

Research Article

Computational Methods for Calculating Multimodal Multiclass Traffic Network Equilibrium: Simulation Benchmark on a Large-Scale Test Case

Mostafa Ameli ,¹ Jean-Patrick Lebacque ,¹ and Ludovic Leclercq ,²

¹University Gustave Eiffel, COSYS, GRETTIA, Paris, France

²University Gustave Eiffel, ENTPE, LICIT, F-69518, Lyon, France

Correspondence should be addressed to Ludovic Leclercq; ludovic.leclercq@entpe.fr

Received 17 May 2020; Revised 17 December 2020; Accepted 28 January 2021; Published 19 February 2021

Academic Editor: Victor L. Knoop

Copyright © 2021 Mostafa Ameli et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study reviews existing computational methods to calculate simulation-based dynamic network equilibrium. We consider a trip-based multimodal 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 multiclass model with several parameters per class. Two different categories of algorithms (heuristic and metaheuristic) are considered in order to solve the discrete dynamic traffic assignment (DTA) problem. Finally, we analyze the equilibrium in a large-scale multimodal 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 metaheuristic 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.

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 routes 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 large numbers 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 path flow distribution [4].

This combination, for large-scale networks, allows us to consider sophisticated traffic models in order to incorporate disaggregate information into the estimates of travel costs [5, 6].

Traffic simulators can be divided into two classes: flow-based models, which consider the flow of each path, and trip-based models, which define how many travelers take each path. Macroscopic traffic flow models fall into the first category, while microscopic models belong to the second. The macroscopic approach and flow-based models are usually fast in equilibrium calculation as the path flow discipline is more flexible (flows are not necessarily equivalent to traveler units). However, they are less realistic for OD pairs without adding integrality constraints with low demand as vehicles are split into small particles in practice. In this study, we decide to focus on the trip-based approach in which each traveler is reproduced individually. The trip-based approach can be considered as a multiagent system, wherein each trip impacts the system by its route (including path and mode(s)) choice. Microscopic traffic simulators are

now widely used for operational studies, and we have chosen to focus on DTA performance for this kind of model. Trip-based DNL attempts to assign particle-discretized time-dependent origin/destination flows in a dynamic network equilibrium framework [7]. The goal of this study is to design and solve a large-scale trip-based traffic network equilibrium model.

The user equilibrium situation, theoretically, is the solution of a fixed-point problem [8]. From a practical point of view, it means that each user perceives no delay or extra cost compared to other users with the same origin and destination [9]. This refers to minimizing the total gap between users' travel cost of each OD pair and the corresponding shortest path (the path(s) with minimum travel cost) travel cost [10]. In the context of DTA, the Method of Successive Average (MSA) algorithm remains by far the most widely used solution method in simulation-based DTA [11]. The simplicity of the MSA implementation and the nonrequirement of derivative information are the main reasons for its widespread use [12]. However, Yang and Jayakrishnan [13] showed that the convergence is not guaranteed to be the user equilibrium solution with the traditional MSA algorithm. Based on this indication, several heuristic approaches are proposed to improve the determination of the step size (e.g., [14–16]). Ameli et al. [17] did a benchmark on most of the heuristic methods in the literature. Here, in addition to heuristic algorithms, we also considered metaheuristic approaches (proposed by [18]) in order to compare different methods to solve the DTA problem.

The travel cost is the criterion for each user to select the travel route and mode(s). It is important to consider generalized travel costs in simulation-based DTA in order to include different features of travelers' behavior. Generally, travelers consider different criteria in selecting their optimal properties of travel, including travel route and travel mode(s). Thus, the total demand of the system can be clustered into different user classes based on the definition of their travel cost function. Many attempts have been made on modeling travelers' behavior in the traffic network. Most multiclass flow models classify travelers by identifying traveler attributes. In the literature, this classification process identifies eight categories of information about travelers: (1) choice model, (2) cost function, (3) economic attributes, (4) generalized cost function, (5) knowledge level of the network, (6) risk taking attitude, (7) social class, and (8) vehicle class. Moreover, some research uses hybrid classification (see, e.g., [19–21]). In this study, we attempt to define a generic cost function for users in order to represent their behavior accurately. For the convenience of the reader, we discuss all categories and identify the different approaches in Appendix A (Table 1) in order to clarify our contribution compared to the previously mentioned papers.

Multiclass DTA models are widely investigated in order to take into account the heterogeneity of the users [22]. While the model becomes more accurate for representing the users' travel cost function, finding the equilibrium becomes more difficult computationally. There are many studies about developing algorithms to find user equilibrium [9]. This study aims to find out which category of solution

algorithms is more efficient for the multiclass DTA problem. There are two main categories of solution methods to solve simulation-based DTA with computational methods: heuristic methods based on fixed-point theory and metaheuristic methods that have been recently introduced [18]. In the literature, these methods are applied individually to networks with different sizes and different levels of details about users and network dynamics [23]. In this study, we apply solution methods with the multiclass model to a real test case (Lyon 6th + Villeurbanne: about 17 km²) in order to benchmark all the solution methods. The goal is to answer which solution method is the most efficient in generalized and advanced numerical simulation settings.

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

2. Dynamic Traffic Assignment Model

2.1. Multiattribute Generalized Travel Cost. In order to describe the traveler's behavior, a generalized cost function that takes into account four types of information mentioned in Section 1: (2), (3), (4), and (7), is used. Typically, the generalized cost (GC) function will integrate travel time (TT) and travel cost (TC). Also, GC will take into account the class-dependent value of time (VOT). Zhan and Ukkusuri [24] denoted α_i for class i and a vector β_i of economic and social class parameters. Therefore, we need to define, for class i and path p ($p \in P$: set of paths), the path generalized cost $GC_p(\alpha_i, \beta_i)$.

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

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

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

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

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

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

where a is the link index ($a \in A$: set of links in the network), π_a^i is the number of users of class i on link a , and $\delta_{ap} = 1$ if path p contains link a and 0 otherwise. The travel time and

TABLE 1: List of notations used in this article.

Equilibrium model:	
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 τ .
$P_{w,\tau}^i$	Set of shortest paths for OD pair w and class i in departure time interval τ .
i	Index of user class, $i \in I$.
τ	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^*	Index of shortest path for class i , $p^* \in P_{w,\tau}^i$.
$\pi_{w,p,\tau}^i$	Number of users in class i from OD pair w , assigned to path p in departure time τ .
$\pi_{w,p,\tau}^{i*}$	Optimal number of users in class i from OD pair w , assigned to path p in departure time τ .
$GC_{p,\tau}(\alpha_i, \beta_i)$	Generalized cost of path p for user class i in departure time τ .
$GC_{w,\tau}^i$	Minimum generalized cost of OD pair w for user class i in departure time τ .
Optimization algorithms:	
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.
$\pi_{p,k}^{i,k}$	The number of users in class i on path p in iteration k .
P^{i*}	The set of shortest paths 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 k .
$AGap(s)$	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_α	The threshold for the exploration rate of the inner loop.
Gap_β	The threshold for the convergence rate of the outer loop.

cost functions depend on the congestion of the link. Therefore, both functions are calculated by the following formulas:

$$\begin{aligned} TT_a &= TT_a(\pi_a), \\ TC_a^i(\beta_i) &= TC_a^i(\beta, \pi_a), \end{aligned} \quad (5)$$

and, consequently, the link generalized cost function is calculated as follows:

$$GC_a(\alpha_i, \beta_i) = TC_a(\beta, \pi_a) + \alpha \cdot TT_a(\pi_a). \quad (6)$$

Equation (6) calculates the generalized cost of each link for each class of users. Finally, the cost of path p is calculated by the following equation:

$$GC_p(\alpha_i, \beta_i) = \sum_{a \in A} GC_a(\alpha, \beta) \delta_{ap}. \quad (7)$$

Now, we can build a DTA model based on the generalized cost defined by equation (7).

