. The trajectories containing less than n_{min} nodes are not considered as valid samples for both trajectory mining approaches. A stand-alone program of DynusT was written in C++ to perform the trajectory mining approaches. Certain amount of random $o-d-t$ combinations was generated and regarded as the queries from ABM. The skim table (in various format in three approaches) is queried for the travel times of those $o-d-t$ combinations.

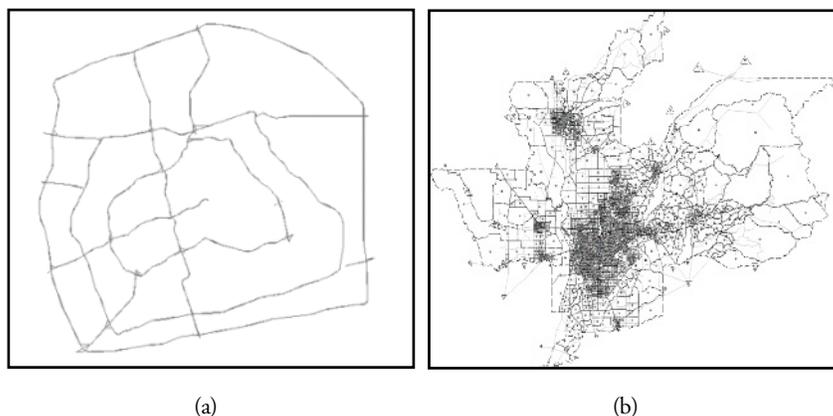


Figure 6. Shanghai Expressway and SACOG network snapshots: (a) Shanghai Expressway network, (b) SACOG network along with TAZ configuration. SACOG network is relatively larger than Shanghai Expressway network in terms of number of nodes/links/zones and number of vehicles.

3.2 Memory usage savings

Table 1 and Table 2 record the major Measures of Effectiveness (MOE) (i.e., memory usage, CPU time and result accuracy) of the three approaches for both network tests. It can be obviously observed that the correlated TM approach is visibly superior to the other two methods in Shanghai Expressway network test, saving almost 90% of memory usage compared to TDSP approach. As for Sacramento network tests, correlated TM also saves almost 90% memory compared to the TDSP approach before querying and more than 60% after 10 million queries from DaySim due to its compact and elegant data structure for the *o-d-t* combinations where travel time information is buried within the scanned trajectories.

Table 1. Performance of the three approaches for Shanghai Expressway test ($n_{min}=10$)

	Single TM	Correlated TM ($\mu=1$ min, $v_{min}=$ 32.2 km/h)	TDSP approach
Scanned trajectories	156,401	156,401	N/A
Memory usage after pre-process (MB)	105	12.14	115
Captured rate after pre-process	72.6%	92.9%	100%
Memory usage after 10 million queries (MB) (travel time of uncaptured <i>o-d-t</i> is maintained in memory)	132	18.32	115
Deviation compared to single TM approach (for all <i>o-d-t</i> combination) (min)	N/A	2.05	1.69
CPU time for pre-process (sec)	22.3	11.4	423
CPU time for 100,000 queries (sec)	39.1	3.1	0.007
CPU time for 1,000,000 queries (sec)	68.9	27.0	0.023
CPU time for 10,000,000 queries (sec)	105.2	255.5	0.265

Table 2. Performance of the three approaches for SACOG test ($n_{min}=10$)

	Single TM	Correlated TM ($\mu=1$ min, $v_{min}=$ 32.2 km/h)	TDSP approach
Examined trajectories	393,249	393,249	N/A
Memory usage after pre-process (MB)	130	20.8	166
Captured rate after pre-process	56.4%	79.9%	100%
Memory usage after 10 million queries (MB) (travel time of uncaptured <i>o-d-t</i> is maintained in memory)	226.5	63.9	166
Deviation compared to single TM approach (for all <i>o-d-t</i> combination) (min)	N/A	3.56	1.42
CPU time for pre-process (sec)	47	32.6	~7200
CPU time for 100,000 queries (sec)	130.6	97.0	0.007
CPU time for 1,000,000 queries (sec)	1201.6	860.5	0.026
CPU time for 10,000,000 queries (sec)	3648.0	5110.9	0.273
CPU time for 50,000,000 queries (sec)	4883.2	5963.1	1.37

The index “capture rate” in Table 1 and Table 2 represents the amount of the total $o-d-t$ combinations where travel times are extracted from trajectories after pre-processing. The TDSP approach applies TDSP for all $o-d-t$ combinations. Therefore, the capture rate is always 100%. It is not guaranteed that all $o-d-t$ combinations can be captured by the two proposed approaches, since there might be some rural TAZs that have never been traversed by any vehicle trajectories, or there might be some $o-d-t$ combinations with corresponding shortest paths that cannot be composed of the existing trajectory segments. Therefore, generally speaking, the capture rate of the proposed trajectory mining approaches cannot reach 100%. However, since the network topology of Shanghai Expressway network is relatively simpler with no rural area. The travel time of almost all (92.9%) $o-d-t$ combinations could be computed by considering correlated multiple trajectories together.

