Formulating multi-class user equilibrium using mixed-integer linear programming

Lin, Xiao; Leclercq, Ludovic; Tavasszy, Lorant; van Lint, Hans

DOI
10.1016/j.ejtl.2022.100097

Publication date
2022

Document Version
Final published version

Published in
EURO Journal on Transportation and Logistics

Citation (APA)
Lin, X., Leclercq, L., Tavasszy, L., & van Lint, H. (2022). Formulating multi-class user equilibrium using mixed-integer linear programming. *EURO Journal on Transportation and Logistics*, *11*, Article 100097. https://doi.org/10.1016/j.ejtl.2022.100097

EURO Journal on Transportation and Logistics 11 (2022) 100097

Contents lists available at ScienceDirect

EURO Journal on Transportation and Logistics

journal homepage: www.elsevier.com/locate/ejtl

Formulating multi-class user equilibrium using mixed-integer linear programming

Xiao Lin [a,*], Ludovic Leclercq [a,b], Lóránt Tavasszy [a], Hans van Lint [a]

[a] Department of Transport and Planning, Delft University of Technology, Stevinweg 1, 2628CN, Delft, The Netherlands
[b] University Gustave Eiffel, ENTPE, LICIT, Lyon, France

ARTICLE INFO

Keywords:
Traffic assignment
User equilibrium
Mathematical programming
Mixed-integer linear programming
Multi-class

ABSTRACT

We introduce an approach to formulate and solve the multi-class user equilibrium traffic assignment as a mixed-integer linear programming (MILP) problem. Compared to simulation approaches, the analytical MILP formulation makes the solution of network assignment problems more tractable. When applied in a multi-class context, it obviates the need to assume a symmetrical influence between classes and thereby allows richer traffic behavior to be taken into account. Also, it integrates naturally in optimization problems such as maintenance planning and traffic management. We develop the model and apply it for the Sioux Falls network, showing that it outperforms the traditional Beckmann-based and MSA approaches in smaller-scale problems. Further research opportunities lie in developing extensions of MILP-based assignment, with different variants of user equilibrium or dynamic assignment, and in improving the model and solution algorithms to allow large-scale application.

1. Introduction

Traffic assignment (TA) has been widely studied over decades. The core concept of TA is to predict how traffic is distributed over a network, given network supply and the demand from road users. Through time, many variations of TA have emerged. Bliemer et al. (2017) summarize these variations according to three capabilities, namely spatial (capacitated or not), temporal (static or dynamic), and behavioral (route choices). Another classifier is whether time variance is involved, dividing the assignment into static traffic assignment (STA) and dynamic traffic assignment (DTA) (Peeta and Ziliaskopoulos, 2001). In terms of user class considerations, TA can be characterized as single-class (SC) assignment and multi-class (MC) assignment, where the latter involves different user classes such as vehicle types.

TA models mostly build on Wardrop's first or second principle, concerning user equilibrium (UE) and system optimal (SO) network states, respectively. While SO scenarios, in which vehicles collectively minimize total travel time, are relevant for specific situations such as emergency evacuation (Sbayti and Mahmassani, 2006), the UE condition expresses route choices by un-coordinated road users. In this situation, equilibrium is reached when no driver can reduce personal travel time by making a different routing decision. Szeto and Wong (2012) review some of the extensions of UE assignment. These include stochastic UE assignment, risk-based UE, reliability-based UE, mean excess traffic equilibrium, robust UE, etc. The UE condition is also used in assignments other than traffic networks (Soumis and Nagurney, 1993). This study focuses on the mathematical formulation of the classic UE captured by the Wardrop's first principle.

Mathematical formulations of UE TA come in many forms, either analytical or via simulation. Simulation-based methods are descriptive and do not aim to optimize. They provide probable results of certain choices and traffic management strategies. As a result simulation methods often lack well-defined solution properties such as optimality and uniqueness (Szeto and Wong, 2012) which could help to assess the validity of calculated network states. For a recent review of simulation based formulations readers are referred to Ameli et al. (2021).

The advantages of describing the TA problem using analytical models, according to Boyce et al. (2001), are three-fold. First, analytical representations are specific and precise. Second, the existence, the uniqueness, and the stability of solutions can be determined with analytical models. Third, solution algorithms to analytical models and their convergence properties can be determined.

In this paper we develop a new analytical formulation for MC UE STA using mixed-integer linear programming (MILP). MILP is widely used in operations research (OR) where a wide range of solving algorithms and well developed software packages are available. Compared with previous approaches based on mathematical programming (mainly the Beckmann transformation approach, BT), this formulation

[*] Corresponding author.
E-mail address: x.lin@tudelft.nl (X. Lin).

https://doi.org/10.1016/j.ejtl.2022.100097
Received 15 July 2021; Received in revised form 26 October 2022; Accepted 1 November 2022
Available online 7 November 2022

is capable of handling multi-class assignment without the strict assumption of symmetrical interaction between classes (further discussed in Section 2). We argue that the MILP formulation opens new doors for the domain of optimization based TA, as it makes mature tooling from OR available for solving and analyzing TA problems. It lays a foundation for further development of more comprehensive and complex models and studies such as DTA and UE solution properties.

The remainder of the paper is organized as follows. Section 2 reviews literature to explain the problems encountered by current formulations. Section 3 develops our new MILP formulation for MC UE STA. Section 4 compares the MILP with other methods in a benchmark network and explores approaches to speed up computation. Section 5 concludes the study and proposes future research directions.

2. Literature review

This section reviews some of the core issues of formulating UE assignments. We start with the multi-class travel cost functions. Then we discuss analytical TA formulations and the assumption of inter-class symmetry, which has been an obstacle to formulate UE with mathematical programming.

2.1. Travel cost functions

Many factors contribute to road users' route choice distributions. These factors range from the length and reliability of travel time to tolls and fuel costs (Bliemer, 2001). This paper focuses only on one factor, travel time, which is often estimated by volume-delay functions such as Bureau of Public Roads (BPR) functions (U.S. Bureau of Public Roads, 1964). A BPR function associates link travel time to the minimum travel time (free flow travel time T_0), link capacity K and link flow x^{link}:

$$T = T_0\left(1+\alpha\left(\frac{x^{\text{link}}}{K}\right)^{\beta}\right), \tag{1}$$

in which α and β are constants and usually $\alpha = 0.15$ and $\beta = 4$.

When considering scenarios with multiple user classes, studies suggest a detailed description of the volume-delay relationship when several user classes are present (Yun et al., 2005; Müller and Schiller, 2015; Lu et al., 2016). Noriega and Florian (2007) on the other hand, adopts a more straightforward extension of the above BPR function, which includes the passenger car equivalent (PCE) value and aggregates the flows of different vehicle classes. The function can be written as follows:

$$c^{\text{link}}_{me} = T_{me,0}\left(1+\alpha\left(\frac{\sum_{m'}\pi_{m'}x^{\text{link}}_{m'e}}{K_e}\right)^{\beta}\right), \qquad \forall m,e, \tag{2}$$

in which c^{link}_{me} represents the travel cost on the link e for class m; $T_{me,0}$ stands for the free-flow travel time on link e by class m; K_e is the capacity of link e; and $\pi_{m'}$ is the PCE value for class m'.