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 τ ($\tau \in T = \{\tau_0, \tau_0 + \sigma, \tau_0 + 2\sigma, \dots, \tau_0 + M\sigma\}$) and

$\tau_0 + M\sigma = H$). σ is the duration of the time intervals. In an interval τ , travel time and traffic conditions do not change. The important notations to introduce the dynamic equilibrium model and optimization algorithms are as follows.

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

$$\begin{cases} GC_{p,\tau}(\alpha_i, \beta_i) - GC_{w,\tau}^i \geq 0, \\ \pi_{p,\tau}^i (GC_{p,\tau}(\alpha_i, \beta_i) - GC_{w,\tau}^i) = 0, \\ \pi_{p,\tau}^i \geq 0. \end{cases} \quad (8)$$

Lu et al. [10] reformulated the problem as a nonlinear 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:

$$\begin{aligned} \sum_{w \in W} \sum_{\tau=1}^T \sum_{i=1}^I \sum_{p \in P_{w,\tau}} GC_{w,p,\tau}^i (\pi_{w,p,\tau}^i - \pi_{w,p,\tau}^{i*}) &\geq 0, \\ \forall \pi_{w,p,\tau}^i, \pi_{w,p,\tau}^{i*} \in \mathcal{C}, \end{aligned} \quad (9)$$

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 assignment, if

Equation (9) holds for all feasible $\pi_{w,p,\tau}^i$. In Equation (9), both $\pi_{w,p,\tau}^i$ and $\pi_{w,p,\tau}^{i*}$ are decision variables and hence the gap function is a function of both variables. Existence of solution(s) to Equation (9) requires the continuity of the generalized cost functions of paths with respect to the numbers of users on paths. In this case, the unicity of the solution requires the strict monotonicity of the OD paths generalized cost functions with respect to OD path flows (i.e., the number of cars) [25]. In general urban transportation networks, these conditions do not hold [22].

3. Solution Algorithm to Solve Trip-Based Dynamic Traffic Assignment

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

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

Figure 1 presents the optimization framework based on the column generation approach. The framework separates the three costly steps in order to minimize the number of shortest path calculations which is very costly in large-scale traffic networks. In other words, the idea of the column generation approach is based on the generation of the set of paths to which users may be assigned. Classical approach starts from an empty set and augments it in every iteration as required. The column generation approach contains two loops: outer loop and inner loop. The outer loop is responsible for path discovery, while the inner loops implement the path flow optimization for a given path set. Note that the classic approach executes both steps in a single loop. In large-scale network problems, it is extremely costly to keep the data of all the possible paths between each OD pair, so, the column generation approach definitely reduces memory requirements [26]. Lu et al. [10] implement and examine the column generation approach in DTA context and show that it not only reduces memory requirements but

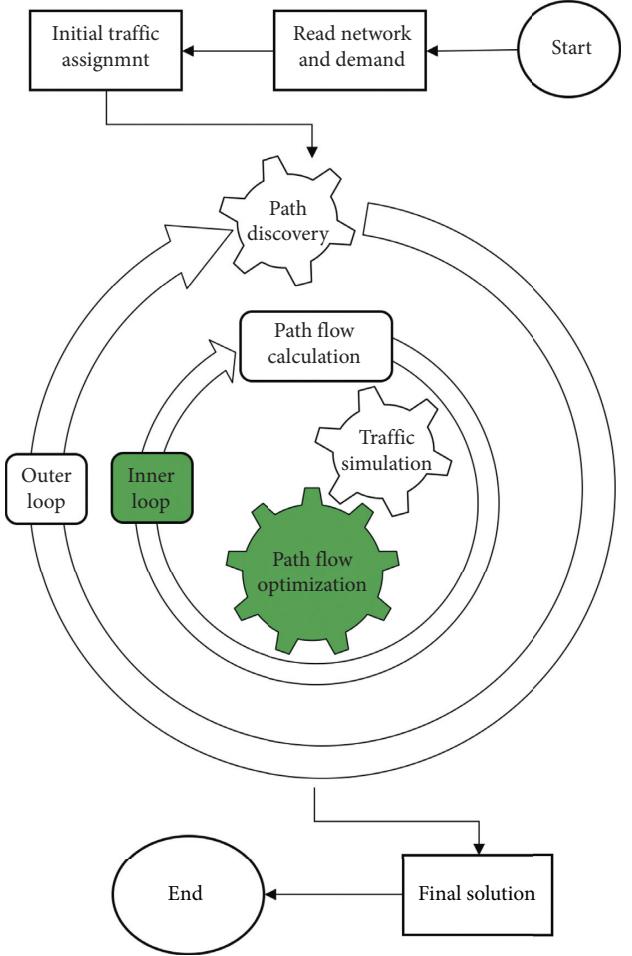


FIGURE 1: Simulation-based DTA process: column generation approach.

also outperforms other algorithms in convergence with a designed swapping algorithm embedded in the inner loop.

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

The total gap as a quality indicator in time interval τ is

$$\text{TGap } (\pi, \text{GC}^*) = \sum_{w \in W} \sum_{i=1}^k \sum_{p \in P_{w,i}} \pi_{w,p}^i (\text{GC}_{w,p}^i - \text{GC}_{w,p}^{i*}). \quad (10)$$

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

$$\text{AGap}(\pi, \text{GC}^*) = \frac{\sum_{w \in W} \sum_{\tau=1}^T \sum_{i=1}^I \sum_{p \in P_{w,\tau}} \pi_{w,p,\tau}^i (\text{GC}_{w,p,\tau}^i - \text{GC}_{w,p,\tau}^{i*})}{\sum_{w \in W} \sum_{\tau=1}^T \sum_{i=1}^I \sum_{p \in P_{w,\tau}} \pi_{w,p,\tau}^i}. \quad (11)$$

The total gap (Equation (10)) and the first quality indicator (Equation (11)) are calculated for each solution with the updated shortest path. The perfect user equilibrium state has zero AGap; thus, the best optimization algorithm obtains minimum AGap. In other words, AGap measures the distance between the algorithms' solution and UE. The second indicator is the assignment violation. The violation indicator is defined in order to count users assigned on nonoptimal path(s). The following steps calculate the violation indicator:

