Transportation Research Part C 130 (2021) 103278

Contents lists available at ScienceDirect

Transportation Research Part C

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

A multi-hop control scheme for traffic management

Hossein Rahimi Farahani [a], Amir Abbas Rassafi [a], Kenan Zhang [b], Yu (Marco) Nie [b,*]

[a] Transportation Planning, Imam Khomeini International University, Qazvin, Iran
[b] Civil and Environmental Engineering, Northwestern University, IL 60208, USA

ARTICLE INFO

Keywords:
Multi-hop control
Intermediate checkpoint
Mathematical program with equilibrium constraints
User equilibrium
System optimal

ABSTRACT

We propose a multi-hop control scheme (MHCS) that aims to route traffic through a set of designated intermediate checkpoints (ICs). Because travelers are allowed to freely choose routes for each "hop" that connects real (origin and destination) and ICs, MHCS promises to keep intervention at a more tolerable level, compared to conventional route-based control schemes. The MHCS problem has a natural bi-level structure: the upper level attempts to minimize congestion by adjusting the hopping ratios, which are then used in the lower level problem to route travelers according to user equilibrium conditions. Accordingly, we formulate the problem as a mathematical program with equilibrium constraints (MPEC), establish its solution existence, and propose to solve it using a sensitivity analysis based algorithm. We examine sixteen heuristic rules for choosing ICs. Results based on five hundred experiments suggest that selecting the most used and most congested nodes at system optimum as the ICs delivered the largest travel time savings. Based on this finding, a set of efficient ICs are identified and adopted to test the potential of a full-scale scheme. The results from numerical experiments indicate that these checkpoints are highly effective in reducing traffic congestion at a reasonable cost of control and unfairness. In particular, they outperform, by a large margin, other choices such as most congested nodes at user equilibrium.

1. Introduction

To modern cities, traffic congestion is a daunting challenge that has major economic, environmental, and societal ramifications. In the United States alone, the price tag for traffic congestion has reached $160 billion in 2014, and that number only accounts for wasted time and fuel (Schrank et al., 2015). It has long been recognized that the selfish route choice of travelers contributes to excessive traffic congestion (Beckmann et al., 1956; Sheffi, 1985). Selfish travelers, seeking to minimize their own travel cost, would settle at a user equilibrium (UE) (Wardrop, 1952), equivalent to the Nash equilibrium of a non-cooperative game. The UE state, however, is inefficient in the sense it is unable to minimize the *total* travel cost. The gap between the total cost at UE and that at the most efficient state, or a system optimum (SO), is known as the *price of anarchy* (Roughgarden, 2002). Achieving SO typically requires *cooperation* orchestrated by a central agent. Yet, implementing such a solution is against human nature because travelers may be asked to sacrifice their own interest, by making a suboptimal travel choice, to benefit others. In the literature, this shortcoming is often referred to as the *unfairness* issue (Jahn et al., 2005).

The rapid development of autonomous vehicle (AV) technology has promised a new tool to influence and even directly control

* Corresponding author.
E-mail addresses: h.r.farahani@edu.ikiu.ac.ir (H.R. Farahani), rasafi@eng.ikiu.ac.ir (A.A. Rassafi), kenanzhang2020@u.northwestern.edu (K. Zhang), y-nie@northwestern.edu (Y.(M. Nie).

https://doi.org/10.1016/j.trc.2021.103278
Received 23 December 2020; Received in revised form 19 May 2021; Accepted 22 June 2021
Available online 10 July 2021

travelers' route choice. For one thing, autonomous vehicles are much more controllable since a robot, not a human, is behind the wheel (Bagloee et al., 2016). Perhaps more importantly, the in-vehicle travel time is much less a cost to travelers because they are able to undertake other activities while sitting in an AV (Steck et al., 2018). The underlying idea is rather simple: if a central agent could control the routes of an AV fleet in an "optimal" manner, a sizeable congestion relief may be achieved at the system level. In fact, not only does such a control scheme work, it often only requires controlling a fraction of all the vehicles in the system (see e.g. Zhang and Nie, 2018; Chen et al., 2020). While controlling the entire route of an AV is feasible, it is still an intrusive measure that may not be well received by its users. For human-driven vehicles that will inevitably still be on the road in the foreseeable future, controlling their routes remains an evasive target, and can only be attempted through nudge and persuasion that have seen rather limited success. Motivated by this limitation, we propose to designate one or more intermediate checkpoints (ICs) that a controlled vehicle must visit in sequence. Thus, instead of delineating the entire route, this multi-hop control scheme (MHCS) provides a rough "sketch" of the route and leaves travelers with the freedom to choose how to route between the ICs.

The above idea was initially explored by Farahani et al. (2019), who formulated and solved the assignment problem resulted from applying the multi-hop control strategy. Given a set of ICs and a vector of hopping ratio (i.e., the proportion of the demand from each origin–destination (OD) pair assigned to each IC), Farahani et al. (2019) show the multi-hop assignment problem can be converted into a standard traffic assignment problem. They also discuss how such a routing scheme may be implemented by utilizing V2X technology that is rapidly maturing in the marketplace (Faez and Khanjary, 2008). However, they leave out the design questions completely. That is, how should the central agent optimally choose the ICs and the hopping ratios? This is the question that we set out to address here.

While jointly optimizing ICs and hopping ratios is a tempting exercise, we note it can result in a highly complex design problem that would pose formidable computational challenges. Instead, this study focuses on the problem of optimizing hopping ratios for a given set of ICs. We formulate the design problem as a bi-level program. The upper level problem, or the leader's problem, aims to minimize the total system cost by setting hopping ratios while anticipating selfish route choices between ICs. The lower level problem, or the follower's problem, is a UE traffic assignment problem created based on the hopping ratios determined by the leader. Since the optimal conditions of the lower level problem are equivalent to UE conditions, our design problem belongs to a broad class of optimization problems known as Mathematical Programs with Equilibrium Constraints (MPEC).

MPECs are notorious for their resistance to efficient solution techniques. The problem, by and large, is attributed to the nature of equilibrium constraints. Not only are these constraints non-convex, but they are also highly nonlinear in general, and especially so in traffic assignment problems. Because of this complexity, no close-form mapping exists between the leader's decision variables and the solution of the follower problem. Another challenge to solving the proposed design problem has to do with the potentially large scale of the upper level problem. The number of hopping ratios grows nonlinearly with the size of the O-D matrix and the number of ICs. In a typical planning network, the number of decision variables in the leader's problem can easily reach hundreds of millions.

In this study, we design a heuristic algorithm based on sensitivity analysis. In each iteration, we first numerically evaluate the derivative of the upper level objective function with respect to (w.r.t) the decision variables (hopping ratios) while considering the interaction between the leader and the follower dictated by the equilibrium conditions. The derivative information obtained from the sensitivity analysis is then plugged in a gradient-based algorithm that iteratively descends to a local minimum.

While the choice of ICs is not formally optimized—which would inevitably lead to a completely intractable combinatorial problem—we investigate the impact of this choice on the effect of the multi-hop strategy, in terms of both efficiency gains and fairness loss, through carefully designed numerical experiments. Specifically, we propose numerous heuristic rules for picking the ICs and demonstrate that some lead to much better results than others.

The remainder of this paper is organized as follows. In Section 2, we briefly review the related studies. The multi-hop problem is presented in Section 3, along with the bi-level formulation. Section 4 describes the sensitivity analysis based solution method. Results of numerical experiments are reported in Section 5, along with heuristic rules for choosing ICs. Section 6 concludes the paper with a summary of findings and directions for future explorations.

2. Related studies

Many have considered the co-existence of travelers following different route choice principles, such as UE and SO, in the context of traffic management. Early studies often cast it as a mixed traffic equilibrium problem (Van Vuren et al., 1990; Haurie and Marcotte, 1985). Yang et al. (2007) propose to formulate such a problem as a Stackelberg game, in which the SO users are considered as being controlled by a central agent for the benefit of the entire system. Because the agent can anticipate the reaction of the UE users, the control strategy based on the Stackelberg game promises greater efficiency. Wang et al. (2016) investigate the conditions under which the Braess paradox (Braess, 1968) arises in mixed traffic equilibrium problem. Yang et al. (2017) consider the impact of some side constraints (such as link capacity constraints) in mixed equilibrium models. Bagloee et al. (2017) propose a general formulation of the mixed traffic equilibrium problem in which the portion of the SO and UE users are determined endogenously by a logit demand function. Li et al. (2018) study the tradeoff between efficiency and stability when controlling a mixed traffic system that consists of both UE and SO users. Zhang and Nie (2018) explore the relationship between the number of SO users in the system and the potential of efficiency gain. The problem is formulated as an MPEC, in which the upper level problem chooses the ratio of SO users to minimize a weighted sum of travel cost and control intensity, and the lower level problem solves a mixed traffic equilibrium. They find a near-optimal solution can often be obtained by designating but a small portion of the travelers as SO users. Built on Zhang and Nie (2018), Chen et al. (2020) study the problem of determining the minimum ratio of SO users required to achieve the overall SO state, which is formulated as a linear program. To reduce control intensity, they further consider the possibility of levying tolls on the UE drivers. Concerning the transition from human-driven vehicles to autonomous driving, several recent studies have considered the pros and cons of forbidding AVs and human-driven vehicles from using the same facility at the same time (Chen et al., 2016; Chen et al., 2017; Liu and Song, 2019; Bahrami and Roorda, 2020).

