Lesson 5

Dynamic Origin-Destination Demand Matrix Estimation (ODME) in DTALite

Please share your comments online for improving this document. If you have any questions or encounter any problems, please feel free to contact us (jiangtao.liu@asu.edu; xzhou74@asu.edu). Your any feedback is greatly appreciated!

Data Set: https://github.com/xzhou99/learning-transportation/tree/master/Lessons/Lesson%205/Data%20Set

Learning Objective:

1. Understanding dynamic OD demand matrix estimation implemented in DTALite;
2. Scenario evaluation in DTALite

This document aims to offer users to understand dynamic OD demand matrix estimation implemented in DTALite and further apply it for scenario evaluation. Section 1 illustrates the data requirement and setting process for performing ODME in DTALite. Section 2 shows the integration of ODME and specific scenario evaluation, which could be work zone, incident, ramp metering, etc., and further provide the guide about how to obtain the user equilibrium result under specific scenarios based on the estimated OD demand matrix/agent data. Section 3 introduces the general methodology of ODME implemented in DTALite.

The core estimation variable of ODME in DTALite is trip production of each zone. Based on the original travel demand in input_demand.csv or the demand file users provide through input_demand_file_list.csv, it is easy to calculate the trip production rate of each zone and the trip distribution ratio of each OD pair. During the ODME process, we fix the original trip distribution ratio of each OD pair and the time profile used to define the percentage of total travel demand departing at some specific time periods. After running the simulation, users can check the calibrated trip production of each zone through ODME_ratio in file ODME_final_result.csv. In addition, the process of each iteration can also be checked in file ODME_zone_based_log.csv and file ODME_link_based_log.csv.

The key input and output files for ODME and scenario evaluation are listed in Table 1. The detail of some files will be explained in the following sections.

Table 1 Key input and output files for ODME

File Group	**Input file list**	**Remark**	**Output file list**	**Remark**
Demand	1. input_demand.csv or input_agent.csv	Demand content for specifying distribution	1. output_agent.csv	Final estimation results represented in terms of agent trajectory format
	2. input_demand_file_list.csv	Demand file specification, specify temporal departure time distribution	2. ODME_zone_based_log.csv 3. ODME_final_result.csv	Zone based production based ratio; Iteration by iteration zone based total production adjustment
Sensor	1. input_link.csv	Store count_sensor_id, speed_sensor_id for referring sensor data	1. ODME_link_based_log.csv	Iteration by iteration link based flow adjustment
	2. sensor_count.csv	Link based sensor count data		
	3. sensor_speed.csv (optional)		2. debug_validation_results.csv	Link based simulated vs. observed results
Scenario	1. input_scenario_settings.csv	ODME and assignment settings	1. output_summary.csv	Iteration-by-iteration UE and ODME statistics
	2. DTASettings.txt	Default settings for sequential ODME run		
	3. scenario_work_zone.csv	Specify the capacity reduction scenarios in the estimation and prediction stages		

1. ODME

The general process of ODME is to adjust the given historical OD demand so that the final assignment results are consistent with observed link traffic measurements, such as, link count, link occupancy, link travel time etc. In addition to the basic traffic network data (input_node.csv, input_link.csv, input_zone.csv, and input_activity_location.csv), it also requires (1) a OD demand matrix seed as our initial demand values, (2) observed sensor data for calibration and (3) scenario settings files for algorithm performance. The demand seed could be a zone-to-zone demand file (such as, input_demand.csv) or an activity-based demand file (such as, input_agent.csv) with time profile in input_demand_file_list.csv, and the sensor utilization involves sensor location information (input_link.csv and sensor_count.csv) and observed traffic data (sensor_count.csv and/or sensor_speed.csv). The specific settings need to be finished in input_secenario_settings.csv and DTASettings.txt, and if necessary, some scenario files also needs to be set up, such as, scenario_work_zone.csv. After ODME and traffic assignment, each vehicle's travel information is stored in output_agent.csv, which can also be treated as the estimated agent-based travel demand file. In addition, the link-based, zone-based and network-based statistics can be found in ODME_zone_based_log.csv, ODME_final_result.csv, ODME_link_based_log.csv, debug_validation_results.csv and output_summary.csv.

(1) Data preparation

As described above, the required basic input data for ODME include the traffic network data (input_node.csv, input_link.csv, input_zone.csv, and input_activity_location.csv), original or historical travel demand data (input_demand.csv and input_demand_file_list.csv), and observed sensor data (sensor_count.csv). The experiment is tested in the West Jordan traffic network, the data set of which can be download at here

The detailed format of sensor_count.csv is listed in Table 2, and sensor location information is also shown in input_link.csv in Table 3.

Table 2. Sample data in sensor_count.csv

count_sensor_id	from_node_id	to_node_id	day_no	start_time_in_min	end_time_in_min	link_count	speed	travel_time_in_min	lane_density
5010->4958	5010	4958	1	990	1050	49.5			
4958->5010	4958	5010	1	990	1050	74.5			
4952->5022	4952	5022	1	990	1050	221.5			

Table 3. Sample data in input_link.csv

link_id	from_node_id	to_node_id	direction	length	number_of_lanes	speed_limit	lane_cap	count_sensor_id	speed_sensor_id
1285	1285	5018	1	0.2384	7	21	1428.6	1285->5018	
1286	1286	11125	1	0.466	7	21	1428.6	1286->11125	
1289	1289	4952	1	0.2427	7	21	1428.6	1289->4952	

In this case, the observed link count from "link_count" is used for ODME, and the start_time_in_min and end_time_in_min defines the corresponding observation time period. If optional density data or travel time data are also available, users can also prepare them in "lane_density" or " travel_time_in_min" in this file. More importantly, to map sensors to links (in file input_link.csv), one can use one of the following two methods to specify a link with sensors: (i) "from_node_id" and "to_node_id", or (ii) field "count_sensor_id", which should be first defined in file input_link.csv. If not, warning messages will be issued.

The traffic network displayed in NeXTA is shown in Figure 1. Those green dots on some links represent the sensors' location in the network.