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

$$\text{UV}_w^i = \begin{cases} 1; & \text{if } \frac{\text{ETT}_w^i - \text{TT}_w^*}{\text{TT}_w^*} \geq 0.1, \\ 0; & \text{o.w.,} \end{cases} \quad (12)$$

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

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

$$\text{ODV}_w = \begin{cases} 1; & \text{if, } \frac{\sum_{i \in I_w} \text{UV}_w^i}{|I_w|} \geq 0.1, \\ 0; & \text{o.w.,} \end{cases} \quad (13)$$

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

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

$$V(G) = \frac{\sum_{w \in W} \text{ODV}_w}{|W|}. \quad (14)$$

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

The framework has two convergence tests to stop the calculation process in order to save computation time. In

both steps, first, the iteration index is checked with the maximum number of loops (j_{\max} and k_{\max}) that we fix for each loop. Suppose that it is not reached ($j < j_{\max}$ or $k < k_{\max}$); then we check the solution's quality at the current loop, but there are three differences between the outer-loop convergence test and inner-loop convergence test. First, in the outer loop, we calculate the quality indicator (Equation (11)) based on the new shortest path(s), but the path set is fixed during the inner-loop operations, and only the shortest path is updated in each iteration. Second, in the inner loop, we calculate the relative total gap (i.e., the difference of the total gap between two iterations) to check the improvement of the process and compare it with Gap_β . Gap_β is a threshold for checking the exploration rate of the optimization algorithm in the inner loop. But, in the outer loop, the goal is to monitor the quality of the solution. Therefore, AGap is calculated and compared with the fixed value (Gap_α). If we have a solution with AGap equal to or less than threshold Gap_α , we terminate the algorithm. Third, in the outer loop, we have one more condition other than checking the solution's quality. The condition is the shortest path checking. If we have no new shortest path per OD and good quality of the solution, then the process is converged. It means we will continue the process when the quality of the solution is not good until we no longer have a new shortest path.

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

3.1. Heuristic Methods. Generally, the heuristic methods are path-swapping descent direction methods that decompose the problem into many (origin-destination, departure time interval, and user class) subproblems. They solve each of the subproblems by adjusting time-varying OD flows between all nonshortest paths and the shortest path(s). We did a cross-comparison of heuristic algorithms in [17] with homogeneous users. Here we present the best algorithm for a large scale, which may have a good performance in multi-class and multimodal contexts.

3.1.1. Method of Successive Average with Ranking Method. The first method is an extension of the Method of Successive Average (MSA). The MSA algorithm is the classical method to solve the traffic assignment problem [27] and it is widely used in theory and application for DTA problems [28]. Equation (15) presents the swapping formula of MSA algorithm [29]:

$$z^k = \left(\frac{k}{k+1} \right) z^{k-1} + \frac{1}{k+1} y^k. \quad (15)$$

where y^k is extracted from the auxiliary path assignments obtained by the all-or-nothing procedure; that is, everyone is placed on the shortest path. Note that the step size in this method is $1/(k+1)$. It means that, during iteration k for this method, a fraction $1/(k+1)$ of users on each nonshortest path swaps to the one of the shortest path in the current

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 ($\text{AGap}(z_j) \geq \text{Gap}_{\alpha}$ **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 $\text{AGap}(z^0) = \text{AGap}(z_j) + 2 \times \text{Gap}_{\beta}$; **While** $k < k_{\max}$ and $|\text{AGap}(z^k) - \text{AGap}(z^{k-1})| > \text{Gap}_{\beta}$ **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.**

ALGORITHM 1: Simulation-based DTA: equilibrium calculation.

iteration. MSA ranking adds a ranking method to choose the users from nonshortest paths to the shortest path(s). In other words, it ranks all users by their generalized cost and swaps the users with higher GC to the shortest path(s). The step size is the same as the MSA method equal to $1/(k+1)$ of all users on nonshortest paths.

3.1.2. Gap-Based Method. To enable the deployment of large-scale (simulation-based) DTA models, [10] 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 ($\text{GC}_{p^{i*}}$):

$$\begin{cases} \pi_p^{i,k+1} = \max \left\{ 0, \pi_p^{i,k} \cdot \left[1 - \rho^k \frac{\text{GC}_p^i - \text{GC}_{p^{i*}}}{\text{GC}_p^i} \right] \right\}; & \text{if } p \notin P^{i*} \\ \pi_p^{i,k+1} = \pi_p^{i,k} + \frac{\sum_{p \in P_u - P^{i*}} [\pi_p^{i,k} \cdot \rho^k ((\text{GC}_p^i - \text{GC}_{p^{i*}})/\text{GC}_p^i)]}{|P^{i*}|}; & \text{if } p \in P^{i*}, \end{cases} \quad (16)$$

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

where j denotes the outer-loop iteration index, k denotes the inner-loop iteration index, and $|P^{i*}|$ is the number of the shortest paths in iteration k . This path assignment updating scheme implies a natural path flow adjustment mechanism:

flows on the nonshortest paths are moved to the shortest path and the volume that moved out from a nonshortest path is proportional to the gap between the nonshortest path and shortest path over nonshortest path cost.

MSA ranking and gap-based method have two main drawbacks which are also the same for all heuristic methods that come from the flow-based traffic assignment:

Using the step size guarantees the algorithm to converge but decreases the flexibility of the method to swap more users to the shortest path(s). Moreover, the final solution is obtained based on the initialization (all-or-nothing) because the algorithm converges (by decreasing the step size) after a while even we use a not good initial assignment which is not really obvious in the large scale.

Many of the methods in the literature (see, e.g., [8, 11, 12, 15, 16, 28, 30–32]) are using the average travel time of each path to calculate the indicators and to distinguish the shortest path(s) and other paths per OD. This means they do not consider users' experienced travel time directly. Moreover, the optimizer is applied for each OD and swap users based on the different values of the paths' average travel time. Note that users in the same departure time interval and on the same path can perceive different travel time values because of the network dynamic evolution.

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

3.1.3. Probabilistic Method. The probabilistic method is a trip-based approach. This method has no step size for optimization [17]. For each user, the probability of swap to the shortest path(s) is calculated by the following formula:

$$\text{Swap indicator} = \frac{\text{EGC}_p - \text{GC}_{p^i}^*}{\text{GC}_p}, \quad (17)$$

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

indicator (the possibility of swapping) for the next inner loop.

3.1.4. Hybrid Methods. Verbas et al. [33] applied an extension of the gap-based method to transit assignment. The method uses Equation (16) to determine the number of swapping users for each path and then the probabilistic method is applied to choose users from each path for swapping. This method is called "gap-based prob." We examined this method in DTA test cases, and it fully dominates the gap-based method in different monoclass test cases [17]. Therefore, we consider this method in our numerical experiments.

