



**HAL**  
open science

# Computational Methods for Calculating Multi-modal Multi-class Traffic Network Equilibrium: Simulation Benchmark on a Large-scale Test Case

Mostafa Ameli, Jean Patrick Lebacque, Ludovic Leclercq

► **To cite this version:**

Mostafa Ameli, Jean Patrick Lebacque, Ludovic Leclercq. Computational Methods for Calculating Multi-modal Multi-class Traffic Network Equilibrium: Simulation Benchmark on a Large-scale Test Case. *Journal of Advanced Transportation*, 2021, 23p. 10.1155/2021/8815653 . hal-03145458

**HAL Id: hal-03145458**

**<https://hal.science/hal-03145458v1>**

Submitted on 18 Feb 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Journal of Advanced Transportation

## Computational Methods for Calculating Multi-modal Multi-class Traffic Network Equilibrium: Simulation Benchmark on a Large-scale Test Case

Mostafa Ameli<sup>1,2</sup>, Jean-Patrick Lebacque<sup>1</sup>, Ludovic Leclercq<sup>2</sup>

<sup>1</sup>Univ. Gustave Eiffel, COSYS, GRETTIA, Paris, France.

<sup>2</sup>Univ. Gustave Eiffel, ENTPE, LICIT, F-69518, Lyon, France.

Correspondence should be addressed to Ludovic Leclercq; ludovic.leclercq@univ-eiffel.fr

### Abstract

This study reviews existing computational methods to calculate simulation-based dynamic network equilibrium. We consider a trip-based multi-modal approach for the dynamic network loading. Mode and path choices are carried out at the same level; therefore, travel times depend on the travel path and the mode attributes of travelers. This study develops a multi-class model with several parameters per class. Two different categories of algorithms (heuristic and meta-heuristic) are considered in order to solve the discrete DTA problem. Finally, we analyze the equilibrium in a large-scale multi-modal DTA test case (Lyon 6th + Villeurbanne) in order to investigate the performance of different optimization approaches to solve trip-based DTA. The results show that, in a multimodal and heterogeneous setting, the meta-heuristic methods provide better solutions than the heuristic methods in terms of optimality and computation time. These improvements are even more significant than in a homogeneous setting.

### keywords

Traffic assignment; User equilibrium; Dynamic network assignment; Multi-class model; Multi-modal Large-scale network; Gap-based; Probabilistic; Genetic algorithm; Simulated annealing

## 1 Introduction

Network user equilibrium is the situation where all users' travel costs in all routes used are equal and lower than those related to the not chosen route [1]. Dynamic traffic assignment (DTA) is a process to assign a given set of trips to a possible path (set of route and modes) following the network user equilibrium by considering the dynamics of the transportation system. In a large-scale network, when we have numerous paths per Origin-Destination (OD) pairs and also a large number of origins and destinations, the problem becomes almost intractable analytically because of multiple flow exchanges at nodes. The simulation-based approach, introduced by [2], is an effective way to address the question of DTA in large-scale networks [3]. Simulation-based DTA models rely on a network performance module called dynamic network loading (DNL). In DNL, the dynamic traffic simulator does the network loading to calculate network states and travel times; then the traffic assignment component determines the route and

29 path flow distribution [4]. This combination, for large-scale networks, allows us to consider sophisticated traffic  
 30 models in order to incorporate disaggregate information into the estimates of travel costs [5].

31 Traffic simulators can be divided into two classes: Flow-based models, which consider the flow of each path and  
 32 Trip-based models, which define how many travelers take each path. Macroscopic traffic flow models fall into the  
 33 first category, while microscopic models belong to the second. The macroscopic approach and flow-based models  
 34 are usually fast in equilibrium calculation as the path flow discipline is more flexible (flows are not necessarily  
 35 equivalent to traveler units). However, they are less realistic for OD pairs without adding integrality constraints with  
 36 low demand as vehicles are split into small particles in practice. In this study, we decide to focus on the trip-based  
 37 approach in which each traveler is reproduced individually. The trip-based approach can be considered as a multi-  
 38 agent system wherein each trip impacts the system by its route (including path and mode(s)) choice. Microscopic  
 39 traffic simulators are now widely used for operational studies, and we have chosen to focus on DTA performance  
 40 for this kind of model. Trip-based DNL attempts to assign particle-discretized time-dependent origin/destination  
 41 flows in a dynamic network equilibrium framework [6]. The goal of this study is to design and solve a large-scale  
 42 trip-based traffic network equilibrium model.

43 The user equilibrium situation, theoretically, is the solution of a fixed-point problem [7]. From a practical point  
 44 of view, it means that each user perceives no delay or extra cost compared to other users with the same origin  
 45 and destination [8]. This refers to minimizing the total gap between users' travel cost of each OD pair and the  
 46 corresponded shortest path (the path(s) with minimum travel cost) travel cost [9]. In the context of DTA, the Method  
 47 of Successive Average (MSA) algorithm remains by far the most widely used solution method in simulation-based  
 48 DTA [10]. The simplicity of the MSA implementation and the non-requirement of derivative information are the  
 49 main reasons for its widespread use [11]. However, [12] showed that the convergence is not guaranteed to be the  
 50 user equilibrium solution with the traditional MSA algorithm. Based on this indication, several heuristic approaches  
 51 are proposed to improve the determination of the step size (e.g., [13–15]). [16] did a benchmark on most of the  
 52 heuristic methods in the literature. Here, in addition to heuristic algorithms, we also considered meta-heuristic  
 53 approaches (proposed by [17]) in order to compare different methods to solve the DTA problem.

54 The travel cost is the criterion for each user to select the travel route and mode(s). It is important to consider  
 55 generalized travel costs in simulation-based DTA in order to include different features of travelers' behavior. Gen-  
 56 erally, travelers consider different criteria in selecting their optimal properties of travel, including travel route and  
 57 travel mode(s). Thus, the total demand of the system can be clustered into different user classes based on the def-  
 58 inition of their travel cost function. Many attempts have been made on modeling travelers' behavior in the traffic  
 59 network. Most multi-class flow models classify travelers by identifying traveler attributes. In the literature, this  
 60 classification process identifies eight categories of information about travelers: (1) Choice model; (2) Cost func-  
 61 tion; (3) Economic attributes; (4) Generalized cost function; (5) Knowledge level of the network; (6) Risk taking  
 62 attitude; (7) Social class; (8) Vehicle class. Moreover, some research uses hybrid classification [see e.g. 18–20].  
 63 In this study, we attempt to define a generic cost function for users in order to represent their behavior accurately.  
 64 For the convenience of the reader, we discuss all categories and identify the different approaches in Appendix A  
 65 (Table 5) in order to clarify our contribution compared to the previously mentioned papers.

66 Multi-class DTA models are widely investigated in order to take into account the heterogeneity of the users [21].  
 67 While the model becomes more accurate for representing the users' travel cost function, finding the equilibrium  
 68 becomes more difficult computationally. There are many studies about developing algorithms to find user equilib-  
 69 rium [8]. This study aims to find out which category of solution algorithms is more efficient for the multi-class  
 70 DTA problem. There are two main categories of solution methods to solve simulation-based DTA with computa-  
 71 tional methods: heuristic methods based on fixed-point theory and meta-heuristic methods that have been recently  
 72 introduced [17]. In the literature, these methods are applied individually to networks with different sizes and dif-  
 73 ferent levels of details about users and network dynamics [22]. In this study, we apply solution methods with the  
 74 multi-class model to a real test case (Lyon 6th + Villeurbanne: about  $17km^2$ ) in order to benchmark all the solution  
 75 methods. The goal is to answer which solution method is the most efficient in generalized and advanced numerical  
 76 simulation settings.

77 In the next section, the multi-class model is presented. The Wardrop condition for the network equilibrium  
 78 model is discussed for trip-based DTA in Section 2. The benchmarked algorithms which find the network equilib-  
 79 rium are explained in Section 3. In Section 4, the numerical experiment and the simulator are presented. Section 5  
 80 presents the numerical results and discusses them based on several indicators. Finally, we conclude this study in  
 81 Section 6.

## 82 2 Dynamic traffic assignment model

### 83 2.1 Multi-attribute generalized travel cost

84 In order to describe the traveler behavior, a generalized cost function that takes into account four types of infor-  
 85 mation mentioned in Section 1: (2), (3), (4) and (7), is used. Typically the generalized cost function ( $GC$ ) will  
 86 integrate Travel Time ( $TT$ ) and Travel Cost ( $TC$ ). Also,  $GC$  will take into account the class-dependent Value of  
 87 time (VOT). [23] denoted  $\alpha_i$  for class  $i$ , and a vector  $\beta_i$  of economic and social class parameters. Therefore, we  
 88 need to define for class  $i$  and path  $p$  ( $p \in P$ : set of paths), the path generalized cost  $GC_p(\alpha_i, \beta_i)$ .

89 The vector of parameters  $\beta_i$  includes the traveler mode(s) and social attributes. For instance, one class in the  
 90 network can be students who pay less to use the transit network and who have the specific value of time. The  
 91 demand of OD pair  $w$  ( $w \in W$ : set of OD pairs) is fixed and:

$$d_w^i = \sum_{p \in P_w} \pi_p^i \quad , \quad (1)$$

$$d_w^i \geq 0 \quad (2)$$

where  $i$  is the user class index ( $i \in 1, 2, \dots, I$ ),  $P_w$  is a set of paths for  $w$  and  $\pi_p^i$  is the number of users of class  $i$  on  
 path  $p$ . After assigning the user to the path between OD pair  $w$ ,  $\pi_a$  (the number of users on link  $a$ ) is:

$$\pi_a^i = \sum_{p \in P} \pi_p^i \delta_{ap} \quad , \quad (3)$$

$$\pi_a = \sum_{i=1}^I \pi_a^i \quad (4)$$

92 where  $a$  is the link index ( $a \in A$ : set of links in the network),  $\pi_a^i$  is the number of users of class  $i$  on link  $a$  and  
 93  $\delta_{ap} = 1$  if path  $p$  contains link  $a$  and 0, otherwise. The travel time and cost functions depend on the congestion of  
 94 the link. Therefore, both functions are calculated by the following formulas:

$$TT_a = TT_a(\pi_a) \quad , \quad (5)$$

$$TC_a^i(\beta_i) = TC_a^i(\beta, \pi_a) \quad (6)$$

Consequently, the link generalized cost function is calculated as follows:

$$GC_a(\alpha_i, \beta_i) = TC_a(\beta, \pi_a) + \alpha_i TT_a(\pi_a) \quad (7)$$

Eq. 7 calculates the generalized cost of each link for each class of users. Finally, the cost of path  $p$  is calculated by the following equation:

$$GC_p(\alpha_i, \beta_i) = \sum_{a \in A} GC_a(\alpha, \beta) \delta_{ap} \quad (8)$$

Now, we can build a DTA model based on the generalized cost defined by Eq. 8.