Another line of inquiry focuses on the tradeoff between the overall efficiency gain and the extra cost imposed on the travelers by the control strategy. Jahn et al. (2005) consider a constrained SO traffic assignment problem, in which the assignment is limited to SO paths no longer than a threshold (i.e., the cost of the UE path plus a tolerable detour cost). Du et al. (2015) propose to influence travelers' decisions by providing "perturbed" information. The idea of perturbation is to exploit the fact that travelers only have bounded rationality in route choice (Mahmassani and Chang, 1987; Lou et al., 2010). Angelelli et al. (2016) propose a hierarchical framework that consists of two stages. The first stage solves a constrained SO routing problem that is similar to that of Jahn et al. (2005) but has linear cost functions, and the second stage attempts to further reduce the level of unfairness.

In all the above studies, the central agent is assumed to have the ability to either dictate the route choice of SO users (as in the Stackelberg game) or to influence it through incentives (as in the mixed equilibrium model). The proposed multi-hop scheme differs from them in that it only controls a few ICs.

The leader–follower structure of our model bears similarities with that of network design problems in transportation (Yang and Bell, 1998; Yang and Yagar, 1995; Yang and Bell, 1997; Dempe and Zemkoho, 2012; Gu et al., 2019). Most network design problems can be cast as an MPEC. Here, the interests of the network designer (the leader) and the network users (the follower) may be at odds with each other. In pursuing its own interest, the network designer must anticipate how the users would react to its decision, which often imposes an equilibrium constraint on the design problem. Being nonlinear and nonconvex optimization problems, MPECs are not amenable to efficient solution methods. One often must rely on heuristics and settle for local optimal solutions. A commonly used approach utilizes the results of sensitivity analysis on the equilibrium condition (Friesz et al., 1990; Yang et al., 1994; Patriksson and Rockafellar, 2002). Specifically, the derivatives of the equilibrium condition are used to approximate the variations in route choice with respect to the changes in decision variables in the leader problem. More recently, Lawphongpanich and Yin (2010) propose to decompose MPEC into a set of nonlinear programs that could then be solved more efficiently. The solution method adopted in this paper is a variant of sensitivity analysis based heuristics.

Sensitivity analysis of traffic equilibria has been investigated extensively in the literature, starting with the seminal works of Tobin and Friesz (1988) and Qiu and Magnanti (1989). Although the first paper above has received broad attention in the transportation community, its limitations are also well known, as noted by Bell and Iida (1997) and Josefsson and Patriksson (2007). The main issue is the special matrix invertibility condition that implies a strong requirement on the topology of the network. Yang and Bell (2007) present a revision of traffic equilibrium sensitivity analysis that resolves some of the limitations. For the fixed-demand traffic assignment problem, Lu (2008) defines derivatives and semi-derivatives under weaker assumptions. The method of Lu (2008) is adopted in this paper as the foundation to develop a sensitivity analysis based solution algorithm for the proposed multi-hop traffic control model.

3. Model

3.1. Preliminary settings

Consider a transportation network $G(N,A)$, where N is the set of nodes, and A the set of links. We denote the set of origins and destinations, respectively by $P\subseteq N$ and $Q\subseteq N$. Let $W\subseteq P\times Q$ be the set of all O-D pairs. The demand between O-D pair (p,q) is denoted by d_{pq} or $d_w>0$, and $\mathbf{d}=(d_w,w\in W)^T$ denotes the vector of all O-D demands. Note that both pq and w are needed: the latter refers to the index of an O-D pair in the demand vector whereas the former emphasizes the origin and the destination (which is especially useful when intermediate checkpoints are involved). We represent a route between (p,q) using r_{pq} or $r_w\in R_{pq}$ or R_w, where $R_w$is the set of routes between the corresponding O-D pair w. The set of all routes in the network would then be defined as $R = (R_w, w\in W)$. The flow on route r_{pq} is denoted by f_{rpq} or f_{rw}, and $\mathbf{f}=(f_{rw},r\in R_w,w\in W)^T$ is the vector of all route flows. Similarly, v_a is the flow on link $a\in A$, and $\mathbf{v}=(v_a,a\in A)^T$ is the vector of all link flows. A list of all notations can be found in Appendix (Table A1).

We introduce the matrix $\Gamma\in\{0,1\}^{|W|\times|R|}$ as the O-D pair-route incidence matrix (i.e., the element γ_{wr} is 1 if route r connects O-D pair w and zero otherwise). The link-route incidence matrix is denoted by $\Delta\in\{0,1\}^{|A|\times|R|}$, where an element δ_{ar} equals 1 if the path r uses link a and zero otherwise. Accordingly, the relationship between route flows and O-D demands is simply stated as $\mathbf{d} = \Gamma\mathbf{f}$, and the relationship between link flows and path flows as $\mathbf{v} = \Delta\mathbf{f}$. Then, the sets of all feasible path flow and link flow vectors may be respectively written as:

$$\Omega_f=\{\mathbf{f}|\mathbf{d}=\Gamma\mathbf{f},\mathbf{f}\geq 0\}, \tag{1}$$

$$\Omega_v=\{\mathbf{v}|\mathbf{v}=\Delta\mathbf{f},\mathbf{d}=\Gamma\mathbf{f},\mathbf{f}\geq 0\} \tag{2}$$

We assume that travel cost functions are separable, i.e., the travel time on a link is the function of the flow on that link only. We also assume that the travel time on a route is the sum of travel times on all links that constitute the route, hence $\mathbf{c} = \Delta^T\mathbf{t}$. In view of $\mathbf{v} = \Delta\mathbf{f}$, we may write path cost vector as $\mathbf{c} = \mathbf{c}(\mathbf{f})$. Consider μ_w as the minimum path cost for O-D pair w and $\boldsymbol{\mu} = (\mu_w, w\in W)^T$, where

$$\mu_w=min\{c_{rw},r\in R_w\},w\in W. \tag{3}$$

Therefore, the route flow vector $\mathbf{f}^*$ satisfies the Wardrop's user equilibrium (UE) condition if and only if:

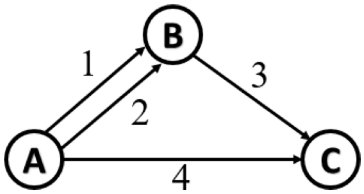


Fig. 1. Network topology of the illustrative example.

$$f_{rw}^{*} > 0 \Rightarrow c_{rw} = \mu_{w}, \tag{4a}$$

$$f_{rw}^{*} = 0 \Rightarrow c_{rw} \geq \mu_{w}. \tag{4b}$$

Condition (4) can be stated as an equivalent variational inequality (UE-VI) problem in terms of path flows (Dafermos, 1980; Smith, 1979): Find a path flow vector $\mathbf{f}^{*} \in \Omega_f$ such that

$$\mathbf{c}(\mathbf{f}^{*})^{T}(\mathbf{f}-\mathbf{f}^{*}) \geq 0, \mathbf{f} \in \Omega_f \tag{5}$$

The above VI problem can also be stated in terms of link flows: Find a link flow vector $\mathbf{v}^{*} \in \Omega_v$ such that

$$\mathbf{t}(\mathbf{v}^{*})^{T}(\mathbf{v}-\mathbf{v}^{*}) \geq 0, \mathbf{v} \in \Omega_v. \tag{6}$$

To define the system optimal (SO) solution, let the marginal link cost function be defined as $s_a = t_a(v_a) + v_a \frac{\mathrm{d}t_a(v_a)}{\mathrm{d}v_a}, a \in A$. The vector of all link marginal travel times is then $\mathbf{s} = (s_a(v_a), a \in A)^{T}$. By the additivity assumption of route costs, we define the vector of route marginal costs as $\mathbf{u} = \Delta^{T}\mathbf{s}$, containing the elements u_{rw} (the marginal cost of route r connecting O-D pair w). The corresponding variational inequality formulation of the SO problem (SO-VI) reads: Find a path flow vector $\mathbf{f}^{*} \in \Omega_f$ such that

$$\mathbf{u}(\mathbf{f}^{*})^{T}(\mathbf{f}-\mathbf{f}^{*}) \geq 0, \mathbf{f} \in \Omega_f \tag{7}$$

Let $IC \subseteq N$ be the set of *intermediate checkpoints* (IC). In a multi-hop control scheme (MHCS) problem, the original O-D demand matrix is converted to a virtual demand matrix according to the ICs. This is achieved using a demand segmentation method detailed in the next subsection. We denote the variables under virtual assignment by adding a prime symbol ($'$) to the original notations. Therefore, the virtual demand vector is denoted by $\mathbf{d'}$ defined on the set of virtual O-D pairs $W' \subseteq \{P \cup IC\} \times \{Q \cup IC\}$, and the corresponding virtual UE link flow vector is represented by $\mathbf{v'} \in \Omega'_{v'} = \{\mathbf{v'} | \mathbf{v'} = \Delta\mathbf{f'}, \mathbf{d'} = \Gamma\mathbf{f'}, \mathbf{f'} \geq 0\}$. Given a solution $\mathbf{v'}^{*}$ to the UE problem, a sensitivity analysis method (like that of Lu (2008)) could be applied to find the Jacobian of link flows w.r.t the virtual demands, denoting as $J' \in \mathbb{R}^{|A| \times |W'|}$. A few other variables, including hopping ratios (the decision variable of the MHCS problem), will be introduced later.

We close by introducing the following assumption to ensure the existence of a unique solution to the UE problem in terms of link flows.

Assumption 1. The link cost function vector $\mathbf{t} = t(\mathbf{v})$ is monotonically increasing and continuously differentiable with respect to $\mathbf{v}$ for $\mathbf{v} \geq 0$. Additionally, the fixed O-D demand vector $\mathbf{d}$ is strictly positive.

We shall see that Assumption 1 also facilitates finding the derivatives of link flows with respect to demand variations.

3.2. Demand segmentation

In MHCS, a portion of traffic between any O-D pair may be guided through certain ICs. Therefore, more O-D pairs must be created corresponding to the movements between original origins/destinations and ICs. Here, we formally describe this procedure, termed *demand segmentation* in this study.