2.2. Analytical TA formulations and asymmetry issues

Given a network and demand, an MC UE STA can be represented by the following complementary conditions:

$$x_{mwp} \geq 0, \qquad \forall m,w,p; \tag{3}$$

$$c_{mwp} \geq c^{*}_{mw}, \qquad \forall m,w,p; \tag{4}$$

$$x_{mwp}\left(c_{mwp} - c^{*}_{mw}\right) = 0, \qquad \forall m,w,p, \tag{5}$$

in which x_{mwp} stands for the number of users in class m, origin–destination (OD) pair w choosing route p; c_{mwp} is the travel cost for a user in class m, OD pair w choosing route p; c^{*}_{mw} is the travel cost of class m, OD pair w under UE condition.

The above UE condition can be expressed using a variational inequality (VI) formulation (Bliemer and Bovy, 2003), a fixed-point formulation (Szeto et al., 2011), or a nonlinear complementary programming formulation (Szeto and Lo, 2004), etc. Nagurney (1998) proves that VI, fixed-point, nonlinear complementary programming are equivalent. It also points out that under the special circumstance where the Jacobian matrix of the cost function is symmetric, the UE condition can be expressed in a mathematical programming formulation, i.e., BT. The symmetry conditions can be inter-user, inter-spatial, or inter-temporal (Bliemer and Bovy, 2003). In a multi-class assignment, inter-class symmetry is a strong assumption, which usually does not hold in reality. This symmetry condition expressed in mathematical terms is:

$$\frac{\partial c_{m'e}}{\partial x^{\text{link}}_{me}} = \frac{\partial c_{me}}{\partial x^{\text{link}}_{m'e}}, \tag{6}$$

in which $x^{\text{link}}_{me} = \sum_{w,p}\delta_{pe}x_{mwp}$ stands for the traffic flow of user class m on link e ($\delta_{pe} = 1$ if link e is used by path p). Eq. (6) describes the inter-user class symmetry, that one user class affect the other user class exactly the same with the other way around (Bliemer and Bovy, 2003). Under such condition, the TA problem can then be expressed in the following formulation as the BT (Beckmann et al., 1956):

$$\min J^{\text{BT}} = \sum_{m,e}\int_0^{x^{\text{link}}_e} c_{me}(x)\,dx. \tag{7}$$

Since the assumption of symmetry usually does not hold in reality, the method of BT is mostly theoretical for MC assignments. Due to this limitation, the researchers have turned to other formulations such as VI or fixed-point instead of the mathematical programming formulation, despite its simplicity and practicality. Bliemer and Bovy (2003) state: "Writing the model as an optimization problem is the most practical formulation in the sense that many literature and algorithms exist for solving this type of problem, however, due to the presence of asymmetries in the cost functions, as an optimization problem and a more general type of problem formulation such as the VI problem formulation should be used instead". Nagurney (1984) states that "multimodal traffic UE problem can only be reduced to a minimization problem if the Jacobian matrix of the travel cost functions is symmetric", which is "not expected to hold".

To bring the statements above into the context of this paper, we now examine the inter-class symmetry assumption (Eq. (6)) using the multi-class cost function (Eq. (2)):

$$\begin{aligned}\frac{\partial c^{\text{link}}_{me}}{\partial x^{\text{link}}_{m'e}} &= \frac{\partial\left(T_{me,0}\left(1+\alpha\left(\frac{\sum_m \pi_m x^{\text{link}}_{me}}{K_e}\right)^{\beta}\right)\right)}{\partial x^{\text{link}}_{m'e}}\\ &= \frac{\partial T_{me,0}\,\alpha\left(\frac{\sum_m \pi_m x^{\text{link}}_{me}}{K_e}\right)^{\beta}}{\partial x^{\text{link}}_{m'e}}\\ &= \frac{\alpha\beta\pi^{\beta-1}_{m'}T_{me,0}}{K^{\beta}_e}(x^{\text{link}}_{m'e})^{\beta-1}.\end{aligned} \tag{8}$$

Similarly,

$$\frac{\partial c^{\text{link}}_{m'e}}{\partial x^{\text{link}}_{me}} = \frac{\alpha\beta\pi^{\beta-1}_{m}T_{m'e,0}}{K^{\beta}_e}(x^{\text{link}}_{me})^{\beta-1}. \tag{9}$$

Therefore, as Eq. (6) indicates, the assumption of inter-class symmetry requires that

$$\frac{\alpha\beta\pi^{\beta-1}_{m'}T_{me,0}}{K^{\beta}_e}(x^{\text{link}}_{m'e})^{\beta-1} = \frac{\alpha\beta\pi^{\beta-1}_{m}T_{m'e,0}}{K^{\beta}_e}(x^{\text{link}}_{me})^{\beta-1}, \tag{10}$$

i.e.,

$$\left(\frac{x^{\text{link}}_{m'e}}{x^{\text{link}}_{me}}\right)^{\beta-1} = \frac{\pi^{\beta-1}_{m}T_{m'e,0}}{\pi^{\beta-1}_{m'}T_{me,0}}. \tag{11}$$

Eq. (11) needs to hold if symmetry is assumed. On the right-hand side, the PCE values and free-flow times are user class property and link property, which are not dependent on link flows. This indicates that if the symmetric assumption holds, there should be a fixed ratio of link flows of different classes, which is usually unrealistic.

To summarize this section: although being a powerful tool, the existing mathematical programming's application in UE TA (mainly BT formulation) is limited as it relies on a strong and unrealistic assumption of symmetry. In this paper, we propose a new formulation for MC STA under UE conditions using mathematical programming. In this formulation we start from the basic UE TA conditions (Eqs. (3)–(5)), and bypass the symmetry assumption to construct the mathematical programming formulation.

3. Methodology

As discussed in the literature review, previous studies provide many options for link cost functions. In this study, we use the MC BPR type of cost function shown in Eq. (2). Note that one can also use more detailed MC link cost functions in this formulation without the need to change the framework of the formulation.

Next we build the MILP formulation for the MC UE STA conditions ((3)–(5)). This section breaks it down into a few steps. First, we rewrite the UE STA problem into an optimization problem. Subsequently, we include a linear approximation of the BPR function into the optimization problem. Lastly, we linearize this optimization problem so that it is reduced to MILP.