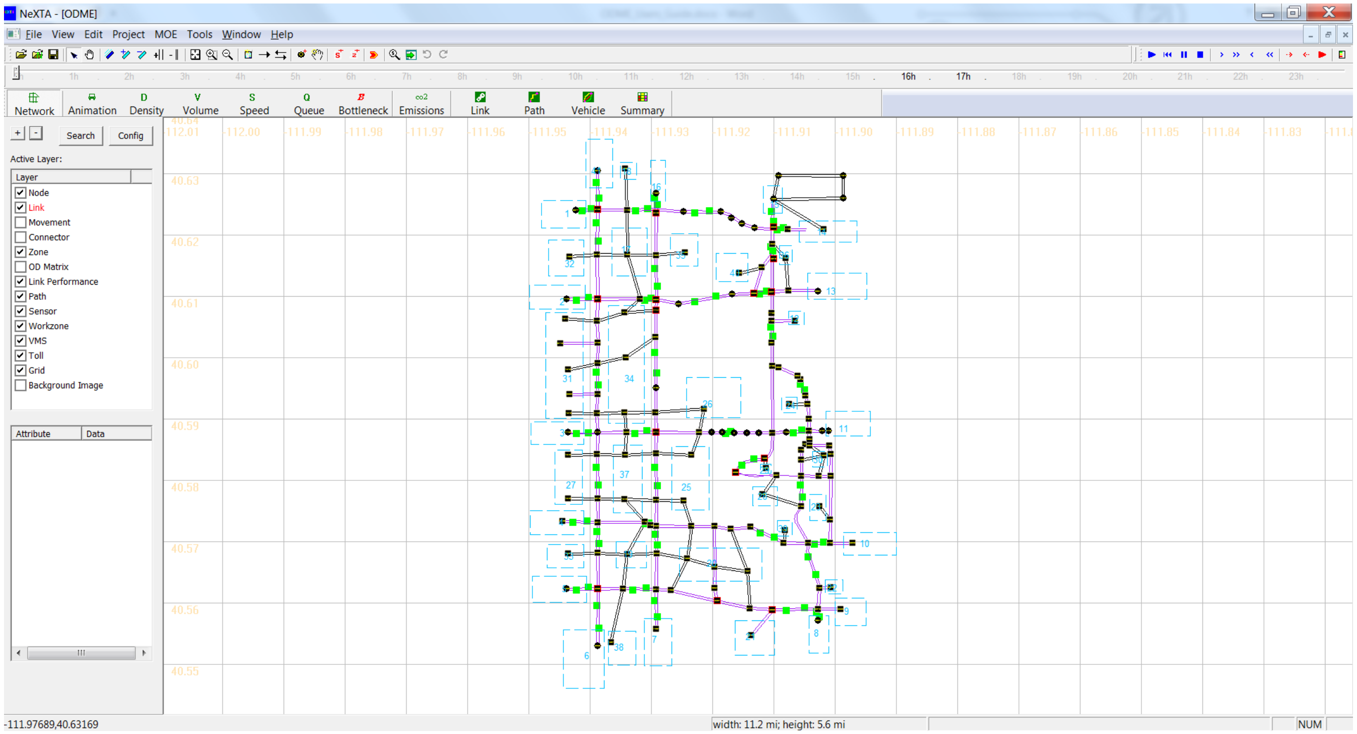


Figure 1. West Jordan traffic network in NeXTA

(2) Parameters settings and ODME running

The required parameter settings are defined in file input_scenario_settings.csv for varying traffic analysis purposes. Table 4 lists those important data fields for ODME and gives corresponding values and explanations for the West Jordan traffic model.

Table 4. Related attributes in file input_scenario_settings.csv for ODME

Data Field	**Value**	**Notes**
number_of_iterations	50	The total number of iterations for ODME
traffic_flow_model	1	This parameter defines a specific traffic flow model used in both assignment and ODME of DTALite; 1 indicates a point queue model in this example. The selection of Newell's KW model is also feasible.
signal_representation_model	0	This parameter defines a specific signal control for DTALite.
traffic_assignment_method	3	This assignment method of "3" is dedicated to ODME
ODME_start_iteration	20	It defines the first iterative assignment period to converge to the user equilibrium state, and could generate a sufficient number of paths for path flow adjustment. The iteration number also indicate that ODME will begin at the 21th iteration.
ODME_end_iteration	50	It defines that ODME will end at the 50the itration.
ODME_max_percentage_deviation_wrt_hist_demand	40	The maximum percentage of demand deviation from base-line dynamic demand.
ODME_step_size	0.05	Moving size of each step in path flow adjustment algorithm
calibration_data_start_time_in_min	990	This and the following parameter specify the time window for ODME to use the sensor data. Note that, users can prepare a long period of sensor data, say from 0 to 24 hours, but only use part of sensor data, say between min 990 and 1050, for calibration.
calibration_data_end_time_in_min	1050	

Remark: in the DTASettings.txt file, you might see the following default settings:
[estimation]
number_of_iterations_per_sequential_adjustment=10
time_period_in_min_per_sequential_adjustment=60

The time_period_in_min_per_sequential_adjustment defines the time period of each sequential adjustment in the algorithm as 60 min (1 hour). The number_of_iterations_per_sequential_adjustment = 10 means that DTALite takes 10 iterations to adjust its demand during the time period of one sequential adjustment above (60 min or 1 hour). In this experiment, the calibrated demand period is 1 hour (990 min to 1050 min), so it requires 10 routing-simulation iterations to complete one global iteration for this demand period. To reach a reasonable converge, at least 3-10 global iterations are usually needed, so at least 30 iterations are required for the path flow adjustment for ODME. Meanwhile, with the starting 20 iterations for reaching the user equilibrium state, a total of 50 iterations will be run for this experiment.

In addition, in order to improve the computational efficiency, users can change the value of the max_number_of_threads_to_be_used in DTASettings.txt as what you want based on your computer configuration to perform parallel computing in DTALite. After ensuring that all data preparation and parameter settings are ready, you can run DTALite directly or call it through the interface of NeXTA.

(3) Result analysis

Though checking output_summary.csv file, users can better understand the process of ODME in DTALite. For the first 20 iterations, a standard dynamic user equilibrium method, MSA, is used. It is expected to see the UE gap (Avg User Equilibirum (UE) gap (min) and Relative UE gap (%)) dramatically decreases and finally reach a stable state, which is shown in Figure 2. In addition, Figure 3 gives the general statistics about different traffic measures for the first 20 iterations