In demand segmentation, the original demand flows are decomposed into equivalent virtual flows and mapped to the set of augmented O-D pairs that includes the ICs, based on the given hopping ratios. For example, assume that drivers who initially travel from origin p to destination q are now guided to pass through the IC m before heading to q. This is equivalent to decomposing the original demand, d_{pq}, to two virtual demands, $d'_{pm} = d'_{mq} = d_{pq}$. Once the virtual O-D matrix is determined for the given ICs and hopping ratios, the outcome of a multi-hop control can be obtained by solving a UE traffic assignment problem based on that matrix. This outcome can then be compared against the original assignment result, in terms of the total link flows and the system cost.

We are now ready to formally introduce the matrix of hopping ratios $\boldsymbol{\lambda}$, which determines the share received by each IC from each entry of the original O-D matrix. Hopping ratios can be arranged into the matrix $\boldsymbol{\lambda} = \{\lambda_{w'w}\}$, $w' \in W'$, $w \in W$ (see Table A1 in Appendix). Each element of the matrix is defined as the proportion of original demand between O-D pair w or (p,q), which is routed among virtual O-D pair w' or (i,j). Thus, the virtual demand matrix could be related to the original demand matrix by the relation $\mathbf{d'} = \boldsymbol{\lambda}\mathbf{d}$.

Consider a simple network with 3 nodes and 4 links as shown Fig. 1. Linear link performance functions are used for the purpose of

Table 1
Demand and link performance functions for the illustrative example.

Link	Link Performance Functions
1	$t_1 = v_1$
2	$t_2 = 2$
3	$t_3 = 1$
4	$t_4 = 1.5v_4$

$d_{AC} = 2.2$ units of demand.

Table 2
UE and SO solutions for the traffic assignment problem defined in Fig. 1 and Table 1.

Optimal conditions	Path Flow			Link Flow			
	1–3	2–3	4	1	2	3	4
UE	0.92	0	1.28	0.92	0	0.92	1.28
SO	1	0.2	1	1	0.2	1.2	1

illustration (see Table 1) and 2.2 units of demand are assumed to travel from node A to node C, that is $\mathbf{d} = [d_{AC}] = (2.2)$. The UE and SO solutions for this network can be found by solving the UE-VI and SO-VI problems, as reported in Table 2. As seen, 1.2 units of demand (54% of the original demand) pass through node B under the SO state while at UE, node B is visited by 0.92 units of demand (42% of the original demand). Let us introduce node B as an IC and assume that 54% of the original demand, as in the SO state, is guided to go through B (the objective of the current study is to optimize this hopping ratio). Accordingly, demand segmentation will yield the (single-dimension) hopping ratio matrix as $\boldsymbol{\lambda}^{(0)} = [\lambda_{AB,AC} \quad \lambda_{BC,AC} \quad \lambda_{AC,AC}]^T = [0.54 \quad 0.54 \quad 0.46]^T$ and the vector of virtual demand as $\mathbf{d}'^{(0)} = [d'_{AB} \quad d'_{BC} \quad d'_{AC}]^T = [1.2 \quad 1.2 \quad 1]^T$.

3.3. Formulation of multi-hop control scheme (MHCS) problem

MHCS aims to minimize the total system cost while satisfying the route choice constraints dictated by the ICs. It achieves this goal by choosing the optimal set of hopping ratios and anticipate the impact of this choice on the distribution of virtual demand among the routes that connect the virtual O-D pairs. The MHCS problem can be readily formulated as a mathematical program with equilibrium constraints (MPEC) as follows:

$$\min_{\boldsymbol{\lambda}} \quad Z(\boldsymbol{\lambda}) = t(\mathbf{v}'^*(\boldsymbol{\lambda}))^T \mathbf{v}'^*(\boldsymbol{\lambda}) \tag{8}$$

$$\text{s.t.} \quad \mathbf{d}' = \boldsymbol{\lambda}\mathbf{d}, \tag{9}$$

$$\sum_{m \in IC \cup \{q\}} \lambda_{pm,pq} = 1, \quad (p,q) \in W, \tag{10}$$

$$\sum_{i \in IC \cup \{p\}} \lambda_{im,pq} = \sum_{j \in IC \cup \{q\}} \lambda_{mj,pq}, \quad m \in IC, \ (p,q) \in W, \tag{11}$$

$$0 \le \boldsymbol{\lambda} \le 1, \tag{12}$$

$$t(\mathbf{v}'^*(\boldsymbol{\lambda}))^T (\mathbf{v}' - \mathbf{v}'^*(\boldsymbol{\lambda})) \ge 0, \quad \forall \mathbf{v}' \in \Omega'_{\mathbf{v}'}. \tag{13}$$

The objective here is to minimize the total travel time experienced by all users of the network. Constraint (9) converts the original demand to virtual demand based on the matrix of hopping ratios. Constraint (10) represents the flow conservation condition for each O-D pair, before and after the conversion. Constraint (11) states the passing flows at each IC must satisfy the conservation condition (i.e., total inflow equals total outflow). Constraint (12) defines the feasible range of each hopping ratio. Finally, Constraint (13) states the link flows used to evaluate the objective (8) must be the solution to the VI-UE problem defined by the virtual O-D demand matrix associated with $\boldsymbol{\lambda}$.

The following result establishes solution existence for the above MPEC problem.

Proposition 1. *A solution to the MHCS problem given in (8)-(13) exists when* Assumption 1 *holds.*

Proof.. The UE problem in (13) has a unique solution whenever Assumption 1 is satisfied. Let $\Omega_{\boldsymbol{\lambda}}$ be the feasible set for $\boldsymbol{\lambda}$, and note that there exists a mapping $\Psi : \Omega_{\boldsymbol{\lambda}} \to \Omega'_{\mathbf{v}'}$ such that $\mathbf{v}' = \Psi(\boldsymbol{\lambda})$ is the solution to the follower UE problem. Thus, Problem (8)–(13) defines a Stackelberg game, in which $\boldsymbol{\lambda}$ is the decision undertaken by the leader, and $\mathbf{v}'$ is the response of the follower. The reaction map

Ψ is bounded because $\mathbf{d}'$, as the multiplication of two positive and bounded variables (i.e., $\mathbf{d}$ and λ), is positive and bounded. In addition, the objective function (8) is continuous and the set Ω_λ is compact. Thus, the MPEC must have a solution as per Corollary 1 of Harker and Pang (1988). □

Although Proposition 1 establishes the existence of a solution to the MHCS problem, it does not exclude the possibility of multiple local optima. However, the computational experiments conducted by Yang and Bell (1998) show that the strong convexity of the objective function and simple linear constraints at the upper-level makes it relatively easy to locate an "efficient" local optimum, i.e., one that is close to the global optimum. Moreover, it is often sufficient for practical purpose if a local MHCS optimum could provide a markedly better performance compared to alternatives such as UE or SO solutions.

One could try to jointly optimize the set of ICs and hopping ratios, which would not change the basic structure of the MPEC, but nevertheless lead to an intractable combinatorial problem. While such a problem can be solved by metaheuristics, we do not see the appeals of going through such a laborious exercise that seems to promise little insights. Instead, we shall focus on optimizing hopping ratios, which lends itself to more effective heuristic algorithms and is a key building block for joint optimization. However, since the choice of ICs does play a crucial role in a multi-hop control scheme, we shall test and compare a few heuristic choice rules in Section 5. Additionally, the current formulation does not directly consider unfairness. A typical approach is to limit the extra delay imposed by the control scheme below a predefine threshold (see Jahn et al. (2005)). Such extra constraints, however, will add another layer of complexity in the lower-level problem, making it much harder to solve.

4. Solution algorithm

4.1. Sensitivity-analysis-based (SAB) algorithm

The underlying idea in the sensitivity-analysis-based (SAB) algorithm is to find the derivatives of link flows with respect to a perturbed parameter (in our case, the virtual demand), and use it to guide the current solution towards a local optimum. In the UE traffic assignment problem, the link flows are viewed as an implicit nonlinear function of demand variations. Once the derivative information becomes available, this function can be approximated by a linear Taylor expansion.

In our algorithm, we employ the method of Lu (2008) to obtain the Jacobian of arc flows with respect to the variation in the fixed virtual demands. It calculates the Jacobian by matrix multiplication together with the solution of a linear system of equations. While the dimension of the linear equation system is bounded by the number of arcs, it is typically much smaller in practice thanks to the compactness of the critical cone. A key requirement of the method is the nondegeneracy condition, which holds as long as the virtual demand vector is strictly positive. This, however, is not a restrictive constraint because traffic assignment problems naturally require strictly positive demand[1]. The reader is referred to Lu (2008) for more details.

The choice of the initial solution is crucial to the performance of the algorithm. Each local optimum has its own *basin of attraction*, and once an initial solution falls into it, the algorithm is "trapped" within that basin, and destined to converge to that local optimum. In this study, we propose to use the SO solution to guide the choice of the initial point of the solution algorithm because, like the SO problem, the MHCS problem also aims to minimize the system cost.

To configure the initial hopping ratios from an SO solution, we need the flows on all paths used at SO. However, it is well known that path flow solutions are not unique in standard static traffic assignment problems. To avoid using arbitrary path flow solutions in different scenarios (which could raise the issue of comparability), we employ the most likely route flow solution, which can be obtained by solving the maximum entropy user equilibrium problem (Lu and Nie, 2010; Xie and Nie, 2019). From SO route flows and the original O-D matrix, we derive the share of ICs, i.e., $\lambda^{(0)}$. Below, Algorithm 1 presents the main steps for the SAB algorithm and Algorithm 2 outlines the steps for setting the initial hopping ratio matrix ($\lambda^{(0)}$).