3.1. Formulating MC UE STA into an optimization problem

In the UE conditions, Eq. (5) specifies that if path p is used by user class m, the path cost c_{mwp} should be equal to c^*_{mw}, the minimal travel cost between OD pair w; if this path is not used, then the cost can be equal to or higher than the minimal travel cost. Here, we include a binary variable a_{mwp} to rewrite this condition:

$$a_{mwp}(c_{mwp} - c^*_{mw}) = 0, \tag{12}$$

in which a_{mwp} is a flag, denoting if path p between OD pair w is used by class m:

$$a_{mwp} = \begin{cases} 0, \text{ if } x_{mwp} = 0, \\ 1, \text{ if } x_{mwp} > 0, \end{cases} \quad \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w. \tag{13}$$

After rewriting (5) into (12), we relax this condition to form an objective function for the optimization problem:

$$\min J|_x = \sum_{m\in\mathcal{M}} \sum_{w\in\mathcal{W}} \sum_{p\in\mathcal{P}_w} a_{mwp}(c_{mwp} - c^*_{mw}). \tag{14}$$

Due to Condition (4), the objective function J is non-negative. Therefore, when J reaches the minimal value 0, all traversed paths have $c_{mwp} - c^*_{mw} = 0$, and all non-traversed paths have $a_{mwp} = 0$. This is equivalent to the user equilibrium condition specified in (5). The conditions of the STA problem are listed as follows.

$$x_{mwp} \geq 0, \quad \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w; \tag{15}$$

$$a_{mwp} = \begin{cases} 0, \text{ if } x_{mwp} = 0, \\ 1, \text{ if } x_{mwp} > 0, \end{cases} \quad \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w; \tag{16}$$

$$c_{mwp} \begin{cases} \geq c^*_{mw}, \text{ if } a_{mwp} = 0, \\ = c^*_{mw}, \text{ if } a_{mwp} = 1, \end{cases} \quad \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w. \tag{17}$$

The above objective function (14) and constraints (15), (16), and (4) formulate the optimization problem for UE STA. The solution to the problem is the flows under UE conditions. Condition (17) specifies in what circumstances $c_{mwp} - c^*_{mw}$ is considered minimal. This condition is not included in the MILP formulation since the objective function (14) is already minimizing $c_{mwp} - c^*_{mw}$. We now focus on the link cost function (BPR) for calculating travel costs.

3.2. BPR linear approximation

To start with the linear approximation of BPR, we need to specify the relationship between path flow/cost and the link flow/cost:

$$x^{\text{link}}_{me} = \sum_{w\in\mathcal{W}} \sum_{p\in\mathcal{P}_w} \delta_{pe} x_{mwp}, \quad \forall m \in \mathcal{M}, e \in \mathcal{E}; \tag{18}$$

$$x^{\text{link}}_{e} = \sum_{m\in\mathcal{M}} \pi_m x^{\text{link}}_{me}, \quad \forall e \in \mathcal{E}; \tag{19}$$

$$c^{\text{link}}_{me} = T_{me,0}\left(1 + \alpha\left(\frac{x^{\text{link}}_e}{K_e}\right)^{\beta}\right), \quad \forall m \in \mathcal{M}, e \in \mathcal{E}; \tag{20}$$

$$c_{mwp} = \sum_{e\in\mathcal{E}} \delta_{pe} c^{\text{link}}_{me}, \quad \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w, \tag{21}$$

in which $\delta_{pe} = 1$ denotes that path p includes link e, otherwise $\delta_{pe} = 0$. With these equations, the MC BPR function (20) is adopted in our study.

Next, we linearize Eq. (20) using piecewise approximation. We use piecewise segments with the universal length for each link. Let L be the number of linear segments to approximate the BPR function. Index $l \in \mathcal{L}$ stands for the lth linear segment in the approximated BPR function, with the range $[x_{l-1}, x_l)$. Hence if the total link flow x^{link} falls into the lth segment, for all class $m \in \mathcal{M}$ and link $e \in \mathcal{E}$, the approximated link cost c^{appr} is

$$\begin{aligned} &c^{\text{appr}}_{me}|_{x^{\text{link}}_e \in [x_{e,l-1}, x_{e,l})} \\ &= \frac{c^{\text{link}}_{me}(x_{e,l}) - c^{\text{link}}_{me}(x_{e,l-1})}{x_{e,l} - x_{e,l-1}}\left(x^{\text{link}}_e - x_{e,l-1}\right) + c^{\text{link}}_{me}(x_{e,l-1}), \end{aligned} \tag{22}$$

in which $x_0 = 0$. To include all possible values of x^{link}_e with linear segments, we use the following form:

$$c^{\text{appr}}_{me} = \sum_{l\in\mathcal{L}} b_e(l) c^{\text{piece}}_{me}(l), \forall e \in \mathcal{E}, m \in \mathcal{M}, \tag{23}$$

in which

$$c^{\text{piece}}_{me}(l) = \frac{c^{\text{link}}_{me}(x_{e,l}) - c^{\text{link}}_{me}(x_{l-1})}{x_{e,l} - x_{e,l-1}}\left(x^{\text{link}}_e - x_{e,l-1}\right) + c^{\text{link}}_{me}(x_{e,l-1}), \tag{24}$$

$$b_{me}(l) = \begin{cases} 1, x \in [x_{l-1}, x_l), \\ 0, \text{otherwise}, \end{cases} \quad \forall m \in \mathcal{M}, e \in \mathcal{E}, l \in \mathcal{L}. \tag{25}$$

In the above equations, $b_e(l), \forall l \in \mathcal{L}$ act as sets of selectors, denoting on which segment the link load falls. Fig. 1 illustrates the piecewise approximation of BPR using 4 segments ($L = L^{\text{left}} + L^{\text{right}}$). The first ($L^{\text{left}} = 2$) segments cover the situation where the total volume (number of all vehicles weighted by PCE) is less than the capacity. The other ($L^{\text{right}} = 2$) segments cover the situation where the total volume is more than the capacity but less than twice of capacity. In reality, the volume should not exceed the capacity. However, we include the situation where the volume is larger than the capacity to investigate the impact of piecewise approximation on the quality of the assignment.

The optimization problem after involving BPR approximation is then with the objective function (14) and constraints (15), (16), (24) and (25).

3.3. MILP

We conduct further linearization to convert the above formulation into MILP. We start from the optimization problem at the end of the previous section and explain the process step by step.

3.3.1. Linearizing the objective function

The objective function (14) is in a quadratic term, with a continuous value ($c_{mwp} - c^*_{mw}$) multiplied by a binary value (a_{mwp}). To linearize this term, we introduce a new value z_{mwp}. Let

$$z_{mwp} = \begin{cases} c_{mwp} - c^*_{mw}, & \text{if } a_{mwp} = 1, \\ 0, & \text{if } a_{mwp} = 0, \end{cases} \quad \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w. \tag{26}$$

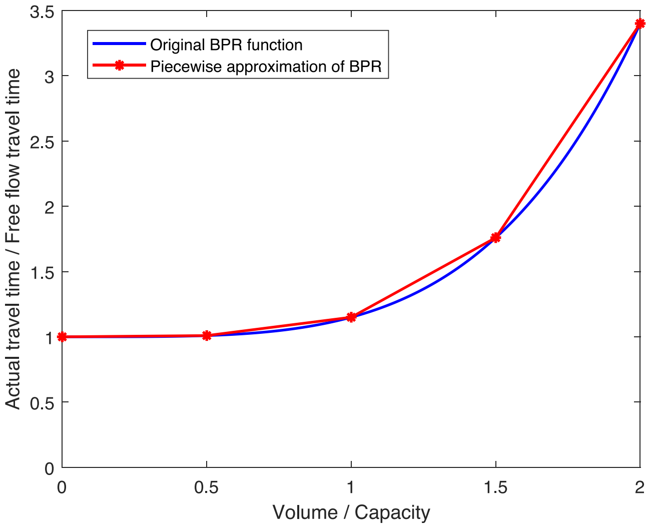

Fig. 1. Piecewise linearized BPR function vs original BPR function with $L = 4$.