## 2.2 Equilibrium Conditions

Consider a network  $G(N, A)$  with a finite set of nodes  $N$  and a finite set of directed links  $A$ . The demand is time-dependent. The period of interest (planning horizon) of duration  $H$  is discretized into a set of small time intervals indexed by  $\tau$  ( $\tau \in T = \{\tau_0, \tau_0 + \sigma, \tau_0 + 2\sigma, \dots, \tau_0 + M\sigma\}$  and  $\tau_0 + M\sigma = H$ ).  $\sigma$  is the duration of the time intervals. In an interval  $\tau$ , travel time and traffic conditions do not change. The important notations to introduce the dynamic equilibrium model and optimization algorithms are as follows:

Table 1: List of notations used in this article

<i>Equilibrium model:</i>	
$W$	OD pairs, subset of origin $\times$ destination nodes, $W \subset N \times N$ .
$I$	Set of user classes.
$P_{w, \tau}$	Set of paths for OD pair $w$ in departure time $\tau$ .
$P_{w, \tau}^{i*}$	Set of shortest paths for OD pair $w$ and class $i$ in departure time interval $\tau$ .
$i$	Index of user class, $i \in I$ .
$\tau$	Index of departure time interval, $\tau \in T$ .
$w$	Index of origin-destination (OD) pair, $w \in W$ .
$p$	Index of path, $p \in P_{w, \tau}$ .
$p^{i*}$	Index of shortest path for class $i$ , $p^{i*} \in P_{w, \tau}^{i*}$ .
$\pi_{w, p, \tau}^i$	Number of users in class $i$ from OD pair $w$ , assigned to path $p$ in departure time $\tau$ .
$\pi_{w, p, \tau}^{i*}$	Optimal number of users in class $i$ from OD pair $w$ , assigned to path $p$ in departure time $\tau$ .
$GC_{p, \tau}(\alpha_i, \beta_i)$	Generalized cost of path $p$ for user class $i$ in departure time $\tau$ .
$GC_{w, \tau}^{i*}$	Minimum generalized cost of OD pair $w$ for user class $i$ in departure time $\tau$ .
<i>Optimization algorithms:</i>	
$j$	Outer loop iteration index.
$k$	Inner loop iteration index.
$j_{max}$	Maximum number of outer loop iterations.
$k_{max}$	Maximum number of inner loop iterations.
$\pi_p^{i, k}$	The number of users in class $i$ on path $p$ in iteration $k$ .
$P^{i*}$	The set of shortest path in iteration $k$ for class $i$ .
$z_j$	The path flow distribution of the outer loop iteration $j$ .
$z^k$	The path flow distribution of the inner loop iteration $j$ .
$AGap(s)$	The travel cost gap per user of the path flow distribution $s$ .
$TGap(s)$	The total travel cost gap of the path flow distribution $s$ .
$TGap_w(s)$	The total travel cost gap of the path flow distribution of OD pair $w$ in solution $s$ .
$Gap_\alpha$	The threshold for the exploration rate of the inner loop.
$Gap_\beta$	The threshold for the convergence rate of the outer loop.

For each class  $i$ , for all OD pairs  $w \in W$  and for all paths  $p \in P_w$ , the dynamic traffic network equilibrium condition with fixed travel demands for the aforementioned traffic network equilibrium problem is

$$\begin{cases} GC_{p,\tau}(\alpha_i, \beta_i) - GC_{w,\tau}^* \geq 0 \\ \pi_{p,\tau}^i (GC_{p,\tau}(\alpha_i, \beta_i) - GC_{w,\tau}^*) = 0 \\ \pi_{p,\tau}^i \geq 0 \end{cases} \quad (9)$$

Lu et al. [9] reformulated the problem as a non-linear problem in order to minimize the gap function. The gap function is defined as the gap between average path travel time and the shortest path travel time. Therefore, the solution of this problem is equivalent to finding the solution to the following variational inequality:

$$\sum_{w \in W} \sum_{\tau=1}^T \sum_{i=1}^I \sum_{p \in P_{w,\tau}} GC_{w,p,\tau}^i \left( \pi_{w,p,\tau}^i - \pi_{w,p,\tau}^{i*} \right) \geq 0 \quad \forall \pi_{w,p,\tau}^i, \pi_{w,p,\tau}^{i*} \in \mathcal{C} \quad (10)$$

104 where  $\mathcal{C}$  denotes the flow constraints based on  $d_w^i$ . Note that  $\pi_{w,p,\tau}^{i*}$ , a feasible assignment, is the optimal assign-  
 105 ment, if Eq. 10 holds for all feasible  $\pi_{w,p,\tau}^i$ . In Eq. 10, both  $\pi_{w,p,\tau}^i$  and  $\pi_{w,p,\tau}^{i*}$  are decision variables and hence  
 106 the gap function is a function of both variables. Existence of solution(s) to Eq. 10 requires the continuity of the  
 107 generalized cost functions of paths with respect to the numbers of users on paths. In this case, the unicity of the  
 108 solution requires the strict monotonicity of the OD paths generalized cost functions with respect to OD path flows  
 109 (i.e. the number of cars) [24]. In general urban transportation networks, these conditions do not hold [21].

### 110 3 Solution Algorithm to Solve Trip-based Dynamic Traffic Assignment

111 The development of efficient algorithms for the numerical computation of equilibria is a topic as important as the  
 112 qualitative analysis of equilibria. The complexity of equilibrium problems, coupled with their increasing scale, is  
 113 precluding their resolution via closed-form analytics. Note that we will use a trip-based simulation-based algorithm  
 114 to look for large-scale user equilibrium. As mentioned before, the solution space is then discrete rather than  
 115 continuous. The optimizer (algorithm) is embedded in the optimization framework of the DTA problem. First, we  
 116 present the optimization framework, which comes from the literature. Then we focus on the optimizer part wherein  
 117 the path assignment is updated by heuristic and meta-heuristic methods.

118 Finding an equilibrium through simulation (when no closed-form analytical solution is available) typically  
 119 involves a solution scheme that relies on an iterative procedure. In large-scale application, this iterative procedure  
 120 contains three costly steps: traffic simulation (network loading), time-dependant shortest path calculation and path  
 121 flow optimization. In order to focus on the path flow optimization, we choose column generation approach.

122 Figure 1 presents the optimization framework based on the column generation approach. The framework sep-  
 123 arates the three costly steps in order to minimize the number of shortest path calculation which is very costly in  
 124 large-scale traffic networks. In other words, the idea of the column generation approach is based on the generation  
 125 of the set of paths to which users may be assigned. Classical approach starts from an empty set and augments it  
 126 every iteration as required. The column generation approach contains two loops: outer loop and inner loop. The  
 127 outer loop is responsible for path discovery while the inner loops implement the path flow optimization for a given  
 128 path set. Note that the classic approach executes both steps in a single loop. In large-scale network problems,  
 129 it is extremely costly to keep the data of all the possible paths between each OD pair, so, the column generation  
 130 approach definitely reduces memory-requirements [25]. Lu et al. [9] implement and examine the column genera-  
 131 tion approach in DTA context and show that it not only reduces memory-requirements but also outperforms other  
 132 algorithms in convergence with a designed swapping algorithm embedded in the inner loop.

133 This approach provides us with a framework to compare different path flow optimization algorithms to address  
 134 which of them is more efficient for the multi-class multi-modal DTA problem. Therefore, we are going to focus on  
 135 the green modules in Figure 1. Before presenting the framework in details, we need to define some solution quality  
 136 indicators to measure the distance between each solution and optimal solution (UE path flow distribution).

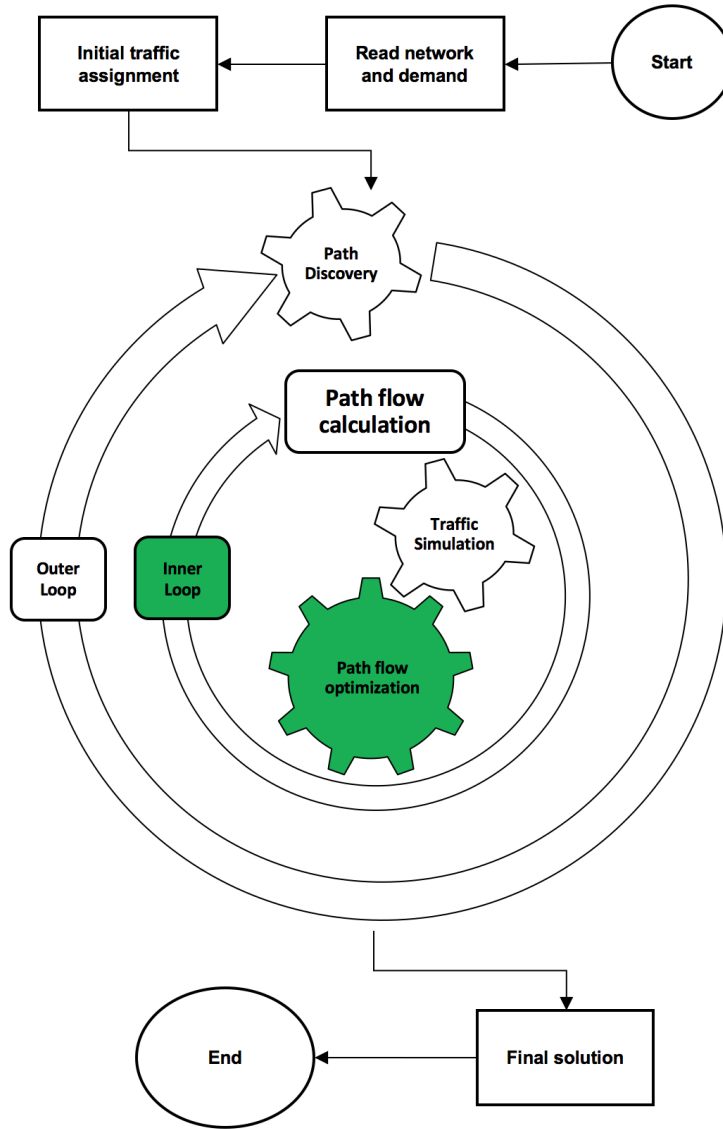


Figure 1: Simulation-based DTA process: Column generation approach

137 The total gap as a quality indicator in time interval  $\tau$  is:

$$TGap(\pi, GC^*) = \sum_{w \in W} \sum_{i=1}^k \sum_{p \in P_{w, \tau}} \pi_{w, p}^i (GC_{w, p}^i - GC_{w, p}^{i*}) \quad (11)$$

