Optimizing Lane Reversals in Transportation Networks to Reduce Traffic Congestion: A global optimization approach

Salomón Wollenstein-Betech, Ioannis Ch. Paschalidis, and Christos G. Cassandras

Division of Systems Engineering and the Center for Information and Systems Engineering, Boston University, Boston, MA 02215 USA

Abstract

This paper studies how to reduce the overall travel time of commuters in a transportation network by reversing the direction of some lanes using a macroscopic network-wide perspective. Similar to the Network Design Problem, the lane reversal problem has been shown to be NP-hard given the dependence of the users' route selection on the lane direction decision. Herein, we propose and compare three efficient methods to solve the routing and lane reversal problem jointly. First, we introduce an alternating method that decouples the routing and lane assignment problems. Second, we propose a Frank-Wolfe method that jointly takes gradient steps to adjust both the lane assignment and routing decisions. Third, we propose a convex approximation method that uses a threshold-based approach to convexify the joint routing and lane reversal objective. The convex approximation method is advantageous since it finds a global optimum solution for the approximated problem and it enables the possibility to include linear constraints. Using this method, we extend the main formulation to be able to limit a maximum number of reversed lanes, as well as to incorporate multiple origin-destination (OD) patterns. We test the proposed methods in a case study using the transportation network of Eastern Massachusetts where our results indicate an overall reduction in travel times of 4.7% by selecting the best 15 reversals. Moreover, using a small test network, we investigate the performance of the lane reversal strategies as a function of the OD demand symmetry. As expected, we observe that when the OD demand is very asymmetric (e.g., for a single OD pair, evacuations, large events), the reduction in travel times is larger than the symmetric case, reaching travel time reductions of 60%.

Keywords: Intelligent Transportation Systems, Smart Cities, Contraflow Lane Reversal, Network Optimization.

1. Introduction

The rise in convenience of human mobility has brought traffic congestion to our cities and thus the imperative need for improving efficiency, reducing greenhouse gas emissions, and reducing travel times. While many advances have been made in this direction, like the creation of new modes of urban mobility services (e.g., bike sharing systems) and the emergence of Connected and Automated Vehicles (CAVs), there are still open questions related to the efficiency of *Intelligent Transportation Systems* (ITS).

One way to reduce traffic congestion without building new roads is to increase the network's capacity by dynamically adjusting the direction of the lanes of the transportation infrastructure. The idea consists of reversing the direction of some lanes in the network during a specific time interval, for example, for the morning peak period. These schemes are known as *Lane Reversals* (or *Contraflow Lane Reversals*) and

This research was supported in part by NSF under grants ECCS-1931600, DMS-1664644, CNS-1645681, IIS-1914792 and CCF-2200052, by AFOSR under grant FA9550-19-1-0158, by DOE under grants DE-AR0001282 and DE-EE0009696, by the MathWorks, by the ONR under grant N00014-19-1-2571, by the NIH under grant R01 GM135930, by NPRP grant (12S-0228-190177) from the Qatar National Research Fund, a member of the Qatar Foundation (the statements made herein are solely the responsibility of the authors). We thank D. Sverdlin-Lisker for proofreading this manuscript.

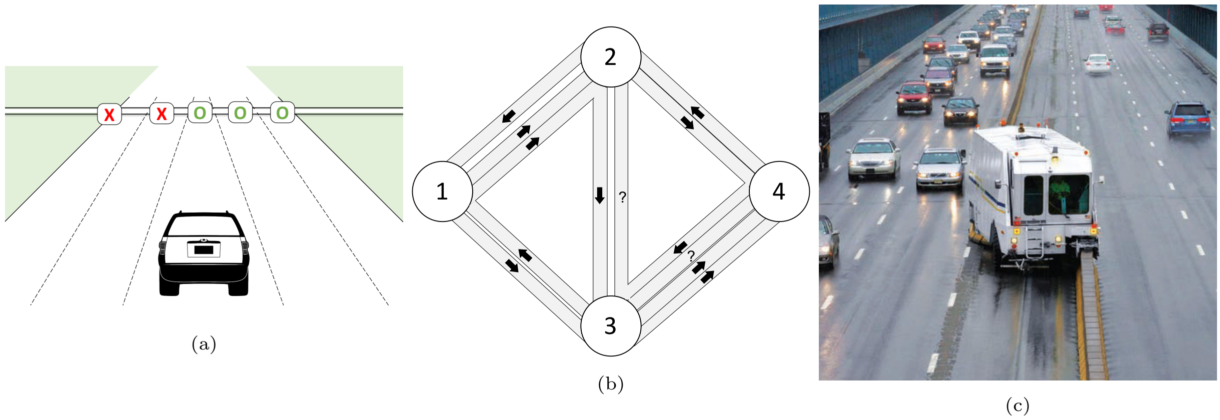


Figure 1: (a) Illustration of a typical lane reversal signal. (b) Diagram of the Braess' network with specific number of lanes and directions. (c) Barrier transfers (or road zippers).

have been already implemented in Mexico City, Montreal, and other cities during rush hour times. Solving the network-wide problem of identifying which are the best lanes to reverse is relevant for transportation agencies as it helps them plan the future transportation infrastructure by providing an estimate of the achievable benefits that lane reversals may provide.

Perhaps one of the most critical practical challenges when operating a lane reversal system is the communication and coordination with the drivers, since changing the direction of a lane too often may cause drivers' confusion. Current lane reversal systems built to reduce traffic congestion indicate the direction of the reversible lanes with overhead signals or road zippers, see Figs. 1a and 1c, respectively. Still, the infrastructure manager has to provide with structured (not too flexible) schedules to avoid confusing the drivers. Propitiously, the rise of CAVs can address such limitations by their ability to communicate with the infrastructure enabling the possibility to implement lane reversals more aggressively, either by doing it for more roads, or by dynamically changing the direction of a single road more often.

In this paper we take a macroscopic network flow planning perspective to address the lane reversal problem. Rather than focusing on when to reverse a particular lane, we are interested in finding the lane configuration that minimizes traffic congestion for a given Origin-Destination (OD) demand pattern. Figure 1b illustrates this view of the problem, where, given a network topology and an OD demand, we seek to find the best *lane directions* to minimize travel times. In Figure 1b, the link connecting nodes 2 and 3 has two lanes. One has been allocated to direction $(2, 3)$ and the other still needs to be determined.

Most of the literature concerned with contraflow lane reversals has concentrated on two main applications: (i) reversing lanes for evacuation routing during emergencies and (ii) alleviating traffic congestion. For evacuation planning, the problem has been studied using simulation and network flow models. Simulation methods by Jha et al. [1] and Theodoulou and Wolshon [2] showed that evacuation route capacity can be improved by 53% and 73%, respectively when designing appropriate lane reversals. These methods selected the reversals using the specific network knowledge from transportation managers. Different than simulation, network flow models presented by Cova and Johnson [3] formulate the problem as a mixed-integer program (MIP) and use a generic solver to find a solution. Their numerical results suggests evacuation time improvements in Salt Lake City of 30% to 40%. Similarly, Zhao et al. [4] and Xie and Turnquist [5] used the MIP heuristic Tabu search to solve the evacuation network problem while considering road intersections in the network representation. In addition to these numerical results, Kim et al. [6] showed that the network flow lane reversal problem is NP-complete and suggested using a greedy heuristic algorithm to find solution.

The research concerned with reducing traffic congestion has focused on reversing lanes for a single bottleneck road (e.g., tunnels and bridges) or for the full network. For single roads, rule-based [7], optimal control [8], discrete model predictive control [9], and fuzzy controllers [10, 11] have been proposed and they typically rely on the fundamental diagram of traffic flow [12]. More recently, and taking into account the advent of CAVs, a lane-free approach was proposed by Malekzadeh et al. [13] where a fluidic boundary is dynamically adapted depending on the number and type of vehicles present in a road. Finally, Levin and Boyles [14] used a microscopic cell-transmission model for which they solved the single lane reversal problem using a heuristic method based on congestion estimates. Their results show a 21.8% reduction in total system travel time. Although these methods provide useful techniques to manage a lane reversal system, they are implemented once a link has been selected to have reversible lanes.

Instead, network-wide methods seek to identify the best links to apply a lane reversal system in the network. To tackle this problem, most researchers have formulated it as a MIP. Unfortunately, these classes of problems are generally NP-hard. To address this issue, Chu et al. [15] used a distributed alternating direction method of multipliers (ADMM) to decompose the problem into smaller instances which are still integer programs. Their numerical results show that travel times for a small subnetwork of New York City could be improved by 61%. Hausknecht et al. [16] and Meng and Khoo [17] solved the networked reversal problem using genetic algorithms and report increases in efficiencies on the order of 72% for the city of Austin. Note that these models do not have guarantees on finding a global optimal solution. However, their results show that these algorithms work well in practice.

Relatedly, the network-wide lane reversal problem shares similarities with the Network Design Problem (NDP) whose objective is to minimize the total network travel time by identifying the best investments for the network given a limited budget. A main difference between the NDP and the lane reversal problem is that the NDP assumes that perturbations on the capacity of a link affect only that link. In contrast, when dealing with lane reversals we have to account for some *shared capacity* between opposite direction links, i.e., (i, j) and (j, i) which involves an additional relationship. A difficulty with NDPs, and the network-wide lane reversals, is that the objective is typically non-convex due to the interaction of the flow and capacity decision variables by a term x^p/z^q where x represents flow; z capacity; and p and q are known integers. Due to this issue, many algorithms have been proposed to find *local* optimal solutions for the NDP using pattern search [18] and gradient-based methods [19]. For a survey on this topic, we refer the reader to [20] and [21].

Closer to our work, four recent articles develop global optimal methods for an approximation of the NDP: Liu and Wang [22], Wang et al. [23], Wang and Lo [24] and Luathep et al. [25]. The first two convexify the x^p/z^q term by approximating it with $\ln(x^p/z^q) = p\ln(x) + z\ln(q)$ which generate two concave terms. Then, they use a tight linear approximation for the logarithmic function that reduces the problem to a mixed integer linear program (MILP). The latter two use a two-dimensional piecewise affine approximation of the term. Although theoretically this approximation works well, it requires many decision variables in the resulting MILP. In turn, our work uses a piecewise affine approximation only for the flow variables and we bundle the relationship with the capacity variable through a threshold-based function. Moreover, we account for additional constrains needed to solve the lane reversal rather than the NDP.

Finally, we note that our discussion so far focuses on the *static* traffic assignment and that there is a set of literature on NDPs for the *dynamic* traffic assignment [26, 27, 28]. The main two advantages of using the static traffic assignment over a dynamic one are that: (i) a solution can be computed efficiently even for large-scale networks; (ii) in many cases we can ensure *global*, rather than *local* optimality. Mostly for these reasons, this modelling framework has prevailed in the transportation community for decades and remains a central model for performing analysis of the network conditions.

The contribution of this paper is threefold. First, we provide three methods to solve the lane reversal problem while considering the routing decisions of commuters. The first method consists of decoupling the routing and lane assignment problems and solving them sequentially to reach a joint solution. Our second method approach uses a Frank-Wolfe algorithm which takes a gradient step for both capacity and routing while maintaining feasibility. Both of these methods can only aspire to *local* optimal solutions. Lastly, we propose a framework that convexifies the objective function such that the resulting problem is tractable and can be optimized to find the global minimum of the convexified problem.