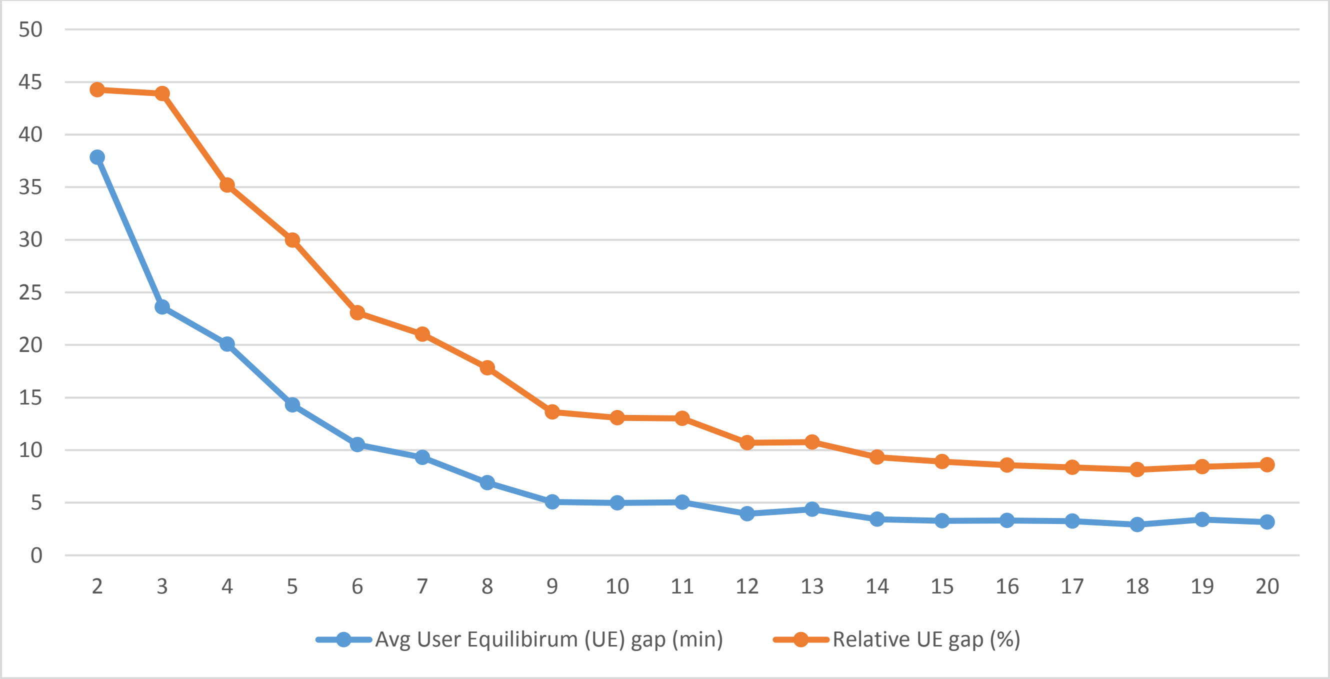


Figure 2. The trend of average UE gap and relative UE gap

Iteration #	CPU Runni	Per Iterati	# of agent	Avg Travel	Avg Waitin	Avg Trip Ti	Avg Distan	Avg Speed	Avg CO (g)	% conside	% switche	% complet	network c	Avg User E	Relative U
1	0:00:03	0:00:03	25282	46.4009	4.36465	9.49976	3.00038	3.87972	25029.7	100	100	100	1350		
2	0:00:06	0:00:03	25282	32.6083	4.36479	6.16369	3.19212	5.87357	19596.1	49.9802	49.9802	100	1350	37.8381	44.2589
3	0:00:10	0:00:04	25282	28.9277	4.36486	5.18874	3.32681	6.90025	18302.9	33.3241	33.3241	100	1350	23.6099	43.8932
4	0:00:12	0:00:02	25282	24.3598	4.36466	4.43344	3.32248	8.1835	16403.7	25.002	25.002	100	1350	20.0816	35.1912
5	0:00:15	0:00:03	25282	24.293	4.36485	4.40716	3.32741	8.21818	16390.7	20.0222	20.0222	100	1350	14.2872	29.9728
6	0:00:17	0:00:02	25282	23.4999	4.36493	4.1995	3.36904	8.60187	16138.9	16.6601	16.6601	100	1350	10.53	23.0463
7	0:00:20	0:00:03	25282	22.0389	4.3649	3.994	3.3556	9.13548	15502.4	14.2948	14.2948	100	1350	9.28808	21.0229
8	0:00:22	0:00:02	25282	21.83	4.36486	3.97921	3.33931	9.17815	15395.2	12.503	12.503	100	1350	6.90259	17.8299
9	0:00:27	0:00:05	25282	21.5712	4.36498	3.95676	3.32313	9.24322	15267.1	11.1502	11.1502	100	1350	5.08769	13.6182
10	0:00:29	0:00:02	25282	21.139	4.36493	3.91005	3.3099	9.39468	15062.2	10.0032	10.0032	100	1350	4.99075	13.0749
11	0:00:32	0:00:03	25282	20.8673	4.36496	3.88687	3.29704	9.48002	14926.6	9.09343	9.09343	100	1350	5.03178	13.0245
12	0:00:34	0:00:02	25282	20.8262	4.36493	3.89094	3.28784	9.47221	14897.7	8.3419	8.3419	100	1350	3.96059	10.7041
13	0:00:37	0:00:03	25282	20.7152	4.3649	3.88885	3.28373	9.51108	14834.1	7.67344	7.67344	100	1350	4.37081	10.7595
14	0:00:40	0:00:03	25282	20.7374	4.36481	3.90482	3.27344	9.47113	14830.1	7.15133	7.15133	100	1350	3.42169	9.3322
15	0:00:42	0:00:02	25282	20.6433	4.36483	3.89946	3.2652	9.49033	14780.3	6.67669	6.67669	100	1350	3.28204	8.89414
16	0:00:45	0:00:03	25282	20.4297	4.36494	3.87381	3.25802	9.56847	14681.3	6.22182	6.22182	100	1350	3.29317	8.55776
17	0:00:47	0:00:02	25282	20.3347	4.36498	3.86672	3.25358	9.60007	14632.7	5.88165	5.88165	100	1350	3.24476	8.35576
18	0:00:52	0:00:05	25282	20.289	4.36494	3.86679	3.24799	9.60515	14605.4	5.56127	5.56127	100	1350	2.91564	8.13619
19	0:00:59	0:00:07	25282	20.1827	4.36496	3.85745	3.24218	9.63851	14552.6	5.26857	5.26857	100	1350	3.39105	8.42188
20	0:01:06	0:00:07	25282	20.1055	4.36489	3.85335	3.23718	9.6606	14511.3	5.00356	5.00356	100	1350	3.17083	8.59123