138 The quality of the solution is evaluated by two indicators. The first indicator that calculates the travel cost gap  
139 per user in seconds is:

$$AGap(\pi, GC^*) = \frac{\sum_{w \in W} \sum_{\tau=1}^T \sum_{i=1}^I \sum_{p \in P_{w, \tau}} \pi_{w, p}^i (GC_{w, p, \tau}^i - GC_{w, p, \tau}^{i*})}{\sum_{w \in W} \sum_{\tau=1}^T \sum_{i=1}^I \sum_{p \in P_{w, \tau}} \pi_{w, p}^i} \quad (12)$$

140 The total gap (Eq. 11) and the first quality indicator (Eq. 12) are calculated for each solution with the updated  
141 shortest path. The perfect user equilibrium state has zero  $AGap$ ; thus, the best optimization algorithm obtains

142 minimum  $AGap$ . In other words,  $AGap$  measures the distance between the algorithms solution and UE. The  
 143 second indicator is the assignment violation. The violation indicator is defined in order to count users assigned on  
 144 (a) non-optimal path(s). The following steps calculate the violation indicator:

145 1. Calculate the user violation: It is defined by considering the gap of each user. The following function ( $UV_w^i$ )  
 146 defines the user violation:

$$UV_w^i = \begin{cases} 1; & \text{if } \frac{ETT_w^i - TT_w^*}{TT_w^*} \geq 0.1 \\ 0; & \text{o.w.} \end{cases} \quad (13)$$

147 where  $ETT_i$  is the experienced travel time of the user  $i$  who travels for OD pair  $w$  and  $TT_w^*$  is the shortest  
 148 path of OD pair  $w$ . In other words, if the gap between the user perceived travel time and shortest path travel  
 149 time is bigger than 10% of the shortest path travel time, the user is in violation.

150 2. Compute the OD violation: The OD pair  $w$  is in violation when there are more than 10% of the users on  $w$   
 151 in violation. Particularly, function  $ODV_w$  defines the OD violation:

$$ODV_w = \begin{cases} 1; & \text{if } \frac{\sum_{i \in I_w} UV_w^i}{|I_w|} \geq 0.1 \\ 0; & \text{o.w.} \end{cases} \quad (14)$$

152 where  $I_w$  is the set of users who travel for OD pair  $w$ .

153 3. The violation indicator of network  $G$  is the share of ODs which are in violation. Therefore, the second  
 154 indicator for the quality of the solution is defined as follows:

$$V(G) = \frac{\sum_{w \in W} ODV_w}{|W|} \quad (15)$$

155 As the first indicator evaluates the overall quality of the solution, the assignment violation evaluates the so-  
 156 lution's quality at the OD level. Based on  $AGap$  and  $V(G)$ , we can measure the quality of the solution at each  
 157 iteration. The  $AGap$  indicator is used for checking the convergence. The steps of the column generation for  
 158 simulation-based DTA is detailed in Algorithm 1.

159 The framework has two convergence tests to stop the calculation process in order to save computation time. In  
 160 both steps, first, the iteration index is checked with the maximum number of loops ( $j_{max}$  and  $k_{max}$ ) that we fix  
 161 for each loop. Suppose it is not reached ( $j < j_{max}$  or  $k < k_{max}$ ), we check the solution's quality at the current  
 162 loop, but there are three differences between the outer loop convergence test and inner loop convergence test. First,  
 163 in the outer loop, we calculate the quality indicator (Eq. 12) based on the new shortest path(s), but the path set is  
 164 fixed during the inner loop operations, and just the shortest path is updated in each iteration. Second, in the inner  
 165 loop, we calculate the relative total gap (i.e., the difference of the total gap between two iterations) to check the  
 166 improvement of the process and compare it with  $Gap_\beta$ .  $Gap_\beta$  is a threshold for checking the exploration rate of  
 167 the optimization algorithm in the inner loop. But in the outer loop, the goal is to monitor the quality of the solution.  
 168 Therefore, the  $AGap$  is calculated and compared with the fixed value ( $Gap_\alpha$ ). If we have a solution with  $AGap$   
 169 equal to or less than threshold  $Gap_\alpha$ , we terminate the algorithm. Third, in the outer loop, we have one more  
 170 condition other than checking the solution's quality. The condition is the shortest path checking. If we have no new



171 shortest path per OD and good quality of the solution, then the process is converged. It means we will continue the  
 172 process when the quality of the solution is not good until we no longer have a new shortest path.

---

**Algorithm 1:** Simulation-based DTA: Equilibrium calculation

---

**Result:** UE path flow distribution

Initialization: Set outer loop iteration index to one ( $j = 1$ ),  $z_j =$  All-or-Nothing assignment

(i.e., everyone is placed on the shortest path);

Execute time-dependant shortest path algorithm and update the path set for all ODs ;

Calculate  $AGap(z_j)$  based on new shortest path(s);

Determine  $j_{max}$  and  $k_{max}$ ;

**while**  $j < j_{max}$  **and** ( $AGap(z_j) \geq Gap_\alpha$  **or** *There is at least one new shortest path for one OD*) **do**

    Set inner loop iteration index to one ( $k = 1$ );

    Set  $z_k = z^j$  and  $AGap(z^0) = AGap(z_j) + 2 \times Gap_\beta$ ;

**while**  $k < k_{max}$  **and**  $|AGap(z^k) - AGap(z^{k-1})| > Gap_\beta$  **do**

        Execute **optimization algorithm** to calculate  $z^{k+1}$ ;

        Execute traffic simulation with  $z^{k+1}$ ;

        Identify the shortest path for all ODs based on the simulation results;

        Update  $AGap(z^{k+1})$  based on updated shortest path(s);

        Keep the best solution of the current inner loop;

        Set  $k = k + 1$ ;

**end**

    Produce the best solution of the inner loop as  $z^k$ ;

    Set  $z_{j+1} = z^k$ ;

    Execute time-dependant shortest path algorithm based on  $z_{j+1}$ ;

    Update the path set for all ODs ;

    Calculate  $AGap(z_{j+1})$  based on new/updated shortest paths;

    Set  $j = j + 1$ ;

**end**

---

174 The optimization algorithm is embedded in the inner loop of the process (Figure 1) that updates the assignment  
 175 plan in each iteration. The heuristic and meta-heuristic algorithms are defined in this step.

### 176 3.1 Heuristic methods

177 Generally, the heuristic methods are path-swapping descent direction methods that decompose the problem into  
 178 many (origin-destination, departure time interval, and user class) sub-problems. They solve each of the sub-  
 179 problems by adjusting time-varying OD flows between all non-shortest paths and the shortest path(s). We did a  
 180 cross-comparison of heuristic algorithms in [16] with homogeneous users. Here we present the best algorithm for  
 181 large-scale, which may have a good performance in multi-class and multi-modal contexts.

#### 182 3.1.1 Method of Successive Average with Ranking method

183 The first method is an extension of the Method of Successive Average (MSA). The MSA algorithm is the classical  
 184 method to solve the traffic assignment problem [26] and widely used in theory and application for DTA problems  
 185 [27]. Eq. 16 presents the swapping formula of MSA algorithm [28]:

$$z^k = \left(\frac{k}{k+1}\right)z^{k-1} + \frac{1}{k+1}y^k \quad (16)$$

186 where  $y^k$  is extracted from the auxiliary path assignments obtained by the All-or-nothing procedure, i.e., everyone  
 187 is placed on the shortest path. Note that the step size in this method is  $\frac{1}{k+1}$ . It means that during iteration  $k$  for

188 this method, a fraction  $\frac{1}{k+1}$  of users on each non-shortest path swap to the one of the shortest path in the current  
 189 iteration. MSA Ranking add a ranking method to choose the users from non-shortest paths to the shortest path(s).  
 190 In other words, it ranks all users by their generalized cost and swaps the users with higher  $GC$  to the shortest  
 191 path(s). The step size is the same as the MSA method equals to  $\frac{1}{k+1}$  of all users on non-shortest paths.

### 192 3.1.2 Gap-based method

To enable the deployment of large-scale (simulation-based) DTA models, [9] proposes a heuristic approach which we called "Gap-based" method in this study. This method updates the path flow in optimization iteration  $k$  based on gap between the path  $p$  generalized cost of user class  $i$  ( $GC_p^i$ ) and that OD shortest path generalized cost of user class  $i$  ( $GC_{p^*}^i$ ):

$$\begin{cases} \pi_p^{i,k+1} = \max\{0, \pi_p^{i,k} \cdot [1 - \rho^k \frac{GC_p^i - GC_{p^*}^i}{GC_p^i}]\}; & \text{if } p \notin P^{i*} \\ \pi_p^{i,k+1} = \pi_p^{i,k} + \frac{\sum_{p \in P_w - P^{i*}} [\pi_p^{i,k} \cdot \rho^k \frac{GC_p^i - GC_{p^*}^i}{GC_p^i}]}{|P^{i*}|}; & \text{if } p \in P^{i*} \end{cases} \quad (17)$$

$$\rho^k = \begin{cases} \frac{1}{k+1}; & \text{if } j = 0 \\ 1; & \text{o.w.} \end{cases} \quad (18)$$

193 where  $j$  denotes the outer loop iteration index,  $k$  denotes the inner loop iteration index, and  $|P^{i*}|$  is the number  
 194 of the shortest paths in iteration  $k$ . This path assignment updating scheme implies a natural path flow adjustment  
 195 mechanism: flows on the non-shortest paths are moved to the shortest path and the volume moved out from a  
 196 non-shortest path is proportional to the gap between the non-shortest path and shortest path over non-shortest path  
 197 cost.

198 MSA ranking and Gap-based method have two main drawbacks which are also the same for all heuristic meth-  
 199 ods that come from the flow-based traffic assignment:

- 200 • Using the step size guarantees the algorithm to converge but decreases the flexibility of the method to swap  
 201 more users to the shortest path(s). Moreover, the final solution is obtained based on the initialization (All-or-  
 202 nothing) because the algorithm converges (by decreasing the step size) after a while even we use a not good  
 203 initial assignment which is not really obvious in large-scale.
- 204 • Many of the methods in the literature (see, e.g., [7, 10, 11, 14, 15, 27, 29–31]) are using the average travel  
 205 time of each path to calculate the indicators and to distinguish the shortest path(s) and other paths per OD.  
 206 It means they do not consider users' experienced travel time directly. Moreover, the optimizer is applied for  
 207 each OD and swap users based on the different values of the paths' average travel time. Note that users in the  
 208 same departure time interval and on the same path can perceive different travel time values because of the  
 209 network dynamic evolution.