Second, we emphasize extensions to our convex approximation approach as it enables writing the problem as a linear program (LP) or an integer linear program (ILP). This is convenient because it permits restricting the maximum number of allowable reversals, as well as selecting the best reversals of a network when considering multiple origin-destination demand patterns.

Our third contribution is to provide several case studies and numerical results for our methods. In particular we observe that for *asymmetric* OD demands, for example in the case of a single OD pair, the benefits of lane reversals are around 65% (matching the results obtained in [15, 16, 17]). In addition to this single OD pair example, we perform a case study using an OD demand matrix estimated using traffic data from the Eastern Massachusetts Area (EMA) transportation network. In this case, our results show that overall travel times in the network can be improved by 4.5%. Besides these numerical results, we observe that the greedy/incremental approach of reversing the best lane, then letting commuters adjust, and subsequently reversing the next best lane can reach a sub-optimal local minimum which is considerably higher than solving the problem jointly.

Building on our model and preliminary analysis in Wollenstein-Betech et al. [29], this paper: (i) Addresses a different and harder problem where we optimize *jointly* the routing decisions of commuters and the lane reversals rather than assuming fixed flows as in our previous analysis. (ii) Devises three families of algorithms to solve the proposed *joint* problem: *alternating*, *Frank-Wolfe*, and a relaxation to a *convex program*. (iii) Provides extensions to our convex approximation method to allow the selection of the best set of *links* to perform reversals for *multiple* OD demand matrices. (iv) Extends our numerical results section by comparing the proposed approach with additional methodologies.

The rest of the paper is structured as follows. In Section 2 we present the model preliminaries and the problem formulation. In Section 3 we introduce the three main methodologies employed to solve the lane assignment problem. Section 4 discusses extensions of our convex model; e.g., constraining the model to a maximum number of reversals and employing the model for multiple origin-destination demand patterns. Section 5 reports numerical results of the different methods over a case study using the Eastern Massachusetts Area (EMA) and a smaller test network, and Section 6 concludes.

2. Model and Problem Formulation

2.1. Preliminaries

We represent the transportation network using a directed graph denoted by $\mathcal{G}$ and composed by the set of *nodes* $\mathcal{V}$ and the set of *links* $\mathcal{A}$. We assume $\mathcal{G}$ to be *strongly-connected* such that every node is reachable from any other node in the network, and let the node-arc incidence matrix of $\mathcal{G}$ be $\mathbf{N} \in \{0, 1, -1\}^{|\mathcal{V}|\times|\mathcal{A}|}$. For every link (i, j) in $\mathcal{A}$, we denote its number of assigned lanes to be z_{ij}; its *capacity per lane* be c_{ij}; and its total capacity be $m_{ij} = z_{ij}c_{ij}$ expressed in vehicles per hour. To establish the relationship between *opposite direction* links, i.e., $(i, j), (j, i) \in \mathcal{A}$, we let n_{ij} be the maximum number of lanes in a *road* (i.e., n_{ij} is the maximum number of lanes a link can have if all lanes in the road are facing the same direction). Hence, $n_{ij} = n_{ji}$.

To model the user trips, let K be the number of Origin-Destination (OD) pairs and $\mathcal{W} = \{\mathbf{w}_k = (s_k, t_k) : k = 1, \ldots, K\}$ the set of all OD pairs. For every $\mathbf{w}_k$, let the demand rate of trips (veh/hr) from its source node $s_k \in \mathcal{V}$ to the target node $t_k \in \mathcal{V}$ be $d_{\mathbf{w}_k} \geq 0$.

In addition to user demand, we use $x_{ij}^{\mathbf{w}_k}$ to track the flow in every link $(i, j) \in \mathcal{A}$ associated with OD pair $\mathbf{w}_k$. Moreover, let x_{ij} represent the (aggregated) total link flow in (i, j), i.e.,

$$x_{ij} = \sum_{k=1}^{K} x_{ij}^{\mathbf{w}_k}, \qquad \forall (i, j) \in \mathcal{A}, \tag{1}$$

and let $\mathbf{x} = (x_{ij};\ (i, j) \in \mathcal{A})$ be the vector of all link flows.

To quantify the travel times in every link, let $t_{ij}(x_{ij}, z_{ij}) : \{\mathbb{R}_{\geq 0}, \mathbb{N}_{\geq 0}\} \mapsto \mathbb{R}_{\geq 0}$ be the *latency cost* (or travel time) function of link (i, j) which depends on the link's flow and on the number of lanes assigned to the link. Using the same structure as in [30], we characterize $t_{ij}(x_{ij}, z_{ij})$ as:

$$t_{ij}(x_{ij}, z_{ij}) = t_{ij}^0 f\Big(\frac{x_{ij}}{c_{ij}z_{ij}}\Big), \tag{2}$$

where $f(\cdot)$ is a strictly increasing, positive, and continuously differentiable function, and t_{ij}^0 is the free-flow travel time on arc (i, j). We set $f(0) = 1$, which ensures that if there is no constraint on the arc's capacity, the travel time t_{ij} equals the free-flow travel time t_{ij}^0. A widely used function by transportation engineers is the *Bureau of Public Roads (BPR)* function [31]

$$f(x_{ij}/m_{ij}) = 1 + \alpha(x_{ij}/m_{ij})^{\beta}. \tag{3}$$

A common choice is $\alpha = 0.15$ and $\beta = 4$. For a discussion on how to estimate these functions using flow data see [32]. Finally, we let the vector of travel latency functions be $\mathbf{t}(\mathbf{x}, \mathbf{z}) = (t_{ij}(x_{ij}, z_{ij});\ \forall (i,j) \in \mathcal{V})$ where $\mathbf{z} = (z_{ij};\ (i,j) \in \mathcal{A})$ is the vector of lane assignment variables.

In this paper, all vectors are column vectors and denoted by bold lowercase letters. We use "prime" to denote transpose, and use $\mathbf{1}$ to denote the vector whose entries are all ones. We refer to the cardinality of a set $\mathcal{S}$ by $|\mathcal{S}|$. For reference, in Appendix A we have included a notation table.

2.2. LASO-TAP Problem

Using the definitions presented above, we now can formulate the *Lane-Assignment System-Optimal Traffic Assignment Problem* (LASO-TAP) as follows:

$$\min_{\mathbf{x},\mathbf{z}} \quad \mathbf{t}(\mathbf{x}, \mathbf{z})'\mathbf{x} \tag{4a}$$

$$\text{s.t.} \quad \sum_{i:(i,j)\in\mathcal{A}} x_{ij}^{\mathbf{w}_k} - \sum_{\ell:(j,\ell)\in\mathcal{A}} x_{j\ell}^{\mathbf{w}_k} = \begin{cases} -d_{\mathbf{w}_k}, & \text{if } j = s_k, \\ 0, & \text{if } j \neq s_k, t_k, \\ d_{\mathbf{w}_k}, & \text{if } j = t_k, \end{cases} \quad \forall k = 1, \ldots, K, \quad \forall j \in \mathcal{V}, \tag{4b}$$

$$z_{ij} + z_{ji} \leq n_{ij}, \quad \forall (i,j) \in \mathcal{A}, \tag{4c}$$

$$\mathbf{x}^{\mathbf{w}_k} \in \mathbb{R}_+^{|\mathcal{A}|}, \quad \mathbf{z} \in \mathbb{N}_+^{|\mathcal{A}|}, \tag{4d}$$

where in the objective (4a) we are minimizing the overall travel time; constraint (4b) ensures that $\mathbf{x}$ is a feasible vector which complies with demand satisfaction and conservation of flow; constraint (4c) ensures that the number of assigned lanes does not exceed the maximum number of available lanes (we have an inequality instead of an equality to give more flexibility to the problem, however, it is possible to impose an equality constraint, if desired); and (4d) restricts $\mathbf{z}$ to be a vector of integer variables.

2.2.1. Origin-based formulation

The total number of variables introduced in the LASO-TAP formulation (4) is $|\mathcal{A}|(|\mathcal{W}| + 1)$, which is typically dominated by the number of OD pairs $|\mathcal{W}|$. In practice, this number can be very large, sometimes up to $|\mathcal{V}|^2$. Hence, solving (4) for real networks require large memory capabilities. To mitigate this problem, we aggregate flows by origin, similar to [33, 34]. This aggregation reduces the number of variables to $|\mathcal{A}|(1 + |\mathcal{V}|)$, which significantly improves the computation time.

Let the set of origins (sources) be $\mathcal{O} = \{s_k : \ d_{\mathbf{w}_k} > 0,\ k = 1, \ldots, K\}$ and let the flow vector with o as it source to all possible destinations be $\mathbf{x}^o$. The total user flow on a link is then $\mathbf{x} = \sum_{o\in\mathcal{O}} \mathbf{x}^o$ and we define the set of user origin-link variables to be $\mathbf{x}^{\mathcal{O}} = \{\mathbf{x}^o : \ o \in \mathcal{O}\}$. For every origin $o \in \mathcal{O}$ and every node $j \in \mathcal{V}$, let ϕ_{oj} be the node *imbalance* describing its excess demand or supply. This is:

$$\phi_{oj} = \begin{cases} \displaystyle\sum_{t_k:\mathbf{w}_k\in\mathcal{W}} -d_{\mathbf{w}_k}, & \text{if } j = o, \\ 0, & \text{if } (o,j) \notin \mathcal{W}, \\ d_{\mathbf{w}_k}, & \text{if } j = t_k \text{ and } s_k = o. \end{cases} \tag{5a}$$

Using these definitions, we can formulate the *origin-based* LASO-TAP as follows:

$$J_{\texttt{MINLP}} = \min_{\mathbf{x}^{\mathcal{O}},\mathbf{z}} \quad \mathbf{t}(\mathbf{x},\mathbf{z})'\mathbf{x} \tag{6a}$$

$$\text{s.t.} \quad \sum_{i:(i,j)\in\mathcal{A}} x_{ij}^o - \sum_{\ell:(j,\ell)\in\mathcal{A}} x_{j\ell}^o = \phi_{oj} \quad \forall j \in \mathcal{V},\ \forall o \in \mathcal{O}, \tag{6b}$$

$$z_{ij} + z_{ji} \leq n_{ij}, \quad \forall (i,j) \in \mathcal{A}, \tag{6c}$$

$$\mathbf{x}^{\mathbf{w}_k} \in \mathbb{R}_+^{|\mathcal{A}|}, \quad \mathbf{z} \in \mathbb{N}_+^{|\mathcal{A}|}. \tag{6d}$$

Despite the reduction from (4) to (6), the LASO-TAP remains hard to solve for several reasons. First, the interaction between the decision variables in (4a) and (6a) makes the objective non-convex. This comes from the fact that when multiplying (3) by x_{ij} we get the term $\gamma_{ij}x_{ij}^{\beta+1}/z_{ij}^{\beta}$ where $\gamma_{ij} = \alpha t_{ij}^0/c_{ij}^{\beta}$. Second, we are optimizing over a set of integer variables $\mathbf{z}$ which increases the computational complexity of the problem. Still, (6) reduces the dimensionality of (4) and makes the problem more manageable to solve. In the following sections we present modifications to this problem formulation to enable efficient solution methods.