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

$$\text{Swap indicator} = \frac{1}{k+1} \frac{\text{EGC}_p - \text{GC}_{p^i}^*}{\text{GC}_p}. \quad (18)$$

This method is called "step-size prob." in this study. It has a good performance in large-scale and congested traffic networks [20]. We developed a smart step-size selection procedure for this method in which we specialize the step size for each OD pair and decrease the step size when the OD gap is not improved compared to the previous iteration [17]. The method is called "smart step-size prob." All the methods that are presented in this section will apply to multiclass multimodal DTA problem in order to evaluate their performance and also compare them with metaheuristic methods.

3.2. Metaheuristic Methods. As discussed in the previous section, in iteration k , the classical (heuristic) algorithms try to improve the path flow distribution by swapping a fraction $1/(k+1)$ of users to the shortest path(s) from each non-shortest path. Then, one simulation is launched based on the updated path flow distribution. We have to wait until the simulation run is finished to know the new link travel times and adjust the path flow distribution to be tested in the next iteration accordingly. This is the serial process of the classical algorithms, which limits the solution space exploration and computational process. We developed two metaheuristic algorithms in [18] in order to overcome the problem of the serial approach. Two parallelized metaheuristic approaches are designed: the first derived from the simulated annealing framework and the second derived from that of the genetic algorithm.

3.2.1. Simulated Annealing Algorithm. The algorithm starts with an initial solution generated randomly; that is, users choose their path from the OD path set randomly. For solution z , the total gap $T\text{Gap}(.)$ between the users' travel cost and the shortest path travel cost (Equation (10)) is considered as the energy of the solution. A set of neighbor solutions are generated with respect to the current one based on the temperature (T) of the current iteration. A neighbor solution is defined as a candidate to replace the current solution. The current phase of the iteration depends on the

temperature of the process. Inspired by the physics of matter, this study distinguishes three different methods to generate a neighbor solution, gas, liquid, and solid; these methods represent the states of matter in nature. When the temperature is high ($T > \lambda$ where λ denotes the boiling temperature), the *gas method* is applied. When running the SA algorithm, by decreasing the temperature, the algorithm enters the liquid phase ($\lambda > T > \lambda'$ where λ' denotes the melting temperature) and then the *liquid method* is applied. When the temperature is quite low ($T < \lambda'$), the *solid method* is applied. The algorithm is presented in Algorithm 2.

3.2.2. Genetic Algorithm. The genetic algorithm (GA) (first formalized by [34]) is inspired by natural selection and genetic variation. The genetic algorithm includes three main operators: selection, crossover, and mutation. The description of the operators is well presented in [35]. The GA framework is adapted to the traffic assignment problem in [18]. The algorithm is designed as a two-layer GA process to search solution space by changing the path flows in **GA Inner** and overcome the drawback of OD impact by considering different combinations of OD assignments in **GA-operators**. In other words, the heuristic algorithms plus a random method are applied in **GA Inner**, and the GA operators in one upper level generate different combinations of OD. **GA Inner** is a new layer in the optimization process compared to the classical GA. It has been added to account for correlations between OD assignments. Moreover, GA considers a set of solutions instead of a single solution at every iteration. The steps of the GA are detailed in Algorithm 3.

4. Numerical Experiments

The multiclass model is applied to a large-scale multimodal network of Lyon 6th + Villeurbanne. This network that covers 25 km² and includes 1,883 nodes, 3,383 links, 94 origins, and 227 destinations is located in Lyon Metropolis and 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 (an open source simulator (<https://github.com/Ifsttar/Open-SymuVia>)) 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 [38]. 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 Equation (6) 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 transportsations. 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 and 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 travel time calculation. The users' routes are determined by the DTA model and the rolling horizon technique [39], which determine the path flow distribution based on a prediction period of 30 minutes and an assignment period of 15 minutes. The multimodal time-dependent shortest path algorithm is implemented based on the multiorigin Dijkstra algorithm [40] with considering hyperpaths based on [41] in order to find minimum cost path(s) in terms of travel time.

The network is loaded with 54,190 travelers of all ODs with given departure times in order to represent the 2.5 morning peak hours of the network between 6 : 30 and 9 : 00. The demand profile comes from the study of [42]. Note that, here, we fixed the departure time in order to focus on travelers' path choices. The main reason is that this study aims to compare the performances of DTA solvers. By fixing the departure time, the input of the problem will be the same for all algorithms. Therefore, the solution space is fixed and given for the route (path and mode) choice problem, and we can compare the performances of different solution algorithms.

Figure 3(a) presents the demand profile of the numerical experiment. The figure presents the inflow rate of the trips to the traffic network. The total length of the Lyon 6th + Villeurbanne network is 496.73 kilometers. Note that each OD has a time-dependent demand profile based on the real data of Lyon city [42]. The dynamic loading represents the saturation state of the network. To show a quick and synthetic overview of the network state, we plot the Macroscopic Fundamental Diagrams (MFD) of the network in Figure 3(b). This diagram represents the overall evolution of the traffic conditions in the network by plotting the total travel distance versus the total travel time [43]. Note that total travel distance is proportional to the mean network flow, while total travel time is equivalent to vehicle accumulation.

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

The trips are divided into nine classes that are presented in Table 2. In this table, the user class's fraction of the total demand is presented by "percentage of trips," and, for each class, we use a specific VOT. The demand level and VOT

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 $\text{AGap}(z^0) = \text{AGap}(z^{-1}) = 2\text{AGap}(z_j)$;
While $k < k_{\max}$ **and** $\text{AGap}(z^k) \neq \text{AGap}(z^{k-1}) \neq \text{AGap}(z^{k-2})$ **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(.) based on updated shortest path(s) for all candidates;
 Calculate the solution acceptance probability for each candidate by the following equation:

$$P(S_{\text{accepted}}' = 1) = e^{-\nabla E/T}$$
 where s' denotes a candidate, S_{accepted}' denotes the binary decision variable and $\nabla E = \text{TGap}(s') - \text{TGap}(s)$;
 Take decision about accepting each candidate to determine z^{k+1} ;
 Decrease the temperature by the following formula:

$$T = T_0 / \ln(k + 1);$$
 Set $k = k + 1$;
end
 Produce the final solution for the outer loop.

ALGORITHM 2: Simulated annealing 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 [36], 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 [37]. 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:
 $MP_w = \text{TGap}_w(s) / \text{TGap}(s)$
 where $\text{TGap}_w(s)$ 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 [18];
 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(.) 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.

ALGORITHM 3: Genetic algorithm.