210 These drawbacks show that we always look for the local optimal based on initialization and our step size. Here  
 211 we present a probabilistic approach to help the heuristic optimization framework in order to find the global optimum  
 212 for trip-based traffic assignment.

### 213 3.1.3 Probabilistic method

The Probabilistic method is a trip-based approach. This method has no step size for optimization [16]. For each user the probability of swap to the shortest path(s) is calculated by the following formula:

$$\text{Swap indicator} = \frac{EGC_p - GC_{p^*}^i}{GC_p} \quad (19)$$

214 where  $EGC_p$  is the experienced generalized cost of the user. Indeed the path swapping approach is applied to  
 215 every user, not paths (number of users on path  $p$ ). Afterward, we use a random number uniformly distributed  
 216 between 0 and 1 for taking a decision about swapping the user or not. If the random number is lower than the swap  
 217 indicator, the user swaps to the shortest path, and otherwise, the user keeps his/her path for the current iteration.  
 218 The probabilistic process enables the algorithm to explore more the solution space without any step size. This  
 219 heuristic method converges intuitively for a large-scale network because the swap indicator decreases by swapping  
 220 based on random numbers. Note that at the end of each outer loop the final assignment plan is the best assignment  
 221 in terms of average gap between different inner loops solutions. It's obvious that because of the swap indicator the  
 222 algorithm converges by random numbers to the solution with a minimum number of possible swaps which means  
 223 the small value of gap for each user. In other words, with every swapping decision, users try to optimize his  $GC$  in  
 224 order to reduce the swap indicator (the possibility of swapping) for next inner loop.

### 225 3.1.4 Hybrid methods

226 Verbas et al. [32] applied an extension of the gap-based method to transit assignment. The method uses Eq. 17  
 227 to determine the number of swapping users for each path and then apply the Probabilistic method to choose users  
 228 from each path for swapping. This method is called "Gap-based Prob.". We examined this method in DTA test  
 229 cases, and it fully dominates the Gap-based method in different mono-class test cases [16]. Therefore we consider  
 230 this method in our numerical experiments.

231 The second hybrid method is similar to the Probabilistic method with a different swap indicator:

$$\text{Swap indicator} = \frac{1}{k+1} \frac{EGC_p - GC_{p^i}^*}{GC_p} \quad (20)$$

232 This method is called "Step-size Prob." in this study. It has a good performance in large-scale and congested  
 233 traffic networks [19]. We developed a smart step-size selection procedure for this method in which we specialize the  
 234 step-size for each OD pair and decrease the step-size when the OD gap is not improved compared to the previous  
 235 iteration [16]. The method is called "Smart Step-size Prob.". All the methods that are presented in this section will  
 236 apply to multi-class multi-modal DTA problem in order to evaluate their performance and also compare them with  
 237 meta-heuristic methods.

## 238 3.2 Meta-heuristic methods

239 As discussed in the previous section, in iteration  $k$ , the classical (heuristic) algorithms try to improve the path flow  
 240 distribution by swapping a fraction  $\frac{1}{k+1}$  of users to the shortest path(s) from each non-shortest path. Then, one  
 241 simulation is launched based on the updated path flow distribution. We have to wait until the simulation run is  
 242 finished to know the new link travel times and adjust the path flow distribution to be tested in the next iteration  
 243 accordingly. This is the serial process of the classical algorithms, which limits the solution space exploration and  
 244 computational process. We developed two meta-heuristic algorithms in [17] in order to overcome the problem of  
 245 the serial approach. Two parallelized meta-heuristic approaches are designed: the first derived from the simulated  
 246 annealing framework and the second from that of the genetic algorithm.

### 247 3.2.1 Simulated annealing algorithm

248 The algorithm starts with an initial solution generated randomly, i.e., users choose their path from the OD path set  
 249 randomly. For solution  $z$ , the total gap  $TGap(\cdot)$  between the users' travel cost and the shortest path travel cost  
 250 (Eq. 11) is considered as the energy of the solution. A set of neighbor solution is generated with respect to the  
 251 current one based on the temperature ( $T$ ) of the current iteration. A neighbor solution is defined as a candidate to  
 252 replace the current solution. The current phase of the iteration depends on the temperature of the process. Inspired

253 by the physics of matter, this study distinguishes three different methods to generate a neighbor solution, gas, liquid,  
 254 and solid; these methods represent the states of matter in nature. When the temperature is high ( $T > \lambda$  where  $\lambda$   
 255 denotes the boiling temperature), the *gas method* is applied. When running the SA algorithm, by decreasing the  
 256 temperature, the algorithm enters the liquid phase ( $\lambda > T > \lambda'$  where  $\lambda'$  denotes the melting temperature) and  
 257 then the *liquid method* is applied. When the temperature is quite low ( $T < \lambda'$ ), the *solid method* is applied. The  
 258 algorithm is presented below:

---

**Algorithm 2:** Simulated annealing algorithm

---

**Result:** Inner loop path flow distribution of iteration  $k$

Initialization: Set  $T = T_0$ , where  $T_0$  denotes the initial temperature;

Set inner loop iteration index to one ( $k = 1$ ) and  $AGap(z^0) = AGap(z^{-1}) = 2AGap(z_j)$ ;

**while**  $k < k_{max}$  **and**  $AGap(z^k) \neq AGap(z^{k-1}) \neq AGap(z^{k-2})$  **do**

**if**  $T > \lambda$  **then**

**Gas Phase:**

      Generate a solution candidate by Randomization method.

      Randomization: users choose their path randomly from the path set;

**else if**  $T > \lambda'$  **then**

**Liquid Phase:**

      Generate the first solution candidate by Randomization method;

      Generate the second and third solution candidates by applying MSA and Gap-based methods to  $z^i$ ;

**else**

**Solid Phase:**

      Generate the first and second solution candidates by applying MSA and Gap-based methods to  $z^i$ ;

**end**

  Execute traffic simulation in parallel for all candidates;

  Identify the shortest path for all ODs based on the simulation results for each candidate;

  Update  $AGap(\cdot)$  based on updated shortest path(s) for all candidates;

  Calculate the solution acceptance probability for each candidate by the following equation:

$$P(S'_{\text{accepted}} = 1) = e^{-\frac{\nabla E}{T}}$$

  where  $s'$  denotes a candidate,  $S'_{\text{accepted}}$  denotes the binary decision variable and

$$\nabla E = TGap(s') - TGap(s);$$

  Take decision about accepting each candidate to determine  $z^{k+1}$ ;

  Decrease the temperature by the following formula:

$$T = \frac{T_0}{\ln(k+1)};$$

  Set  $k = k + 1$ ;

**end**

  Produce the final solution for the outer loop.

---

### 260 3.2.2 Genetic algorithm

261 The Genetic Algorithm (GA) (first formalized by [33]) is inspired by natural selection and genetic variation. The ge-  
 262 netic algorithm includes three main operators: selection, crossover, and mutation. The description of the operators  
 263 is well presented in [34]. The GA framework is adapted to the traffic assignment problem in [17]. The algorithm is  
 264 designed as a two-layer GA process to search solution space by changing the path flows in **GA Inner** and overcome  
 265 the drawback of OD impact by considering different combinations of OD assignments in **GA-operators**. In other  
 266 words, the heuristic algorithms plus a random method are applied in **GA Inner**, and the GA operators in one upper  
 267 level generate different combinations of OD. **GA Inner** is a new layer in the optimization process compared to the  
 268 classical GA. It has been added to account for correlations between OD assignments. Moreover, GA considers a  
 269 set of solutions instead of a single solution at every iteration. The steps of the GA are as follows:

**Algorithm 3:** Genetic algorithm

**Result:** Inner loop path flow distribution of iteration  $k$

Initialization: Generate the initial solution population based on Randomization;

**if**  $j > 1$  **then**

  | Add the solution comes from the outer loop.

Set inner loop iteration index to one ( $k = 1$ );

$M(p^0) = M(p^{-1}) = 0$ , where  $M(p^k)$  denotes the mean average gap of the population of iteration  $k$  ( $p^k$ );

**while**  $k < k_{max}$  **and**  $M(p^k) \neq M(p^{k-1}) \neq M(p^{k-2})$  **do**

**1. GA-operators:** Apply the GA operators to each pair of selected solutions.

**1.1. Selection:** Random selection based on the crossover rate ( $Cr$ ) and population size ( $PS$ ) to compute the number of selected solutions for the crossover process:  $SS = PS \times Cr$  (Note that based on [35], we set  $Cr = 0.5$  and the  $PS$  is set to 10 individuals in this study.);

**1.2. Crossover:** Apply a non-uniform crossover by using a bit-vector mask method [36]. We select two different solutions (parents) from the set of selected solutions. Then apply the crossover between each pair of solutions. Two new solutions are generated.;

**1.3. Mutation:** Apply the mutation operator by replacing one OD assignment of the solution by another corresponding OD assignment from another solution. The possibility of the mutation for one OD assignment is calculated based on the quality of the OD assignment:

$$MP_w = \frac{TGap_w(s)}{TGap(s)}$$

where  $TGap_w(s)$  denotes the total gap of the OD assignment  $w$  in solution  $s$  (note that in application of each mutation operator, two new solutions are generated).;

**2. GA Inner:** Apply the GA Inner operators to every selected solutions

    Adaptive random method [17];

    Execute traffic simulation in parallel for all new solutions obtained from the previous steps;

    Identify the shortest path for all ODs based on the simulation results for every new solutions;

    Update  $AGap(\cdot)$  based on updated shortest path(s);

    Apply the selection operator to keep the size of population fix by a replacement strategy.;

    Set  $k = k + 1$ ;

**end**

Produce the final solution (a solution with the minimum  $AGap$  in the population  $p^k$ ) for the outer loop.

## 4 Numerical Experiments

The multi-class model is applied to a large-scale multi-modal network of Lyon 6th + Villeurbanne. This network covers  $25 \text{ km}^2$  and includes 1,883 Nodes, 3,383 Links, 94 Origins, 227 Destinations and located in Lyon Metropolis is described in Figure 2. All the signalized intersections in the real field have been implemented in the simulator with their actual signal timing. In this work, we use Symuvia<sup>1</sup> as a trip-based simulator for calculating the needed variables in the network. Symuvia is a microscopic simulator based on a Lagrangian discretization of the LWR model [37]. We get the simulation time-step to 1 second and collect the link traffic information (travel times) every 1 minute. Note that the dynamic simulator uses Eq. 7 to calculate the generalized travel cost on a link at each time interval. There are buses (with/ without a dedicated line), trams, and metro lines considering public transportations. In the simulation process, all these patterns, in addition to the private car and walking option, are simulated. For instance, for using a metro line, a user walks (walking time) to the station, then waits for the next train (waiting time). When the train arrives at the station, if the train has free capacity, the user can take the train (boarding time) and then take the next patterns until arriving at the destination point. All of these actions are considered for the

<sup>1</sup>Symuvia is an open source simulator (<https://github.com/Ifsttar/Open-SymuVia>).