With the memory saving results in mind, one can tell that for the SACOG network after pre-processing, single TM approach used 78% of the memory used by the TDSP approach and captures travel times of 56% of all possible $o-d-t$ combinations, while correlated TM approach used only 12.5% of the memory used by TDSP and captures travel times of almost 80% of all possible $o-d-t$ combinations. One thing to note is that due to the fact that correlated TM approach takes the correlation between different trajectories into consideration, the capture rate is always higher than single TM approach.

3.3 Travel time accuracy

In terms of the travel time accuracy, the result generated by single TM approach is regarded as the benchmark. The deviation to the result by single TM is calculated as:

$$d_2 = \sqrt{\frac{\sum_{j=1}^n (tt_{2,j} - tt_{1,j})^2}{n}} \quad (1)$$

$$d_3 = \sqrt{\frac{\sum_{j=1}^n (tt_{3,j} - tt_{1,j})^2}{n}} \quad (2)$$

where d_2 denotes the deviation between the results of the two mining approaches, and d_3 denotes the deviation between the results of TDSP approach and single TM approach. n is the total amount of $o-d-t$ combinations, and tt_{kj} is the corresponding travel time of the j th $o-d-t$ generated by the k th approach.

The travel time calculation procedures vary for each of the three approaches and they are definitely related to the intrinsic mechanisms of each approach. Relatively speaking, the results generated by single TM approach are more representative than the other two methods. It is because the single TM approach considers the average of all trajectory segments representing the same $o-d-t$. Whereas in the TDSP approach, a randomly selected generation link within the origin zone and the beginning of the skim aggregation interval are utilized as the o and t of the TDSP. This o and t are just one sample within the aggregated space and time. Such a strategy may not be as representative as necessary in order to represent the average travel time for the entire zone, especially when zones are relatively large. As for correlated TM approach, deviation from the single TM approach is mainly brought in from the aggregated time perspective. This approach aggregates a range of time (i.e., the whole time period within one aggregation interval) into one point in the compact time-space diagram shown in Figure 4, and this aggregate time interval length is different from the skim generation interval length. This fact undoubtedly introduces deviation when compared with the single TM approach. Besides, to consider combined shortest path in

correlated TM may also introduce additional deviation compared to single TM. From Table 1 and Table 2 one can tell that the deviation between the results of the correlated TM approach and that of the single TM approach is higher than the deviation between the TDSP approach and the single TM approach.

Note that the term “deviation” is used here rather than “error,” because it is difficult to tell which result is the ground-true interzonal travel time.

3.4 CPU time performance

Next, CPU time performance is recorded. The CPU time is mainly composed of CPU times for pre-processing and querying. The pre-processing times of both trajectory mining approaches claim commanding leads over the TDSP approach. The trajectory mining approaches won't involve TDSP during the pre-process, resulting in less CPU time. However, the CPU time of trajectory mining approaches is higher than TDSP approaches in the query procedure since not 100% of the travel times of *o-d-t* combinations are stored in memory when utilizing mining approaches. For those missing *o-d-t* combinations, TDSP will be called during querying, and a great portion of CPU time is actually spent on those TDSP calls. For the correlated TM approach specifically, the BFS conducted in the compact time-space diagram also takes extra CPU time, although the time complexity for worst-case unweighted BFS is much lower than TDSP.

After a certain large amount of queries, eventually the travel times of all *o-d-t* combinations will be stored in either the time-space diagram or the two-dimensional matrix. Results for later *o-d-t* queries can be extracted directly from either of those two storages. Therefore, the total CPU time for both trajectory mining approaches is not proportional to the amount of queries but more like a concave curve, as one can tell from Figure 7.