The problem as stated in (4) and (6) is a lane *assignment* problem rather than a lane *reversal* problem. This distinction is negligible when we are capable to reverse all lanes in the network. In reality, the transportation infrastructure is not very flexible and might not be able to handle many lane reversals. Hence, we are interested in considering a *sparse* lane assignment problem in which we limit the number of lane reversals. We provide this extension in Section 4.

Remark 1. Note that the LASO-TAP as stated in (4) and (6) is seeking a System-Optimal (SO) solution to the Traffic Assignment Problem (TAP). We can expect a SO user-behavior when vehicles collaborate with each other when deciding their routes. This SO user-behavior has been investigated for the presence of CAVs showing promising time improvements [35, 36]. An interesting theoretical result establishes that by slightly modifying the objective function, the same algorithms that solve a SO TAP are capable of solving the User-Centric (UC) TAP. In particular, Dafermos and Sparrow [37, Thm. 1.5] discusses the relationship between the solution to these two problems where they show that if a solver is available for the SO problem, then it can be used to find a solution to the UC problem by replacing every element of the objective $x_{ij}t_{ij}(x_{ij}, z_{ij})$ with $\int_0^{x_{ij}} t_{ij}(s, z_{ij})ds$. Analogously this can also be seen by employing the Beckmann et al. [30] UC formulation where the objective is to minimize $\sum_{(i,j)\in\mathcal{A}} \int_0^{x_{ij}} t_{ij}(s, z_{ij})ds$. For the classical BPR function specified in Equations (2) and (3) we have that $\int_0^{x_{ij}} t_{ij}(s, z_{ij})ds = t_{ij}^0 x_{ij}\big(1 + \frac{\alpha}{\beta+1}(\frac{x_{ij}}{c_{ij}z_{ij}})^{\beta}\big)$. In this paper we will mainly focus on the SO case, but our framework can handle both SO and UC TAPs. A comparison between the solution of these two is provided in Section 5.8.

3. Methods

We have discussed how the LASO-TAP problem is hard to solve for several reasons. First, the objective is non-convex. Second, the optimization is made over a set of integer variables $\mathbf{z}$. Thus, let us consider three main strategies to find effective solutions to the problem. To describe these strategies, we first define a condition that will help us identify if a current solution is at an *equilibrium point* or not. Intuitively, we say that Problem (6) is at an equilibrium point if the objective does not improve when we perform a single reversal in the network. We now state this formally.

Definition 1. *An equilibrium point of problems* (4) *and* (6) *is found if for all* $(i,j) \in \mathcal{A}$ *the following conditions holds*

$$\psi_{ij} \geq 0 \quad \forall z_{ij} = 0, \ldots, n_{ij} - 1, \ \textit{and}$$
$$\psi_{ij} \leq 0 \quad \forall z_{ij} = n_{ij},$$

where ψ_{ij} captures the difference on the overall travel times between the current solution $(\mathbf{x}, \mathbf{z})$ and the solution when we perform a reversal at link (i, j). Formally, let $\mathbf{x}^{+ij}$ indicate the updated flow vector when we add one lane to (i, j) and subtract a lane from (j, i). Similarly, let $\mathbf{z}^{+ij}$ be the updated lane configuration where $z_{kl}^{+ij} = z_{kl} + \mathbb{1}_{(k,l)=(i,j)} - \mathbb{1}_{(l,k)=(j,i)}$ for $(k, l) \in \mathcal{A}$. Then, ψ_{ij} is defined as

$$\psi_{ij} = \mathbf{t}(\mathbf{x}^{+ij}, \mathbf{z}^{+ij})'\mathbf{x}^{+ij} - \mathbf{t}(\mathbf{x}, \mathbf{z})'\mathbf{x}. \tag{7}$$

Note that to check whether this condition holds or not, we have to run a TAP to get $\mathbf{x}^{+ij}$. Therefore, to evaluate the vector $\boldsymbol{\psi} = (\psi_{ij}; (i, j) \in \mathcal{A})$ we require solving $|\mathcal{A}|/2$ TAPs. Checking this condition in a sequential algorithm is computationally expensive, but could be used regularly as a measure of closeness to an equilibrium point. Note that many equilibrium points may be encountered and that assessing the global optimality of such a point may be a difficult and often an intractable task.

In the following subsections we present three families of algorithms that could be used to efficiently solve the LASO-TAP problem. First, we introduce an alternating method which consists of decoupling the lane and traffic assignment problems. Next, we provide a *feasible direction* (or *Frank-Wolfe*) algorithm which reduces the complexity of the problem by solving a fluidic version of it. Finally, we formulate the problem using a convex approximation for which we develop a piecewise-linear method to find its global minimizer.

3.1. Alternating method

We consider the method of sequentially and iteratively solving two disjoint and easier problems. The idea consists of first solving the traffic assignment problem (TAP) using any of the standard methods (e.g., Method of Successive Averages, Frank-Wolfe, among others) and then solving for the best *lane assignment* (LA) for those specific flows. With the new lane allocation, we re-solve the TAP and LA problems and repeat this procedure until convergence. The LA problem of allocating lanes for a fixed flow vector $\mathbf{x}$ is defined as

$$J_{\text{IP}}^{\mathbf{z}} = \min_{\mathbf{z}} \quad \mathbf{t}(\mathbf{x}, \mathbf{z})'\mathbf{x} \tag{8a}$$

$$\text{s.t.} \quad z_{ij} + z_{ji} \leq n_{ij}, \quad \forall (i, j) \in \mathcal{A}, \tag{8b}$$

$$\mathbf{z} \in \mathbb{N}_{+}^{|\mathcal{A}|}, \tag{8c}$$

where the objective (8a) only depends on $\mathbf{z}$. Problem (8) is an integer programming problem with a convex objective and linear constraints. The convexity comes from the fact that each element $J_{ij}^{\mathbf{z}}$ is convex in z_{ij} when we use a BPR function of the form of (3). This is because its second derivative is nonnegative for all $z_{ij} \geq 0$, and by the fact that the summation of convex functions is convex. Moreover, $J_{ij}^{\mathbf{z}}$ is monotonically decreasing as it decreases when we increase the capacity of arc (i, j).

For a deeper discussion on how to efficiently solve (8) we refer to [29]. However, one can see that (8) is separable since both the objective and constraints are separable for each pair of opposite direction links (i, j) and (j, i). This enables the possibility of distributively solving this problem and obtaining the global optimal solution. Solving the lane assignment problem is thus on order of $\mathcal{O}(\|\mathbf{n}\|_\infty)$ where $\mathbf{n} = (n_{ij};\ (i, j) \in \mathcal{A})$. This complexity is only true when we allow to reverse any lane in the network. However, if we are interested in including other coupling constraints, for example constraining a maximum number of reversals, then we would have to solve the full convex integer programming problem. In [29] we show how this problem can be reformulated into a linear programming problem.

The family of algorithms presented herein consists of sequentially solving TAP and LA until convergence. We refer to a *family* of algorithms since we can restrict the LA problem to have a maximum of reversals at every iteration. For example if we restrict the LA to only select the best reversal at every iteration, we would end up with a greedy algorithm. This *greedy* approach will switch the best lane, then will optimize flows, and then will look for the next best reversal, and so forth. This approach is natural when urban planners reverse a lane, i.e., wait to see how traffic responds, re-assess the network conditions, and select the next most congested link.

3.2. Frank-Wolfe method

A different approach to solve the LASO-TAP problem is to relax the integer variables to continuous ones. This relaxation is often employed when dealing with integer programming problems and has the benefit of generating a lower bound. This relaxed problem leads to a non-convex continuous programming problem for which standard methods can be used to find a local stationary point. In our context, we first derive an equilibrium condition for the relaxed case which is analogous to the one described in Definition 1. We quantify the impact on the objective when we reverse an infinitesimal fraction of a lane while assuming that flows $\mathbf{x}$ remain unchanged. Its impact on the objective can be estimated using

$$\frac{\partial J^{\mathbf{z}}}{\partial z_{ij}} = \frac{\partial}{\partial z_{ij}}\Big(x_{ij}t_{ij}(x_{ij}, z_{ij}) - x_{ji}t_{ji}(x_{ji}, n_{ij} - z_{ij})\Big). \tag{9}$$

Then, an equilibrium point for the capacity variables $\mathbf{z}$ is that for all (i, j) in $\mathcal{A}$:

$$\begin{aligned}
\partial J^{\mathbf{z}}/\partial z_{ij} = 0, &\quad \text{for } z_{ij} \in (0, n_{ij}),\\
\partial J^{\mathbf{z}}/\partial z_{ij} \geq 0, &\quad \text{for } z_{ij} = 0,\\
\partial J^{\mathbf{z}}/\partial z_{ij} \leq 0, &\quad \text{for } z_{ij} = n_{ij}.
\end{aligned}$$

That is, there is no benefit to reversing an infinitesimal unit of capacity for any arc (i, j) for a current flow solution.

Since the relaxed problem is not convex due to the interaction between the flow and capacity variables in $f(x_{ij}/m_{ij})$, we cannot ensure that the stable point will be a *global* optimal but rather a *local* optimal solution. We argue that this local solution is better than the initial allocation since at every step we aim to improve the overall travel time by modifying $\mathbf{z}$.

To solve the joint $(\mathbf{x}, \mathbf{z})$ problem we consider an *enlarged* Frank-Wolfe algorithm which is similar to the one used for solving the TAP. At every iteration we find the best routing decisions based on the current status of the infrastructure and then we immediately take a step of adjusting the capacity for the current flow solution. Algorithm 1 provides a detailed description of this methodology. When the algorithm terminates, we simply project the fluidic (continuous) variables to its nearest integer by applying $\Pi(\mathbf{z})$ defined in Step 11 of Algorithm 1. Note that, similarly to the Frank-Wolfe methodology for solving TAP, this method is memory-efficient [38].

3.3. Convex approximation

The goal of this section is to develop an approximation to the problem such that efficient algorithms can be employed to find a *global* optimal solution instead of a *local* solution. The key idea is transform the interaction in the objective of x_{ij} with the capacity m_{ij} such that we construct a convex objective. To do so, we set each link's capacity using a fixed (nominal or current) number of lanes z^0_{ij} (its vectorized version $\mathbf{z}^0$) where $z^0_{ij} + z^0_{ji} = n_{ij}$. Then, we relate the new lane assignment with the flow in a threshold-based fashion. Specifically, we propose the following objective:

$$\min_{\mathbf{x},\mathbf{z}} \quad \mathbf{t}(\mathbf{x}, \mathbf{z}^0)'\mathbf{x} + \lambda\|\max\{\mathbf{0}, \mathbf{x} - \boldsymbol{\Theta}(\mathbf{z})\}\|_2. \tag{10}$$

where λ is a regularizer that trades off routing efficiency (first term) against not exceeding a link's capacity (second term) and $\boldsymbol{\Theta} = (\Theta_{ij}(z_{ij}); (i, j) \in \mathcal{A})$ is a vector of affine functions pointing to capacity thresholds. Note that the objective in (10) seeks a joint solution that avoids that the flow x_{ij} exceeds the capacity threshold $\Theta_{ij}(z_{ij})$. An intuitive function would be to consider $\Theta_{ij}(z_{ij}) = c_{ij}z_{ij}$, which points to the capacity of the link $(i, j) \in \mathcal{A}$ when z_{ij} lanes are assigned to it. However, we allow the more general case in which any other affine $\Theta_{ij}(z_{ij})$ functions can be employed, e.g., a fraction of the link's capacity that the practitioner would like to avoid exceeding.