284 travel time calculation. The users' routes are determined by the DTA model and the rolling horizon technique [38]  
 285 which determine the path flow distribution based on a prediction period of 30 minutes and an assignment period  
 286 of 15 minutes. The multi-modal time-dependent shortest path algorithm is implemented based on the multi-origin  
 287 Dijkstra algorithm [39] with considering hyper paths based on [40] in order to find minimum cost path(s) in terms  
 288 of travel time.

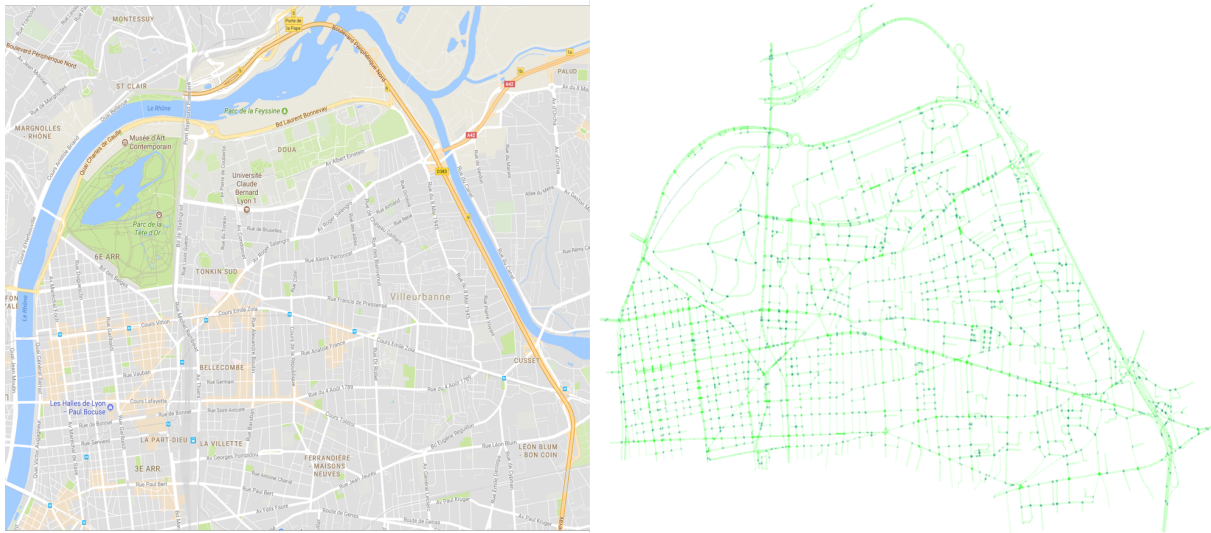


Figure 2: Lyon 6th + Villeurbanne: Mapping data ©Google 2020 and the traffic network using by Symuvia

289 The network is loaded with 54,190 travelers of all ODs with given departure times in order to represent the 2.5  
 290 morning peak hours of the network between 6:30 to 9:00. The demand profile comes from the study of [41]. Note  
 291 that here, we fixed the departure time in order to focus on travelers' path choices. The main reason is that this study  
 292 aims to compare the performance of DTA solvers. By fixing the departure time, the input of the problem will be  
 293 the same for all algorithms. Therefore, the solution space is fixed and given for the route (path and mode) choice  
 294 problem, and we can compare the performance of different solution algorithms.

295 Figure 3(a) presents the demand profile of the numerical experiment. The figure presents the inflow rate of the  
 296 trips to the traffic network. The total length of the Lyon 6th + Villeurbanne network is 496.73 kilometers. Note  
 297 that each OD has a time-dependent demand profile based on the real data of Lyon city [41]. The dynamic loading  
 298 represents the saturation state of the network. To show a quick and synthetic overview of the network state, we  
 299 plot the Macroscopic Fundamental Diagrams (MFD) of the network in Figure 3(b). This diagram represents the  
 300 overall evolution of the traffic conditions in the network by plotting the total travel distance vs. the total travel time  
 301 [42]. Note that total travel distance is proportional to the mean network flow, while total travel time is equivalent  
 302 to vehicle accumulation.

303 The MFD shows the network state for every 5 minutes between 7:00 to 9:00 during the full simulation. Before  
 304 7:00, the demand is low, and the network is almost empty. As such, we filtered out the first 30 minutes in order  
 305 to present the critical part of the evolution of the network traffic conditions. For instance, point number 15 of the  
 306 MFD (Figure 3(b)) shows the total travel distance and total travel time of all the travelers in the network between  
 307 8:10 to 8:15. First, the MFD curve increases from (0, 0). The traffic states remain under-saturated (point numbers  
 308 1 to 8) when demand is light, in this case, from 6:30 to 7:40. Afterwards, travel production, which is equivalent to  
 309 the total travel distance for a given period of time, stabilizes while the accumulation (or total travel time) continues  
 310 to increase (point numbers 9 to 13). This corresponds to the saturation level occurring from 7:40 to 8:05. The  
 311 decrease in travel production and accumulation (point numbers 14 to 24) shows that the network starts to exit the

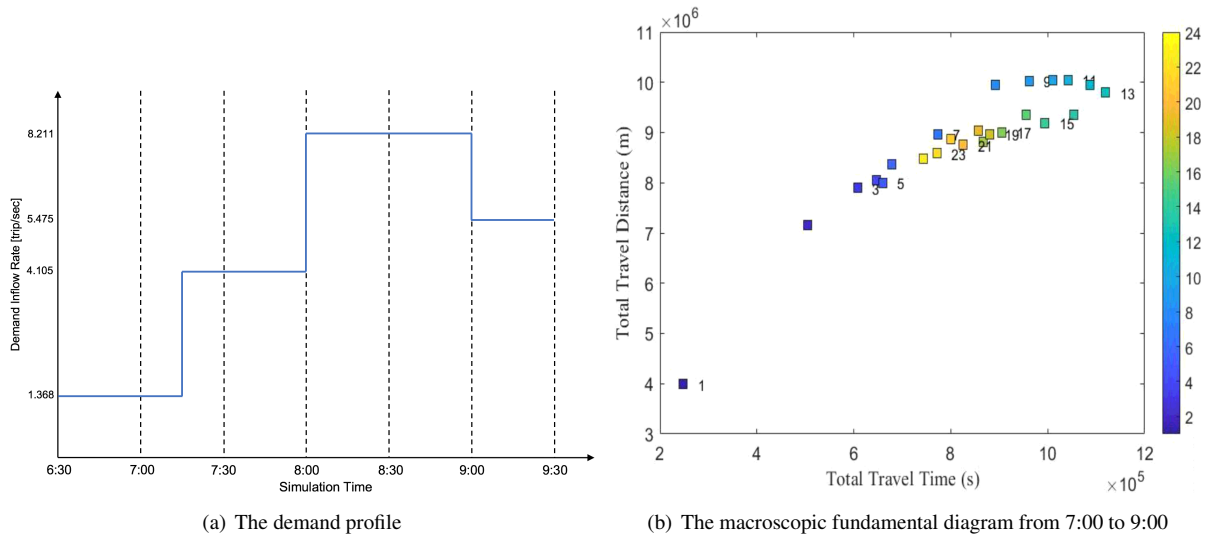


Figure 3: The demand scenario of Lyon 6th + Villeurbanne.

312 saturation level at the end of the simulation period and slowly returns to the under-saturation level from 8:05 to  
 313 9:00.

314 The trips are divided into nine classes that are presented in Table 2. In this table, the user class's fraction of the  
 315 total demand is presented by "percentage of trips", and for each class, we use a specific VOT. The demand level  
 316 and VOT values are extracted from the real data of the full Lyon networks [41]. The optimization goal is to find the  
 317 user equilibrium based on the generalized cost functions.

Table 2: User classes and their value of time for the Lyon 6th + Villeurbanne network

Percentage of trips	Class name (job)	Value of time [euro/hour]
3.20%	worker category 1	52.56
6.40%	worker category 2	39.24
35.1%	worker category 3	18.72
26.70%	student	7.92
0.70%	worker-student	10.80
18.80%	retired	9.36
3.2%	at home	6.84
5.9%	unemployed and other	15.36

318 The optimization process is executed for a maximum of 10 outer loops for five assignment periods. At each  
 319 outer loop iteration, (a) new shortest path(s) can be discovered; therefore, if at least a new shortest path is found per  
 320 outer loop iteration, we have 11 paths per OD. In each outer loop, a maximum of 40 Inner loops has been executed  
 321 to find the best assignment pattern. Note that, in heuristic and SA algorithms, there is one more simulation per  
 322 outer loop for keeping the best assignment pattern if the process is converged by the maximum number of the inner  
 323 loop. It means if the converge does not happen during all inner loop, we take the best solution and simulate the  
 324 network to prepare the result for the outer loop convergence test. The convergence tests of two loops are fixed for  
 325 all algorithms. In the inner loop convergence test, the optimization will be stopped when we have less than 1%  
 326 improvement ( $Gap_\beta = 0.01$ ). Improvement is related to the average gap between the current and previous inner  
 327 loops. For the outer loop, we will stop the process when we do not find a new shortest path for all the ODs and also  
 328 have a small value of gap or have no improvement in terms of total gap ( $Gap_\alpha = 0$ ). In the next section, we will

329 compare the performance of heuristic and meta-heuristic algorithms in the multi-class multi-modal DTA problem  
 330 based on the numerical results.

## 331 5 Numerical Results and Discussion

332 The full optimization process is done for all methods. The aim of the experiments performed on the Lyon 6th +  
 333 Villeurbanne network is to examine the algorithms convergence pattern of the solutions for a large-scale network  
 334 case. Thus, the UE is calculated for the considered network seven times with the heuristic and meta-heuristic  
 335 algorithms defined in Section 3. The  $AGap$  indicator is used to evaluate the quality of the solution.