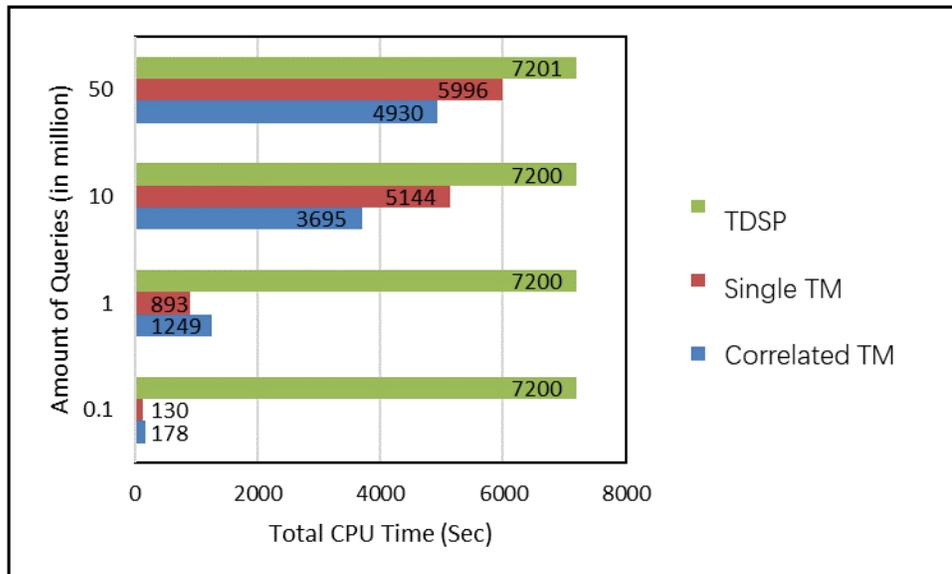


Figure 7. Total CPU time vs. amount of queries in SACOG test. Both trajectory mining approaches spend less CPU time compared to TDSP even in extreme cases where amount of queries is extremely high (the normal amount of queries for this SACOG model is 2.5-7.5 million)

One noteworthy note is that in a typical ABM-SBDTA integration, it is less likely that *o-d-t* combinations associated with rural TAZs or off-peak times that have never been covered by any trajectory segments will be queried. Essentially, the *o-d-t* queries are somehow non-uniformly distributed rather

than in a uniformly distributed manner used in this numerical test. The possibility that the queried $o-d-t$ is already captured by trajectory mining approach will be higher in real integration practices, and this fact further benefits both proposed trajectory mining approaches. Meanwhile, for a normal ABM model (e.g., DaySim, CT-RAMP, etc.) with a similar size to SACOG, roughly 5-15 destinations are sampled depending on the activity types for destination choice models, and a 5-30 min departure time window is usually considered for the departure time choice model. Therefore, for the tested SACOG network, there are roughly 2.5-7.5 million times that the skim is queried within each iteration in practice and it's very unlikely that the skim will be queried for as much as 50 million times. As a result, one can conclude that both trajectory mining approaches tend to save CPU time in practice, especially in larger networks (SACOG network) with smaller amounts of queries.

3.5 Impact of v_{min} and μ on travel time accuracy of correlated TM

As concluded in previous subsections, correlated TM approach greatly outperforms single TM and TDSP approach in terms of both CPU time and memory usage, at the price of travel time accuracy. The deviation of travel time compared to single TM is due to two reasons: 1) the combination of multiple trajectory segments may not be the true shortest path from the origin zone to the destination zone, and 2) the introduction of aggregation interval in order to construct the compact time-space diagram. Therefore, a sensitivity analysis is conducted herein to evaluate the impact of v_{min} and μ on the travel time accuracy performance of correlated TM.

Figure 8 presents the impact of v_{min} and μ on performance of Correlated TM in SACOG network. One can tell that as v_{min} goes up, the capture rate drops, as the scan eligible set of BFS would shrink. Furthermore, the deviation of the resulting interzonal travel times to single TM drops. It is because the leftover $o-d-t$ that are still captured by correlated TM are now of high accuracy, which is ensured by the high v_{min} . The capture rate would further impact the CPU time as the more $o-d-t$ combinations captured during pre-processing, the less TDSP would be run for missing $o-d-t$ combinations.

On the other hand, a higher μ would lead to higher deviation to single TM and lower CPU time, which is understandable as a higher μ means less number of columns in the compact time-space diagram in correlated TM. It means less depth the BFS would go over to find the destination, whereas the accuracy would be off.

As a conclusion, both v_{min} and μ should be chosen wisely at the discretion of modelers considering the trade-off between efficiency and accuracy.