Algorithm 1 LASO Frank-Wolfe

1: *Initialization.* Set counter $l := 1$. Perform all-or-nothing assignment with $\mathbf{t}^l = \mathbf{t}(\mathbf{0}, \mathbf{z}^0)$. This yields $\mathbf{x}^l$.
2: *Update SO travel time.* $\mathbf{t}^{l+1} \leftarrow \mathbf{t}(\mathbf{x}^l, \mathbf{z}^l) + \mathbf{x}^l \nabla_{\mathbf{x}^l} \mathbf{t}(\mathbf{x}^l, \mathbf{z}^l)$.
3: *Capacity direction finding.* Obtain $\nabla J^{\mathbf{z}}(\mathbf{z}^l)$ using (9).
4: *Capacity step size selection.* Use $\alpha_1^l = \alpha_1^0 / l$.
5: *Move capacity.* $\mathbf{z}^{l+1} \leftarrow \mathbf{z}^l - \alpha_1^l \nabla J^{\mathbf{z}}(\mathbf{z}^l)$.
6: *Flow direction finding.* Perform All-or-nothing assignment with $\mathbf{t}^{l+1}$ and $\mathbf{z}^{l+1}$ and get *auxiliary* flows $\mathbf{y}^{l+1}$.
7: *Flow step size selection.* Use line search and select α_2 by solving

$$\min_{0 \le \alpha_2 \le 1} \sum_{(i,j) \in \mathcal{A}} \int_0^{x_{ij}^l + \alpha_2 (y_{ij}^l - x_{ij}^l)} t_{ij}^{l+1}(\omega, z_{ij}^{l+1}) d\omega$$

.

8: *Move flow.* $\mathbf{x}^{l+1} \leftarrow \mathbf{x}^l + \alpha_2(\mathbf{y}^l - \mathbf{x}^l)$.
9: *Relative gap.* Calculate the Relative Gap (RG) using

$$\text{RG} = \frac{\big(\mathbf{x}^{l+1}\big)' \big(\mathbf{t}(\mathbf{x}^{l+1}, \mathbf{z}^{l+1}) + \nabla_{\mathbf{x}^{l+1}} \mathbf{t}(\mathbf{x}^{l+1}, \mathbf{z}^{l+1})\big)}{\sum_{k=1}^K d_k h_w} - 1$$

where h_k is the SO shortest travel time (i.e., $\mathbf{t}(\mathbf{x}^l, \mathbf{z}^l) + \nabla_{\mathbf{x}^l} \mathbf{t}(\mathbf{x}^l, \mathbf{z}^l)$) from w_{sk} to w_{tk}.

10: *Stopping criterion.* If RG $< \xi_1$ and $\|\nabla J^{\mathbf{z}}(\mathbf{z})\|_2 \le \xi_2$ or $l > \mathcal{L}$ then continue to Step 11. Otherwise, let $l = l + 1$ and go to Step 2. ξ_1, ξ_2, and $\mathcal{L}$ are input parameters that restrict the level of accuracy of the solution. ξ_1 and ξ_2 identify the maximum desired closeness to a routing equilibrium and a lane assignment equilibrium, respectively; in turn, $\mathcal{L}$ indicates the maximum number of iterations to terminate the algorithm.
11: *Project* $\mathbf{z}^{l+1}$ *to closest integer*: $\mathbf{z}^{\text{int}} = \Pi(\mathbf{z}^{l+1})$ and output $(\mathbf{x}^{l+1}, \mathbf{z}^{\text{int}})$. To avoid computational issues, we will let $\mathbf{z} \ge 1$. To do so let $\Pi(\mathbf{z}) = (\pi(z_{ij})$ for $(i,j) \in \mathcal{A})$ and $\pi(z_{ij}) = \lfloor z_{ij} \rceil + \mathbb{1}_{\lfloor z_{ij} \rceil = 0} - \mathbb{1}_{\lfloor z_{ji} \rceil = 0}$.

Using this objective and the inclusion of slack variables $\mathbf{s} = (s_{ij};\ (i,j) \in \mathcal{A})$ pointing to the flow exceeding the threshold $\mathbf{\Theta}(\mathbf{z})$, we formulate an approximation to the LASO-TAP problem as follows:

$$J_{\texttt{MICVX}} = \min_{\mathbf{x}, \mathbf{z}, \mathbf{s}} \quad \mathbf{t}(\mathbf{x}, \mathbf{z}^0)'\mathbf{x} + \lambda \|\mathbf{s}\|_2 \tag{11a}$$

$$\text{s.t.} \quad \sum_{i:(i,j) \in \mathcal{A}} x_{ij} - \sum_{\ell:(j,\ell) \in \mathcal{A}} x_{j\ell} = \phi_j \quad \forall j \in \mathcal{N}, \tag{11b}$$

$$z_{ij} + z_{ji} \le n_{ij}, \quad \forall (i,j) \in \mathcal{A}, \tag{11c}$$

$$\mathbf{s} \ge \mathbf{x} - \mathbf{\Theta}(\mathbf{z}), \tag{11d}$$

$$z_{ij} \in \{0, 1, \ldots, n_{ij}\}, \quad \forall (i,j) \in \mathcal{A}, \tag{11e}$$

$$\mathbf{x},\ \mathbf{s} \ge 0. \tag{11f}$$

which results in a convex mixed integer program with linear constraints.

3.3.1. Piecewise-affine approximation

Following a similar approach as in [36], we consider approximating the travel latency function, i.e., $t_{ij}(x_{ij}, z_{ij}^0)$, with a piecewise affine function as in Figure 2. For every piecewise segment in the range $\theta_{ij}^{(l)} \le x_{ij} \le \theta_{ij}^{(l+1)}$ we introduce a slack variable $\varepsilon_{ij}^{(l)}$. Hence, $t_{ij}(x_{ij}, z_{ij}^0)$ is approximated by

$$\hat{t}_{ij}(\boldsymbol{\varepsilon}_{ij}, z_{ij}^0) := t_{ij}^0 \left(1 + \sum_{l=1}^L \frac{a_l}{z_{ij}^0} \varepsilon_{ij}^{(l)}\right),$$

where $a_1 < a_2 < \cdots < a_L$ are the slopes of the L segments in the piecewise-affine approximation and $\boldsymbol{\varepsilon}_{ij} = (\varepsilon_{ij}^{(l)}; l = 1, \ldots, L)$. Using this function we obtain the quadratic objective

$$\hat{T}_{ij}(\boldsymbol{\varepsilon}_{ij}) := \hat{t}_{ij}(\boldsymbol{\varepsilon}_{ij}, z_{ij}^0)(\mathbf{1}'\boldsymbol{\varepsilon}_{ij}) \approx t_{ij}(x_{ij}, z_{ij}^0)x_{ij},$$

where $\mathbf{1}'\boldsymbol{\varepsilon}_{ij} = x_{ij}$ (recall that $\mathbf{1}$ is a vector of all ones) and we define $T_{ij} := t_{ij}(x_{ij}, z_{ij}^0)x_{ij}$. In [36] we showed that this piecewise affine approximation for the classical TAP results in a convex quadratic program (QP) that could be further approximated by a linear program (LP). Using this piecewise affine approximation, we formulate the LASO-TAP problem as a mixed integer quadratic programming problem (MIQP). To ease notation we let $\hat{\mathbf{T}}(\mathcal{E}) = (\hat{T}_{ij}(\boldsymbol{\varepsilon}_{ij});\ (i,j) \in \mathcal{A})$. Then, the problem is written as

$$\begin{aligned} J_{\texttt{MIQP}} = \min_{\boldsymbol{\varepsilon},\mathbf{z},\mathbf{s}} \quad & \mathbf{1}'\hat{\mathbf{T}}(\mathcal{E}) + \lambda\|\mathbf{s}\|_2 && \text{(12a)}\\ \text{s.t.} \quad & \sum_{i:(i,j)\in\mathcal{A}} \mathbf{1}'\boldsymbol{\varepsilon}_{ij} - \sum_{k:(j,k)\in\mathcal{A}} \mathbf{1}'\boldsymbol{\varepsilon}_{jk} = \phi_j, \quad \forall j \in \mathcal{N}, && \text{(12b)}\\ & 0 \le \epsilon_{ij}^{(l)} \le \theta_{ij}^{(l+1)} - \theta_{ij}^{(l)}, \quad l = 1, \ldots, L, && \text{(12c)}\\ & z_{ij} + z_{ji} \le n_{ij}, \quad \forall (i,j) \in \mathcal{A}, && \text{(12d)}\\ & \mathbf{s} \ge \mathbf{x} - \Theta(\mathbf{z}), && \text{(12e)}\\ & z_{ij} \in \{0, 1, \ldots, n_{ij}\}, \quad \forall (i,j) \in \mathcal{A}, && \text{(12f)}\\ & \mathbf{s} \ge 0. && \text{(12g)} \end{aligned}$$

An interesting result of this formulation is that we can formulate this problem as a mixed integer linear program (MILP) by approximating the routing part of the objective ($\mathbf{1}'\hat{\mathbf{T}}(\mathcal{E})$) as in [36] and by penalizing the slack variables $\mathbf{s}$ with an ℓ_1-norm instead of an ℓ_2-norm, we let the objective of the MILP be $J_{\texttt{MILP}}$.

Although the problem we are dealing with is NP-hard due to the integrality of $\mathbf{z}$, exact methods such as branch and bound have been found to perform well in practice. However, in order to guarantee tractability, we can always relax $\mathbf{z}$ by letting it be a non-negative continuous variable. When considering this relaxation, we can handle it in two different ways. First, we can think of the non-integer part of the solution as a percentage of the time in which the lane is reversed. For example, if the solution indicates that an optimal lane configuration is equal to $z_{12} = 4/3$ and $z_{21} = 5/3$ for a 3 hours period, we can think that for lane $(1,2)$ will have 1 lane assigned for the full period and only for one hour will be assigned a second lane. Similarly, link $(2,1)$ will have 1 lane assigned during the whole period and a second lane will be assigned for 2 hours. This interpretation can also be employed for the continuous solution of the FW method in Section 3.2 and Algorithm 1 before performing the projection step. The second approach for which this continuous solution can be mapped to a feasible solution is by projecting the continuous solution to the closest integer (as in the proposed Frank-Wolfe method). We let the cost of the *relaxed* MIQP and MILP be $J_{\texttt{QP}}$ and $J_{\texttt{LP}}$, respectively. Moreover, we let the objective of the projections of the solution of QP and LP to the closest integer be $J_{\Pi(\texttt{QP})}$ and $J_{\Pi(\texttt{LP})}$, respectively.