Algorithm 1. Sensitivity analysis based (SAB) algorithm to MHCS

1: **Inputs**:
2: Network $G(N,A)$, link cost function $t_a, a \in A$, demand vector $\mathbf{d}$, set of intermediate checkpoints IC, gap tolerances $\varepsilon_1, \varepsilon_2$, maximum iteration Y.
3: **Output**: Optimal matrix of hopping ratios λ^*
4: **Initialize**:
5: Find the most likely SO route flow solution $\mathbf{f}^{(0)}$ that has a relative gap smaller than ε_1. Determine $\lambda^{(0)}$ using Algorithm 2 and then $\mathbf{d}'^{(0)}$ using the relationship $\mathbf{d}'^{(0)} = \lambda^{(0)}\mathbf{d}$. Set $k = 0$.
6: **Main Loop**:
7: **while** $k < Y$ and $e > \varepsilon_2$ **do**
8: Find the UE traffic pattern $\mathbf{v}'^{*(k)}$ by solving the UE-VI problem (13) with the set of virtual demands $\mathbf{d}'^{(k)}$ determined by (9). The relative gap of the solution should be smaller than ε_1.
9: Perform the sensitivity analysis of the UE at the solution $\mathbf{v}'^{*(k)}$ and define $J'(k)$.

(*continued on next page*)

[1] If an equilibrium solution contains at least one path with zero flow, it is a degenerate solution, which calls the differentiablity into question. In our algorithm, when such a degenerate solution is found, we compute semi-derivatives for the coordinated directions, as suggested in Lu (2008). It is worth noting that, because a UE state in a general network typically corresponds to infinitely many path flow solution, it is rare that these solution are all degenerate (Yang and Bell, 2007).

(*continued*)

Algorithm 1. Sensitivity analysis based (SAB) algorithm to MHCS

10: Using the first-order Taylor's expansion formula, write $\widetilde{\mathbf{v}}'(\boldsymbol{\lambda})$ as follows: $\widetilde{\mathbf{v}}'(\boldsymbol{\lambda}) = \mathbf{v}'^{*(k)} + J'^{(k)}(\mathbf{d}'(\boldsymbol{\lambda}) - \mathbf{d}'^{(k)})$.
11: Replace $\mathbf{v}'^{*}$ in (8) with $\widetilde{\mathbf{v}}'(\lambda)$ and solve the upper-level problem (8)-(12) to determine $\boldsymbol{\lambda}^{(k+1)}$.
12: Update the convergence criterion $e = \|\boldsymbol{\lambda}^{(k+1)} - \boldsymbol{\lambda}^{(k)}\|$. Set $k = k + 1$.
13: **end while**
14: set $\boldsymbol{\lambda}^{*} = \boldsymbol{\lambda}^{(k)}$.

Algorithm 2. Setting initial hopping ratio matrix ($\boldsymbol{\lambda}^{(0)}$) from a given SO route flow vector.

Inputs:
Network $G(N,A)$, set of entropy-maximizing SO routes R, vector of all SO route flows $\mathbf{f}$, set of all O-D pairs W, demand vector $\mathbf{d}$, set of intermediate checkpoints IC.
Output: Initial matrix of hopping ratio $\boldsymbol{\lambda}^{(0)}$
Configure $\boldsymbol{\lambda}$ matrix through a row-generation technique where only the used virtual O-D pairs (rows) are added to $\boldsymbol{\lambda}$:
For each $w \in W$ and $r_w \in R_w$
  Define IC_{rw} as the sequence (set) of intermediate checkpoints on path r_w. Begin the sequence with the origin of w and finish it with the destination of w.
  For every two consecutive elements of IC_{rw} denoted by m, n
  Define $w' = (m, n)$ if it has not already been defined.
    Set $\lambda_{w'w} = f_{rw}/d_w$, if $\lambda_{w'w}$ has not already been defined; otherwise, set $\lambda_{w'w} = \lambda_{w'w} + f_{rw}/d_w$.
  end for
  Augment the wth column of $\boldsymbol{\lambda}$ using the hopping ratios $\lambda_{w'w}$.
end for
set $\boldsymbol{\lambda}^{(0)} = \boldsymbol{\lambda}$.

Algorithm 1 decomposes the MPEC into two simpler problems and solves them iteratively until a desired level of convergence is met. The first problem is the UE-VI problem (line 8) with a set of fixed virtual demands. The second problem is the upper level problem (line 11) that seeks to update the hopping ratios, with linearly approximated link flows (lines 9 and 10). We solve both the UE-VI and SO-VI problems using the gradient projection algorithm (Jayakrishnan et al., 1994) coded in MATLAB. For the upper level problem on line 11, which is a nonlinear problem with linear constraints, the built-in interior-point algorithm in MATLAB is used.

Note that in the SAB algorithm the derivatives (or sensitivity) are only valid within a close neighborhood of the fixed demands. Specifically, the deviations (or perturbation) from the fix demands must not be large enough to change the set of used UE routes in the current iterations ($R^{*(k)}$). This requirement may be violated in the early iterations of the algorithm, when the perturbations tend to be large. However, as the algorithm converges to a local optimum, the conditions are usually satisfied easily. Thus, while the performance of the algorithm in the first few iterations could be affected by this potential violation, its overall impact is expected to be negligible.

4.2. Illustrative example

Let us revisit the illustrative example introduced in Section 3 where we obtained $\boldsymbol{\lambda}^{(0)}$and $\mathbf{d}'^{(0)}$. That means we have followed Algorithm 1 up to line 6. Now, we continue by solving the UE problem using the virtual demand (i.e., $\mathbf{d}'^{(0)}$), which produces the link flows specified at $k = 0$ according to line 8 of Algorithm 1. The results are given in Table 3. The derivatives of the link flows w.r.t virtual fixed demand rates (line 9 of Algorithm 1) can be obtained as follows (see Lu, 2008):

$$J'^{(0)} = \frac{\partial \mathbf{v}^{*(0)}}{\partial \mathbf{d}'^{(0)}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Solving the upper-level problem (8)-(12) (lines 10 and 11 of Algorithm 1), the values of $\boldsymbol{\lambda}^{(k+1)}$ is determined as: $\lambda^{(1)}_{AB,AC} = \lambda^{(1)}_{BC,AC} = 0.509$ and $\lambda^{(1)}_{AC,AC} = 0.491$, yielding $d'^{(1)}_{AB} = d'^{(1)}_{BC} = 1.12$and $d'^{(1)}_{AC} = 1.08$. In iteration $k = 1$, the UE link flow solution corresponding to this new virtual demand matrix is $\mathbf{v}'^{*(1)} = [1.12, 0, 1.12, 1.08]^T$ (Table 3). In the next step, we compute the derivatives again, and find they are the same as those from the previous iteration, i.e., $J'(1) = J'(0)$. Hence, the solution to the upper-level problem would still be $\mathbf{v}'^{*} = [1.12, 0, 1.12, 1.08]^T$. This suggests that the convergence criterion is met, and the UE traffic pattern obtained in this iteration ($k = 1$) is the final MHCS solution.

We proceed to compare the MHCS's solution with the UE and the SO solutions. For this purpose, we first introduce a number of

Table 3
MHCS Solution for the illustrative example.

	Sub-OD pair/ Link of the connecting path				Links			
	(A,B)/ 1	(A,B)/ 2	(B,C)/ 3	(A,C)/ 4	1	2	3	4
Flow at k = 0	1.2	0	1.2	1	1.2	0	1.2	1
Flow at k = 1*	1.12	0	1.12	1.08	1.12	0	1.12	1.08

Table 4
List of performance measures.

N. O.	Measure	Description
1	$\mathbf{TT'} = t(\mathbf{v'}^*)^T\mathbf{v'}^*$	*total travel time* of the network with virtual demands.
2	$\mathbf{RTTS}^{\dagger} = \dfrac{(\mathrm{TT}_{UE} - \mathrm{TT}')}{(\mathrm{TT}_{UE} - \mathrm{TT}_{SO})}$	*relative total travel time saving* as the ratio between the travel time saving achieved by the MHCS solution (compared to the UE solution) to the maximum possible travel time saving.
3	$\mathbf{TU} = \sum_{w\in W}\sum_{r\in R_w}(c_{rw} - \mu_w)f_{rw}$	*total unfairness* as the sum of differences between each vehicle's actual travel time and the least possible travel time at a given solution.
4	$\mathbf{STU} = \dfrac{(\mathrm{TT}_{UE} - \mathrm{TT}')}{\mathrm{TU}}$	*saving-to-unfairness* as the ratio between the total travel time savings of a solution (SO or MHCS) and its TU.
5	$\mathbf{RSTG} = \dfrac{RTTS}{Share\ of\ guided\ flow} = \dfrac{\mathrm{RTTS}}{\frac{\text{guided flow}}{\text{total demand}}}$	*relative saving-to-guided flow* as the amount of RTTS achieved for each unit share of guided demand under MHCS.
6	$\mathbf{DTTS}^{\ddagger} = \dfrac{(\mathrm{TT}_{UE} - \mathrm{TT}')}{d_{w_{exp}}}$	*demand-normalized total travel time saving* that normalizes the time savings of MHCS based on the demand of the selected O-D pair.
7	$\mathbf{TTTS}^{\dagger\dagger} = \dfrac{(\mathrm{TT}_{UE} - \mathrm{TT}')}{TT_{w_{exp}}}$	*travel time-normalized total saving* that normalizes the time savings of MHCS based on the total travel time of the selected O-D pair at UE.
8	$\mathbf{RTU} = \dfrac{\sum_{r\in R_w}(c_{rw} - \mu_w)f_{rw}}{TT_{w_{exp}}}$	*relative total unfairness* as the total unfairness divided by the total travel time contributed by the selected O-D pair.

†TT_{UE} and TT_{SO} are the total travel time of the network under the UE and SO states, respectively.
‡ As used in Sec. 5.1, $d_{w_{exp}}$ is the demand of the O-D pair selected for the experiment, indexed as w_{exp}.
†† As used in Sec. 5.1, $TT_{w_{exp}}$ is the total travel time of the O-D pair selected for the experiment at the UE state.