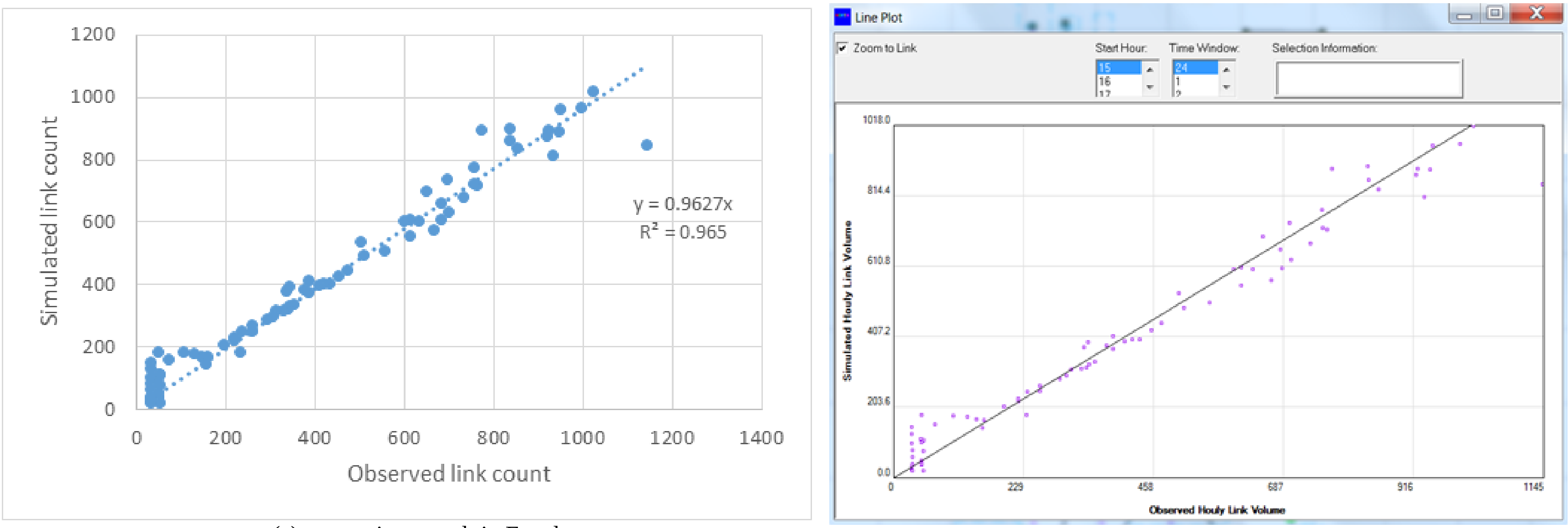
Figure 3. The statistics of different traffic measurements for the first 20 iterations

In the following 30 iterations, users can check the R_squared values from the iterative adjustment process, which should show an increasing pattern toward a reasonable statistics of 0.7, 0.8, or 0.9 in output_summary.csv. The other measures related to ODME include the link count estimation absolute and percentage errors. The summary result of the following 30 iterations is shown in Figure 4, where the R_squared values are marked in the red rectangle.

21	0:01:11	0:00:05	17010	6.50203	0	1.20774	2.76409	25.5067	6232.65	0	0	100	1350	3.52555	0	78	133.545	125.328	0.34322	0.37856
22	0:01:16	0:00:05	21659	14.4455	3.06763	3.08206	2.95332	12.2668	11119.2	0	0	100	1350	0.18716	0	78	98.0192	96.0266	0.68721	0.58957
23	0:01:21	0:00:05	19291	8.07208	0.45665	1.58006	2.78878	20.729	7120.12	0	0	100	1350	1.49182	0	78	80.062	82.1507	0.64038	0.73476
24	0:01:27	0:00:06	21963	11.642	2.48209	2.58609	2.83621	14.6171	9537.29	0	0	100	1350	0.30803	0	78	71.6667	78.5259	0.77311	0.7861
25	0:01:32	0:00:05	20948	9.17211	1.23052	1.94097	2.7754	18.1555	7891.01	0	0	100	1350	0.57174	0	78	60.1966	67.711	0.7885	0.85985
26	0:01:38	0:00:06	22357	10.8444	2.52397	2.46838	2.81553	15.5777	9186.72	0	0	100	1350	0.42242	0	78	57.5171	68.2546	0.79747	0.88253
27	0:01:44	0:00:06	21706	10.1887	1.52853	2.19436	2.77464	16.3396	8436.2	0	0	100	1350	0.43764	0	78	50.1239	62.0388	0.84603	0.90465
28	0:01:50	0:00:06	21930	9.57042	1.9526	2.18755	2.74057	17.1815	8306.9	0	0	100	1350	0.34603	0	78	48.6624	59.5303	0.83237	0.92082
29	0:01:56	0:00:06	22172	9.81169	2.55412	2.35793	2.73379	16.7175	8649.83	0	0	100	1350	0.29671	0	78	43.6944	57.0831	0.86821	0.92156
30	0:02:03	0:00:07	22420	10.3105	1.85105	2.31259	2.73158	15.8958	8572.43	0	0	100	1350	0.32408	0	78	40.1774	52.3117	0.87664	0.93724
31	0:02:09	0:00:06	22063	9.70522	2.30154	2.31501	2.69853	16.683	8459.84	0	0	100	1350	0.31273	0	78	38.4808	49.6109	0.87088	0.93976
32	0:02:15	0:00:06	22522	10.598	2.00165	2.40664	2.71655	15.3796	8743.31	0	0	100	1350	0.21752	0	78	34.6816	46.2085	0.89558	0.95001
33	0:02:21	0:00:06	21883	9.28344	2.23503	2.23901	2.68181	17.3329	8225.06	0	0	100	1350	0.31232	0	78	34.4081	44.305	0.88027	0.95212
34	0:02:27	0:00:06	22738	10.5047	2.21046	2.44983	2.69738	15.4067	8763.12	0	0	100	1350	0.20315	0	78	31.4637	41.0526	0.90981	0.95907
35	0:02:34	0:00:07	22097	9.64962	2.26388	2.32208	2.67122	16.6093	8384.01	0	0	100	1350	0.25626	0	78	32.2756	40.1318	0.89939	0.95542
36	0:02:40	0:00:06	22610	10.1841	2.28271	2.42108	2.68049	15.7922	8631.06	0	0	100	1350	0.19969	0	78	30.3568	37.5642	0.91043	0.96214
37	0:02:47	0:00:07	22510	10.3201	2.66667	2.52272	2.68291	15.5982	8850.98	0	0	100	1350	0.22691	0	78	30.9124	38.7419	0.90943	0.95887
38	0:02:53	0:00:06	22674	10.4931	2.38299	2.51006	2.66751	15.2529	8792.28	0	0	100	1350	0.18223	0	78	28.2842	35.5185	0.91884	0.96566
39	0:03:00	0:00:07	22360	9.62653	2.3591	2.35104	2.65534	16.5501	8393.35	0	0	100	1350	0.21915	0	78	29.2885	38.1582	0.90262	0.964
40	0:03:06	0:00:06	22818	10.4377	2.48416	2.52336	2.66255	15.3054	8805.87	0	0	100	1350	0.17825	0	78	27.7137	35.347	0.91756	0.96861
41	0:03:12	0:00:06	22647	10.1195	2.3669	2.4417	2.66253	15.7866	8616.27	0	0	100	1350	0.21673	0	78	27.4615	36.4246	0.91299	0.966
42	0:03:18	0:00:06	22631	10.0093	2.48874	2.45156	2.65126	15.8928	8611.02	0	0	100	1350	0.19539	0	78	25.5897	32.7038	0.91819	0.96934
43	0:03:25	0:00:07	22874	10.3002	2.71117	2.54419	2.66482	15.5229	8839.33	0	0	100	1350	0.19751	0	78	27.7885	35.9076	0.91616	0.96597
44	0:03:31	0:00:06	22848	10.4554	2.57168	2.56026	2.65179	15.2177	8829.35	0	0	100	1350	0.20329	0	78	25.344	33.195	0.92116	0.96921
45	0:03:37	0:00:06	22540	9.98122	2.61195	2.46698	2.65651	15.969	8655.69	0	0	100	1350	0.19387	0	78	27.4188	35.2209	0.89742	0.96857
46	0:03:43	0:00:06	23105	10.6808	2.59008	2.59674	2.65928	14.9387	8947.38	0	0	100	1350	0.16706	0	78	25.938	35.7663	0.91765	0.97041
47	0:03:49	0:00:06	22690	10.6968	2.54469	2.59303	2.65593	14.8975	8932.31	0	0	100	1350	0.19135	0	78	26.9765	34.814	0.90868	0.96904
48	0:03:56	0:00:07	23109	10.5509	2.64003	2.58911	2.65057	15.0731	8903.27	0	0	100	1350	0.18079	0	78	24.8462	33.565	0.91635	0.9717
49	0:04:02	0:00:06	22693	10.5336	2.56554	2.58654	2.64169	15.0472	8844.35	0	0	100	1350	0.18096	0	78	26.6517	34.3345	0.922	0.96428
50	0:04:09	0:00:07	22891	10.5918	2.68017	2.61181	2.64202	14.9664	8929	0	0	100	1350	0.15897	0	78	25.1538	32.9018	0.90595	0.97112