336 The results for the performance indicators of all the algorithms are presented in Table 3. GA leads to  $\sim 10\times$   
 337 more simulations, and SA leads to only  $\sim 1.3\times$  simulations compared to the  $\sim 550$  simulations required in heuristic  
 338 algorithms. However, the computation times are significantly lower because of the parallel simulation framework.  
 339 Moreover, the solutions obtained by the meta-heuristic algorithms are significantly closer to the optimal UE than  
 340 other methods. The meta-heuristic algorithms dominate all heuristic methods in the final solution  $AGap$ ; indeed,  
 341 they are significantly better than the Gap-based Prob. algorithm, which is the best heuristic algorithm in this  
 342 experiment. The GA and SA algorithms manage to reduce the UE  $AGap$  of the Gap-based Prob. method by more  
 343 than 50% (Table 3). The Violation indicator also shows that meta-heuristic algorithms work much better than the  
 344 heuristics methods (reduction of 54%).

Table 3: Solution quality and performance indicators [CT: Computation time].

Indicator/ Method	Number of simulations	Incomplete travels (%)	Violation $V(G)$	$AGap$ [euros]	Improvement to $AGap$ compared to GBP	CT [Hours]	Improvement to CT compared to SSP	MAX number of cores used
MSA Ranking	566	5,74%	13.77%	1.38	-24.32%	132.26	-9.74%	1
Gap-based Prob. (GBP)	594	4.08%	9.03%	1.11	-	137.52	-14.10%	1
Probabilistic (Prob.)	598	4.66%	11.64%	1.23	-10.81%	143.19	-18.81%	1
Step-size Prob. (SSP)	514	9.11%	14.77%	2.61	-135.14%	120.52	-	1
Smart Step-size Prob.	591	5.16%	10.22%	1.27	-14.41%	139.59	-15.81%	1
Simulated Annealing	778	3.01%	4.82%	0.32	71.17%	91.30	24.25%	3
Genetic Algorithm	5866	3.79%	6.09%	0.54	51.35%	83.78	30.49%	18

345 Regarding the percentage of incomplete trips, also the meta-heuristic algorithms dominate the others. The  
 346 incomplete trips indicator denotes the share of trips, which is not finished by the end of the simulation in the final  
 347 path flow distribution of each algorithm. A lower number of incomplete trips means a lower total travel time spent  
 348 in the system over the simulation period. The SA algorithm finds the closest solution to UE (minimum  $AGap$ ) in  
 349 this study. Moreover, the final solution of SA has the best value for other quality indicators that are highlighted in  
 350 green in the SA row in Table 3.

351 Moreover, we measure the computation time (CT) of algorithms to evaluate the performance of them. The CTs  
 352 of SA algorithm and GA are significantly better than heuristic methods. In particular, it is significantly better than  
 353 the SSP method (green values in GA row inside Table 3). Note also that the DTA process over the full simulation  
 354 period requires considerable computational resources because of the network size. Therefore, the computational  
 355 improvement obtained by the meta-heuristic methods is high, on average, 47 hours (two days) for the SA and GA  
 356 compared to the heuristic algorithms. The GA reduces the CT by one day and a half compared to SSP as the fastest  
 357 heuristic method.

358 Regarding the computation resources, every simulation usually needs one central processing unit (core) of  
 359 the processor. The heuristic algorithms, as mentioned before, are run in series, so they need only one core per  
 360 iteration. The SA algorithm generates a maximum of three new path flow patterns in his *liquid method* per iteration.  
 361 Therefore, we need a maximum of three cores for the simulation process. For the GA, according to population size,  
 362 crossover rate and the mutation rate, this value is maximum 30 new solutions from the GA in this study. Because  
 363 all the experiments are conducted on a 64-bit personal computer with 18 cores, the number of cores is limited to



364 18. If the number of new solutions in GA is bigger than 18, the algorithm executes in two successive phases, the  
 365 first 18 simulations in the first phase and the remaining solutions in the second phase.

366 Figure 4 presents the convergence pattern for all seven algorithms in the last assignment period. The conver-  
 367 gence pattern is drawn based on the average delay of all users. By the best heuristic methods, the final user average  
 368 gap is around 30 sec, but this value for the meta-heuristic method is around 10 sec. It means that the meta-heuristic  
 369 methods also obtain the better solutions in term of delay. In Figure 4, the convergence pattern of all heuristic algo-  
 370 rithms first decrease and then variate in a small range for the last iterations, where they seem to stick to the local  
 371 optimum. Meta-heuristic algorithms convergence pattern is always decreasing, which means the algorithms are  
 372 exploring and looking for the global optimum. Therefore, both meta-heuristic algorithms produce not only better  
 373 solutions in terms of optimality but also better (i.e., not increasing) convergence patterns than classical algorithms.

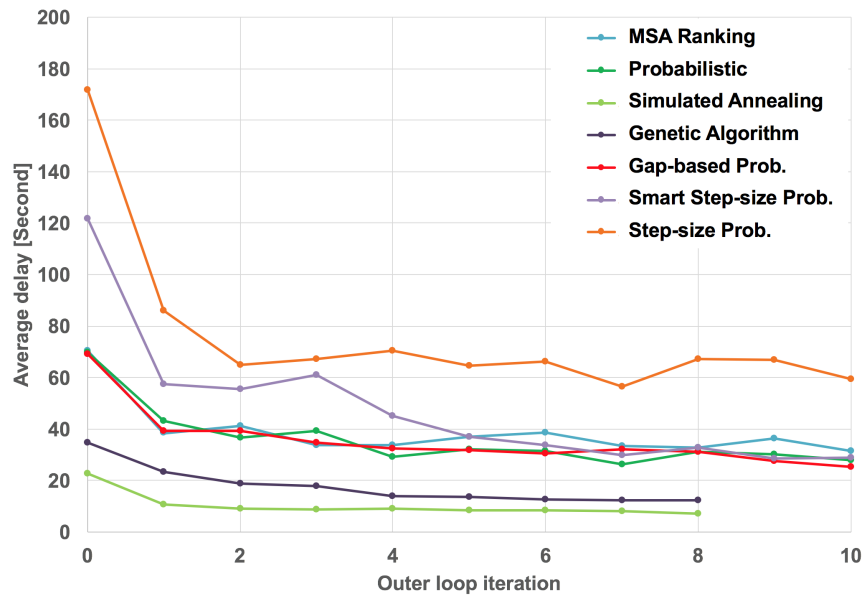


Figure 4: Convergence patterns for the outer loops

374 In order to consider the complexity of the problem on the performance of algorithms, a mono-class demand  
 375 scenario is also designed. The volume of the demand and the traffic network are the same as the multi-class  
 376 scenario. For mono-class users, we consider the weighted average VOT of the multi-class scenario (15.84 euro per  
 377 hour), extracted from Table 2, and the same monetary cost for public transportation. The setting for the optimization  
 378 framework, including the maximum inner and outer loops' iterations, is fixed to the same values as the multi-class  
 379 scenario. Table 4 presents the mono-class scenario results with the multi-class scenario. Same as the previous  
 380 experiment, meta-heuristic algorithms dominate heuristic algorithms. The results show that heuristic algorithms  
 381 cannot guarantee the user equilibrium for the large-scale network.

Table 4: Comparison between the performance of algorithms with homogeneous and heterogeneous users.

Demand profile; Quality indicator /Method	Mono-class users				Multi-class users			
	<i>AGap</i> [euros]	Improvement to <i>AGap</i> compared to Prob.	CT [Hours]	Improvement to CT compared to Smart SSP	<i>AGap</i> [euros]	Improvement to <i>AGap</i> compared to GBP	CT [Hours]	Improvement to CT compared to SSP
MSA Ranking	0.47	-40.03%	105.62	-9.28%	1.38	-24.32%	132.26	-9.74%
Gap-based Prob. (GBP)	0.42	-23.88%	101.81	-5.33%	1.11	-	137.52	-14.10%
Probabilistic (Prob.)	0.34	-	98.12	-1.53%	1.23	-10.81%	143.19	-18.81%
Step-size Prob. (SSP)	0.55	-64.78%	108.71	-12.48%	2.61	-135.14%	120.52	-
Smart Step-size Prob.	0.37	-10.45%	96.65	-	1.27	-14.41%	139.59	-15.81%
Simulated Annealing	0.21	36.36%	85.62	11.41%	0.32	71.17%	91.30	24.25%
Genetic Algorithm	0.27	18.48%	72.84	24.64%	0.54	51.35%	83.78	30.49%

382 The results for  $AGap$  and CT of the multi-class setting are worse than the mono-class because of the VOT of  
 383 different classes that change the scale of  $AGap$  and also require more CT to explore the solution space of the multi-  
 384 class setting. However, regarding the  $AGap$ , it is even more beneficial to use meta-heuristic methods over heuristic  
 385 methods in a multi-class environment, as the improvements in  $AGap$  are significantly better than the mono-class  
 386 problem (Table 4). The same conclusion can be drawn for CT. The GA algorithm is the fastest algorithm in both  
 387 scenarios. The performance comparison of solution algorithms in two demand scenarios shows that meta-heuristic  
 388 algorithms are more efficient when the problem's complexity is increased.

## 389 6 Conclusion

390 This study develops the multi-class model by designing the hybrid generalized cost function VOT and cost function,  
 391 including mode and economic classes). The model is extended for DTA, and variational inequality formulation is  
 392 presented for the dynamic user equilibrium problem. This problem is also challenging from the computation point  
 393 of view in the large-scale application. We compared two different assignment frameworks (heuristic and meta-  
 394 heuristic) to execute the model on the large-scale network in order to investigate which algorithm is more efficient.  
 395 Seven optimization methods are discussed to calculate the user equilibrium. The model and assignment framework  
 396 are applied to the Lyon 6th + Villeurbanne network, and the two optimization approaches are compared on the large-  
 397 scale network. The multi-attribute multi-class model and optimization process have proven to be successful when  
 398 applied to our multi-modal large-scale network. The results show that the meta-heuristic methods (SA and GA) with  
 399 parallel computation work significantly better than the heuristic methods. The average user's gap in generalized  
 400 cost for meta-heuristic algorithms is about 4.8 times in average better than the heuristic methods. Furthermore, the  
 401 meta-heuristic algorithms save a minimum of 36 hours in computation time. GA is the fastest algorithm, and SA  
 402 provides the best solution for the network user equilibrium problem. Moreover, we can conclude that all heuristic  
 403 methods do not guarantee the user equilibrium solution in large-scale DTA problems.