Linearizing the above constraint (26), we have:

$$z_{mwp} \le c_{mwp} - c^*_{mw}, \qquad \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w; \tag{27}$$

$$z_{mwp} \ge c_{mwp} - c^*_{mw} - M(1 - a_{mwp}), \qquad \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w; \tag{28}$$

$$z_{mwp} \le M a_{mwp}, \qquad \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w. \tag{29}$$

Constant M stands for a sufficiently large number. In practice, this can be the upper bound of possible value of z_{mwp}. The objective function then becomes:

$$\min J = \sum_{m\in\mathcal{M}} \sum_{w\in\mathcal{W}} \sum_{p\in\mathcal{P}_w} z_{mwp}, \tag{30}$$

with the additional linear constraints (27)–(29).

3.3.2. Linearizing conditions related to route choices

To linearize Constraint (16), we use the similar approach with the "big M" method, as follows:

$$M x_{mwp} \ge a_{mwp}, \qquad \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w; \tag{31}$$

$$x_{mwp} \le M a_{mwp}, \qquad \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w. \tag{32}$$

3.3.3. Linearizing conditions related to the BPR approximation

Linearizing the quadratic form of c^{appr}_{me} in Constraint (23) we introduce $h_{me}(l)$:

$$c^{\text{appr}}_{me} = \sum_{l\in\mathcal{L}} h_{me}(l), \forall m \in \mathcal{M}, e \in \mathcal{E}, \tag{33}$$

in which

$$h_{me}(l) = \begin{cases} c^{\text{piece}}_{me}(l), & \text{if } b_e(l) = 1, \\ 0, & \text{if } b_e(l) = 0, \end{cases} \quad \forall m \in \mathcal{M}, e \in \mathcal{E}, l \in \mathcal{L}. \tag{34}$$

Performing the linearization on Constraint (34), for all $m \in \mathcal{M}, e \in \mathcal{E}, l \in \mathcal{L}$ we have the following:

$$h_{me}(l) \le M b_e(l); \tag{35}$$

$$h_{me}(l) \ge -M b_e(l); \tag{36}$$

$$h_{me}(l) \le c^{\text{piece}}_{me}(l) + M(1 - b_e(l)); \tag{37}$$

$$h_{me}(l) \ge c^{\text{piece}}_{me}(l) - M(1 - b_e(l)). \tag{38}$$

To linearize Constraint (25), we need to use additional auxiliary binary variables $\hat{b}_e(l)$:

$$\hat{b}_e(l) = \begin{cases} 1, \text{if } x^{\text{link}}_e > x_l, \\ 0, \text{if } x^{\text{link}}_e \le x_l, \end{cases} \quad \forall e \in \mathcal{E}, l \in \{1, 2, \dots, L-1\}. \tag{39}$$

Linearizing the constraints:

$$x^{\text{link}}_e - x_l \le M \hat{b}_e(l), \qquad \forall e \in \mathcal{E}, l \in \{1, 2, \dots, L-1\}; \tag{40}$$

$$x^{\text{link}}_e - x_l \ge M(\hat{b}_e(l) - 1), \qquad \forall e \in \mathcal{E}, l \in \{1, 2, \dots, L-1\}. \tag{41}$$

The relation between $b_e(l)$ and $\hat{b}_e(l)$ is:

$$b_e(l) = \hat{b}_e(l-1) - \hat{b}_e(l), \forall e \in \mathcal{E}, l \in \mathcal{L}, \tag{42}$$

in which $\hat{b}_e(0) = 1$ and $\hat{b}_e(L) = 0$.

3.4. Final MILP formulation

We use a kth Dijkstra shortest path method to exogenously enumerate the paths before building the model. Note that this may generate different shortest paths for different vehicle classes. Consequently, the model is formulated and fed to an MILP solver. The final form of the MILP formulation for UE STA is listed as follows:

$$\min J = \sum_{m\in\mathcal{M}} \sum_{w\in\mathcal{W}} \sum_{p\in\mathcal{P}_w} z_{mwp},$$

s.t.

$$\begin{aligned}
& z_{mwp} \le c_{mwp} - c^*_{mw}, && \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w; \\
& z_{mwp} \ge c_{mwp} - c^*_{mw} - M(1 - a_{mwp}), && \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w; \\
& z_{mwp} \le M a_{mwp}, && \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w; \\
& x_{mwp} \ge 0, && \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w; \\
& M x_{mwp} \ge a_{mwp}, && \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w; \\
& x_{mwp} \le M a_{mwp}, && \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w; \\
& c_{mwp} - c^*_{mw} \ge 0, && \forall m \in \mathcal{M}, w \in \mathcal{W}, p \in \mathcal{P}_w; \\
& c^{\text{piece}}_{me} = \frac{c^{\text{link}}_{me}(x_{e,l}) - c^{\text{link}}_{me}(x_{e,l-1})}{x_{e,l} - x_{e,l-1}} && \\
& \quad \times \left(x^{\text{link}}_e - x_{e,l-1}\right) + c^{\text{link}}_{me}(x_{e,l}), && \forall m \in \mathcal{M}, e \in \mathcal{E}, l \in \mathcal{L}; \\
& c^{\text{appr}}_{me} = \sum_{l\in\mathcal{L}} h_{me}(l), && \forall m \in \mathcal{M}, e \in \mathcal{E}; \\
& h_{me}(l) \le M b_e(l), && \forall m \in \mathcal{M}, e \in \mathcal{E}, l \in \mathcal{L}; \\
& h_{me}(l) \ge -M b_e(l), && \forall m \in \mathcal{M}, e \in \mathcal{E}, l \in \mathcal{L}; \\
& h_{me}(l) \le c^{\text{piece}}_{me}(l) + M(1 - b_e(l)), && \forall m \in \mathcal{M}, e \in \mathcal{E}, l \in \mathcal{L}; \\
& h_{me}(l) \ge c^{\text{piece}}_{me}(l) - M(1 - b_e(l)), && \forall m \in \mathcal{M}, e \in \mathcal{E}, l \in \mathcal{L}; \\
& x^{\text{link}}_e - x_l \le M \hat{b}_e(l), && \forall e \in \mathcal{E}, l \in \{1, 2, \dots, L-1\}; \\
& x^{\text{link}}_e - x_l \ge M(\hat{b}_e(l) - 1), && \forall e \in \mathcal{E}, l \in \{1, 2, \dots, L-1\}; \\
& b_e(l) = \hat{b}_e(l-1) - \hat{b}_e(l), && \forall e \in \mathcal{E}, l \in \mathcal{L}.
\end{aligned}$$

3.5. Special order sets

Some commercial solvers offer the option of using the special order set (SOS) type of constraints in solving MILP problems with sets of binary variables that satisfy certain conditions with accelerated computations. SOSs are suitable for constraints specifying piecewise approximation and can generally accelerate computations. In our implementations we make use of this feature with the experiments. We compare the performance of the implementation of MILP and the implementation with SOSs (MILP-SOS) in our numerical examples in the next section.