Figure 4. The summary result of the following 30 iterations

The comparison between simulation link measurements and observed link measurements is available in file debug_validation_results.csv. Figure 5 (a) illustrates the comparison result in link counts. Users can also check the comparison result in NeXTA through clicking “Tools”→ “Sensor Data Management” → “View Validation Plot for Link Count”, shown in Figure 5(b).



(a) comparison result in Excel

(b) comparison result in NeXTA

Figure 5. Comparison result between observed link count and simulated link count after ODME

2. Scenario analysis using estimated OD demand in one simulation

For performing scenario analysis, such as, work zone, incident, ramp metering, etc., the travel demand should be calibrated in advance. In this section, the integration of the two parts above is realized in DTALite through just one simulation, the process of which can be illustrated in Figure 6.

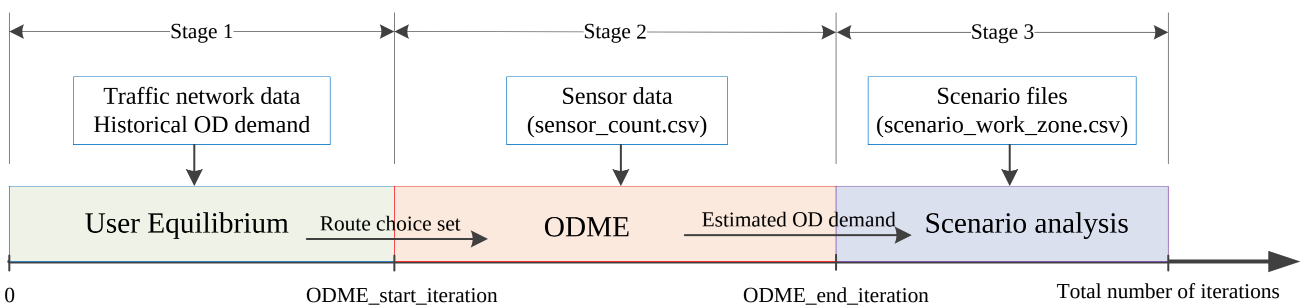


Figure 6. The process of one simulation for scenario analysis using estimated OD demand

At stage 1, the user equilibrium is reached on the basis of traffic network data and historical OD demand and the route choice set is generated for the path flow adjustment at stage 2, where observed sensor data is input and used for OD demand estimation. Based on the estimated OD demand, the scenario analysis is performed for traffic state prediction or new user equilibrium condition searching.

(1) Evaluate specific scenarios using estimated OD demand

The experiment is still the West Jordan traffic network and its data set can be download at here. In this case, it is assumed that the work zone occurs on links 5112→5589 and 5589→5114. The detailed data are input in file Scenario_Work_Zone.csv shown in Table 5.

Table 5. Input data of Scenario_Work_Zone.csv

Link	Scenario No	Start Day	End Day	Start Time in Min	End Time in min	Capacity Reduction Percentage (%)	Speed Limit
[5112,5589]	0	51	51	0	1440	50	50
[5589,5114]	0	51	51	0	1440	50	50

As listed in Table 5, both "Start Day" and "End Day" have same day value of 51, which indicates that the work zone will happen at the 51th iteration and the first 50 iterations is the process of ODME in Section 1. As a result, the vaule of "number_of_assignment_days" should be set as 51 in file input_scenario_settings.csv.

The simulation result of the 51th iteration in output_summary.csv is shown in Figure 7. It is observed that the average speed decreases and the average travel time increases due to the link capacity and speed limit reduction in Scenario_Work_Zone.csv.