Table 5
Comparison between MHCS, UE and SO.

Measure	UE	SO	MHCS
TT (unit of time)	4.224	4.100	4.124
TU(unit of time)	0	0.80	0.56
RTTS	0	1	0.81
STU	–	0.155	0.179
RSTG	–	1.818	1.588

performance measures, as summarized in Table 4. The MHCS problem solely focuses on reducing the total cost of the network as measured by TT′. However, to gauge the overall performance of ICs, we need to include the unfairness measures (e.g., TU), too. Note that for RTTS, STU, and RSTG measures, the larger is the value, the better.

Table 5 reports the results in the illustrative example. We see that the MHCS solution has achieved an RTTS of 0.81, suggesting it has gained 81% of the possible improvement. It has a total unfairness of about 0.56, compared to 0.8 at SO, a 30% reduction. Accordingly, MHCS outperforms SO with respect to the saving-to-unfairness (STU) index by about 15%. That is, MHCS strikes a better balance between the total travel time savings and the overall unfairness. For every unit share of traffic flow under control, MHCS renders a travel time saving of about 1.591, compared to 1.833 for the SO solution. Thus, SO delivers a better RTSG value than MHCS in this case, i.e., it generates greater travel time savings for each unit of traffic under control (here the share of traffic under control is $\frac{1.12}{2.2} \cong 0.51$ for MHCS, and $\frac{1.2}{2.2} \cong 0.55$ for SO). It is worth emphasizing, however, that each vehicle's route is fully controlled at SO, whereas MHCS merely mandates a few intermediate checkpoints. In a large network where a path may contain hundreds of links, the difference in the control burden brought by the two schemes can be rather significant.

5. Results of numerical experiments

All experiments are conducted on the Sioux Falls network (Fig. 2), obtained from the GitHub repository "Transportation Networks for Research"[2]. This network is well known in transportation research, and for computationally demanding problems like MHCS, it amounts to a reasonable benchmark (Di et al., 2018; López-Ramos et al., 2019; Zhao and Zhang, 2020). All experiments are run on a Windows 64-bit computer with 2.5 GHz Intel Core-i5 CPU and 6 GB 1600 MHz DDR3 RAM. Three sets of experiments are conducted. In Section 5.1, we explore how the choice of ICs affects the MCHS solution when only one O-D pair is under control. Section 5.2 examines the case when traffic between multiple O-D pairs can be guided through selected sets of ICs. In Section 5.3, we first describe a heuristic procedure used to rank ICs in a general network. Then, we test a full version of MHCS, i.e., when traffic from all O-D pairs are under control, using various ICs selected by the procedure.

[2] https://github.com/bstabler/TransportationNetworks. Accessed on March 13, 2019.

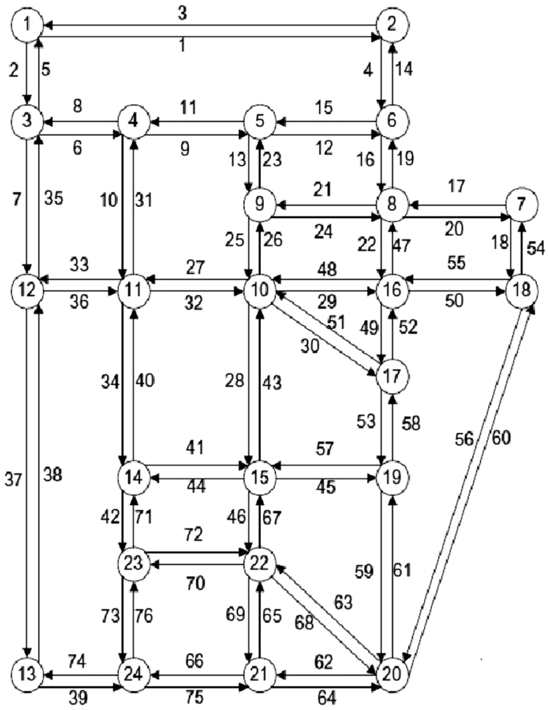


Fig. 2. Topology of Sioux Falls network.

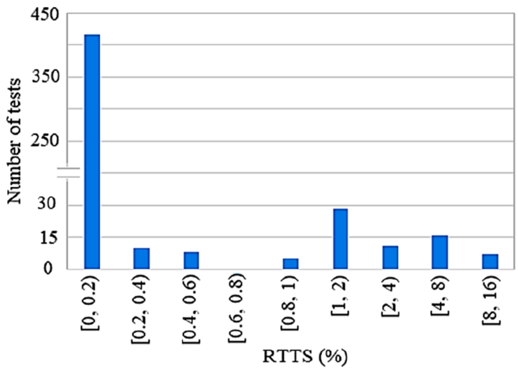
Table 6
Heuristic rules for choosing ICs.

Number	Rule
1	Most congested node on the UE routes
2	Least congested node on the UE routes
3	Most used node on the UE routes
4	Least used node on the UE routes
5	Most congested node on the SO routes
6	Least congested node on the SO routes
7	Most used node on the SO routes
8	Least used node on the SO routes
9	Most congested node on the exclusive* UE routes
10	Least congested node on the exclusive UE routes
11	Most used node on the exclusive UE routes
12	Least used node on the exclusive UE routes
13	Most congested node on the exclusive SO routes
14	Least congested node on the exclusive SO routes
15	Most used node on the exclusive SO routes
16	Least used node on the exclusive SO routes

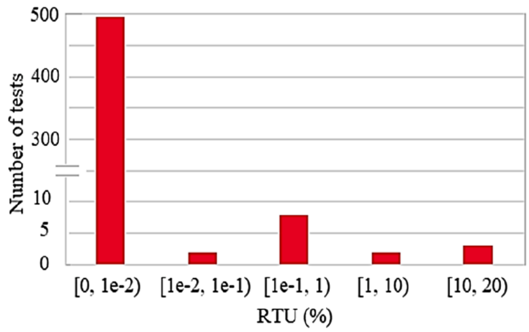
*The UE (or SO) route is exclusive if it is only used under the UE (or SO) traffic pattern.

5.1. Experiments with different rules for choosing intermediate checkpoints

To choose the ICs, we focus on the differences and similarities between maximum entropy route sets corresponding to UE and SO solutions. Table 6 lists 16 heuristic rules that form the basis for ranking ICs in Sec. 5.3. Specifically, the level of congestion at a node is measured by the amount of the flow traversing through it, and the usage of a node is defined by the number of times it lies on used routes. For example, Rule #1 picks the node that is most congested, and Rule #3 picks the node with highest usage. With these heuristic rules, Algorithm 3 outlines the main steps for the MHCS single-OD tests, i.e., when traffic between one O-D pair is controlled and the other O-D pairs follow the regular UE conditions. The purpose of Algorithm 3 is to obtain the values of heuristic rules with



(a) Histogram of RTTS.



(b) Histogram of RTU.

Fig. 3. RTTS and RTU profile of the single-OD tests.

respect to a set of desired measures. This will be used later in Section 5.3 to select the set of ICs in a general network.

Algorithm 3. Finding the values of heuristic rules by single-OD tests of MHCS

1: **Inputs**: inputs of Algorithms 1 and 2.
2: **Output**: the values of the heuristic rules for choosing ICs.
3: **Initialize**
4: Determine a set of used measures, denoted by $M = \{M_n | n = 1, 2, \dots, h\}$
5: Choose an appropriate sample of O-D pairs W_s.
6: **Main loop**
7: **For** each O-D pair w in W_s
8: Form L_w as the set of applicable rules for w.
9: **For** each rule l in L_w
10: Define I_l^w as the set of ICs corresponding to rule l.
11: If l is a UE-based rule, define the starting point ($\lambda^{(0)}$) w.r.t the UE routes; otherwise define it w.r.t the SO routes, using Algorithm 2[3]
12: **For** each i inI_l^w
13: Conduct a single-OD MHCS experiment which only controls the traffic from w to get, for each measure h, $M^w_{h,l,i}$.
14: **end for**
15: Set $M^w_{h,l} = \sum_i M^w_{h,l,i}/|I_l^w|$.
16: **end for**
17: **end for**
18: Average the values of the measures for each rule over all O-D pairs, i.e., set $M_{h,l} = \sum_w M^w_{h,l}/N_l, \ \forall h, l.$ where N_l is the number of O-D pairs in W_s for which the lth rule was applicable.

[3]Algorithm 2 defines the starting point w.r.t the SO routes, but the approach is also applicable w.r.t the UE routes.

The following remarks are in order about Algorithm 3.

- The algorithm first identifies the performance measures against which the rules are valued. The MHCS outputs are evaluated based on time savings and total fairness. However, because we conduct each experiment on a single O-D pair that may have a very different level of demand from other O-D pairs, the normalized measures (measures 6 to 8 in Table 4, or DTTS, TTTS, and RTU) are adopted in our experiments.
- For O-D pairs with few routes, many rules in Table 6 may produce the same node. To limit the impact of this problem in choosing W_s (the sample of O-D pairs in line 5), our experience suggests that focusing on O-D pairs with at least three or more UE or SO routes is enough to obtain sufficiently diverse intermediate checkpoints. Intuitively, MHCS is more effective when it controls the O-D pairs with a more diverse set of routes. In Sioux Falls network, 41 O-D pairs have three or more UE or SO routes and thus, we include all of them in the test.
- We note that not all rules are applicable for a given O-D pair. For instance, if the UE routes of an O-D pair are a subset of its SO routes, then Rules 9 to 12 are not applicable because that O-D pair has no "exclusive UE route". By the same token, when the SO routes are a subset of UE routes, Rules 13 to 16 are inapplicable. Thus, we need to identify the applicable rules for the tested O-D pair (line 8). In Sioux Falls network, the total of 500 valid cases of experiments are obtained from applying sixteen rules to 41 O-D pairs.
- On line 10, we identify the IC(s) corresponding to the current rule l. Sometimes, a choice rule produces more than one node. In such an event, all nodes are tested (lines 12–13) and the average of all test results is taken as the output (line 15). In each test (as in line 11), if the selected node belongs to the UE rules (Rules 1–4 and 9–12), the initial solution ($\lambda^{(0)}$) is configured based on the UE rather than SO path flows. In line 13, the single-OD MHCS test is conducted, i.e., MHCS is solved for one IC and one O-D pair under control. All other O-D pairs are not subject to any control and thus simply follow the regular UE conditions. This setup helps us focus on the

Table 7
Choice rules ranking based on the average DTTS.