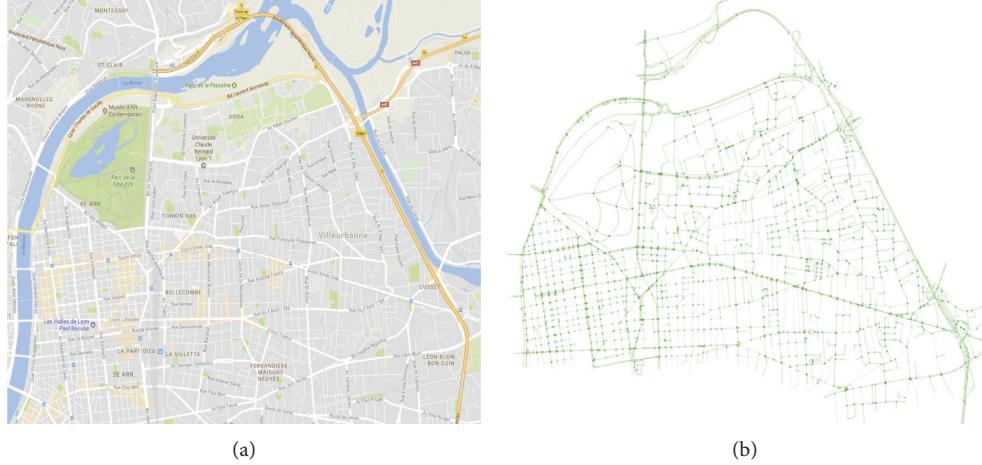


FIGURE 2: Lyon 6th + Villeurbanne: (a) mapping data ©Google 2020; (b) the traffic network considered in Symuvia.

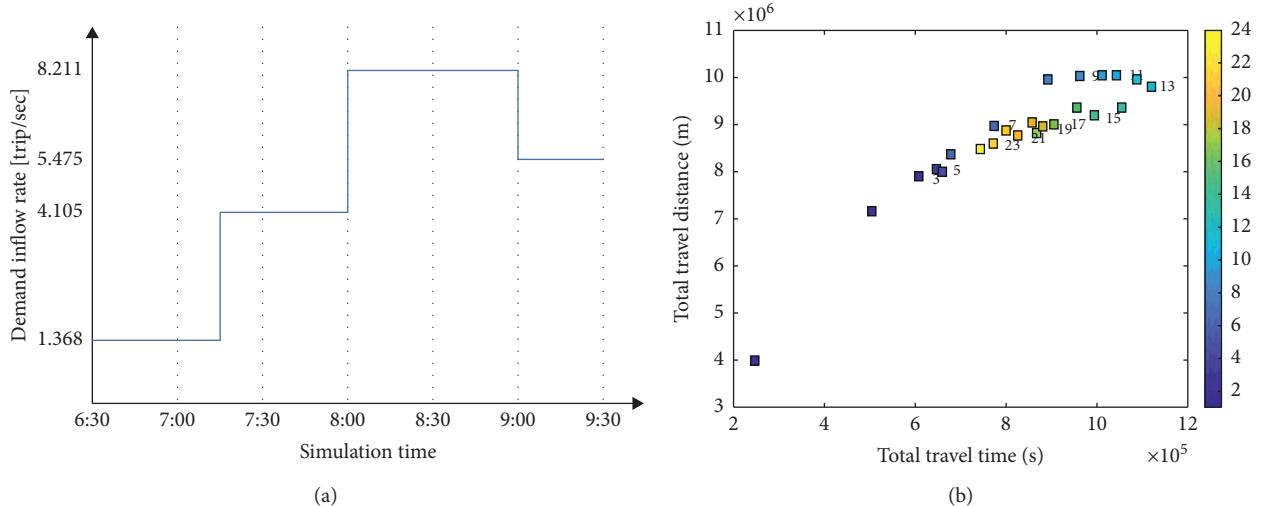


FIGURE 3: The demand scenario of Lyon 6th + Villeurbanne. (a) The demand profile. (b) The macroscopic fundamental diagram from 7:00 to 9:00.

values are extracted from the real data of the full Lyon networks [42]. The optimization goal is to find the user equilibrium based on the generalized cost functions.

The optimization process is executed for a maximum of 10 outer loops for five assignment periods. At each outer-loop iteration, new shortest path(s) can be discovered; therefore, if at least a new shortest path is found per outer-loop iteration, we have 11 paths per OD. In each outer loop, a maximum of 40 inner loops have been executed to find the best assignment pattern. Note that, in heuristic and SA algorithms, there is one more simulation per outer loop for keeping the best assignment pattern if the process is converged by the maximum number of the inner loops. This means that if the convergence does not happen during all inner loops, we take the best solution and simulate the network to prepare the result for the outer-loop convergence test. The convergence tests of two loops are fixed for all algorithms. In the inner-loop convergence test, the optimization will be stopped when we have less than 1%

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

improvement ($\text{Gap}_\beta = 0.01$). Improvement is related to the average gap between the current and previous inner 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 have a small

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

value of gap or have no improvement in terms of total gap ($\text{Gap}_\alpha = 0$). In the next section, we will compare the performances of heuristic and metaheuristic algorithms in the multiclass multimodal DTA problem based on the numerical results.

5. Numerical Results and Discussion

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

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

Regarding the percentage of incomplete trips, also the metaheuristic algorithms dominate the others. The incomplete trips indicator denotes the share of trips, which is not finished by the end of the simulation in the final path flow

distribution of each algorithm. A lower number of incomplete trips means a lower total travel time spent in the system over the simulation period. The SA algorithm finds the closest solution to UE (minimum AGap) in this study. Moreover, the final solution of SA has the best value for other quality indicators that are highlighted in bold in the SA row in Table 3.

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

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

Figure 4 presents the convergence pattern for all seven algorithms in the last assignment period. The convergence

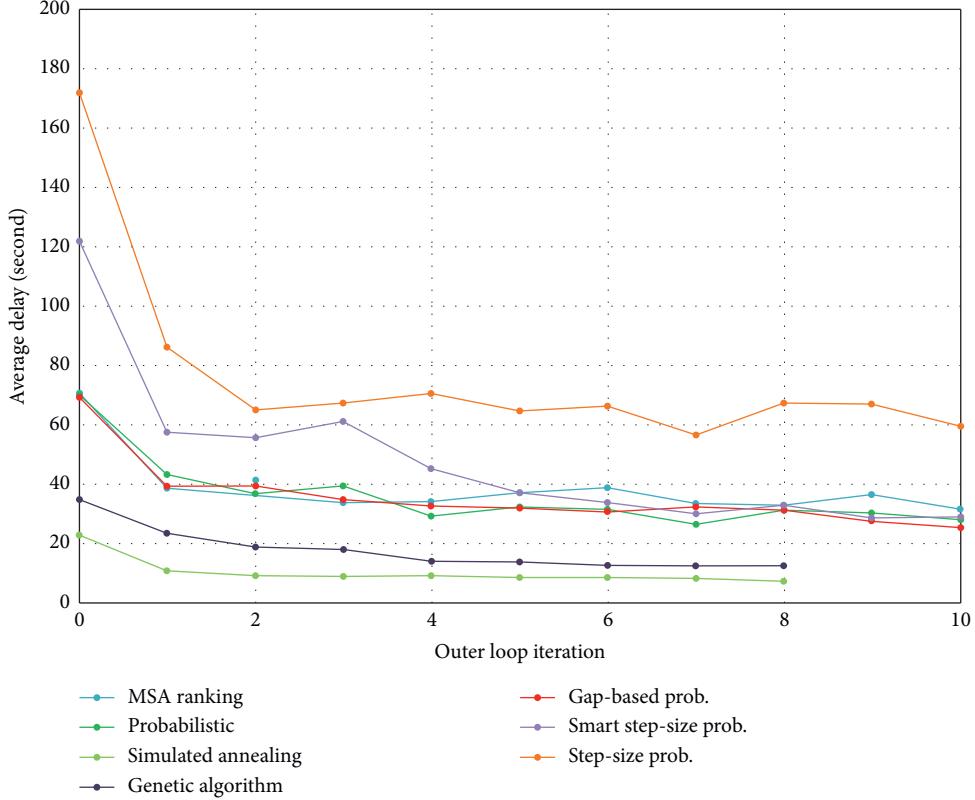


FIGURE 4: Convergence patterns for the outer loops.