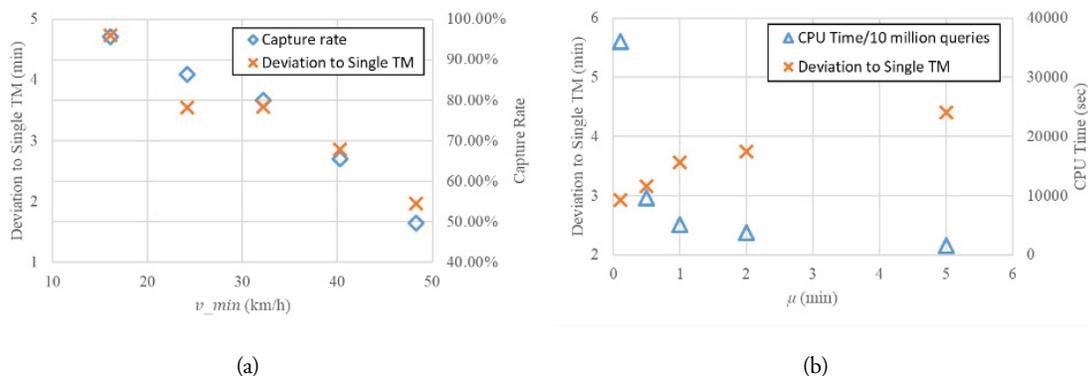


Figure 8. Impact of v_{min} and μ on performance of Correlated TM in SACOG network. (a) Capture rate and Deviation to Single TM VS. v_{min} . Both MOEs are negative correlated to v_{min} . (b) CPU Time and deviation to Single TM VS. μ . CPU time is negative correlated to μ whereas deviation is positive correlated to μ .

4 Conclusions

The paper proposed an innovative approach to acquire interzonal travel impedance skim values. Traditional skim generation methods call for multiple TDSPs, resulting in intolerable computational times and memory usage. The major contributions of this paper are the innovative data mining approaches used to extract travel impedance skims from the existing massive vehicle trajectories produced by SB-DTA tools.

In the first proposed method, data is pulled out from vehicle trajectories and stored in a two dimensional matrix. Pre-processing CPU time is saved, as well as total CPU time. The second proposed strategy nicely builds a compact time-space diagram based on the trajectory data, and the travel time is obtained by performing an unweighted breadth-first-search upon the compact time-space diagram.

Numerical tests are conducted on a newly-built Shanghai expressway network and a U.S. metropolitan network. The correlated-trajectory mining approach shows attractive properties of saving almost 90% of the memory after the pre-process and more than 60% of the memory after relatively large amounts of queries when compared to the TDSP approach in large real-life network tests, mainly due to the elegant and compact data storage format. At the same time, a large portion of CPU time is saved using both trajectory mining approaches when compared to the TDSP approach, especially when the amount of queries is low. The numerical experiments highlight huge potential advantages of the proposed approach in terms of memory usage and CPU time. The two proposed methods are simple in their methodology yet outstanding in performance.

Acknowledgments

This research was sponsored by the National Key Research and Development Program of China (2018YFB1600805), the Shanghai Sailing program (19YF1451200), and the Science and Technology Commission Shanghai Municipality (19692108700, 19DZ1208800).

Data availability statement

Data of Shanghai Expressway system is not available to public since the model itself is one of the deliverables of an on-going project sponsored by the Science and Technology Commission of Shanghai Municipality. Data of SACOG network is available upon request to the corresponding authors.