Iteration #	CPU Runni	Per Iterati	# of agent:	Avg Travel	Avg Waitin	Avg Trip Ti	Avg Distan	Avg Speed
40	0:03:58	0:00:05	22818	10.4377	2.48416	2.52336	2.66255	15.3054
41	0:04:03	0:00:05	22647	10.1195	2.3669	2.4417	2.66253	15.7866
42	0:04:08	0:00:05	22631	10.0093	2.48874	2.45156	2.65126	15.8928
43	0:04:14	0:00:06	22874	10.3002	2.71117	2.54419	2.66482	15.5229
44	0:04:19	0:00:05	22848	10.4554	2.57168	2.56026	2.65179	15.2177
45	0:04:23	0:00:04	22540	9.98122	2.61195	2.46698	2.65651	15.969
46	0:04:28	0:00:05	23105	10.6808	2.59008	2.59674	2.65928	14.9387
47	0:04:33	0:00:05	22690	10.6968	2.54469	2.59303	2.65593	14.8975
48	0:04:37	0:00:04	23109	10.5509	2.64003	2.58911	2.65057	15.0731
49	0:04:42	0:00:05	22693	10.5336	2.56554	2.58654	2.64169	15.0472
50	0:04:47	0:00:05	22891	10.5918	2.68017	2.61181	2.64202	14.9664
51	0:04:53	0:00:06	22891	13.4408	2.68009	3.17244	2.64202	11.794

Figure 7. Summary of the 51th iteration for work zone scenario

(2) Find the user equilibrium state for specific scenarios

DTALite can also continue to run to obtain the user equilibrium condition for specific scenarios through just one simulation. We still choose the West Jordan traffic network as our test case, which can be downloaded at here. The updated file Scenario_Work_Zone.csv is listed in Table 6.

Table 6 Updated input in Scenario_Work_Zone.csv

Link	Scenario No	Start Day	End Day	Start Time in Min	End Time in min	Capacity Reduction Percentage (%)	Speed Limit
[5112,5589]	0	51	70	0	1440	50	50
[5589,5114]	0	51	70	0	1440	50	50

As shown in Table 6, “Start Day” and “End Day” are set as 51 and 70, respectively, which means that DTALite will run 20 more iterations for reaching the user equilibrium condition under work zone scenario after the first 50 iterations for ODME. Therefore, the vaule of “number_of_assignment_days” will be set as 70 in file input_scenario_settings.csv. The general summary result of the last 20 iterations is shown in Figure 8, and one stable user equilibrium condition is observed in this case.

Iteration #	CPU Runni	Per Iterati	# of agent:	Avg Travel	Avg Waitin	Avg Trip Ti	Avg Distan	Avg Speed
51	0:04:21	0:00:03	22891	13.4408	2.68009	3.17244	2.64202	11.794
52	0:04:24	0:00:03	22891	13.4408	2.68009	3.17244	2.64202	11.794
53	0:04:26	0:00:02	22891	12.3731	2.6801	2.9606	2.64429	12.8227
54	0:04:28	0:00:02	22891	11.4698	2.68016	2.78215	2.6447	13.8347
55	0:04:31	0:00:03	22891	10.8707	2.68015	2.66469	2.64397	14.5931
56	0:04:33	0:00:02	22891	10.2261	2.68012	2.54077	2.64173	15.4999
57	0:04:36	0:00:03	22891	9.75895	2.68011	2.45227	2.63857	16.2225
58	0:04:38	0:00:02	22891	9.4334	2.68012	2.39166	2.63454	16.7566
59	0:04:41	0:00:03	22891	9.09226	2.68008	2.32702	2.63164	17.3662
60	0:04:43	0:00:02	22891	8.83613	2.6801	2.28032	2.6278	17.8435
61	0:04:46	0:00:03	22891	8.68021	2.68008	2.25209	2.62498	18.1446
62	0:04:48	0:00:02	22891	8.60879	2.68013	2.24193	2.62135	18.2698
63	0:04:51	0:00:03	22891	8.5458	2.6801	2.23266	2.61821	18.3824
64	0:04:53	0:00:02	22891	8.44776	2.68011	2.21583	2.61397	18.5657
65	0:04:56	0:00:03	22891	8.3939	2.68008	2.20947	2.6101	18.6571
66	0:04:58	0:00:02	22891	8.35054	2.6801	2.20423	2.60686	18.7307
67	0:05:01	0:00:03	22891	8.31091	2.68006	2.20066	2.603	18.7922
68	0:05:03	0:00:02	22891	8.28268	2.68006	2.1989	2.59991	18.8338
69	0:05:06	0:00:03	22891	8.25539	2.68013	2.19735	2.59785	18.8811
70	0:05:08	0:00:02	22891	8.22562	2.68014	2.19473	2.59526	18.9306

Figure 8. Summary of the last 20 iterations for work zone scenario

3. Methodology

3.1 Introduction

ODME implemented in DTALite is based on a single-level nonlinear optimization model proposed by Lu, Zhou and Zhang (2013). The general introduction on OD demand estimation problem can be found at here.

Our model has the following key features:

- The model is a path flow-based optimization model, which incorporates heterogeneous sources of traffic measurements and does not require explicit dynamic link-path incidences.
- The objective is to minimize (i) the deviation between observed and estimated traffic states and (ii) the deviation between aggregated path flows and target OD flows, subject to the dynamic user equilibrium (DUE) constraint represented by a gap-function-based reformulation.
- A Lagrangian relaxation-based algorithm which dualizes the difficult DUE constraint to the objective function is proposed to solve the model.
- This algorithm integrates a gradient-projection-based path flow adjustment method within a column generation-based framework.
- DTALite, a dynamic network loading (DNL) model which is based on Newell's simplified kinematic wave theory, is employed in the DUE assignment process to realistically capture congestion phenomena and shock wave propagation.
- This optimization also derives analytical gradient formulas for the changes in link flow and density due to the unit change of time-dependent path inflow in a general network under congestion conditions.

Reference: Lu C-C, Zhou*, X. Zhang, K. (2013) Dynamic Origin-Destination Demand Flow Estimation under Congested Traffic Conditions. Transportation Research Part C. 34, 16-37. (A short version of the paper)

2.2 Mathematical model and solution algorithm

Given sensor data (i.e. observed link flows and densities) and target (aggregated historical) OD demands, the proposed single-level time-dependent path flow estimation model is a nonlinear program with the path flows $r(w,\tau,p), \forall w,\tau,p$ and least path travel times $\pi = \{\pi(w,\tau), w, \tau\}$ as the decision variables. Denote $c = \{c(w,\tau,p), \forall w,\tau,p\}$, $q = \{q(l,t), \forall l,t\}$ and $k = \{k(l,t), \forall l,t\}$. The objective function, Eq.(1), minimizes the weighted sum of the deviation between estimated time-dependent OD demands (or aggregated path flows) and target demands and the deviation between estimated and observed link flows and densities, where β_d, β_q and β_k are the weights reflecting different degrees of confidence on target OD demands and observed link flows and densities, respectively.