For each link $e \in \mathcal{E}$, the aggregated flow $x^{\text{link}}_e = \sum_m \sum_w \sum_p \delta_{pe} x_{mwp} \pi_m$ monotonically determines the travel time of each vehicle class via this link according to the BPR function. In MILP-SOS, the piecewise BPR function is represented by SOS type 2 constraints. A type 2 SOS constraint specifies that in a set of variables $[b_1, b_2, \dots, b_n]$, only 2 (consecutively) of them can take values other than 0 (Keha et al., 2004). Denote cost function with linearized BPR as $\bar{c}^{\text{link}}_{me}(x)$, the number of segments $L = L^{\text{left}} + L^{\text{right}}$, and the capacity of link e as K_e we have the following:

$$\text{SOS: } \left[(b_1, 0), (b_2, K_e/L^{\text{left}}), \dots, (b_{L+1}, K_e/L^{\text{left}} \times L)\right], \quad \forall m \in \mathcal{M}, e \in \mathcal{E}; \tag{43}$$

$$\sum_{l=1}^{L+1} b_{me,l} = 1, \quad \forall m \in \mathcal{M}, e \in \mathcal{E}; \tag{44}$$

$$\sum_{l=1}^{L+1} b_{me,l} K_e/L^{\text{left}} \times (l-1) = x_e^{\text{link}}, \quad \forall m \in \mathcal{M}, e \in \mathcal{E}; \tag{45}$$

$$\sum_{l=1}^{L+1} b_{me,l} c_{me}^{\text{link}} (K_e/L^{\text{left}} \times (l-1)) = \tilde{c}_{me}^{\text{link}}, \quad \forall m \in \mathcal{M}, e \in \mathcal{E}. \tag{46}$$

Condition (43) declares the SOSs for the solver. Each of the sets has $L+1$ pairs of auxiliary decision variables b_l and their weights $K_e/L^{\text{left}} \times (l-1)$. Constraints (44)–(46) complete the SOS constraints in our implementation MILP-SOS. Constraints (43)–(46) replace the previous BPR linearization (Constraints (22)–(25), and (33)–(42)) in MILP.

3.6. Capability in handling asymmetry

Here we briefly discuss the relations between this mathematical programming formulation and the inter-class symmetry assumption. In the BT formulation, the symmetry assumption (6) makes the BT equivalent to the UE condition expressed in (5). In our MILP formulation, this UE condition is directly transformed to the optimization problem (14) with (13) specifying the auxiliary binary variable a. Hence, the condition of (5) is equivalent to $J = 0$ in (13) and (14). The MILP formulation does not require this symmetric assumption (6) and can handle situations with inter-class asymmetry.

4. Numerical examples

In this section, we conduct a series of numerical TA experiments on the network Sioux Falls[1] (see Fig. 2). Methods of MILP, BT, and MSA are compared in a single class case. Subsequently, we investigate the performance of MILP formulation in multi-class scenarios and compare it to that of MSA. The experiments are coded in Matlab 2018b on a Windows 10 server with INTEL Xeon 6148 2.4 GHz CPU, and 32 GB RAM. The MILP and the BT are solved by IBM Cplex 12.10, while the MSA gets its results from link-based iterations.

We use the average gap (shown in Eq. (47)) of the results to measure the quality of the assignments. This indicator denotes the distance from the solution to the perfect UE (Ameli et al., 2021). The average gap of a perfect assignment is 0. Note that the lowest path costs $\tilde{c}_{mw}^*$ are calculated over the loaded network after the assignments, which may include different paths from the ones enumerated before the assignment. In order to isolate the influence of possible inefficient path enumerations, we also use Agap-P to calculate the average gap value on the pre-assigned paths using c_{mw}^*:

$$AGap = \frac{\sum_m \sum_w \sum_p (c_{mwp} - \tilde{c}_{mw}^*) \pi_m x_{mwp}}{\sum_m \sum_w \sum_p \pi_m x_{mwp}}; \tag{47}$$

$$AGap\text{-}P = \frac{\sum_m \sum_w \sum_p (c_{mwp} - c_{mw}^*) \pi_m x_{mwp}}{\sum_m \sum_w \sum_p \pi_m x_{mwp}}. \tag{48}$$

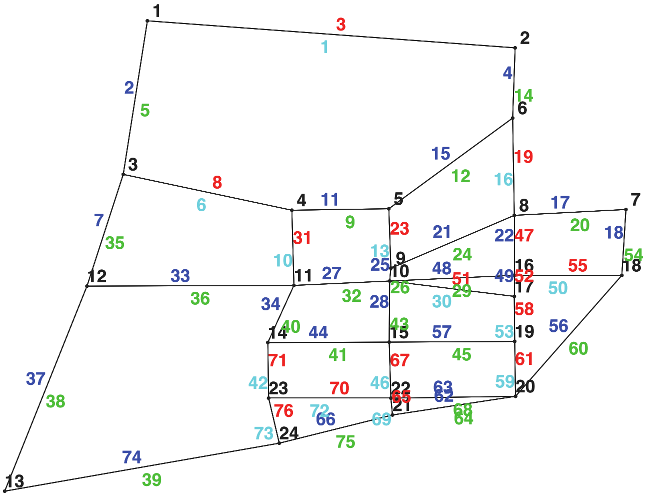

Fig. 2. Sioux Falls network, online version with colors. Digits in black represent node numbers; digits in other colors represent link numbers.

Table 1
Single-class OD matrix.

fromNode	toNode	Demand
1	7	6250
1	20	7500
13	2	7500
13	18	5000
19	1	10000
24	2	10000

Table 2
Comparison of single class assignments: MILP, BT, and MSA, in terms of computation time and average gap.

Method	MILP	BT	MSA[a]
Time (s)	20.7	31.7	351.7
Average Gap	0.7085	4.5012	13.4686

[a] The gap of link flow summation between the last two iterations is smaller than 10^{-8}.

With the 2 indicators (Agap and Agap-P), the inaccuracies (deviation from the perfect UE) brought by pre-assigned paths and BPR linearization are isolated. In particular, if Agap-P = 0, then the BPR linearization brings no inaccuracy; while if Agap = Agap-P, the pre-assigned paths remain the shortest after the assignment.

4.1. Single-class assignment

We first compare MILP with BT and MSA in a single-class assignment using the OD pairs and demand in Table 1. Both MILP and BT are using 6 pre-assigned paths and 5 segments for BPR linearization. MSA converges after 351.7 s. The results of assignments are shown in Figs. 3–5. Table 2 lists computation time and average gap of each assignment method. The results show that the MILP takes the least time and has the lowest average gap. BT and MSA both use more time than MILP and yield larger gaps. The advantage of the MILP in this SC assignment is obvious.