404 As meta-heuristic algorithms are new, there is room to improve them and also apply other kinds of meta-  
 405 heuristic algorithms. In addition to this, the authors propose the following topics for future works:

406 1- Consider departure time choice models. One possibility can be adding another layer of optimization for the  
 407 departure time adjustment process. It can also be interesting to use the proposed parallel framework to optimize  
 408 departure time and route choices simultaneously. 2- Extend the multi-attribute, multi-class trip-based model to  
 409 activity-based models in order to consider travel with activity chains. 3- Compare and analyze the performance of  
 410 the meta-heuristic method in other large-scale networks with different typologies and the other types of optimization  
 411 methods. 4- Combine and investigate the meta-heuristic methods with other kinds of traffic simulators and shortest  
 412 path calculation methods.

## 413 Acknowledgement

414 We thank the two anonymous reviewers whose comments and suggestions helped improve and clarify this  
 415 manuscript. This work has received funding from the European Research Council (ERC) under the European  
 416 Union's Horizon 2020 research and innovation program (Grant agreement No 646592 – MAGnUM project).

## 417 References

- 418 [1] J. G. Wardrop, "Road paper. some theoretical aspects of road traffic research.," *Proceedings of the institution*  
 419 *of civil engineers*, vol. 1, no. 3, pp. 325–362, 1952.
- 420 [2] R. Jayakrishnan, W. K. Tsai, and A. Chen, "A dynamic traffic assignment model with traffic-flow relation-  
 421 ships," *Transportation Research Part C: Emerging Technologies*, vol. 3, no. 1, pp. 51–72, 1995.

- 422 [3] T. Schreiter, V. Wageningen-Kessels, Y. Yuan, J. Van Lint, S. Hoogendoorn, *et al.*, “Fastlane: Traffic flow  
423 modeling and multi-class dynamic traffic management,” in *Trail-Beta Congress 2012: Mobility and Logistics-  
424 Science Meets Practice, Rotterdam, The Netherlands, 30-31 October 2012*, Citeseer, 2012.
- 425 [4] W. Song, K. Han, Y. Wang, T. Friesz, and E. Del Castillo, “Statistical metamodeling of dynamic network  
426 loading,” *Transportation research procedia*, vol. 23, pp. 263–282, 2017.
- 427 [5] X. Xu, A. Zockaie, H. S. Mahmassani, H. Halat, Ö. Verbas, M. Hyland, P. Vovsha, and J. Hicks, “Schedule  
428 consistency for daily activity chains in integrated activity-based dynamic multimodal network assignment,”  
429 *Transportation Research Record*, vol. 2664, no. 1, pp. 11–22, 2017.
- 430 [6] R. Jayakrishnan and C. R. Rindt, “Distributed computing and simulation in a traffic research test bed,”  
431 *Computer-Aided Civil and Infrastructure Engineering*, vol. 14, pp. 429–443, nov 1999.
- 432 [7] T. L. Friesz and K. Han, “The mathematical foundations of dynamic user equilibrium,” *Transportation Re-  
433 search Part B: Methodological*, sep 2018.
- 434 [8] S. Peeta and A. K. Ziliaskopoulos, “Foundations of dynamic traffic assignment: The past, the present and the  
435 future,” *Networks and spatial economics*, vol. 1, no. 3-4, pp. 233–265, 2001.
- 436 [9] C.-C. Lu, H. S. Mahmassani, and X. Zhou, “Equivalent gap function-based reformulation and solution algo-  
437 rithm for the dynamic user equilibrium problem,” *Transportation Research Part B: Methodological*, vol. 43,  
438 no. 3, pp. 345–364, 2009.
- 439 [10] G. Peque, T. Miyagi, and F. Kurauchi, “Adaptive learning algorithms for simulation-based dynamic traffic user  
440 equilibrium,” *International Journal of Intelligent Transportation Systems Research*, vol. 16, no. 3, pp. 215–  
441 226, 2018.
- 442 [11] H. Sbayti, C.-C. Lu, and H. S. Mahmassani, “Efficient implementation of method of successive averages in  
443 simulation-based dynamic traffic assignment models for large-scale network applications,” *Transportation  
444 research record*, vol. 2029, no. 1, pp. 22–30, 2007.
- 445 [12] I. Yang and R. Jayakrishnan, “Gradient projection method for simulation-based dynamic traffic assignment,”  
446 *Transportation research record*, vol. 2284, no. 1, pp. 70–80, 2012.
- 447 [13] M. Mahut, M. Florian, and N. Tremblay, “Comparison of assignment methods for simulation-based dynamic-  
448 equilibrium traffic assignment,” in *Proceeding of the Transportation Research Board 87th Annual Meeting  
449 (DVD), Washington, DC*, 2008.
- 450 [14] M. Florian, M. Mahut, and N. Tremblay, “Application of a simulation-based dynamic traffic assignment  
451 model,” *European journal of operational research*, vol. 189, no. 3, pp. 1381–1392, 2008.
- 452 [15] H. X. Liu, X. He, and B. He, “Method of successive weighted averages (mswa) and self-regulated averaging  
453 schemes for solving stochastic user equilibrium problem,” *Networks and Spatial Economics*, vol. 9, no. 4,  
454 p. 485, 2009.
- 455 [16] M. Ameli, J.-P. Lebacque, and L. Leclercq, “Cross-comparison of convergence algorithms to solve trip-based  
456 dynamic traffic assignment problems,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 3,  
457 pp. 219–240, 2020.
- 458 [17] M. Ameli, J.-P. Lebacque, and L. Leclercq, “Simulation-based dynamic traffic assignment: meta-heuristic  
459 solution methods with parallel computing,” *Computer-Aided Civil and Infrastructure Engineering*, p. In press,  
460 2020.

- 461 [18] Y. Jiang, W. Szeto, J. Long, and K. Han, "Multi-class dynamic traffic assignment with physical queues:  
462 intersection-movement-based formulation and paradox," *Transportmetrica A: Transport Science*, vol. 12,  
463 no. 10, pp. 878–908, 2016.
- 464 [19] H. Halat, A. Zockaie, H. S. Mahmassani, X. Xu, and O. Verbas, "Dynamic network equilibrium for daily  
465 activity-trip chains of heterogeneous travelers: application to large-scale networks," *Transportation*, vol. 43,  
466 no. 6, pp. 1041–1059, 2016.
- 467 [20] X. Xu, F. Fakhroosavi, A. Zockaie, and H. S. Mahmassani, "Estimating path travel costs for heterogeneous  
468 users on large-scale networks: Heuristic approach to integrated activity-based model–dynamic traffic assign-  
469 ment models," *Transportation Research Record*, vol. 2667, no. 1, pp. 119–130, 2017.
- 470 [21] J. Wang, S. Peeta, and X. He, "Multiclass traffic assignment model for mixed traffic flow of human-driven  
471 vehicles and connected and autonomous vehicles," *Transportation Research Part B: Methodological*, vol. 126,  
472 pp. 139–168, 2019.
- 473 [22] M. Ameli, *Heuristic Methods for Calculating Dynamic Traffic Assignment*. PhD thesis, IFSTTAR and Uni-  
474 versité de Lyon, 2019.
- 475 [23] X. Zhan and S. V. Ukkusuri, "Multiclass, simultaneous route and departure time choice dynamic traffic as-  
476 signment with an embedded spatial queuing model," *Transportmetrica B: Transport Dynamics*, pp. 1–23,  
477 2017.
- 478 [24] M. Ameli, J. P. Lebacque, and L. Leclercq, "Day-to-day multimodal dynamic traffic assignment: Impacts of  
479 the learning process in case of non-unique solutions," in *DTA 2018, 7th International Symposium on Dynamic  
480 Traffic Assignment, Jun 2018, Hong Kong*, 2018.
- 481 [25] M. W. Levin, M. Pool, T. Owens, N. R. Juri, and S. T. Waller, "Improving the convergence of simulation-based  
482 dynamic traffic assignment methodologies," *Networks and Spatial Economics*, vol. 15, no. 3, pp. 655–676,  
483 2015.
- 484 [26] K. Nagel and G. Flötteröd, "Agent-based traffic assignment," in *The Multi-Agent Transport Simulation MAT-  
485 Sim*, pp. 315–326, Ubiquity Press, aug 2016.
- 486 [27] R. Mounce and M. Carey, "On the convergence of the method of successive averages for calculating equilib-  
487 rium in traffic networks," *Transportation Science*, vol. 49, pp. 535–542, aug 2015.
- 488 [28] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*,  
489 pp. 400–407, 1951.
- 490 [29] S. Boyles, J. Duthie, C. Melson, and T. Rambha, "Diverge models and dynamic traffic equilibria," in *IN-  
491 FORMS Annual Meeting*, 2013.
- 492 [30] C. Tong and S. Wong, "Heuristic algorithms for simulation-based dynamic traffic assignment," *Transportmet-  
493 rica*, vol. 6, no. 2, pp. 97–120, 2010.
- 494 [31] R. Mounce and M. Carey, "Route swapping in dynamic traffic networks," *Transportation Research Part B:  
495 Methodological*, vol. 45, no. 1, pp. 102–111, 2011.
- 496 [32] Ömer Verbas, H. S. Mahmassani, and M. F. Hyland, "Gap-based transit assignment algorithm with vehicle  
497 capacity constraints: Simulation-based implementation and large-scale application," *Transportation Research  
498 Part B: Methodological*, vol. 93, pp. 1–16, nov 2016.
- 499 [33] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to  
500 biology, control, and artificial intelligence*. MIT press, 1992.