P1: Nonlinear program

$$Min\ Z = \beta_d \sum_w \left[\sum_{\tau \in H_d} \sum_p r(w,\tau,p) - \bar{d}(w)\right]^2 + \sum_{l \in S} \sum_{t \in H_0} \{\beta_q [q(l,t) - \bar{q}(l,t)]^2 + \beta_q \left[k(l,t) - \bar{k}(l,t)\right]^2\} \quad (1)$$

Subject to

$$(c,q,k) = DNLF(r) \quad (2)$$

$$g(r,\pi) = \sum_w \sum_\tau \sum_p \{r(w,\tau,p)[c(w,\tau,p) - \pi(w,\tau,p)]\} = 0 \quad (3)$$

$$c(w,\tau,p) - \pi(w,\tau) \geq 0, \forall w,\tau,p \quad (4)$$

$$\pi(w,\tau) \geq 0, \forall p \in P(w,\tau), \forall w,\tau \quad (5)$$

$$r(w,\tau,p) \geq 0,, \forall w,\tau,p \quad (6)$$

where

A: set of links

W: set of OD pairs

P: set of paths

S: set of links with sensors, $S \subseteq A$

H_d: set of discretized departure time intervals

H_o: set of discretized observation time intervals

Index:

t: index of simulation time intervals, $t = 0, \ldots, T$. This paper refers to any particular time interval t as the time t.

τ: index of departure time intervals, $\tau \in H_d$

w: index of OD pairs, $w \in W$

p: index of paths for each OD pair, $p \in P$

l: index of links, $l \in A$

Traffic measurements inputs

$\bar{q}(l,t)$: observed number of vehicles passing through an upstream detector on link l during observation interval t

$\bar{k}(l,t)$: observed density on link l during observation interval t

$\bar{d}(w)$: target demand, which is the total traffic demand for OD pair w over a planning horizon

Estimation variables

$r(w,\tau,p)$: estimated path flow on path p of OD pair w and departure time interval τ

$c(w,\tau,p)$: estimated path travel time on path p of OD pair w and departure time interval τ

$\pi(w,\tau)$: estimated least path travel time of OD pair w and departure time interval τ

$q(l,t)$: estimated number of vehicles passing through an upstream detector on link l during observation interval t

$k(l,t)$: estimated density on link l during observation interval t

$d(w,\tau)$: estimated demand of OD pair w and departure time interval τ

Solution algorithm

This section describes the Lagrangian relaxation-based heuristic for solving the single-level time-dependent path flow estimation model. We propose the following heuristic solution method to efficiently obtain good solutions for problem instances on road networks of practical sizes. The heuristic integrates Lagrangian relaxation and column generation methods to solve the time-dependent path flow estimation model, P1. The gap function constraint Eq.(3) is relaxed to the objective function Eq.(1) with a non-negative Lagrange multiplier l. The resulting Lagrangian subproblem is given as follows.

$$P_2: Min_{r,\pi} L(r,\pi,\lambda) = Z + \lambda\{g(r,\pi)\} \quad (7)$$

Subject to constraints (2), (4), (5) and (6),

where r and π are the vectors of path flows and least path times respectively. For a given l, the solution to P_2 provides a lower bound to P_1. The Lagrangian dual problem is given as follows.

$$P_3: Max_{\lambda} Min_{r,\pi} L(r,\pi,\lambda)$$

$$\text{Subject to } \lambda \geq 0 \quad (8)$$

The heuristic consists of two major algorithmic steps: at each iteration n, (i) given a Lagrange multiplier $\lambda(n)$, find an optimal path assignment $r(n)$ and least path travel times $\pi(n)$ by solving the Lagrangian subproblem, **P2**, and (ii) given a vehicle path assignment $r(n)$ and least path travel times $\pi(n)$, update the Lagrange multiplier $\lambda(n+1)$by using the following rule.

$$\lambda(n+1) = \max\{0, \lambda(n) + \alpha(n)\{\sum_w \sum_\tau \sum_p r(w,\tau,p)[c(w,\tau,p) - \pi(w,\tau)]\}\} \quad (9)$$

where $\alpha(n)$ is the step size for updating the Lagrange multiplier.

Accordingly, this heuristic has two loops. The outer loop is for updating the Lagrange multiplier using the rule described in Eq.(9). For each outer loop iteration n (i.e., corresponding to a given Lagrange multiplier $\lambda(n)$), a column generation-based approach is used to solve the Lagrangian subproblem P_2. This approach forms an inner loop for solving a DUE assignment problem under a restricted feasible solution space. In each inner loop iteration m, a time-dependent shortest path algorithm (Ziliaskopoulos and Mahmassani, 1993) is adopted to generate time-dependent least time paths and to augment the restricted path set. In light of the time-dependent shortest path algorithm, the least path travel times $\pi(m)$ are obtained to satisfy the constraints Eq. (4) and (5), thus, these definitional constraints of the least travel times can be dropped in solving the restricted Lagrangian subproblem. To solve the restricted subproblem, a gradient-projection-based descent direction method (Lu et al., 2009) is used to update path flows $r(m+1)$, while maintaining the feasibility of non-negativity constraints Eq.(6). Specifically,

$$r(w,\tau,p)^{m+1} = Max\{0, r(w,\tau,p)^m - \gamma^m[\beta_d \nabla h^d(r)|_{r=r^m} + \beta_q \nabla h^q(r)|_{r=r^m} + \beta_k \nabla h^k(r)|_{r=r^m} + \lambda(n)\nabla g(r,\pi)|_{r=r^m}]\} \quad (10)$$

where γ^m is the step size, and the gradients, which consist of the first-order partial derivatives with respect to a path flow variable $r(w,\tau,p)$, can be derived as follows.

$$\nabla h^d(r) = \frac{\partial\{\sum_{\tau\in H_d}\sum_{p\in P}(r_{w,\tau,p}) - \bar{d}_w\}}{\partial r(w,\tau,p)} = 2(\sum_{\tau\in H_d}\sum_{p\in P} r(w,\tau,p) - \bar{d}_w) \quad (11)$$

$$\nabla h^q(r) = \frac{\partial\{\sum_{l\in S}\sum_{t\in H_o}[q_{l,t}(r) - \bar{q}(l,t)\}}{\partial r(w,\tau,p)} = 2(\sum_{l\in S}\sum_{t\in H_o}[q_{l,t}(r) - \bar{q}(l,t)] \times \frac{\partial q(l,t)(r)}{\partial r(w,\tau,p)}) \quad (12)$$

$$\nabla h^k(r) = \frac{\partial\{\sum_{l\in S}\sum_{t\in H_o}[k_{l,t}(r) - \bar{k}(l,t)\}}{\partial r(w,\tau,p)} = 2(\sum_{l\in S}\sum_{t\in H_o}[k_{l,t}(r) - \bar{k}(l,t)] \times \frac{\partial k(l,t)(r)}{\partial r(w,\tau,p)}) \quad (13)$$

$$\nabla g(r,\pi) = \frac{\partial g(r,\pi)}{\partial r(w,\tau,p)} = c(w,\tau,p) - \pi(w,\tau) + r(w,\tau,p)\frac{\partial c(w,\tau,p)}{\partial r(w,\tau,p)} \quad (14)$$

Estimated link flows, densities, and link/path travel times and the corresponding partial derivatives, namely $\nabla h^q(r)$, $\nabla h^k(r)$ and $\frac{\partial c(w,\tau,p)}{\partial r(w,\tau,p)}$ are obtained from the DNL model presented in Section 2.