4.2. Multi-class assignment

Next, we look further into the performance of the MILP formulation in a scenario with two user classes. The MC demand is given in Table 3. Several configurations are used in the experiments to better understand the performance of the MILP formulation, with assignment from the MSA as a reference (see Table 4). These configurations are built by

[1] The network properties of Sioux Falls can be found at the following link: https://github.com/bstabler/TransportationNetworks/tree/master/SiouxFalls; The visualization is made by the Matlab package developed by KU Leuven: https://www.mech.kuleuven.be/en/cib/traffic/downloads

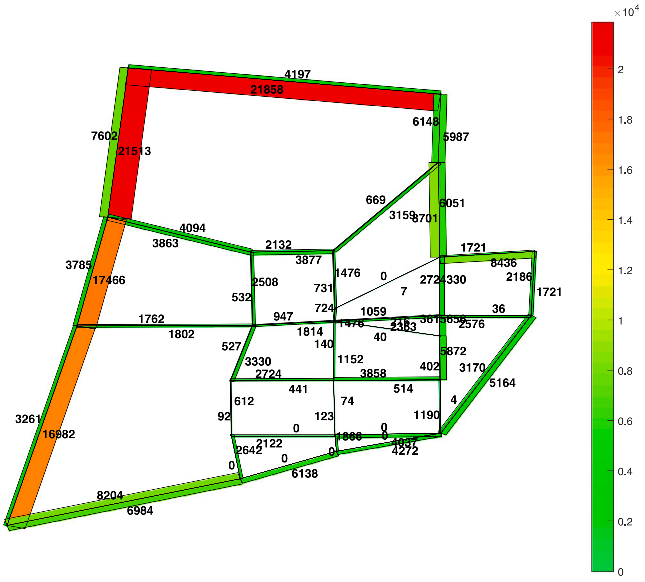

Fig. 3. MSA results single class assignment.

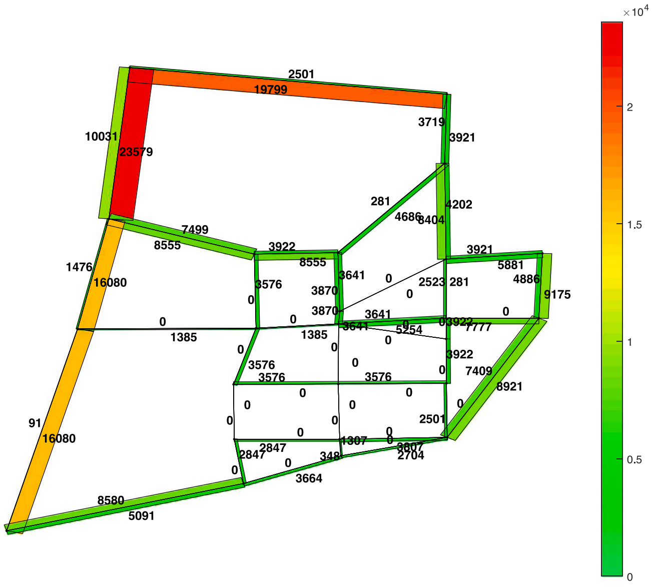

Fig. 4. BT results single class assignment.

altering 3 inputs: the number of enumerated paths, the number of segments in the piecewise linearization, and a multiplier to the demand of cars (×1, ×2, ×3, ×5). The demands given by Table 3 are considered as ×1, the "original demand". Scenarios of ×2, ×3, ×5 are with 2,3, and 5 times the original car demand. The truck demands remain the same across all configurations. The configuration also includes the different numbers of pre-assigned paths and the number of piecewise segments considered in the MILP formulation. Note that in reality, the traffic flows on a link should not be larger than the capacity, and BPR also becomes less accurate in such a situation. However, we still include the scenarios with excessively high demand (×5) and BPR segments outside the capacity. This is to gain a full insight into the performances of the proposed mathematical formulation.

Table 4 shows the performances (time, Agap, Agap-P) of the multi-class assignments using MILP and MSA methods under different conditions (demand, number of pre-assigned paths, and number of segments for BPR approximation). With the MSA method, the higher the demand the more iterations are needed for the algorithm to converge and hence more time is required. MSA assignments that do not converge within 1500 iterations are stopped. In the MILP approach, the solving time generally increases if more paths/segments are considered. It is not uncommon that solving time becomes less feasible to find an exact solution to a larger size optimization problem. This makes an interesting next step to develop specific algorithms tailored to this type of optimization problem to increase the efficiency of finding exact or near exact solutions.

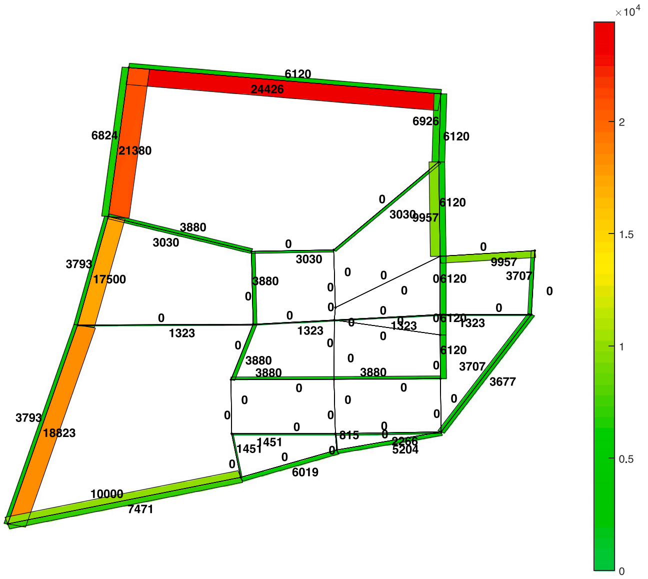

Fig. 5. MILP results single class assignment.

Table 3
Multi-class OD matrix.

fromNode	toNode	Demand Car	Demand Truck
1	7	2500	1500
3	20	3000	800
13	2	3000	500
19	1	2000	300
24	2	2400	500
12	18	2000	700

4.3. *Discussions on the performance of MILP*

4.3.1. *Solution quality reflected by average gaps*

In Table 4, under the Demand ×1 scenario, the MILP provides excellent solution quality, keeping both the Agap-P and Agap at 0. This outperforms MSA in terms of both solving time and precision. On average, the time required to solve the UE is less than MSA at the current scale. However, the solution becomes less accurate when the demand increases. This inaccuracy has at least two sources.

The first source is the piecewise linear approximation, reflected by the value of Agap-P. In the current MILP formulation, the inaccuracy increases when the demand of a link exceeds its capacity, since the BPR function contains a 4th-order link volume argument. With higher demand, it is more likely that the piecewise approximation becomes less precise in representing the link costs given by the BPR function. Interestingly, increasing the number of segments used in the assignments does not guarantee preciser results. See experiments 3–6, 10–13, and 14–17. In theory, more segments likely will result in a piecewise approximation that fits better the BPR function. From the experiments we see that the accuracy gain may vary in different situations, depending on the distribution of link flows.