TABLE 4: Comparison between the performances of algorithms with homogeneous and heterogeneous users.

Demand profile; quality indicator/ method	Monoclass users				Multiclass users			
	AGap (euros)	Improvement to AGap compared to prob.	CT (hours)	Improvement to CT compared to smart SSP	AGap (euros)	Improvement to AGap 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%

pattern is drawn based on the average delay of all users. By the best heuristic methods, the final user average gap is around 30 sec, but this value for the metaheuristic method is around 10 sec. It means that the metaheuristic methods also obtain the better solutions in terms of delay. In Figure 4, the convergence pattern of all heuristic algorithms first decreases and then variates in a small range for the last iterations, where they seem to stick to the local optimum. Metaheuristic algorithms' convergence pattern is always decreasing, which means that the algorithms are exploring

and looking for the global optimum. Therefore, both metaheuristic algorithms produce not only better solutions in terms of optimality but also better (i.e., not increasing) convergence patterns than classical algorithms.

In order to consider the complexity of the problem on the performance of algorithms, a monoclass demand scenario is also designed. The volume of the demand and the traffic network are the same as those in the multiclass scenario. For monoclass users, we consider the weighted average VOT of the multiclass scenario (15.84 euros per

TABLE 5: The sample of different studies on multiclass traffic assignment in the literature.

Research (year)	Traffic network			Traffic assignment			Multimodal			Multiclass		
	Flow-based	Trip-based	Static	Dynamic	Economic	Risk taking	General cost function	Knowledge level	Cost function	Choice model	Social class	Vehicle class
Huang and Li [46]	×	×	×	×	×	×	×	×	×	×	×	×
Zhang et al. [48]	×	×	×	×	×	×	×	×	×	×	×	×
Xu et al. [53]	×	×	×	×	×	×	×	×	×	×	×	×
Meng et al. [57]												
Florian and Morosan [58]	×											
Ryu et al. [59]	×	×	×	×	×	×	×	×	×	×	×	×
Zheng and Geroliminis [60]	×	×	×	×	×	×	×	×	×	×	×	×
Jiang et al. [19]	×	×	×	×	×	×	×	×	×	×	×	×
Halat et al. [20]	×	×	×	×	×	×	×	×	×	×	×	×
Xu et al. [21]												
Yu and Jayakrishnan [52]	×											
Andrade et al. [61]	×											
Zhan and Ukkusuri [24]	×											
Hu et al. [50]	×											
Pi et al. [62]	×	×	×	×	×	×	×	×	×	×	×	×
Wang et al. [22]												
This study (2021)	×						×	×	×	×	×	×

hour), extracted from Table 2, and the same monetary cost for public transportation. The setting for the optimization framework, including the maximum inner- and outer-loops' iterations, is fixed to the same values as the multiclass scenario. Table 4 presents the monoclass scenario results with the multiclass scenario. Same as the previous experiment, metaheuristic algorithms dominate heuristic algorithms. The results show that heuristic algorithms cannot guarantee the user equilibrium for the large-scale network.

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

6. Conclusion

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

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

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

Appendix

A. Literature Review on Travel Cost Definition in DTA Problems

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

- (1) Choice model: users are divided into different groups based on the behavioral model that they take into account to choose their route and mode(s). Usually, utility functions are formulated to express the choices of each class of users [44]. Random functions are used in some cases when there is less information about users' choice model [45]. This classification considers the path (transportation mode included) choice model and users' preferences.
- (2) Cost function: link cost or path cost can be different for each class of users; for example, the toll of the links can be different for each class. Also, path cost can be defined per user class [46]. In the dynamic case, the cost function is variable in each time interval per user class.
- (3) Economic attributes: users are categorized into different classes based on their income [47]. Considering the 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 that aggregates time and monetary cost:

$$GC = f(TT, TC). \quad (A.1)$$

Typically, the generalized cost (GC) function integrates travel time (TT) and travel cost (TC). GC can take into account the class-dependent value of time (VOT) [48]:

$$GC = TC + \alpha_i \cdot TT, \quad (A.2)$$

where α_i is VOT for class i . Another approach to generalize the travel cost is to add a weighting parameter for cost summation [49]:

$$GC = w_i \cdot TC + (1 - w_i) \cdot TT, \quad (A.3)$$

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

- (5) Knowledge level of the network: users can be classified by their ability to access real-time traffic information (ATIS). Each class of users attempts to optimize the travel cost based on their information from the network status [46, 50, 51].
- (6) Risk taking attitude: generally speaking, risk, uncertainty, and reliability are three concepts that can be defined for both sides: supplier (traffic network system) and demander (traveler) [52]. Reliability is defined as the perceived probability of favorable outcome. Risk is defined as the expected value of the unfavorable outcome. For traffic network as a supplier, uncertainty can be defined on demand. Travelers treat travel time variability as a risk in their travel choices because it introduces uncertainty for an on-time arrival at the destination [52]. Using travel time budget is an approach to classify the users in order to deal with the arrival time uncertainty. Therefore, the late arrival time will be penalized [53]. Moreover, user cost function can be developed by Value of Uncertainty (VOU) to consider the uncertainties of the travel [54].
- (7) Social class: classifying users into social classes does not make a difference in the equilibrium calculation. The advantage is when we have an elastic demand, and the goal is to have a better estimation of the demand in traffic assignment problems [55].
- (8) Vehicle class: in multimodal traffic network, the cost of travel depends not only on the traffic flow of one mode but also on the traffic loads of other modes. Vehicle classes are defined in order to calculate the travel time and cost based on mode attributes [56].

Table 5 presents a comparison between different studies in the literature and this study on multiclass traffic assignment.

Data Availability

No data were used for this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work received funding from the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Program (Grant agreement no. 646592 – MAGnUM project).