The steps of this algorithm are presented as follows in Figure 9.

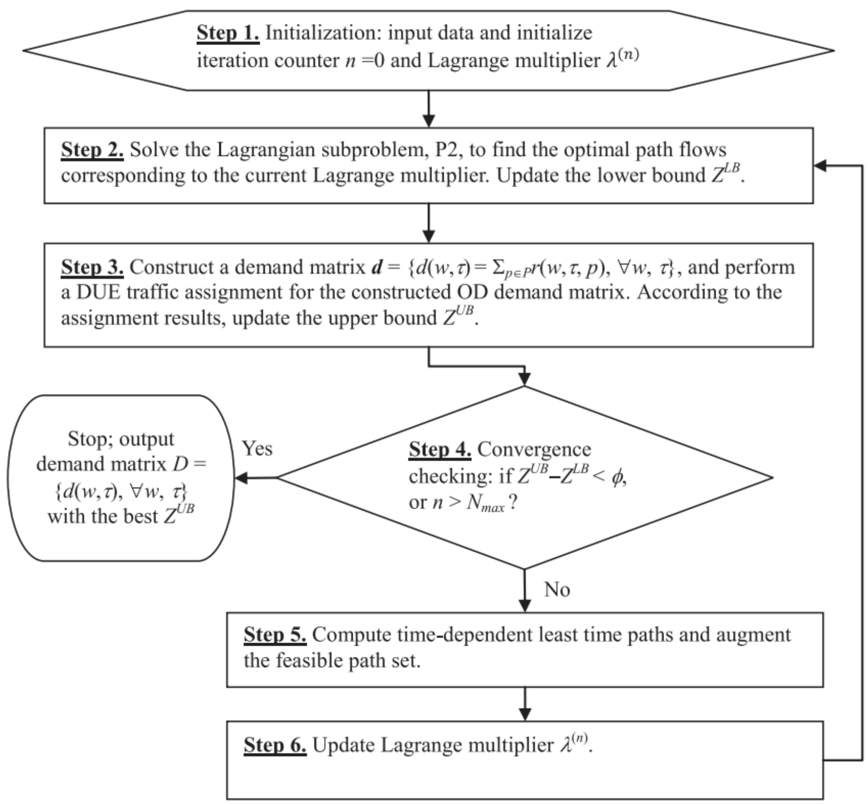


Figure 9. The procedure of proposed algorithm

Typically, a Lagrangian solution framework requires obtaining exact solutions to relaxed subproblems. It should be remarked that, analyzing existence and uniqueness of solutions to the DUE problem for multiple OD pairs are very challenging, and the gradient-based algorithm through Eqs. (10)-(14) cannot guarantee that the relaxed (nonlinear) problem P2 is solved to its optimality. Thus, when no global optimum solution is available for P2, the proposed overall Lagrangian solution algorithm is still a heuristic method in nature.

Solving the proposed single-level dynamic OD estimation model requires the evaluation of the partial derivatives with respect to time-varying path flows, i.e., $\frac{\partial q(l,t)(r)}{\partial r(w,\tau,p)}$, $\frac{\partial k(l,t)(r)}{\partial r(w,\tau,p)}$ and $\frac{\partial c(w,\tau,p)}{\partial r(w,\tau,p)}$. These partial derivatives represent the marginal effects of an additional unit of path inflow on link flow and density and path travel time. This section delineates the evaluation of these partial derivatives due to path flow perturbation in a congested network, based on cumulative link inflow and outflow curves. The following notation is used throughout this section.

L: the number of links on the path

l: link index $l = 1, 2, \ldots L$

t_l': the time when an additional unit of perturbation flow arrives at link l

t_l'': the time when an additional unit of perturbation flow departs at link l

t_l^{qs}: the time when the queue starts to form on link l

t_l^B: the time when the queue vanishes on link l

$t_l^A := t_l^B - FFTT(l)$

t_l^{q*}: the time when the queue on link l starts to spillback to its upstream link $l-1$

n^A: cumulative arrivals at time t_l^A

n': cumulative arrivals at time t_l'

The following propositions can be directly induced from Figure 10 for deriving the marginal effects on link flow (inflow and outflow) and density.

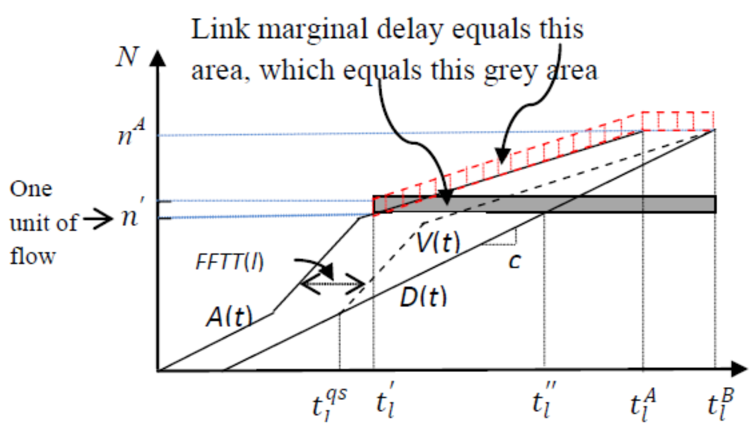


Figure 10. Impact of adding a unit of flow on a congested link

Proposition 1: Under *free-flow* conditions, an extra unit of flow arriving at the upstream end of link l at time t_l': results in the following: (i) the link inflow and outflow increase by 1 at times t_l' and t_l'', respectively, and the flow rates at other time intervals do not change; (ii) the link density increases by 1 from t_l' to t_l''; (iii) the individual travel times are not changed, and $t_l'' = t_l' + FFTT(l)$

Proposition 2: Under *partially* congested conditions and constant link (outflow) capacity c, an extra unit of flow arriving at the upstream end of link l at time $t_l{}'$ results in the following: (i) the link inflow and outflow increase by 1 at times t_l' and t_l^B, respectively, and the flow rates at other time intervals do not change; (ii) the link density increases by 1 from t_l' and t_l^B ; (iii) the flows arriving between t_l' and t_l^A experience the additional delay $1/c$, because it takes $1/c$ to discharge this perturbation flow.