We conclude by observing that this convex approximation approach gives us four different methodologies: the integer-based NP-hard methods MIQP and MILP; and the polynomial-time methods QP and LP with their appropriate projections. In addition, in the following lemmata we formalize the relationship of the solutions of these convex approximation models using classical results from integer programming theory [39] and affine approximations [36].

Lemma 3.1. *For the QP we have that $J_{QP} \le J_{MIQP} \le J_{\Pi(QP)}$. Similarly, for the LP we have $J_{LP} \le J_{MILP} \le J_{\Pi(LP)}$.*

Proof. Since the LP and QP are relaxations of the original MILP and MIQP, $J_{\texttt{QP}}$ will generate a lower-bound to $J_{\texttt{MIQP}}$ since the feasible set of the MIQP is a subset of the feasible set of QP. Moreover, when projecting to the closest integer, $J_{\Pi(\texttt{QP})}$ will provide an upper-bound for $J_{\texttt{MIQP}}$ since any feasible solution is an upper-bound to the optimal solution. The same argument follows for the LP models. □

Using these results, we can solve a relaxed problem (LP or QP) and its projection in polynomial time and use the results to assess the performance of the solution by computing the ratio between the upper and lower bounds, i.e., $J_{\Pi(\mathrm{LP})}/J_{\mathrm{LP}}$. This metric provide an upper bound on the suboptimality of the solution $J_{\Pi(\mathrm{LP})}$.

Lemma 3.2. *Let the OD demands $d_{\mathbf{w}_k}$, thresholds $\theta_{ij}^{(l)}$, the capacities n_{ij} be non-negative and bounded by above. Let the slope parameters in $\hat{t}_{ij}$ follow $a_1 < a_2 < \cdots < a_L < \infty$; and $\hat{T}_{ij}^{LP}(\boldsymbol{\varepsilon}_{ij})$, $\hat{T}_{ij}^{QP}(\boldsymbol{\varepsilon}_{ij})$ be the affine approximation functions for the LP and QP models as defined in [36], respectively. Then, as $L \to \infty$, we have that $\hat{T}_{ij}^{LP}(\boldsymbol{\varepsilon}_{ij}) \to \hat{T}_{ij}^{QP}(\boldsymbol{\varepsilon}_{ij})$ for all $(i,j) \in \mathcal{A}$.*

Proof. The proof follows Lemma 1 and Theroem 2 in [36]. □

This result establishes the relationship between the first element of the objective in (12) for the LP and QP models. It suggests that the QP yields a better approximation of the original function $T(x)$ than the LP problem. Moreover, it indicates that as the number of linear segments, L, increases, the LP approximation is closer to the QP approximation. Hence, for small L's, it is desirable to use the QP formulation in terms of the solution's quality. However, the LP provide benefits with respect to computational times (as it is later shown in Section 5.7).

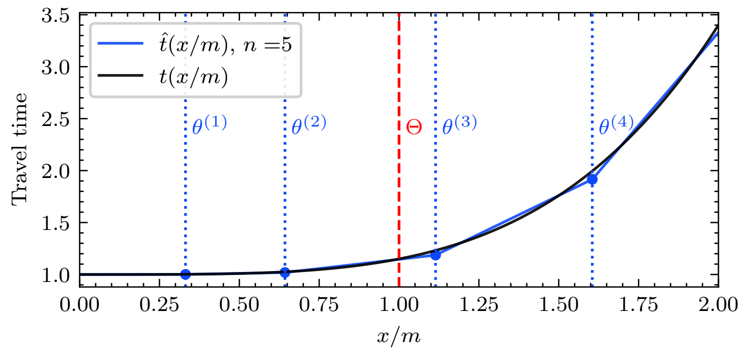


Figure 2: Piecewise affine approximation of the travel time function

3.4. Discussion

A few comments are relevant to consider. First, one should note that contraflow lane reversals could violate our assumption regarding the strong connectivity of $\mathcal{G}$. This may happen since we could disconnect certain nodes in the network while performing a lane reversal. One way to avoid this issue is to restrict the solution to maintain at least one lane in each direction of the road. While this approach maintains connectivity, it may sacrifice some efficiency. An alternative way to handle this issue is by assuming a tiny amount of demand for every pair of nodes in the network. In this case, the algorithm will avoid disconnecting the nodes as this tiny flow will incur an infinite cost. In this way, we ensure that there is a path connecting any two nodes. Note however, that this trick may increase the dimensionality of the problem by increasing the number of variables. This is because now every node will become an origin in (6).

Second, the convexified version requires tuning λ. Although there is no specific way to set λ, we observe that it is trading off routing with not exceeding a link's capacity. Fortunately, these two objectives, routing and not exceeding a link's capacity, are somehow aligned because as flows get closer to a link's capacity, they also become less attractive for routing purposes due to higher travel times (see Figure 2).

Third, we would like to argue the flexibility that the convexified (the MIQP, MILP, QP and LP) approaches provide in comparison with the alternating and Frank-Wolfe methods. Their main advantage is that they allow the inclusion of additional linear constraints to our problem. This offers the possibility to extend the framework to more realistic and practical examples. Another advantage of the convex approaches, and more specifically the LP approach, is the ability to provide with very low effort sensitivity analysis of the solution with respect to the demand vector. Up to now, we have assumed that we have perfect knowledge of the OD demand. However, this is typically not the case. In addition to sensitivity analysis, the LP method also allows us to extend this work to handle *robust* (or uncertainty-aware) instances of the problem in which the OD demand lies inside an uncertainty set and we aim to choose the best lane configuration of the worst selection of OD demands [40]. Our LP formulation can accommodate this with relatively low effort using [41]. Finally, commercial software for solving LPs and QPs is widely available and has been improved over the years such that it is quite efficient to compute a solution to the problem using this strategy.

4. Extensions

In the previous section we introduced three main methodologies to solve the lane assignment problem while considering the routing decisions of commuters. Now, we exploit the convex approximation methodology to extend the framework to be suitable for more practical implementations, especially to limit the number of reversals either by limiting the number of *lanes* to reverse or limiting the number of *links* (roads) to reverse. Imposing a maximum number of reversals becomes relevant for several reasons. First, when considering the case of human-driven vehicles, we would like to have fewer reversible roads in the network to avoid confusion. Second, if the transportation agency is planning to invest on infrastructure in these roads, e.g., surveillance cameras, barrier transfer machines (see Fig. 1c), then the agency will limit the number of reversible lanes/links in the network due to budget constraints. The extensions presented herein are suitable for the MIQP and MILP frameworks but are not suitable for LP and QP. This is because we rely on inequalities that are well-defined for the integer variables but not for the continuous ones.

4.1. Maximum number of lane reversals

So far we have selected the lane configuration such that it complies with the constraint $z_{ij} + z_{ji} = n_{ij}$. Note that this does not consider the *current* infrastructure status and it assigns lanes to links regardless of implementation costs. However, transportation infrastructure is, in general, not flexible to perform many reversals during a day. Therefore, we would like to limit the number of allowable lane reversals. To achieve that, we seek to ensure that our solution does not deviate too much from the *nominal* (or *current*) configuration. This can be modeled as $|z_{ij}^0 - z_{ij}| \leq \xi$ where z_{ij}^0 is the current lane allocation in (i, j) and ξ is the maximum number of lane reversals allowed. Then, using the convex formulation (12) we can simply add for each link $(i, j) \in \mathcal{A}$, a slack variable $r_{ij} \geq 0$ and the linear constraints $r_{ij} \geq z_{ij}^0 - z_{ij}$; $r_{ij} \geq -(z_{ij}^0 - z_{ij})$; and $\sum_{(i,j)\in\mathcal{A}} r_{ij} \leq \xi$.

4.2. Maximum number of link reversals

In a similar way to the previous subsection, suppose now that the planner would like to limit the number of allowable *links* (instead of lane) reversals. This is relevant because when investing in reversal infrastructure for a link/road, the cost of reversing one versus multiple lanes may not be significant as compared to the cost of investing in infrastructure for multiple lanes in different roads. To model this, we introduce for every arc $(i, j) \in \mathcal{A}$ the variable

$$q_{ij} = \min\{1, \ |z_{ij}^0 - z_{ij}|\} = \max\{-1, -r_{ij}\}. \tag{13}$$

where r_{ij} is defined as in Section 4.1. To introduce this to our MIQP or MILP, we let ξ be the number of allowable *link/road* reversals and we add the slack variables q_{ij} with constraints $q_{ij} \leq 0$; $q_{ij} \leq r_{ij}$; $q_{ij} \geq -1$; and $\sum_{(i,j)\in\mathcal{A}} q_{ij} \leq \xi$. Note that the extension to limit a planning *budget*, rather than a fixed number of link reversals, is an immediate result of this formulation. We can simply multiply by a constant b_{ij}, denoting the cost of investing in a reversal at link (i, j), before each q_{ij} in the previous summation, and restrict the right-hand side by a total budget rather than a maximum number of link reversals. A similar approach can be followed for the lane reversal extension in Section 4.1, if desired.

4.3. Multi-period optimization

Until now, we have considered the problem of finding the best lane configuration for a single OD demand matrix. However, in practice we would like to decide where to invest based on *multiple* traffic patterns. For example, by considering the morning and the afternoon peak traffic.

To achieve this, suppose that we have an OD demand matrix for every time interval $t \in \mathcal{T}$. For example, let $\mathcal{T} = \{\text{AM}, \text{MD}, \text{PM}, \text{NT}\}$ corresponding to morning, midday, afternoon and night traffic patterns, respectively. Then, by using the slack variables $\mathbf{r} = (r_{ij};\ (i,j) \in \mathcal{A})$ and $\mathbf{q} = (q_{ij};\ (i,j) \in \mathcal{A})$ defined in sections 4.1 and 4.2, respectively, we can formulate this problem as:

$$
\begin{aligned}
\min_{\{\mathcal{E}_t, \mathbf{c}_t, \mathbf{s}_t \ \mid\ t \in \mathcal{T}\}} \quad & \sum_{t \in \mathcal{T}} \mathbf{1}'\hat{\mathbf{T}}_t(\mathcal{E}_t) + \lambda \|\mathbf{s}_t\|_2, && \text{(14a)} \\
\text{s.t.} \quad & \sum_{i:(i,j)\in\mathcal{A}} x_{ijt} - \sum_{\ell:(j,\ell)\in\mathcal{A}} x_{j\ell t} = \phi_{jt},\ \forall j \in \mathcal{N},\ \forall t \in \mathcal{T}, && \text{(14b)} \\
& z_{ijt} + z_{jit} \leq n_{ij}, \quad \forall (i,j) \in \mathcal{A},\ \forall t \in \mathcal{T}, && \text{(14c)} \\
& \mathbf{s}_t \geq \mathbf{x}_t - \Theta(\mathbf{z}_t), \quad \forall t \in \mathcal{T}, && \text{(14d)} \\
& \mathbf{r}_t \geq \mathbf{z}_t^0 - \mathbf{z}_t, \quad \forall t \in \mathcal{T}, && \text{(14e)} \\
& \mathbf{r}_t \geq -(\mathbf{z}_t^0 - \mathbf{z}_t), \quad \forall t \in \mathcal{T}, && \text{(14f)} \\
& \mathbf{q} \geq -\mathbf{1}, && \text{(14g)} \\
& \mathbf{q} \leq -\sum_{t \in \mathcal{T}} \mathbf{r}_t, && \text{(14h)} \\
& \mathbf{1}'\mathbf{q} \leq \xi, && \text{(14i)} \\
& \boldsymbol{\varepsilon}_t, \mathbf{s}_t \geq 0,\ \mathbf{z}_t \in \mathbb{N}_+, \quad \forall t \in \mathcal{T}, && \text{(14j)}
\end{aligned}
$$