Rule #	Rule description	Ave. DTTS
15	Most used exc. SO	18.699
13	Most cong. exc. SO	15.818
7	Most used SO	5.266
6	Least cong. SO	3.992
16	Least used exc. SO	3.455
5	Most cong. SO	3.161
8	Least used SO	2.658
10	Least cong. exc. UE	1.354
12	Least used exc. UE	1.294
4	Least used UE	0.992
14	Least cong. exc. SO	0.878
2	Least cong. UE	0.851
3	Most used UE	0.789
9	Most cong. exc. UE	0.457
11	Most used exc. UE	0.382
1	Most cong. UE	0.247

Table 8
Choice rules ranking based on the average TTTS.

Rule #	Rule description	Ave. TTTS
7	Most used SO	33.423
8	Least used SO	22.99
13	Most cong. exc. SO	22.773
15	Most used exc. SO	22.658
6	Least cong. SO	4.587
5	Most cong. SO	4.094
16	Least used exc. SO	2.529
2	Least cong. UE	1.515
4	Least used UE	1.309
10	Least cong. exc. UE	0.845
12	Least used exc. UE	0.838
3	Most used UE	0.719
14	Least cong. exc. SO	0.425
1	Most cong. UE	0.356
11	Most used exc. UE	0.275
9	Most cong. exc. UE	0.257

impact of different rules, by excluding the potential interactions between O-D pairs. Thus, Constraints (10) and (11) in the MHCS formulation can be simplified as:

$$\lambda_{pm,pq} + \lambda_{pq,pq} = 1, \quad p = P_{exp},\ q = Q_{exp},\ m = IC_{exp}, \tag{14}$$

$$\lambda_{pm,pq} = \lambda_{mq,pq}, \quad p = P_{exp},\ q = Q_{exp},\ m = IC_{exp,} \tag{15}$$

where (P_{exp}, Q_{exp}) is the selected O-D pair and IC_{exp} is the selected intermediate checkpoint. After all the O-D pairs are examined, we average the values of each measure for each rule (line 18).

Fig. 3 give the histograms of the experiment results of experiments in terms of RTTS and RTU respectively. RTTS (measure 2 in Table 4) is reported here because the reader may find it more relatable than normalized measures such as DTTS and TTTS. When we rank the choice rules, however, DTTS and TTTS measures will be applied.

RTTS values reported in Fig. 3a are small in most cases, since we only control traffic between one O-D pair with one IC in each experiment. As a first-order estimation, the average share of each O-D pair in the total travel time saving at SO would be about $\frac{1}{528} = 0.00189 \cong 0.19\%$. This explains why most RTTS values lie within [0, 0.2) (see Fig. 3a). What is more noteworthy is the fact that in certain cases controlling a single O-D with one IC could achieve a saving almost two orders of magnitude greater than the average (for seven O-D pairs MHCS achieves an RTTS value in the range of 8–16%). In 34 cases (over 6% of all cases), MHCS reduces the total system travel time by 2% or more by merely guiding traffic from one O-D pair through one IC.

The maximum value of RTTS is 13.5%, achieved by O-D pair (16,19). The O-D pair has a demand of 1300 veh (about 0.36% of the total demand). The total travel time contributed by this O-D pair is 22019 min (about 0.3% of the total travel time) at UE, and 18051 min at SO (a total saving of 3968 min, corresponding to an RTTS of 1.39%). Clearly, even though MCHS just guides traffic from O-D pair (16,19) through one intermediate node, its impact extends far beyond the O-D pair itself. This finding generally agrees with the literature (e.g., Zhang and Nie, 2018) on the high value of certain O-D pairs for traffic management. Yet, it provides a striking example that highlights how little control is needed (in this case, 0.36% of total demand is asked to add one required checkpoint in their route)

Table 9
Choice rules ranking based on the average RTU.

Rule #	Rule description	Ave. RTU (%)
11	Most used exc. UE	2.64E-05
5	Most cong. SO	5.06E-05
9	Most cong. exc. UE	6.48E-05
10	Least cong. exc. UE	8.70E-05
3	Most used UE	1.02E-04
12	Least used exc. UE	4.68E-04
1	Most cong. UE	1.00E-03
4	Least used UE	1.03E-03
8	Least used SO	2.02E-02
7	Most used SO	2.46E-02
6	Least cong. SO	2.99E-02
16	Least used exc. SO	8.26E-02
13	Most cong. exc. SO	8.66E-02
14	Least cong. exc. SO	9.09E-02
15	Most used exc. SO	1.14E-01
2	Least cong. UE	1.60E-01

Table 10
O-D pairs that exhibit high potential to improve network efficiency.

Rank	Origin	Destination	Demand	RTTS (%)	Rank	Origin	Destination	Demand	RTTS (%)
1	16	19	1300	13.49	4	14	16	700	7.95
2	16	15	1200	12.74	5	20	10	700	6.61
3	13	15	700	9.18	6	6	17	2000	4.88

Table 11
Multi-OD tests of MHCS.

Test N.O.	OD pairs	ICs	RTTS (%)	TU (min)	Guided vehicles (%)
1	(16,19)	18	13.49	6074.60	0.36
2	(16,19), (16,15)	18, 20	15.69	14816.12	0.63
3	(16,19), (16,15), (13,15)	10, 18, 20	20.96	16773.89	0.79
4	(16,19), (16,15), (13,15), (14,16)	10, 18, 20	23.41	17331.20	0.91
5	(16,19), (16,15), (13,15), (14,16), (20,10)	10, 18, 20	29.61	25512.57	1.62
6	(16,19), (16,15), (13,15), (14,16), (20,10), (6,17)	10, 18, 20	31.36	30799.07	1.71
7	All (SO)	All	100	195045.73	14.11*

*This value is reported by Chen et al. (2020) as the minimum ratio of the fully controlled vehicles to achieve the SO state in Sioux Falls network. In MHCS, vehicles are not fully controlled and hence more vehicles might be needed to achieve SO. However, we keep using the above value as it defines a useful lower bound for the share of guided vehicles to achieve the SO state under MHCS.

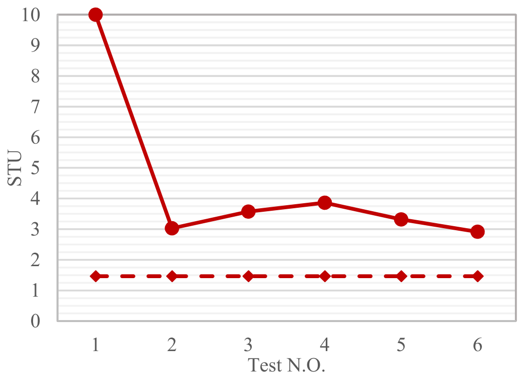
to achieve a substantial efficiency gain. Fig. 3b suggests that in most cases MCHC leads to very small relative total unfairness (RTU) values. There are only 5 cases for which RTU is larger than 1%, and the maximum is 12.2%, achieved by O-D pair (14,12).

Tables 7 to 9 attempt to rank the 16 heuristic choice rules listed in Table 6 in accordance with the values of measures achieved in line 18 of Algorithm 3. Tables 7 and 8 rank the rules based on the DTTS and TTTS measures, respectively. As seen, the rules based on SO routes perform much better in saving the total travel time than those based on the UE rules. The performance of the top-ranking rules from both tables is impressive. Rule #15 has a DTTS of 18.7, suggesting it saves about 19 min per each guided vehicle. On the other hand, Rule #7 saves about 33 min on average per every minute spent by travelers associated with the corresponding O-D (TTTS = 33.4). Note that the DTTS and TTTS values in some cases are large because guiding traffic from one O-D pair could also impact the flows from other O-D pair. But the other O-D pairs are not counted in the denominators of the corresponding measures (Measures 6 and 7 in Table 4) since they are still on the shortest routes.

Another notable point from the above experiments is the effect of the initial point on the results of MHCS. It is not uncommon that an SO and a UE rule produce the same candidate node as IC. However, the final MHCS results would be very different because different initial points ($\lambda^{(0)}$) are used for these rules: SO routes are used to generate the initial point for an SO rule, and UE routes are used for a UE rule (see line 11 of Algorithm 3). For example, for O-D pair (21, 17), the intermediate checkpoint corresponding to both Rule 4 (a UE rule) and Rule 5 (an SO rule) is node 18. While the solution algorithm and the tested IC are the same in both cases, the SO-based initial point leads to a time saving almost twice as large as the UE-based initial point (350 vs. 177 min).

Table 9 ranks the rules in the increasing order with respect to the average RTU measure. As expected, the rules based on UE routes generally perform better with respect to fairness, in most cases yielding RTU values at least an order of magnitude smaller than those obtained by SO rules. A remarkable exception is Rule #5, which is an SO based rule but is ranked the second in fairness, much higher

(a) Change in RSTG. All cases perform much better than SO (dashed line represent the SO).



(b) Change in STU. All the cases performs better than SO (dashed line represent the SO).