Table 4
Experiment results with different configurations of MILP and MSA.

MILP	Demand	No. Paths	$L^{\text{left}}/L^{\text{right}}$	Time (s)	Agap	Agap-P
1	×1	3	2/1	0.172	0	0
2	×2	3	2/1	0.953	0.2424	0.0789
3	×2	4	2/1	1.546	0.0808	0.0808
4	×2	4	3/1	6.032	0.1254	0.1254
5	×2	4	2/2	3.688	0.4998	0.4998
6	×2	4	3/2	4.187	0.1724	0.1724
7	×2	5	2/1	8.063	0.0605	0.0605
8	×3	3	2/1	0.922	4.3927	0.4705
9	×3	3	2/2	8.532	4.3034	0.8919
10	×3	4	2/1	3.016	2.5195	0.1972
11	×3	4	2/2	2.782	2.6697	0.8721
12	×3	4	3/2	22.734	2.4623	0.2608
13	×3	4	3/3	4.375	2.9379	1.0039
14	×3	5	2/1	51.750	2.3011	0.9958
15	×3	5	2/2	4.969	2.9384	1.1770
16	×3	5	3/2	455.469	2.2962	0.6861
17	×3	5	3/3	226.500	3.0163	1.6120
18	×3	M	2/1	0.578	2.1113	0.1599
19	×3	M	3/2	1.782	2.1700	0.2860
20	×5	3	2/1	1.391	44.5027	4.3988
21	×5	4	2/1	16.437	35.0108	4.2701
22	×5	5	2/1	85.297	31.2293	15.2133
23	×5	5	2/2	658.203	26.0172	8.5628
24	×5	M	2/2	282.469	4.0837	2.6640
MSA	Demand	Iterations	Convergence	Time	Agap	
25	×1	831	Y	106.33	0.2901	
26	×2	1285	Y	257.03	0.3995	
27	×3	>1500	N	365.95	0.5127	
28	×5	>1500	N	330.02	0.6622	

Table 5
Used paths in experiment 14.

	Route 1	Route 2	Route 3	Route 4	Route 5	Route 6
Pre-assigned Path usage	1	1	1	1	1	1
	1	1	0	0	0	0
	0	1	0	0	0	0
	0	1	0	0	0	0
	0	0	0	1	0	0

Table 6
Used paths in experiment 18.

	Route 1	Route 2	Route 3	Route 4	Route 5	Route 6
Re-assigned Paths	2	4	1	8	2	2
Re-assigned Path usage	1	1	1	1	1	1
	1	1	–	0	0	0
	–	1	–	0	–	–
	–	1	–	0	–	–
	–	0	–	1	–	–
	–	–	–	1	–	–
	–	–	–	0	–	–
	–	–	–	1	–	–

The second source of inaccuracy is related to the path pre-assignment, reflected by the value of Agap − Agap-P. In an empty Sioux Falls network, the number of possible paths from one node to another can be many, and their travel costs do not differ largely. When loaded with higher demand, vehicles can quickly fill up the first several pre-assigned paths in one OD pair. Cheaper paths can emerge after assignment. In this situation, increasing the number of pre-assigned paths can increase the performance, as seen in Table 4 with demand ×2, ×3 and ×5. However, increasing the number of paths also increases the size of the problem and the computation time.

When the demand is low (such as ×1 or ×2) and vehicles do not fill up the pre-assigned paths, or when there are only limited paths between each OD in the network (such as in a motorway network), increasing the number of pre-assigned paths does not provide significant improvements to the solution: see experiment 1 or compare experiment 3 and 7. In this situation, all potential paths are included in the solving process and Agap = Agap-P, indicating that the path enumeration does not bring any inaccuracy. On the other hand, as our experiments with higher demands (such as × 5) have shown, the inaccuracy becomes higher when it considers a congested urban environment (such as Sioux Falls). Nevertheless, these limits are not related to the MILP formulation itself, but to the procedures built around it, namely the BPR function, the piecewise linearization and path pre-assignment, thus not diminishing the potential of the MILP formulation.

4.3.2. Assignment with initial results

In an exploration for better path enumeration mechanism, we manually pick paths according to the results from previous assignments (marked by “M” in Table 4). In these experiments, different number of paths may be manually assigned to each OD pair to improve solution quality and reduce solving time. For example, kth dijkstra assigned 5 paths to each OD pair in experiment 14 (Table 4). It took 51.75 s to solve. Table 5 shows which of the assigned paths are actually used (indicated by $a_{mwp} = 1$). This information can be used for a second assignment: if between 1 OD pair only the first path is used, we reduce the number of pre-assigned paths to 1. If the kth path is used, we set the number of pre-assigned paths larger than k. Applying this method to a second assignment (experiment 18), we have the results of path usage listed in Table 6. Compared with experiment 14 in Table 4, we see reductions in both solving time as well as Agap. Similar results are obtained from experiment 19, applying the path usage information from experiment 16; and from experiment 24, re-assigned with the results from experiment 23. The results indicate that pre-selecting the number of considered paths can provide much more precise solutions with shorter computation times (by a factor up to 250). This feature is particularly useful when conducting STA with an initial solution. It applies to assignments that evaluate different traffic management strategies and/or under different travel demand.

Table 7
Sizes and solving difficulties of MILP and MILP-SOS with different numbers of OD pairs.

Experiment	Total DV	Binary DV	Linear constraints	SOSs	Nodes
MILP (50-OD)	2900	832	5852	–	2593
MILP (100-OD)	3900	1132	7752	–	43943
MILP-SOS (50-OD)	1912	300	2356	152	1749
MILP-SOS (100-OD)	2912	600	4256	152	19292

Table 8
Computation time of MILP, MILP-SOS and MSA.

No. ODs	20	40	60	80	100	120
Solving time MILP (s)	0.19	6.08	20.67	73.84	374.34	3566
Solving time MILP-SOS (s)	0.11	1.83	9.33	3.92	66.67	48.48
Speed up factor	1.70	3.32	2.22	18.84	5.61	117.11
Solving time MSA (s)	59.2	119.97	144.87	156.10	209.23	233.84
Iterations MSA	700	955	1022	1035	1171	1213

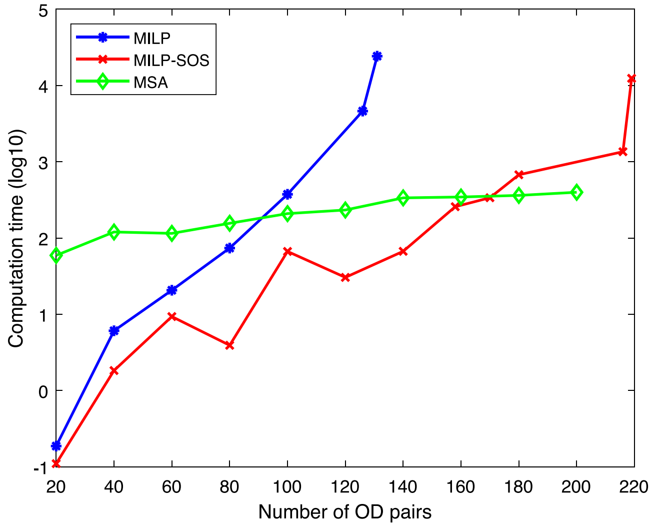

Fig. 6. Computation time of MILP with/without SOSs.