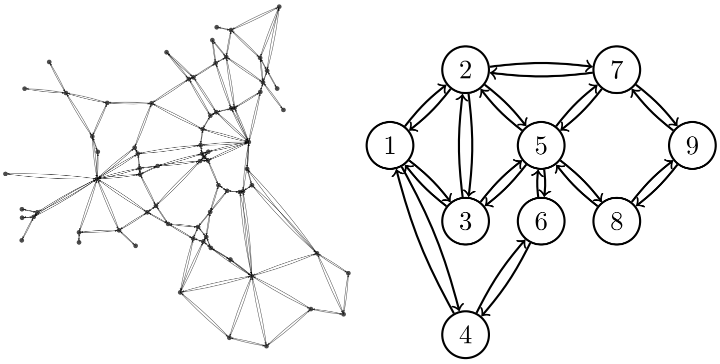
where we have written an augmented version of (12) with an additional coupling constraint (14h) which limits the number of links to invest over all the OD demand setting in $\mathcal{T}$. Notice that constraint (14h) is the only coupling constraint over $\mathcal{T}$. Therefore, this formulation is suitable for using decomposition techniques, such as Dantzig-Wolfe decomposition [42].

5. Numerical Results and Case Studies

To validate and compare the methods described above, we consider several numerical examples over two different transportation networks shown in Figure 3. The *test* network is a small example that is useful to test our methods. In addition, we perform a case study using the Eastern Massachusetts interstate highways (EMA) subnetwork (Figure 3a). The EMA road network is relevant in the context of lane reversals as it captures the dynamics of suburban/urban mobility where we expect lane reversal strategies to be beneficial. This is because arterial roads typically have a large number of lanes and, at the same time, they experience high traffic congestion. The values of the network topology (e.g., capacities, free flow speeds), as well as the code used to perform the experiments is publicly available in our online repository.[1]

We present our numerical results in the same order as our methods were presented in Section 3. First, we analyze the convergence of different alternating methods and the Frank-Wolfe algorithm. Then, we report numerical results for different selections of the λ parameter for the convex programming approach and we compare all approaches. Then, we provide an example of one of our extensions which limits the number of link reversals and we experiment with the symmetry of the OD demand. Finally, we show the computation effort of the methods as well as the differences between the user-centric and system-optimal solutions of the problem.

[1] https://github.com/salomonw/contraflow-lane-reversal



(a) (b)

Figure 3: (a) EMA transportation network composed of 74 nodes, 258 arcs, 581 lanes, and 1113 OD pairs; (b) Test network consists of 9 nodes, 26 arcs, 61 lanes, and 5 OD pairs.

5.1. Alternating and Greedy

Using the alternating method described in Section 3.1 we solve the LASO-TAP problem using three different methodologies. First, we introduce the *greedy* (or *One*) approach in which we first solve the TAP. Then, find the best possible reversal and switch it. Once we have changed the lane, we re-solve the TAP and carry out this process iteratively until we optimistically converge (since cannot be guaranteed) to a value. The second approach, which we called *Five*, implements the same idea, but instead of changing the best possible lane, it changes the top 5 lane reversals at every iteration. Similarly, our last alternating approach referred to as *Full* follows the same procedure but at each step reverses all the possible lanes that improve the travelling times. Figure 4b indicates the convergence of these three approaches where we observe two main patterns. First, the convergence of the algorithm is reached within a few (two to three) iterations which is consistent with many bi-level formulations involving the traffic assignment problem. Second, and more interestingly, the greedy approach performs poorly compared to the other methods. Our justification for this behavior is that when flows (or routes) are re-optimized, they are trying to use the links of the network infrastructure efficiently. Hence, we find a good allocation of flows to lanes such that the whole system is closer to a stationary point.

5.2. Convergence of Alg. 1

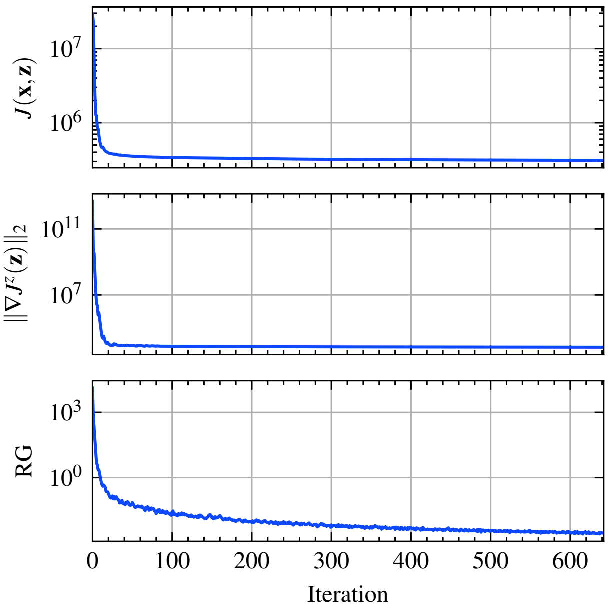
With the aim of validating our Frank-Wolfe methodology described in Section 3.2, we observe its trajectory over the iterates using the EMA transportation network. Similar to the other examples, we run the algorithm using the EMA transportation network with $t_{ij}(x_{ij}, z_{ij}) = t^0_{ij}(1+0.15(x_{ij}/c_{ij}z_{ij})^4)$, and $\alpha^0_1 = 30$. Hence, we can estimate Equation (9) with

$$\frac{J^{\mathbf{z}}(\mathbf{z})}{\partial z_{ij}} = -\frac{0.6t^0_{ij}x^5_{ij}}{c^4_{ij}z^5_{ij}} - \frac{0.6t^0_{ji}x^5_{ji}}{c^4_{ji}(n_{ij}-z_{ij})^5}.$$

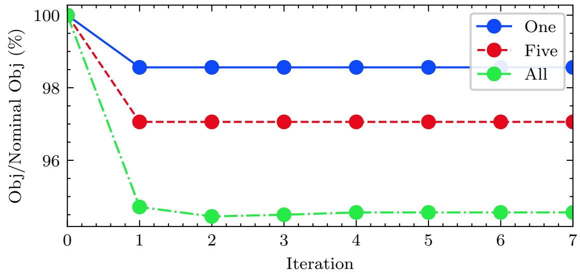
Figure 4a shows (i) the value of the objective function $J(\mathbf{x},\mathbf{z})$; (ii) the ℓ_2 norm of the gradient with respect to the lane reversals, i.e., $\|\nabla J^{\mathbf{z}}(\mathbf{z})\|_2$. When this value equals zero, we know that an infinitesimal change in reversing a lane will not be beneficial to the system; (iii) the relative gap (RG) pointing to the closeness to a solution that follows a Wardrop equilibrium. Hence, these results shows the effectiveness of Alg. 1 in the sense that it is minimizing the overall travel times by adjusting the routing and the links capacity at every iteration.

5.3. Dependence on λ

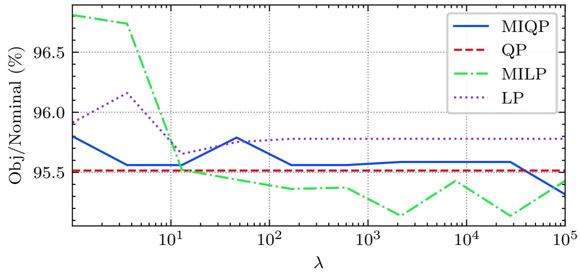
As we discussed in Subsection 3.4, one of the main drawbacks of our convex formulation is the inclusion of the parameter λ. This parameter serves as a trade-off between the routing and the lane assignment



(a) Convergence of the Frank-Wolfe algorithm proposed in Section 3.2



(b) Convergence of the alternating method for different numbers of reversals per iteration. *"One"* corresponds to reversing only one lane at every iteration while *"All"* reverse all the beneficial lanes at every step.



(c) Objective function for different values of λ and for different convex methods over the EMA network.

Figure 4

decisions. Hence, we would like to observe the performance of our solution for different values of λ. To perform the experiment, we use the EMA network and we solve the problem for $\lambda \in (0, 1 \times 10^5)$. Figure 4c reports the objective for different values of λ and for different algorithms. The numerical results show that the model is robust for different values of λ, i.e., the performance does not change substantially for different λ's. This is a positive result, since it suggest that calibrating λ does not play an important role in the performance of this method.

5.4. Comparison between methods

Now, we compare all the methods proposed in Section 3. We solve the routing problem, i.e., the TAP, with the current lane configuration and call it the *nominal* solution. Then, we solve the LASO-TAP problem for the test and the EMA networks using all the methodologies described earlier, i.e., Frank-Wolfe (FW), the three variations of the alternating method, and the different convex integer and continuous programs. In Table 1 we report the computational time of each method, as well as their performance compared to the nominal lane allocation. Specifically, we take $1 - (\text{OBJ}_{\text{method}}/\text{OBJ}_{\text{nominal}})$ to obtain a *relative improvement* (RI) metric with respect to the nominal allocation.

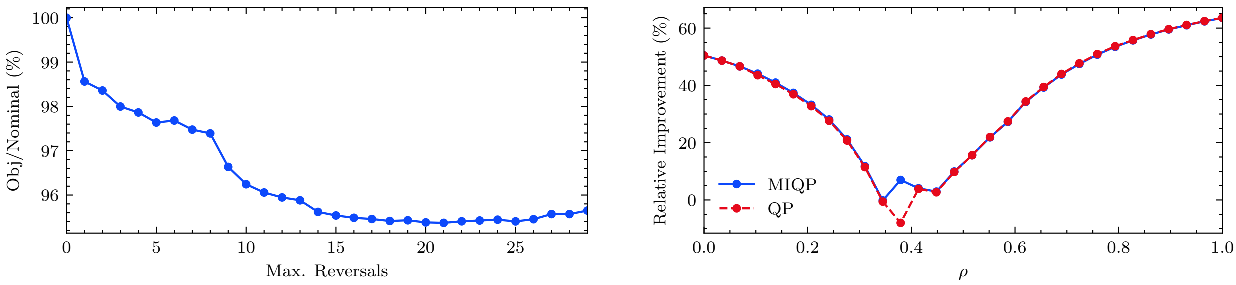
We observe that the convex approximation methods perform very well in terms of obtaining efficient solutions to the problem. In particular LP, QP and MILP compute the solution relatively fast compared with the FW or the alternating methods and achieve good solutions while providing the flexibility to add linear constraints.

5.5. Maximum number of reversed lanes

This experiment computes the Pareto optimal frontier using the MIQP and the extension of our convex approximation that limits the number of link reversals presented in Section 4.2. We compute the performance of different values of the maximum number of allowable link reversals in Figure 5a for the EMA network. We observe that the first lanes are the ones that contribute the most to the improvement of the overall travel times since we expect certain diminishing returns behavior.