Fig. 4. Efficiency of multi-OD cases of MHCS compared to SO.

than most UE based rules. A close look reveals that the reason behind the abnormality is that the node selected by this rule happens to be used by nearly all flows in most cases. As a result, there is no discrepancy among the users about whether to pass through the node, which means no one is treated unfairly under the multi-hop scheme.

5.2. Experiments with a limited set of O-D pairs

Among the 41 O-D pairs tested in the previous section, some O-D pairs exhibited extraordinary potential to improve network efficiency through the multi-hop strategy. This motivates us to ask the question, what if we could simultaneously control traffic from several such promising O-D pairs? In this section, we solve a version of MHCS problem that includes a subset of O-D pairs selected from six O-D pairs that have the highest RTTS values based on the results from Section 5.1 (see Table 10). The selected ICs are also those with the highest RTTS values in the O-D pairs tests. Table 11 reports the results of six tests, in which the number of high potential O-D pairs included increases from one to six. The first experiment has only one IC (18), the second has two (18 and 20), and the other four experiments all have three ICs (18, 20, 10). The last row in Table 11 reports the results corresponding to SO solution for comparison.

We expect that, controlling more O-D pairs through more ICs would achieve greater efficiency, but at the expense of worsening the unfairness issue. The results presented in Table 11 largely confirm this expectation. When all six O-D pairs (out of 528), or about 1.71% of all traffic, are under control, more than 31% of all travel time savings could be obtained. Meanwhile, the Total Unfairness (TU) measure increases from 6,075 min in test 1 to 30,799 min in test 6, which amounts to about one sixth of the TU value at SO. A close look reveals that in test 6, the maximum unfairness value (incurred for O-D pair (6,17)) is 12 min, which constitutes 27% of the entire journey time for that O-D pair. Thus, under MHCS, some travelers might be asked to endure fairly large sacrifices, even though the overall TU measures may seem modest, especially in lieu of the gains in travel time savings. As stated earlier, since the unfairness measure is not included in the objective or constraints of the MHCS formulation, the above observation is hardly surprising.

Another finding from Table 11 is that the “marginal” gain of travel time savings from controlling additional O-D pairs decreases rather quickly. By controlling just one O-D pair (16,19), test 1 achieves a saving of 13.49%. Adding the O-D pair with the second highest potential, (16, 15), into the control list, however, merely increases the saving by about two percentage points. Even worse, the addition worsens all efficiency measures. In this particular case the reason for the poor performance is obvious: the two O-D pairs have to compete with each other for using the intermediate nodes (18 and 20), which means none of them could achieve its full potential in the single-OD case. This result is expected, and it highlights the importance of a systematic approach to optimally choosing both a set of controlled O-D pairs and a set of ICs.

Finally, we use STU(saving-to-unfairness) and RSTG (relative saving-to-guided flow) indexes (measures 4 and 5 in Table 4) to evaluate the control efficiency of various multi-OD MHCS instances against that of the SO solution. Fig. 4 plots these measures in each MHCS instance and SO. The latter is portrayed as a dashed straight line in both subplots. As seen from the plots, all MHCS instance are more efficient than SO based on both STU and RSTG. Although test 6 achieves the greatest time saving, it is not the most efficient case. The highest STU and RSTG are obtained by test 1 that controls only one O-D pair through one IC. The second highest STU is achieved by test 4. As for RSTG, tests 3 and 4 are the next favorable choices. Overall, tests 3 and 4 both seem to strike a good balance between travel time saving, unfairness and control efficiency.

5.3. MHCS experiments with all O-D pairs

In this section we first present a procedure to guide the choice of ICs in a general network, and then experiment with the full version of MHCS that controls all O-D pairs. Algorithm 4 explains how the nodes in a network are ranked based on their “desirability” for being

Table 12
Results of full-scale MHCS experiments with one intermediate checkpoint.

Test name	IC	IS selection logic/Traffic pattern	RTTS (%)	TU (min)	Guided vehicles (%)	Guided O-D pairs (%)	STU	RSTG
IC-3	3	1st rank of the choice rules	56.20	80867.50	6.11	10.61	1.99	9.21
IC-12	12	2nd rank of the choice rules	31.50	23035.83	3.04	5.30	3.91	10.36
IC-18	18	3rd rank of the choice rules	30.39	47636.58	4.44	6.81	1.82	6.85
IC-10	10	Most congested	27.89	42286.77	4.60	7.77	1.89	6.07
SO	All	SO traffic pattern	100	195045.73	14.11	30.30	1.47	7.09

Table 13
Results of full-scale MHCS experiments with three intermediate checkpoints.

IC	Selection logic/Traffic pattern	RTTS (%)	TU (min)	Guided vehicles (%)	Guided O-D pairs (%)	STU	RSTG
3, 12, 18	Choice rules (efficient set)	77.14	105716.38	10.22	18.56	2.09	7.55
10, 15, 18	Most cong. (congested set)	53.13	75260.13	13.22	17.42	2.02	4.02
All	SO traffic pattern	100	195045.73	14.11	30.30	1.47	7.09

used as intermediate checkpoints.

Algorithm 4. Choose desired set of ICs.

1: **Inputs:** Inputs of Algorithm 3, the number of desired ICs in the network x.
2: **Outputs:** The desired set of ICs.
3: Call Algorithm 3 to obtain $M_{h,l}$, the value of each rule l for each measure h.
4: Normalize $M_{h,l}$ between 1 and 100 to get scaled value $\overline{M}_{h,l}$.
5: Assign a positive weight B_n to each measure h such that $\sum_h B_h = 1$.
6: For each node $i \in N$ and rule l, find $E_{i,l}$, the rule-specific value of node i under rule l.
7: Normalize $E_{i,l}$ between 1 and 100 to get a scaled value $\overline{E}_{i,l}$
8: Compute the desirability score of node i as $O_i = \sum_l (\overline{E}_{i,l} \sum_h B_h \overline{M}_{h,l})$
9: Choose the first x nodes following the descending order of O_i.

Algorithm 4 first obtains $M_{h,l}$ by calling Algorithm 3 (see Tables 7–9 for the results for our example). Because different measures have different scales, they are first normalized into the same range (see line 4 of Algorithm 4). The normalized value $\overline{M}_{h,l} = \left(M_{h,l} - \min_j M_{h,j}\right) \Big/ \left(\max_j M_{h,j} - \min_j M_{h,j}\right) \times 100$, when a measure is considered "better" if its value is higher (e.g., DTTS and TTTS measures) and $\overline{M}_{h,l} = \left(\max_j M_{h,j} - M_{h,l}\right) \Big/ \left(\max_j M_{h,j} - \min_j M_{h,j}\right) \times 100$ otherwise (like RTU measure). On line 5, the algorithm assigns a pre-defined weight to each measure. For example, if the primary goal is to manage traffic congestion, then travel time saving measures (e.g., DTTS and TTTS) are of primary concern and would receive a larger weight.

The next step requires elaboration. On line 6, the rule-specific value of each node under each rule is determined. Take Rule 3 (the most used node on the UE routes) as an example. The rule-specific value of a node is obtained as follows. We first identify all O-D pairs for which the node is the most used node. Then, adding the number of routes passing through that node for each of those O-D pairs yields the value we need. Note that this raw value too should be normalized using the method described above (line 7).

In this section, we consider two sets of ICs. The first set, called the *efficient set*, is obtained using Algorithm 4 by setting the weights on line 5 as $B_{DTTS} = B_{TTTS} = 0.5$. The top three nodes identified this way are nodes 3, 18 and 12, which then forms the efficient set. We limit the number of nodes included in the set to three because a larger number leads to excessively long computation time. In the second set, called the *congested set*, we simply choose the first three most congested nodes (either at SO or UE) as the ICs. This set consists of nodes 10, 15 and 18.

In the first group of experiments, all O-D traffic is routed based on the hopping ratio, through one IC. In total, four single-IC cases are examined, including all three nodes from the efficient set and one from the congested set (10). Table 12 shows using intermediate nodes from the efficient set outperforms that from the congested set in all measures. The results of MHCS with IC-3 (with node 3 as the IC) is especially noteworthy. The relative time saving of MHCS is over 55% in this case, more than twice as much as that obtained by routing traffic through the most congested node (IC-10). Importantly, this significant efficiency improvement is achieved by merely telling 6% of all drivers from 56 O-D pairs (about 11% of all O-D pairs) to use node 3 as an IC in their way to destination. Although the total unfairness in IC-3 also doubles that in IC-10, only 2% of the traffic (i.e., about a third of all guided traffic) actually experience an unfair treatment (i.e., their travel time is higher than the others with the same O-D).

The results of other efficiency measures are more nuanced. IC-12 is the fairest scheme (with TU = 23036): it achieves 31% of the potential time savings, while only 3% of the total traffic between 28 O-D pairs (about 5% of all O-D pairs) is routed through node 12. As a result, IC-12 has the best STU and RSTG measures of all schemes, significantly outperforming the SO solution. IC-3 is the close second, but IC-10 (the most congested node) falls far behind.

Our second and final batch of experiments in this section involve the MHCS instances that control all O-D pairs and consider three

ICs. Table 13 reports results of two tests, using the efficient set and the congested set as the ICs respectively.

Similarly, the efficient set performs much better than the congested set. It achieves about 78% of the potential total travel time saving by guiding only 10% of all traffic between 98 O-D pairs (19% of all O-D pairs) through the three nodes. In this case, the average number of ICs visited by the guided traffic is 1.21. This means that the majority of the guided traffic visit only one IC before arriving at their intended destination. While the total unfairness measure is higher in the efficient set than the congested set, only about 4% of all traffic (40% of the guided traffic) actually suffers an unfair treatment.