4.3.3. Scalability

We discuss the scalability of both MILP without and with the SOS feature (MILP, MILP-SOS). The benchmark experiment uses demand ×1, 4 segments for piecewise approximation, and 3 pre-assigned paths. Table 7 shows the size of the problem formulated in MILP and MILP-SOS with 50/100 ODs in terms of numbers of all decision variables, binary decision variables, linear constraints, SOSs, as well as numbers of nodes needed to solve the mixed-integer problems. The table indicates that MILP-SOS can largely reduce the problem size. Note that the number of binary decision variables in MILP-SOS is only related to the number of all considered paths; the number of SOSs is related to the number of links and the number of segments used for piecewise BPR. When the number of OD pairs doubles, the number of nodes required for MILP increases by 16 times; while for MILP-SOS it is by 10 times.

We gradually increase the size of the problem by adding more OD pairs into the problem. Fig. 6 and Table 8 show the evolution of computation time of solving MILP, MILP-SOS, and MSA. From them we can see that for smaller problems, MILP and MILP-SOS is faster than MSA. In terms of the speed-up factor brought by the SOS feature, in general we can say the speed-up factor is larger when solving larger problems. Both MILP and MILP-SOS will need more time solving larger scale problems. To scale up this method, specifically designed heuristics algorithms (such as column generation (Gamache et al., 1999)) should be studied.

5. Conclusion

Solving multi-class UE with mathematical programming formulations brings many benefits. However, the existing method, namely Beckmann Transformation, depends on a strong assumption of symmetry. This paper presents a novel MILP formulation based on mathematical programming but does not depend on such assumptions. The method makes use of a piecewise linearized BPR function and a kth shortest path enumeration along with the assignment process. We compare this formulation with other existing methods in the Sioux Falls network in terms of solving time and solution quality. The results suggest that the developed MILP is capable of finding good UE solution for smaller to medium networks. With a realistic demand, the Average Gap of MILP approach remains below 1.

We also explore approaches to improve the solving performance: a path-assignment approach is developed and can substantially improve the performance when an initial solution is available; a special order set method is used to reduce the number of the binary variables in the MILP problem and can accelerate computation. Nevertheless, our experiments show that the current solving methods do not scale. To apply the MILP formulation to larger problems more supplementary investigations are needed to speed up the computation, such as designing better path enumeration and solving algorithms. It is also interesting to apply this formulation in settings with more complexity such as dynamic traffic assignment.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work described in this paper is supported by the Dutch Research Council (NWO, The Netherlands) under the project “Freight Traffic Management as a Service” (FTMAAS, 439.18.456). No conflict of interests is declared. The authors deeply appreciate the comments given by the anonymous reviewers, which have been inspiring and have greatly improved the formulation and the quality of the manuscript.

References

Ameli, M., Lebacque, J.-P., Leclercq, L., 2021. Computational methods for calculating multimodal multiclass traffic network equilibrium: Simulation benchmark on a large-scale test case. J. Adv. Transp. 2021.

Beckmann, M.J., McGuire, C.B., Winsten, C.B., 1956. Studies in the economics of transportation. Tech. rep., Yale University Press.

Bliemer, M.C.J., 2001. Analytical Dynamic Traffic Assignment with Interacting User-Classes-Theoretical Advances and Applications using a Variational Inequality Approach (Ph.D. thesis). Delft University of Technology.

Bliemer, M.C., Bovy, P.H., 2003. Quasi-variational inequality formulation of the multiclass dynamic traffic assignment problem. Transp. Res. B 37 (6), 501–519.

Bliemer, M.C., Raadsen, M.P., Brederode, L.J., Bell, M.G., Wismans, L.J., Smith, M.J., 2017. Genetics of traffic assignment models for strategic transport planning. Transp. Rev. 37 (1), 56–78.

Boyce, D., Lee, D.-H., Ran, B., 2001. Analytical models of the dynamic traffic assignment problem. Netw. Spat. Econ. 1 (3–4), 377–390.

Gamache, M., Soumis, F., Marquis, G., Desrosiers, J., 1999. A column generation approach for large-scale aircrew rostering problems. Oper. Res. 47 (2), 247–263.

Keha, A.B., de Farias Jr., I.R., Nemhauser, G.L., 2004. Models for representing piecewise linear cost functions. Oper. Res. Lett. 32 (1), 44–48.

Lu, Z., Meng, Q., Gomes, G., 2016. Estimating link travel time functions for heterogeneous traffic flows on freeways. J. Adv. Transp. 50 (8), 1683–1698.

Müller, S., Schiller, C., 2015. Improvement of the volume-delay function by incorporating the impact of trucks on traffic flow. Transp. Plan. Technol. 38 (8), 878–888.

Nagurney, A.B., 1984. Comparative tests of multimodal traffic equilibrium methods. Transp. Res. B 18 (6), 469–485.

Nagurney, A., 1998. Network Economics: A Variational Inequality Approach. Vol. 10, Kluwer Academic Publishers.

Noriega, Y., Florian, M., 2007. Algorithmic approaches for asymmetric multi-class network equilibrium problems with different class delay relationships. Tech. rep., CIRRELT.

Peeta, S., Ziliaskopoulos, A.K., 2001. Foundations of dynamic traffic assignment: The past, the present and the future. Netw. Spat. Econ. 1 (3–4), 233–265.

Sbayti, H., Mahmassani, H.S., 2006. Optimal scheduling of evacuation operations. Transp. Res. Rec. 1964 (1), 238–246.

Soumis, F., Nagurney, A., 1993. A stochastic, multiclass airline network equilibrium model. Oper. Res. 41 (4), 721–730.

Szeto, W.Y., Jiang, Y., Sumalee, A., 2011. A cell-based model for multi-class doubly stochastic dynamic traffic assignment. Comput.-Aided Civ. Infrastruct. Eng. 26 (8), 595–611.

Szeto, W.Y., Lo, H.K., 2004. A cell-based simultaneous route and departure time choice model with elastic demand. Transp. Res. B 38 (7), 593–612.

Szeto, W., Wong, S., 2012. Dynamic traffic assignment: model classifications and recent advances in travel choice principles. Cent. Eur. J. Eng. 2 (1), 1–18.

U.S. Bureau of Public Roads, 1964. Traffic Assignment Manual for Application with a Large, High Speed Computer. US Department of Commerce.

Yun, S., White, W.W., Lamb, D.R., Wu, Y., 2005. Accounting for the impact of heavy truck traffic in volume–delay functions in transportation planning models. Transp. Res. Rec. 1931 (1), 8–17.