Table 1: Results indicating the relative improvement in the overall travel time between a method and the nominal capacity.

Method	Test		EMA	
	RI (%)	Time (s)	RI (%)	Time (s)
Nominal	0.0	0.01	0.0	0.42
FW	12.7	11.5	0.7	956.2
Alt. (1)	10.3	0.62	1.4	12.2
Alt. (5)	14.6	0.60	2.9	12.4
Alt. (full)	14.7	0.59	5.4	12.4
MIQP	17.5	0.02	4.7	101.37
QP	17.4	0.02	4.5	0.71
MILP	17.5	0.02	4.6	5.67
LP	17.3	0.01	4.2	0.36



(a) Performance on the overall travel times as the number of allowed link reversals increases. We employed the MIQP method in the EMA network

(b) The relative improvement for different OD demand distributions in the Test network. The x-axis ρ indicate the fraction of demand travelling from west to east. Similarly, $(1-\rho)$ captures the fraction of vehicles travelling from east to west. The extremes ($\rho = 0$ and $\rho = 1$) are the most assymetric travelling patterns

Figure 5

This extension to the model, and corresponding results, are of interest to urban planners that might need to prioritize the most critical roads and may have a budget that depends on the overall benefit of reducing travel times for the system. In addition, our numerical results imply that it is not necessary to invest in too many lane reversals to achieve a large fraction of achievable improvement. For example, by investing between 10 and 15 links in the network (out of 129 possible links) the solution has already reached most of the benefit.

5.6. *Effect of OD demand symmetry*

A characteristic of lane reversal strategies is that larger benefits occur when the demand is *not symmetric.* Consider the demand patterns of a large metropolitan area on a weekday morning. In most cases, we expect a large fraction of the demand to be travelling towards the city center. Hence, we expect high traffic heading towards the city center and low traffic traveling to suburban areas. We refer to this as an *asymmetric* demand. In contrast, the traffic flows between two major cities could serve as an example of a *symmetric* OD demand. We generated an experiment using the Test network (Figure 3b) to exemplify the effect of symmetry in the performance of lane reversals. To do so, let $\rho \in (0, 1)$ be a parameter describing the fraction of the OD demand travelling from west to east and $(1 - \rho)$ the fraction of vehicles going from east to west. More explicitly, we defined the OD matrix to be $d_{(1,9)} = 15000\rho$, $d_{(1,9)} = 15000(1 - \rho)$ and $d_{(s,t)} = 0$ for all $(s, t) \in \mathcal{W} \setminus \{(1, 9), (9, 1)\}$.

Figure 5b shows the *relative improvement* of the lane reversal solution, using MIQP and QP, with respect to the nominal capacity solution. We observe that for the cases in which demand is not symmetric, the improvements in travel time can be as high as to 65%. In contrast, when the demand is symmetric, there are fewer benefits (considering uniform capacities across the network). Notice that negative values can be

Table 2: Computational times for convex approximation methods

Network	Num. of lanes	Num. of constraints	L	Num. of vars.	Num. of integer vars. in IPs	Computational Times (ms) LP	QP	MILP	MIQP
Test	61	236	3	286	26	2.1	3.2	5.6	4.9
		288	5	338	26	2.1	4.3	5.8	7.7
		340	7	390	26	2.6	5.3	4.4	7.7
EMA Small	105	257	3	384	24	1.7	3.5	5.2	122.4
		305	5	432	24	2.2	5.0	4.0	48.1
		353	7	480	24	2.3	5.5	12.2	22.9
Sioux Falls	506	1,185	3	2,432	76	36.3	91.3	40.2	481.7
		1,337	5	2,584	76	46.9	123.1	47.4	511.8
		1,489	7	2,736	76	53.6	120.6	65.5	550.7
EMA	581	6,209	3	16,512	258	154.0	402.9	315.7	2,599.8
		6,725	5	17,028	258	276.8	478.0	490.4	1,413.1
		7,241	7	17,544	258	290.9	588.1	492.2	1,445.7
Anaheim	4,214	24,891	3	56,912	1,268	1,073.9	1,421.0	1,104.6	25,964.0
		26,719	5	58,740	1,268	1,882.7	1,523.0	2,020.4	11,204.2
		28,547	7	60,568	1,268	2,021.6	1,559.7	2,685.1	10,983.4
NYC	17,870	164,811	3	458,538	4,330	30,179.0	67,486.6	69,788.6	NA
		171,085	5	464,812	4,330	56,658.5	85,796.0	250,996.9	NA
		177,359	7	471,086	4,330	68,534.7	74,356.9	143,835.1	NA

observed (as in the QP case) since we are solving an approximation method (both by the convex approach and by projecting the continuous QP solution) that could deviate from the optimal value in certain cases. These results provide a tractable example to understand the potential of lane reversals depending on the symmetry of the demand and suggest the instances for which these interventions provide large benefits. Some real-life examples in which asymmetric demands can be found include: morning and afternoon peaks, massive events, and holiday travel.

5.7. Computational effort

This experiment aims to compare the computational times of the convex approximation methods described in Section 3.3.1 for different network sizes and levels of L (number of piecewise linear segments). The motivation is to provide an empirical sensitivity analysis of the main methods and parameters of the algorithms. To do so, we apply the methods to benchmark network topologies available in Stabler et al. [43] and in Wollenstein-Betech et al. [36] ranging the number of lanes between 61 and 17870. Table 2 reports the number of lanes, constraints, variables and integer variables needed for each of the problems, as well as the computational times for the LP, QP, MILP and MIQP methods. We highlight the fact that increasing L does not have a large impact in the computational times but we can expect to affect the solutions' quality. Moreover, we note that for the NYC case, the largest network, the MIQP was unable to converge while the MILP was able to obtain an exact optimal solution almost in comparable times to those of the QP method. This provides empirical evidence that it is attainable to find an exact global integer optimal solution for considerably large networks.

5.8. System-Optimal and User-Centric comparison

We show that our method is capable of solving the Lane Assignment User Centric TAP (LAUC-TAP) and compare the quality of its solution with the LASO-TAP; observing small differences between their solutions. To do so, let us first provide intuition on how our convex approximation method could fail to improve the total system travel time. Let us consider Braess' paradox whose main insight is that the addition of new capacity could worsen the objective due to the routing decisions of UC users. In the most classical example of Braess' paradox, the capacity of the link connecting nodes 2 to 3 in Figure 1b is assumed to be infinite.

Table 3: UC vs SO solutions

Max. Lane Reversals	$\text{TAP}^{UC}(\mathbf{z}^*_{UC})$	$\text{TAP}^{UC}(\mathbf{z}^*_{SO})$	$\text{TAP}^{SO}(\mathbf{z}^*_{SO})$
0	**1.038**	**1.038**	1.000
1	**1.020**	**1.031**	0.986
5	1.002	1.002	0.959
10	0.992	0.992	0.945
15	**0.979**	**0.982**	0.934
18	**0.976**	**0.988**	0.935

Hence, the second element of the objective of our method, i.e., $\lambda\|\max\{0, x_{23}\} - c_{23}z_{23}\|$, will be equal to zero since c_{23} is very large. Therefore, there is no *signaling* to the optimization program indicating that link $(2, 3)$ is congested. Hence, the optimization will decide to keep the lane open and will not be able to improve the solution.

However, in more realistic scenarios (e.g., using a larger network and the BPR function), we observe that our method finds a better solution when assuming UC behavior and solving the LAUC-TAP rather than the LASO-TAP. To show this, we report the normalized total system travel times of the UC- and SO-TAP with lane configurations coming from the solution of the LASO-TAP and LAUC-TAP. Specifically, in Table 3, $\text{TAP}^{UC}(\mathbf{z}^*_{UC})$ reports the normalized total system travel time (dividing by $TAP^{SO}(\mathbf{z}^*_{SO})$ with zero allowable reversals) of a network of user-centric agents and by assigning the lane configuration according to the LAUC-TAP. In contrast, $\text{TAP}^{UC}(\mathbf{z}^*_{SO})$ assumes a user-centric behavior but implements the solution of the LASO-TAP. Finally, $\text{TAP}^{SO}(\mathbf{z}^*_{SO})$ reports results for a network with socially-optimal agents evaluated at the solution of $\mathbf{z}^*_{SO}$.

When the maximum number of reversed lanes is equal to zero, we have under the $\text{TAP}^{UC}(\mathbf{z}^*_{UC})$ the value of the Price of Anarchy. Moreover, for 1 allowable reversal, we see that $\text{TAP}^{UC}(\mathbf{z}^*_{UC}) \leq \text{TAP}^{UC}(\mathbf{z}^*_{SO})$ showing how our UC approach finds a better lane allocation by anticipating the agents' selfish behavior. However, as the maximum number of lane reversals increases, we observe that the difference between $\text{TAP}^{UC}(\mathbf{z}^*_{UC})$ and $\text{TAP}^{UC}(\mathbf{z}^*_{SO})$ becomes most of the times negligible. We believe this happens because in practice the SO flows are a good predictor of the UC flows [44].

6. Conclusion

The problem of identifying the best lanes to reverse in a congested network is challenging because it requires to solve a mixed integer non-convex programming problem. The literature dealing with this problem has been focused on using heuristic algorithms for solving it. In this work we propose three strategies to reduce the complexity of the problem. Our first method uses the principle of *decomposition* or *divide and conquer* by separating the joint routing and assignment problem into separate ones. Our second method uses the idea of *relaxation* by converting the integer variables to continuous ones. Lastly, our third method *convexifies* the objective by modifying the objective function. This last method is interesting since it allows including additional constraints to the problem, e.g., a maximum number of reversals.

We provide numerical results for all our methods showing their performance over a test network and a case study using the transportation network of Eastern Massachusetts. Interestingly our results show that the greedy approach, the one that reverses the most relevant lane at every iteration, can result in near-optimal solutions. Moreover, our results suggest that the convexification approach is efficient in both the quality of the solution and the computational burden.

We identify the following future research directions. First, herein we consider the *static* TAP which is a good model for transportation planning purposes. However, extending this work to a *dynamic* traffic setup would be beneficial to design real-time network-wide lane reversal controllers which could consider the dynamic intersection management similar to Xie et al. [45]. Second, as explained in Section 3.3, our convex approximation method finds an assignment that avoids generating congested links by setting up a threshold $\Theta_{ij}(z_{ij})$. Although this approach follows our intuition on what lanes to reverse, it may fail to provide good solutions for certain examples of the user-centric behavior; for example, the Braess' paradox. Therefore, extending this method to identify these cases would be advantageous to improve the allocation for user-centric behavior. Third, the LP and QP formulations that we have described require many variables (and memory) for large networks. Hence, designing decomposition techniques can reduce the memory requirements and speed up the computational times for large networks with many OD pairs. Finally, our convex approximation approach can be extended to include robustness with the aim to provide the best allocation for an allowable set of OD demands rather than a single vector.

References

[1] M. Jha, K. Moore, B. Pashaie, Emergency evacuation planning with microscopic traffic simulation, Transportation Research Record: Journal of the Transportation Research Board 1886 (2004) 40–48.