- 501 [34] J. M. Johnson and Y. Rahmat-Samii, "Genetic algorithm optimization and its application to antenna design," in  
502 *Proceedings of IEEE Antennas and Propagation Society International Symposium and URSI National Radio*  
503 *Science Meeting*, vol. 1, pp. 326–329, IEEE, 1994.
- 504 [35] M. Srinivas and L. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE*  
505 *Transactions on Systems, Man, and Cybernetics*, vol. 24, pp. 656–667, apr 1994.
- 506 [36] H. Maini, K. Mehrotra, C. Mohan, and S. Ranka, "Knowledge-based nonuniform crossover," *Proceedings of*  
507 *the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelli-*  
508 *gence*, 1994.
- 509 [37] L. Leclercq, J. A. Laval, and E. Chevallier, "The lagrangian coordinates and what it means for first order  
510 traffic flow models," in *Transportation and Traffic Theory 2007. Papers Selected for Presentation at ISTTT17,*  
511 2007.
- 512 [38] H. S. Mahmassani, "Dynamic network traffic assignment and simulation methodology for advanced system  
513 management applications," *Networks and spatial economics*, vol. 1, no. 3, pp. 267–292, 2001.
- 514 [39] Y. Hawas, H. Mahmassani, R. Taylor, A. Ziliaskopoulos, G.-L. Chang, and S. Peeta, "Development of  
515 dynasmart-x software for real-time dynamic traffic assignment," 1998.
- 516 [40] İ. Ö. Verbas and H. S. Mahmassani, "Finding least cost hyperpaths in multimodal transit networks: Method-  
517 ology, algorithm, and large-scale application," *Transportation Research Record*, vol. 2497, no. 1, pp. 95–105,  
518 2015.
- 519 [41] J. Krug, A. Burianne, and L. Leclercq, "Reconstituting demand patterns of the city of lyon by fusing multiple  
520 gis data sources," tech. rep., University of Lyon, ENTPE, LICIT, 2019.
- 521 [42] N. Alisoltani, L. Leclercq, M. Zargayouna, and J. Krug, "Optimal fleet management for real-time ride-sharing  
522 service considering network congestion," in *The 98th annual meeting of the Transportation Research Board,*  
523 *Washington, DC*, 2019.
- 524 [43] S.-G. Li and Y.-M. Su, "A multi-mode, multi-class dynamic network model with queues for advanced trans-  
525 portation information systems," *EJTIR*, vol. 5, no. 1, pp. 55–80, 2005.
- 526 [44] M. Patriksson, *The traffic assignment problem: models and methods*. Courier Dover Publications, 2015.
- 527 [45] H.-J. Huang and Z.-C. Li, "A multiclass, multicriteria logit-based traffic equilibrium assignment model under  
528 atis," *European Journal of Operational Research*, vol. 176, no. 3, pp. 1464–1477, 2007.
- 529 [46] H. Yang and X. Zhang, "Multiclass network toll design problem with social and spatial equity constraints,"  
530 *Journal of Transportation Engineering*, vol. 128, no. 5, pp. 420–428, 2002.
- 531 [47] K. Zhang, H. S. Mahmassani, and C. C. Lu, "Dynamic pricing, heterogeneous users and perception error:  
532 Probit-based bi-criterion dynamic stochastic user equilibrium assignment," *Transportation Research Part C:*  
533 *Emerging Technologies*, vol. 27, pp. 189–204, 2013.
- 534 [48] A. Nagurney and J. Dong, "A multiclass, multicriteria traffic network equilibrium model with elastic demand,"  
535 *Transportation Research Part B: Methodological*, vol. 36, no. 5, pp. 445–469, 2002.
- 536 [49] T.-Y. Hu, C.-C. Tong, T.-Y. Liao, and L.-W. Chen, "Dynamic route choice behaviour and simulation-based  
537 dynamic traffic assignment model for mixed traffic flows," *KSCE Journal of Civil Engineering*, vol. 22, no. 2,  
538 pp. 813–822, 2018.

- 539 [50] B. Ran, D.-H. Lee, and M. S.-I. Shin, “New algorithm for a multiclass dynamic traffic assignment model,”  
540 *Journal of Transportation Engineering*, vol. 128, no. 4, pp. 323–335, 2002.
- 541 [51] J. G. Yu and R. Jayakrishnan, “Multiclass, multicriteria dynamic traffic assignment with path-dependent link  
542 cost and entropy-based risk preference,” *Transportation Research Record*, vol. 2667, no. 1, pp. 108–118,  
543 2017.
- 544 [52] X. Xu, A. Chen, Z. Zhou, and L. Cheng, “A multi-class mean-excess traffic equilibrium model with elastic  
545 demand,” *Journal of Advanced Transportation*, vol. 48, no. 3, pp. 203–222, 2014.
- 546 [53] M. A. Abdel-Aty, R. Kitamura, and P. P. Jovanis, “Using stated preference data for studying the effect of ad-  
547 vanced traffic information on drivers’ route choice,” *Transportation Research Part C: Emerging Technologies*,  
548 vol. 5, no. 1, pp. 39–50, 1997.
- 549 [54] M. Florian, J. H. Wu, and S. He, “A multi-class multi-mode variable demand network equilibrium model with  
550 hierarchical logit structures,” in *Transportation and network analysis: Current trends*, pp. 119–133, Springer,  
551 2002.
- 552 [55] W. H. Lam and H.-J. Huang, “A combined trip distribution and assignment model for multiple user classes,”  
553 *Transportation Research Part B: Methodological*, vol. 26, no. 4, pp. 275–287, 1992.
- 554 [56] R. Garcia and A. Marin, “Network equilibrium with combined modes: models and solution algorithms,”  
555 *Transportation Research Part B: Methodological*, vol. 39, no. 3, pp. 223–254, 2005.
- 556 [57] M. Meng, C. Shao, Y. D. Wong, and J. Zhang, “A multiclass, multimodal dynamic traffic assignment model  
557 with departure time,” *Mathematical Problems in Engineering*, vol. 2014, 2014.
- 558 [58] M. Florian and C. D. Morosan, “On uniqueness and proportionality in multi-class equilibrium assignment,”  
559 *Transportation Research Part B: Methodological*, vol. 70, pp. 173–185, 2014.
- 560 [59] S. Ryu, A. Chen, and K. Choi, “Solving the stochastic multi-class traffic assignment problem with asymmetric  
561 interactions, route overlapping, and vehicle restrictions,” *Journal of Advanced Transportation*, vol. 50, no. 2,  
562 pp. 255–270, 2016.
- 563 [60] N. Zheng and N. Geroliminis, “Modeling and optimization of multimodal urban networks with limited parking  
564 and dynamic pricing,” *Transportation Research Part B: Methodological*, vol. 83, pp. 36–58, 2016.
- 565 [61] G. R. de Andrade, Z. Chen, L. Elefteriadou, and Y. Yin, “Multiclass traffic assignment problem with flow-  
566 dependent passenger car equivalent value of trucks,” *Transportation Research Record*, vol. 2667, no. 1,  
567 pp. 131–141, 2017.
- 568 [62] X. Pi, W. Ma, and Z. S. Qian, “A general formulation for multi-modal dynamic traffic assignment considering  
569 multi-class vehicles, public transit and parking,” *Transportation Research Part C: Emerging Technologies*,  
570 vol. 104, pp. 369–389, 2019.

## 571 **A Appendix**

### 572 **A.1 Literature review on travel cost definition in DTA problems**

573 In the literature, this classification process identifies eight categories of information about travelers:

- 574 1. Choice model: users are divided into different groups based on the behavioral model that they take into  
 575 account to choose their route and mode(s). Usually, utility functions are formulated to express the choices  
 576 of each class of users [43]. Random functions are used in some cases when there is less information about  
 577 users' choice model [44]. This classification considers the path (transportation mode included) choice model  
 578 and users' preferences.
- 579 2. Cost function: link cost or path cost can be different for each class of users, e.g. the toll of the links can  
 580 be different for each class. Also, path cost can be defined per user class [45]. In the dynamic case, the cost  
 581 function is variable in each time interval per user class.
- 582 3. Economic attributes: users are categorized into different classes based on their income [46]. Considering the  
 583 money budget for each class of users is another way to consider economic attributes.
4. Generalized cost function: the cost can be expressed in terms of travel time or monetary cost. The goal of  
 the generalized cost function is to propose a function which aggregates time and monetary cost:

$$GC = f(TT, TC) \quad (21)$$

Typically, the generalized cost function ( $GC$ ) integrates Travel Time ( $TT$ ) and Travel Cost ( $TC$ ).  $GC$  can take into account the class-dependent Value of time (VOT) [47]:

$$GC = TC + \alpha_i.TT \quad (22)$$

where  $\alpha_i$  is VOT for class  $i$ . Another approach to generalize the travel cost is to add a weighting parameter for cost summation [48]:

$$GC = w_i.TC + (1 - w_i).TT \quad (23)$$

584 where  $w_i$  is a weight parameter and  $0 < w_i < 1$  for class  $i$ .

- 585 5. Knowledge level of the network: users can be classified by their ability to access real-time traffic information  
 586 (ATIS). Each class of users attempts to optimize the travel cost based on their information from the network  
 587 status [49], [45] and [50].
- 588 6. Risk taking attitude: general speaking, risk, uncertainty, and reliability are three concepts that can be defined  
 589 for both sides: supplier (traffic network system) and demander (traveler) [51]. Reliability is defined as the  
 590 perceived probability of favorable outcome. Risk is defined as the expected value of the unfavorable outcome.  
 591 For traffic network as a supplier, uncertainty can be defined on demand. Travelers treat travel time variability  
 592 as a risk in their travel choices because it introduces uncertainty for an on-time arrival at the destination [51].  
 593 Using travel time budget is an approach to classify the users in order to deal with the arrival time uncertainty.  
 594 Therefore the late arrival time will be penalized [52]. Moreover, user cost function can be developed by Value  
 595 of Uncertainty (VOU) to consider the uncertainties of the travel [53].
- 596 7. Social class: to classify users into social classes does not make a difference in the equilibrium calculation.  
 597 The advantage is when we have an elastic demand and the goal is to have a better estimation of the demand  
 598 in traffic assignment problems [54].
- 599 8. Vehicle class: in multi-modal traffic network, the cost of travel depends not only on the traffic flow of one  
 600 mode but also on the traffic loads of other modes. Vehicle classes are defined in order to calculate the travel  
 601 time and cost based on mode attributes [55].

Table 5: The sample of different studies on multi-class traffic assignment in the Literature.

Research (year)	Traffic Network		Traffic Assignment		Multi modal	Multi-class							
	Flow based	Trip based	Static	Dynamic		Risk taking attitude	Economic class	General cost function	Knowledge level	Cost function	Choice model	Social class	Vehicle class
Lam and Huang (1992) [55]	x		x		x								x
Florian et al. (2002) [54]	x		x		x								x
Ran et. al (2002) [50]	x			x				x					
Yang and Zhang (2002) [46]	x		x										
Li and Su (2005) [43]	x			x				x					
Garcia and Marin (2005) [56]	x		x					x					x
Huang and li (2007) [45]	x		x					x					
Zhang et al. (2013) [47]	x		x					x					
Xu et al. (2014) [52]	x		x										
Meng et al. (2014) [57]	x			x				x					x
Florian and Morosan (2014) [58]	x		x										
Ryu et al. (2015) [59]	x		x										x
Zheng et al. (2016) [60]				x									
Jiang et al. (2016) [18]	x			x				x					x
Halat et al. (2016) [19]				x									
Xu et al. (2017) [20]				x									
Yu and Jayakrishnan (2017) [51]	x			x									
Andrade et al. (2017) [61]	x			x									
Zhan and Ukkusuri (2017) [23]	x		x										
Hu et al. (2018) [49]				x									
Pi et al. (2019) [62]	x			x									
Wang et al. (2019) [21]	x			x									
This Study (2020)				x									