References

- [1] J. G. Wardrop, "Road paper. some theoretical aspects of road traffic research," *Proceedings of the Institution of Civil Engineers*, vol. 1, no. 3, pp. 325–362, 1952.
- [2] R. Jayakrishnan, W. K. Tsai, and A. Chen, "A dynamic traffic assignment model with traffic-flow relationships," *Transportation Research Part C: Emerging Technologies*, vol. 3, no. 1, pp. 51–72, 1995.
- [3] T. Schreiter, V. Wageningen-Kessels, Y. Yuan, J. Van Lint, S. Hoogendoorn et al., "Fastlane: traffic flow modeling and multi-class dynamic traffic management," in *Proceedings of the Trail-Beta Congress 2012: Mobility and Logistics-Science Meets Practice*, pp. 30–31, Rotterdam, The Netherlands, October 2012.
- [4] W. Song, K. Han, Y. Wang, T. Friesz, and E. Del Castillo, "Statistical metamodeling of dynamic network loading," *Transportation Research Procedia*, vol. 23, pp. 263–282, 2017.
- [5] X. Xu, A. Zockaei, H. S. Mahmassani et al., "Schedule consistency for daily activity chains in integrated activity-based dynamic multimodal network assignment," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2664, no. 1, pp. 11–22, 2017.
- [6] N. Alisoltani, M. Zargayouna, and L. Leclercq, "A sequential clustering method for the taxi-dispatching problem considering traffic dynamics," *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 4, pp. 169–181, 2020.
- [7] R. Jayakrishnan and C. R. Rindt, "Distributed computing and simulation in a traffic research test bed," *Computer-Aided Civil and Infrastructure Engineering*, vol. 14, no. 6, pp. 429–443, nov 1999.
- [8] T. L. Friesz and K. Han, "The mathematical foundations of dynamic user equilibrium," *Transportation Research Part B: Methodological*, 2018.
- [9] S. Peeta and A. K. Ziliaskopoulos, "Foundations of dynamic traffic assignment: the past, the present and the future," *Networks and Spatial Economics*, vol. 1, no. 3-4, pp. 233–265, 2001.
- [10] C.-C. Lu, H. S. Mahmassani, and X. Zhou, "Equivalent gap function-based reformulation and solution algorithm for the dynamic user equilibrium problem," *Transportation Research Part B: Methodological*, vol. 43, no. 3, pp. 345–364, 2009.
- [11] G. Peque, T. Miyagi, and F. Kurauchi, "Adaptive learning algorithms for simulation-based dynamic traffic user equilibrium," *International Journal of Intelligent Transportation Systems Research*, vol. 16, no. 3, pp. 215–226, 2018.
- [12] H. Sbayti, C.-C. Lu, and H. S. Mahmassani, "Efficient implementation of method of successive averages in simulation-based dynamic traffic assignment models for large-scale network applications," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2029, no. 1, pp. 22–30, 2007.
- [13] I. Yang and R. Jayakrishnan, "Gradient projection method for simulation-based dynamic traffic assignment," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2284, no. 1, pp. 70–80, 2012.
- [14] M. Mahut, M. Florian, and N. Tremblay, "Comparison of assignment methods for simulation-based dynamic-equilibrium traffic assignment," in *Proceeding of the Transportation Research Board 87th Annual Meeting (DVD)*, Washington, DC, USA, January 2008.
- [15] M. Florian, M. Mahut, and N. Tremblay, "Application of a simulation-based dynamic traffic assignment model," *European Journal of Operational Research*, vol. 189, no. 3, pp. 1381–1392, 2008.
- [16] H. X. Liu, X. He, and B. He, "Method of successive weighted averages (mswa) and self-regulated averaging schemes for solving stochastic user equilibrium problem," *Networks and Spatial Economics*, vol. 9, no. 4, p. 485, 2009.

- [17] M. Ameli, J. P. Lebacque, and L. Leclercq, "Cross-comparison of convergence algorithms to solve trip-based dynamic traffic assignment problems," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 3, pp. 219–240, 2020.
- [18] M. Ameli, J.-P. Lebacque, and L. Leclercq, *Simulation-based Dynamic Traffic Assignment: Meta-Heuristic Solution Methods with Parallel Computing*, Hoboken, NJ, USA, 2020.
- [19] Y. Jiang, W. Y. Szeto, J. Long, and K. Han, "Multi-class dynamic traffic assignment with physical queues: intersection-movement-based formulation and paradox," *Transportmetrica A: Transport Science*, vol. 12, no. 10, pp. 878–908, 2016.
- [20] H. Halat, A. Zockaie, H. S. Mahmassani, X. Xu, and O. Verbas, "Dynamic network equilibrium for daily activity-trip chains of heterogeneous travelers: application to large-scale networks," *Transportation*, vol. 43, no. 6, pp. 1041–1059, 2016.
- [21] X. Xu, F. Fakhrmoosavi, A. Zockaie, and H. S. Mahmassani, "Estimating path travel costs for heterogeneous users on large-scale networks: heuristic approach to integrated activity-based model-dynamic traffic assignment models," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2667, no. 1, pp. 119–130, 2017.
- [22] J. Wang, S. Peeta, and X. He, "Multiclass traffic assignment model for mixed traffic flow of human-driven vehicles and connected and autonomous vehicles," *Transportation Research Part B: Methodological*, vol. 126, pp. 139–168, 2019.
- [23] M. Ameli, *Heuristic Methods for Calculating Dynamic Traffic Assignment*, PhD Thesis, IFSTTAR and Université de Lyon, Lyon, France, 2019.
- [24] X. Zhan and S. V. Ukkusuri, "Multiclass, simultaneous route and departure time choice dynamic traffic assignment with an embedded spatial queuing model," *Transportmetrica B: Transport Dynamics*, pp. 1–23, 2017.
- [25] M. Ameli, J. P. Lebacque, and L. Leclercq, "Day-to-day multimodal dynamic traffic assignment: impacts of the learning process in case of non-unique solutions," in *Proceedings of the DTA 2018, 7th International Symposium On Dynamic Traffic Assignment*, Hong Kong, June 2018.
- [26] M. W. Levin, M. Pool, T. Owens, N. R. Juri, and S. Travis Waller, "Improving the convergence of simulation-based dynamic traffic assignment methodologies," *Networks and Spatial Economics*, vol. 15, no. 3, pp. 655–676, 2015.
- [27] K. Nagel and G. Flötteröd, "Agent-based traffic assignment," in *The Multi-Agent Transport Simulation MATSim*, pp. 315–326, Ubiquity Press, London, UK, aug 2016.
- [28] R. Mounce and M. Carey, "On the convergence of the method of successive averages for calculating equilibrium in traffic networks," *Transportation Science*, vol. 49, no. 3, pp. 535–542, aug 2015.
- [29] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.
- [30] S. Boyles, J. Duthie, C. Melson, and T. Rambha, "Diverge models and dynamic traffic equilibria," in *Proceedings of the INFORMS Annual Meeting*, Austin, TX, USA, November 2013.
- [31] C. O. Tong and S. C. Wong, "Heuristic algorithms for simulation-based dynamic traffic assignment," *Transportmetrica*, vol. 6, no. 2, pp. 97–120, 2010.
- [32] R. Mounce and M. Carey, "Route swapping in dynamic traffic networks," *Transportation Research Part B: Methodological*, vol. 45, no. 1, pp. 102–111, 2011.
- [33] Ö. Verbas, H. S. Mahmassani, and M. F. Hyland, "Gap-based transit assignment algorithm with vehicle capacity constraints: simulation-based implementation and large-scale application," *Transportation Research Part B: Methodological*, vol. 93, no. 1–16, 2016.
- [34] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT press, Cambridge, MA, USA, 1992.
- [35] J. M. Johnson and Y. Rahmat-Samii, "Genetic algorithm optimization and its application to antenna design," in *Proceedings Of IEEE Antennas and Propagation Society International Symposium and URSI National Radio Science Meeting*, IEEE, Seattle, WA, USA, June 1994.
- [36] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 4, pp. 656–667, apr 1994.
- [37] H. Maini, K. Mehrotra, C. Mohan, and S. Ranka, "Knowledge-based nonuniform crossover," in *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, Seattle, WA, USA, June 1994.
- [38] L. Leclercq, J. A. Laval, and E. Chevallier, "The Lagrangian coordinates and what it means for first order traffic flow models," in *Proceedings of the Transportation And Traffic Theory 2007. Papers Selected for Presentation at ISTTT17*, London, UK, 2007.
- [39] H. S. Mahmassani, "Dynamic network traffic assignment and simulation methodology for advanced system management applications," *Networks and Spatial Economics*, vol. 1, no. 3, pp. 267–292, 2001.
- [40] Y. Hawas, H. Mahmassani, R. Taylor, A. Ziliaskopoulos, G.-L. Chang, and S. Peeta, *Development of Dynasmart-X Software for Real-Time Dynamic Traffic Assignment*, University of Maryland, College Park, MD USA, 1998.
- [41] İ. Ö. Verbas and H. S. Mahmassani, "Finding least cost hyperpaths in multimodal transit networks," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2497, no. 1, pp. 95–105, 2015.
- [42] J. Krug, A. Burianne, and L. Leclercq, "Reconstituting demand patterns of the city of Lyon by fusing multiple gis data sources," Tech. Rep., University of Lyon, Lyon, UK, 2019.
- [43] N. Alisoltani, L. Leclercq, M. Zargayouna, and J. Krug, "Optimal fleet management for real-time ride-sharing service considering network congestion," in *Proceedings of the 98th Annual Meeting of the Transportation Research Board*, Washington, DC, USA, January 2019.
- [44] S.-G. Li and Y.-M. Su, "A multi-mode, multi-class dynamic network model with queues for advanced transportation information systems," *EJTR*, vol. 5, no. 1, pp. 55–80, 2005.
- [45] M. Patriksson, *The Traffic Assignment Problem: Models and Methods*, Courier Dover Publications, Mineola, NY, USA, 2015.
- [46] H.-J. Huang and Z.-C. Li, "A multiclass, multicriteria logit-based traffic equilibrium assignment model under atis," *European Journal of Operational Research*, vol. 176, no. 3, pp. 1464–1477, 2007.
- [47] H. Yang and X. Zhang, "Multiclass network toll design problem with social and spatial equity constraints," *Journal of Transportation Engineering*, vol. 128, no. 5, pp. 420–428, 2002.
- [48] K. Zhang, H. S. Mahmassani, and C.-C. Lu, "Dynamic pricing, heterogeneous users and perception error: probit-based bicriterion dynamic stochastic user equilibrium assignment,"