References

- Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H., & Mishalani, R. (1998). *DynaMIT: a simulation-based system for traffic prediction*. Paper presented at the DACCORD Short Term Forecasting Workshop, Delft, Netherlands.
- Boyce, D., O'Neill, C., & Scherr, W. (2008). Solving the sequential travel forecasting procedure with feedback. *Transportation Research Record*, 2077, 129–135. <https://doi.org/10.3141/2077-17>
- Boyce, D., Zhang, Y., & Lupa, M. (1994). Introducing “feedback” into four-step travel forecasting procedure versus equilibrium solution of combined model. *Transportation Research Record*, 1443, 65–74.
- Bradley, M. A., Bowman, J. L., & Griesenbeck, B. (2007). *Development and application of the SACSIM activity-based model system*. Paper presented at the 11th World Conference on Transportation Research, Berkeley, CA.
- Caliper. (2011). Traffic simulation models. Retrieved from <http://www.caliper.com/TransModeler/Simulation.htm>
- Castiglione, J., Bradley, M., & Gliebe, J. (2015). *Activity-based travel demand models: A primer*. Washington, DC: Transportation Research Board.
- Chiu, Y.-C., Bottom, J., Mahut, M., Paz, A., Balakrishna, R., Waller, T., & Hicks, J. (2011). *Dynamic traffic assignment: A primer*. Washington, DC: Transportation Research Board.
- Chiu, Y.-C., Nava, E., Zheng, H., & Bustillos, B. (2011). *DynusT user's manual*. Retrieved from <http://dynust.net/wikibin/doku.php>.
- Davidson, W., Donnelly, R., Vovsha, P., Freedman, J., Ruegg, S., Hicks, J., . . . Picado, R. (2007). Synthesis of first practices and operational research approaches in activity-based travel demand modeling. *Transportation Research Part A: Policy and Practice*, 41(5), 464–488. <https://doi.org/10.1016/j.tra.2006.09.003>
- Davidson, W., Vovsha, P., Freedman, J., & Donnelly, R. (2010). *CT-RAMP family of activity-based models*. Paper presented at the 33rd Australasian Transport Research Forum (ATRF), Canberra, Australia.
- Florian, M. A., Mahut, M., & Tremblay, N. (2006). *A simulation-based dynamic traffic assignment model: Dynameq*. Montreal, Canada: Centre for Research on Transportation.
- Hao, J., Hatzopoulou, M., & Miller, E. (2010). Integrating an activity-based travel demand model with dynamic traffic assignment and emission models. *Transportation Research Record*, 2176, 1–13. <https://doi.org/10.3141/2176-01>
- Isard, W. (1956). *Location and space-economy*. New York: Wiley .
- Lam, W. H. K., & Yin, Y. (2001). An activity-based time-dependent traffic assignment model. *Transportation Research Part B: Methodological*, 35(6), 549–574. [https://doi.org/10.1016/S0191-2615\(00\)00010-2](https://doi.org/10.1016/S0191-2615(00)00010-2)
- Levinson, D. M., & Kumar, A. (1994). A multi-modal trip distribution model. *Transportation Research Record*, 1466, 124–131. Retrieved from SSRN 1091827.
- Li, Y., Zhu, L., Sun, J., & Tian, Y. (2019). Generating a spatiotemporal dynamic map for traffic analysis using macroscopic fundamental diagram. *Journal of Advanced Transportation*, 2019. Retrieved from <https://doi.org/10.1155/2019/9540386>
- Lin, D.-Y., Eluru, N., Waller, S., & Bhat, C. (2008). Integration of activity-based modeling and dynamic traffic assignment. *Transportation Research Record*, 2076, 52–61. <https://doi.org/10.3141/2076-06>
- Loudon, W., Parameswaran, J., & Gardner, B. (1997). Incorporating feedback in travel forecasting. *Transportation Research Record*, 1607, 185–195. <https://doi.org/10.3141/1607-25>
- Mahmassani, H. S., & Khaled, F. A. (2003). Dynasmart-ip: Dynamic traffic assignment meso-simulator for intermodal networks. In *Advanced modeling for transit operations and service planning* (200–229). Amsterdam: Elsevier.

- Meyer, M. D. (2009). *Transportation planning handbook* (Third edition). Washington, DC: Institute of Transportation Engineers.
- Mitchell, R. B., & Rapkin, C. (1954). *Urban traffic: A function of land use*. New York, NY: Columbia University Press.
- Park, J. I., Lee, Y. P., & Yoon, S. (2013). Partition-based hybrid MIMO decoding schemes with combined depth- and breath-first search. *Applied Mechanics and Materials*, 284–287, 2652–2656.
- Peeta, S., & Ziliaskopoulos, A. K. (2001). Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics*, 1, 233–265.
- Reinhart, C. F. (2006). *Tutorial on the use of daysim simulations for sustainable design*. Ottawa, Ontario: Institute for Research in Construction, National Research Council Canada.
- Tian, Y., & Chiu, Y.-C. (2014). A variable time-discretization strategies-based, time-dependent, shortest-path algorithm for dynamic traffic assignment. *Journal of Intelligent Transportation Systems*, 18(4), 339–351. <https://doi.org/10.1080/15472450.2013.806753>
- Tian, Y., Chiu, Y.-C., & Sun, J. (2019). Understanding behavioral effects of tradable mobility credit scheme: An experimental economics approach. *Transport Policy*, 81, 1–11. <https://doi.org/10.1016/j.tranpol.2019.05.019>
- Zhou, X., Taylor, J., & Pratico, F. (2014). DTALite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration. *Cogent Engineering* 1, 1. <https://doi.org/10.1080/23311916.2014.961345>