[2] G. Theodoulou, B. Wolshon, Alternative methods to increase the effectiveness of freeway contraflow evacuation, Transportation Research Record: Journal of the Transportation Research Board 1865 (2004) 48–56.

[3] T. J. Cova, J. P. Johnson, A network flow model for lane-based evacuation routing, Transportation Research Part A: Policy and Practice 37 (2003) 579–604.

[4] X. Zhao, Z.-y. Feng, Y. Li, A. Bernard, Evacuation network optimization model with lane-based reversal and routing, Mathematical Problems in Engineering (2016). doi:10.1155/2016/1273508.

[5] C. Xie, M. A. Turnquist, Lane-based evacuation network optimization: An integrated lagrangian relaxation and tabu search approach, Transportation Research Part C: Emerging Technologies 19 (2011) 40–63.

[6] S. Kim, S. Shekhar, M. Min, Contraflow transportation network reconfiguration for evacuation route planning, IEEE Transactions on Knowledge & Data Engineering 20 (2008) 1115–1129.

[7] K. Ampountolas, J. A. dos Santos, R. C. Carlson, Motorway tidal flow lane control, IEEE Transactions on Intelligent Transportation Systems 21 (2019) 1687–1696.

[8] K. Ampountolas, R. C. Carlson, Optimal control of motorway tidal flow, in: 2019 18th European Control Conference (ECC), IEEE Press, 2019, pp. 3680–3685.

[9] J. R. D. Frejo, I. Papamichail, M. Papageorgiou, E. F. Camacho, Macroscopic modeling and control of reversible lanes on freeways, IEEE Transactions on Intelligent Transportation Systems 17 (2015) 948–959.

[10] W. Zhou, P. Livolsi, E. Miska, H. Zhang, J. Wu, D. Yang, An intelligent traffic responsive contraflow lane control system, in: Proceedings of VNIS'93-Vehicle Navigation and Information Systems Conference, IEEE, 1993, pp. 174–181.

[11] D. Xue, Z. Dong, An intelligent contraflow control method for real-time optimal traffic scheduling using artificial neural network, fuzzy pattern recognition, and optimization, IEEE Transactions on Control Systems Technology 8 (2000) 183–191.

[12] M. J. Lighthill, G. B. Whitham, On kinematic waves ii. a theory of traffic flow on long crowded roads, Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 229 (1955) 317–345.

[13] M. Malekzadeh, I. Papamichail, M. Papageorgiou, K. Bogenberger, Optimal internal boundary control of lane-free automated vehicle traffic, Transportation Research Part C: Emerging Technologies 126 (2021) 103060.

[14] M. W. Levin, S. D. Boyles, A cell transmission model for dynamic lane reversal with autonomous vehicles, Transportation Research Part C: Emerging Technologies 68 (2016) 126–143.

[15] K.-F. Chu, A. Y. Lam, V. O. Li, Dynamic lane reversal routing and scheduling for connected and autonomous vehicles: Formulation and distributed algorithm, IEEE Transactions on Intelligent Transportation Systems 21 (2019) 2557–2570.

[16] M. Hausknecht, T.-C. Au, P. Stone, D. Fajardo, T. Waller, Dynamic lane reversal in traffic management, in: 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), IEEE Press, 2011, pp. 1929–1934. doi:10.1109/ITSC.2011.6082932.

[17] Q. Meng, H. L. Khoo, Optimizing contraflow scheduling problem: model and algorithm, Journal of Intelligent Transportation Systems 12 (2008) 126–138.

[18] M. Abdulaal, L. J. LeBlanc, Continuous equilibrium network design models, Transportation Research Part B: Methodological 13 (1979) 19–32.

[19] S.-W. Chiou, Bilevel programming for the continuous transport network design problem, Transportation Research Part B: Methodological 39 (2005) 361–383.

[20] T. L. Magnanti, R. T. Wong, Network design and transportation planning: Models and algorithms, Transportation Science 18 (1984) 1–55.

[21] H. Yang, M. G. H. Bell, Models and algorithms for road network design: a review and some new developments, Transport Reviews 18 (1998) 257–278.

[22] H. Liu, D. Z. Wang, Global optimization method for network design problem with stochastic user equilibrium, Transportation Research Part B: Methodological 72 (2015) 20–39.

[23] D. Z. Wang, H. Liu, W. Szeto, A novel discrete network design problem formulation and its global optimization solution algorithm, Transportation Research Part E: Logistics and Transportation Review 79 (2015) 213–230.

[24] D. Z. Wang, H. K. Lo, Global optimum of the linearized network design problem with equilibrium flows, Transportation Research Part B: Methodological 44 (2010) 482–492.

[25] P. Luathep, A. Sumalee, W. H. Lam, Z.-C. Li, H. K. Lo, Global optimization method for mixed transportation network design problem: a mixed-integer linear programming approach, Transportation Research Part B: Methodological 45 (2011) 808–827.

[26] S. V. Ukkusuri, S. T. Waller, Linear programming models for the user and system optimal dynamic network design problem: formulations, comparisons and extensions, Networks and Spatial Economics 8 (2008) 383–406.
[27] S. T. Waller, A. K. Ziliaskopoulos, Stochastic dynamic network design problem, Transportation Research Record: Journal of the Transportation Research Board 1771 (2001) 106–113.
[28] R. Z. Farahani, E. Miandoabchi, W. Y. Szeto, H. Rashidi, A review of urban transportation network design problems, European Journal of Operational Research 229 (2013) 281–302.
[29] S. Wollenstein-Betech, I. C. Paschalidis, C. G. Cassandras, Planning Strategies for Lane Reversals in Transportation Networks, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 2021, pp. 2131–2136. doi:10.1109/ITSC48978.2021.9564808.
[30] M. J. Beckmann, C. B. McGuire, C. B. Winsten, Studies in the Economics of Transportation, Yale University Press, 1955. doi:10.1057/jors.1980.83.
[31] Traffic Assignment Manual, Bureau of public roads, US Department of Commerce (1964).
[32] S. Wollenstein-Betech, C. Sun, J. Zhang, C. G. Cassandras, I. C. Paschalidis, Joint Data-Driven Estimation of Origin-Destination Demands and Travel Latency Functions in Multi-Class Transportation Networks, IEEE Transactions on Control of Network Systems (2022). doi:10.1109/TCNS.2022.3161200.
[33] H. Bar-Gera, Origin-based algorithm for the traffic assignment problem, Transportation Science 36 (2002) 398–417.
[34] F. Rossi, R. Iglesias, M. Alizadeh, M. Pavone, On the Interaction Between Autonomous Mobility-on-Demand Systems and the Power Network: Models and Coordination Algorithms, IEEE Transactions on Control of Network Systems 7 (2020) 384–397. doi:10.1109/TCNS.2019.2923384.
[35] A. Houshmand, S. Wollenstein-Betech, C. G. Cassandras, The Penetration Rate Effect of Connected and Automated Vehicles in Mixed Traffic Routing, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 1755–1760. doi:10.1109/ITSC.2019.8916938.
[36] S. Wollenstein-Betech, M. Salazar, A. Houshmand, M. Pavone, I. C. Paschalidis, C. G. Cassandras, Routing and Rebalancing Intermodal Autonomous Mobility-on-Demand Systems in Mixed Traffic, IEEE Transactions on Intelligent Transportation Systems (2021). doi:10.1109/TITS.2021.3112106.
[37] S. C. Dafermos, F. T. Sparrow, The traffic assignment problem for a general network, Journal of Research of the National Bureau of Standards B 73 (1969) 91–118.
[38] M. Patriksson, The Traffic Assignment Problem: Models and Methods, volume 54, Dover Publications, 1994.
[39] L. A. Wolsey, Integer programming, John Wiley & Sons, 2020.
[40] A. Ben-Tal, L. El Ghaoui, A. Nemirovski, Robust optimization, Princeton university press, 2009.
[41] D. Bertsimas, M. Sim, Robust discrete optimization and network flows, Mathematical Programming 98 (2003) 49–71.
[42] G. B. Dantzig, P. Wolfe, Decomposition principle for linear programs, Operations Research 8 (1960) 101–111.
[43] B. Stabler, H. Bar-Gera, E. Sall, Transportation Networks for Research, 2019. URL: `https://github.com/bstabler/TransportationNetworks`.
[44] L. J. LeBlanc, M. Abdulaal, A comparison of user-optimum versus system-optimum traffic assignment in transportation network design, Transportation Research Part B: Methodological 18 (1984) 115–121.
[45] C. Xie, D.-Y. Lin, S. T. Waller, A dynamic evacuation network optimization problem with lane reversal and crossing elimination strategies, Transportation Research Part E: Logistics and Transportation Review 46 (2010) 295–316.

Appendix A: Notation table

Table 4: Notation Table of main variables and parameters

Sets		
$\mathcal{A}$	:=	Links of the transportation network.
$\mathcal{V}$	:=	Nodes of the transportation network.
$\mathcal{W}$	:=	Origin-Destination pairs.
$\mathcal{O}$	:=	Set of origins with positive demand.
Parameters		
$\mathcal{G}$	:=	Graph representing a transportation network.
$\mathbf{N}$	:=	Node-arc incidence matrix of $\mathcal{G}$.
K	:=	Number of OD pairs.
$\mathbf{w}_k$	:=	OD pair k with source $s_k \in \mathcal{V}$ and target $t_k \in \mathcal{V}$.
$d_{\mathbf{w}_k}$	:=	Demand rate of OD pair $\mathbf{w}_k \in \mathcal{W}$.
n_{ij}	:=	Maximum number of lanes that could be assigned to (i, j).
n_{ij}^0	:=	Nominal (current) lane configuration.
c_{ij}	:=	Capacity per lane in link $(i, j) \in \mathcal{A}$.
m_{ij}	:=	Total capacity in link $(i, j) \in \mathcal{A}$.
t_{ij}^0	:=	Free-flow travel time for link $(i, j) \in \mathcal{A}$.
ϕ	:=	Node imbalance describing excess demand or supply.
ψ_{ij}	:=	Difference in the overall travel time between current network lane assignment and when we reverse a link in (i, j).
λ	:=	Regularizer term trading off routing efficiency and not exceeding a links capacity.
$\Theta_{ij}(z_{ij})$	:=	Threshold in capacity of (i, j) for the convex approach.
L	:=	Number of piecewise affine segments.
a_l	:=	Slope of the piecewise affine segment $l = 1, \ldots, L$.
$\theta_{ij}^{(l)}$	:=	Piecewise affine function breakpoints.
Decision Variables		
$x_{ij}^{\mathbf{w}_k}$	:=	Flow on link (i, j) associated with OD pair $\mathbf{w}_k$.
x_{ij}^o	:=	Flow on link (i, j) whose origin is node o.
x_{ij}	:=	Total flow on link (i, j).
z_{ij}	:=	Number of lanes assigned to link (i, j).
$\varepsilon_{ij}^{(l)}$	:=	Flow exceeding threshold $\theta_{ij}^{(l)}$ and up to $\theta_{ij}^{(l+1)}\theta_{ij}^{(l)}$.
s_{ij}	:=	Slack variable pointing to the flow exceeding threshold $\Theta_{ij}(z_{ij})$.