- Transportation Research Part C: Emerging Technologies*, vol. 27, pp. 189–204, 2013.
- [49] A. Nagurney and J. Dong, “A multiclass, multicriteria traffic network equilibrium model with elastic demand,” *Transportation Research Part B: Methodological*, vol. 36, no. 5, pp. 445–469, 2002.
 - [50] T.-Y. Hu, C.-C. Tong, T.-Y. Liao, and L.-W. Chen, “Dynamic route choice behaviour and simulation-based dynamic traffic assignment model for mixed traffic flows,” *KSCE Journal of Civil Engineering*, vol. 22, no. 2, pp. 813–822, 2018.
 - [51] B. Ran, D.-H. Lee, and M. S.-I. Shin, “New algorithm for a multiclass dynamic traffic assignment model,” *Journal of Transportation Engineering*, vol. 128, no. 4, pp. 323–335, 2002.
 - [52] J. G. Yu and R. Jayakrishnan, “Multiclass, multicriteria dynamic traffic assignment with path-dependent link cost and entropy-based risk preference,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2667, no. 1, pp. 108–118, 2017.
 - [53] X. Xu, A. Chen, Z. Zhou, and L. Cheng, “A multi-class mean-excess traffic equilibrium model with elastic demand,” *Journal of Advanced Transportation*, vol. 48, no. 3, pp. 203–222, 2014.
 - [54] M. A. Abdel-Aty, R. Kitamura, and P. P. Jovanis, “Using stated preference data for studying the effect of advanced traffic information on drivers’ route choice,” *Transportation Research Part C: Emerging Technologies*, vol. 5, no. 1, pp. 39–50, 1997.
 - [55] M. Florian, J. H. Wu, and S. He, “A multi-class multi-mode variable demand network equilibrium model with hierarchical logit structures,” in *Transportation And Network Analysis: Current Trends*, pp. 119–133, Springer, Berlin, Germany, 2002.
 - [56] W. H. K. Lam and H.-J. Huang, “A combined trip distribution and assignment model for multiple user classes,” *Transportation Research Part B: Methodological*, vol. 26, no. 4, pp. 275–287, 1992.
 - [57] M. Meng, C. Shao, Y. D. Wong, and J. Zhang, “A multiclass, multimodal dynamic traffic assignment model with departure time,” *Mathematical Problems in Engineering*, vol. 2014, Article ID 812614, 9 pages, 2014.
 - [58] M. Florian and C. D. Morosan, “On uniqueness and proportionality in multi-class equilibrium assignment,” *Transportation Research Part B: Methodological*, vol. 70, pp. 173–185, 2014.
 - [59] S. Ryu, A. Chen, and K. Choi, “Solving the stochastic multi-class traffic assignment problem with asymmetric interactions, route overlapping, and vehicle restrictions,” *Journal of Advanced Transportation*, vol. 50, no. 2, pp. 255–270, 2016.
 - [60] N. Zheng and N. Geroliminis, “Modeling and optimization of multimodal urban networks with limited parking and dynamic pricing,” *Transportation Research Part B: Methodological*, vol. 83, pp. 36–58, 2016.
 - [61] G. R. de Andrade, Z. Chen, L. Elefteriadou, and Y. Yin, “Multiclass traffic assignment problem with flow-dependent passenger car equivalent value of trucks,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2667, no. 1, pp. 131–141, 2017.
 - [62] X. Pi, W. Ma, and Z. Qian, “A general formulation for multimodal dynamic traffic assignment considering multi-class vehicles, public transit and parking,” *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 369–389, 2019.
 - [63] R. Garcia and A. Marin, “Network equilibrium with combined modes: models and solution algorithms,” *Transportation Research Part B: Methodological*, vol. 39, no. 3, pp. 223–254, 2005.