Interestingly, the performance of the congested set is markedly worse than that of the single IC-3 case (see Table 12): it has a smaller relative time savings and controls more than twice as much of traffic. Compared to the SO solution, MHCS with either the efficient or the congested set produces a better STU measure. However, in terms of RSTG measure, the efficient set is the best and the congested set is the worst (even worse than SO). Clearly, the performance of MHCS depends heavily on ICs, and a poor choice could even do harm than good.

To summarize, the above experiments indicate that (i) MHCS is a promising approach to managing network traffic, as it is capable of reducing congestion at a relatively low control cost, and (ii) MHCS based on the efficient set identified by ranking the heuristic single-node choice rules consistently outperforms the SO solution in terms of the trade-off between travel time savings, control efficiency and unfairness. Choosing the most congested nodes as ICs seems a good strategy as a more congested node could influence more traffic. However, this intuition does not bear out in our experiments. We caution, however, the relative performance of the efficient vs. congested ICs may depend on network topology and hence could vary from problem to problem.

6. Conclusions

We propose and test a multi-hop control scheme (MHCS) that aims to route traffic through a set of designated ICs. Because travelers are allowed to freely choose routes for each “hop” that connects real (origin and destination) and ICs, MHCS promises to keep intervention at a more tolerable level, compared to conventional route-based control schemes. This problem has a natural bi-level structure: the upper level attempts to minimize congestion by adjusting the hopping ratios, which are then used in the lower level problem to route travelers according to user equilibrium conditions. Accordingly, we formulated the MHCS problem as a mathematical program with equilibrium constraints (MPEC), established its solution existence, and solved it using a sensitivity analysis based algorithm.

While no attempt is made here to optimize the choice of the ICs, a significant portion of the numerical experiments are devoted to this issue. Specifically, we examine sixteen heuristic rules for choosing ICs. Results based on single-OD experiments suggest that the most used and most congested nodes on SO routes deliver the largest travel time savings. In the best case, controlling merely one O-D pair (out of 528) and routing its traffic through one IC could achieve about 13.5% of the maximum possible travel time savings. Based on these findings, we identify a set of efficient ICs, which are then adopted to test the potential of a full-scale MHCS in which all O-D pairs are allowed to be controlled. The results from other numerical experiments are summarized below.

1. The efficient ICs are highly effective in reducing traffic congestion at a reasonable cost of control and unfairness. Routing about 7% of all traffic from all O-D pairs through the most efficient node could achieve more than 50% of travel time savings.
2. When the top three efficient nodes are included, the relative time savings reaches 77% by controlling 11% of all traffic.
3. The efficient ICs outperform, by a large margin, other choices such that most congested nodes.

While this study provides a general framework for studying the multi-hop control scheme, it leaves a few important issues to be addressed in future research. First and foremost, the choice of ICs should be optimized jointly with the hopping ratios in the upper-level problem. It then leads to a combinatorial optimization problem that is notoriously hard to solve. Hence, one possible future direction is to develop *meta*-heuristic solution methods for the combined problem. Second, the current solution algorithm for MHCS may not be appropriate for larger network. The number of hopping ratios grows nonlinearly with the size of O-D matrix and the number of ICs. Therefore, more scalable solution algorithms are needed if we are to consider real-world application. Thirdly, the current version of MHCS completely excludes the consideration of unfairness. A future study could incorporate it either as part of the objective function or as a constraint. Last but not least, our study ignores the potential impact of the proposed control measure on other travel choices (e.g., mode, departure time) or activity participation patterns. Indeed, a guided traveler may switch to another mode if they experience a much longer travel time. Moreover, multi-hop control may also be integrated with users’ daily activities like carpooling, refueling, shopping, etc. Incorporating such interactions into the modeling framework will pose new theoretical and computational challenges that require further research.

Author Statement

The authors confirm contribution to the paper as follows: study conception and design: H.R. Farahani, A.A. Rassafi and Y. Nie. Model specification, algorithm development and data collection: H.R. Farahani, A.A. Rassafi and Y. Nie. Analysis and interpretation of results: H.R. Farahani, K. Zhang, Y. Nie. Draft manuscript preparation: H.R. Farahani, Y. Nie, K. Zhang, and A. A. Rassafi.

Appendix A

See Table A1.

Table A1
List of notations.

Notation	Description
Sets	
N	set of network nodes
A	set of network links
P	set of origin nodes ($P \subseteq N$)
Q	set of destination nodes ($Q \subseteq N$)
W	set of O-D pairs ($W \subseteq P \times Q$)
IC	set of intermediate checkpoints ($IC \subseteq N$)
W'*	set of virtual O-D pairs ($W' \subseteq \{P \cup IC\} \times \{Q \cup IC\}$)
R	set of all routes in the network ($R = (R_w, w \in W)$)
R_{pq} or R_w	set of routes between O-D pair (p,q) or w
Ω_f	set of all feasible route flows ($\Omega_f = \{\mathbf{f} \mid \mathbf{d} = \Gamma\mathbf{f}, \mathbf{f} \geq 0\}$)
Ω_v	set of all feasible link flows ($\Omega_v = \{\mathbf{v} \mid \mathbf{v} = \Delta\mathbf{f}, \mathbf{d} = \Gamma\mathbf{f}, \mathbf{f} \geq 0\}$)
Ω_λ	set of all feasible hopping ratios
M	set of used performance measures to evaluate the results of single-OD tests ($M = \{M_n \mid n = 1, 2, \ldots, h\}$)
W_s	a sample of O-D pairs for the single-OD tests ($W_s \subseteq W$)
L	set of all heuristic rules for choosing the ICs (see Table 6)
L_w	set of applicable rules for the O-D pair w
I_l^w	set of ICs corresponding to rule $l \in L_w$
Parameters	
a	a link ($a \in A$)
(p,q) or w	an O-D pair ($p \in P, q \in Q, w$ or $(p,q) \in W$)
d_{pq} or d_w	original demand between the O-D pair (p,q) or w
$\mathbf{d}$	vector of all original O-D demands ($\mathbf{d} = (d_w, w \in W)^T$)
d'_{ij} or $d'_{w'}$	virtual demand between the O-D pair (i,j) or w'
$\mathbf{d'}$	vector of all virtual O-D demands ($\mathbf{d'} = (d'_{w'}, w' \in W')^T$)
r_{pq} or r_w	route between the O-D pair (p,q) or w (r_{pq} or $r_w \in R_{pq}$ or R_w)
f_{rpq} or f_{rw}	flow on route r between O-D pair (p,q) or w
$\mathbf{f}$	vector of all route flows ($\mathbf{f} = (f_{rw}, r \in R_w, w \in W)^T$)
v_a	flow on link a
$\mathbf{v}$	vector of all link flows ($\mathbf{v} = (v_a, a \in A)^T$)
v'_a	flow on link a resulted from assigning virtual demands $\mathbf{d'}$
$\mathbf{v}'$	vector of all link flows resulted from assigning virtual demands ($\mathbf{v'} = (v'_a, a \in A)^T$)
t_a	travel time on link a
$\mathbf{t}$	vector of all links' travel time
c_{rw}	cost on route r connecting O-D pair w
$\mathbf{c}$	vector of all route costs
μ_w	minimum path cost for O-D pair w ($\mu_w = min\{c_{rw}, r \in R_w\}, w \in W$)
s_a	marginal link cost function on link a ($s_a = t_a(v_a) + v_a \frac{\mathrm{d}t_a(v_a)}{\mathrm{d}v_a}, a \in A$)
$\mathbf{s}$	vector of all links' marginal cost functions ($\mathbf{s} = (s_a(v_a), a \in A)^T$)
u_{rw}	marginal cost on route r connecting O-D pair w
$\mathbf{u}$	vector of all marginal route costs
Γ	O-D pair-route incidence matrix ($\in \{0,1\}^{\|W\| \times \|R\|}$), i.e., the element γ_{wr} is 1 if route r connects O-D pair w and zero otherwise; so $\mathbf{d'} = \Gamma'\mathbf{f'}$
Δ	link-route incidence matrix ($\in \{0,1\}^{\|A\| \times \|R\|}$), i.e., the element δ_{ar} equals 1 if the path r uses link a and 0 otherwise; so $\mathbf{v'} = \Delta'\mathbf{f'}$ and $\mathbf{c'} = \Delta' T \mathbf{t'}$
J'	Jacobian of link flows with respect to the virtual demands ($J' \in \mathbb{R}^{\|A\| \times \|W'\|}$)
k	the iteration number in Algorithm 1
Ψ	a mapping from the set of feasible hopping ratios onto the set of virtual link flows $\Psi : \Omega_\lambda \to \Omega'_{v'}$ such that $\mathbf{v'} = \Psi(\lambda)$
M_n or M_h	a performance measure used to evaluate the results of single-OD tests
h	the number of measures used to evaluate the results of single-OD tests
l	a heuristic rule for choosing the ICs ($l \in L$ or $l \in L_w$)
$M^w_{h,l,i}$	the value of measure h corresponding to testing ith IC in I_l^w (see Algorithm 3)
$M^w_{h,l}$	the average value of measure h over all the ICs corresponding to rule l and O-D pair w (i.e., $M^w_{h,l} = \sum_i M^w_{h,l,i} / \|I_l^w\|$)
$M_{h,l}$	the average value of measure h over all O-D pairs corresponding to testing rule l (i.e., $M_{h,l} = \sum_w M^w_{h,l} / N_l$)
N_l	the number of O-D pairs in W_s for which the lth rule is applicable
P_{exp}	the origin node of the experimented O-D pair
Q_{exp}	the destination node of the experimented O-D pair

(continued on next page)

Table A1 (*continued*)

Notation	Description
IC_{exp}	the selected intermediate node to be tested as IC
x	number of desired ICs in the network
$\overline{M}_{h,l}$	the normalized value of $M_{h,l}$
B_n	a positive weight corresponding to measure n
$E_{i,l}$	the rule-specific value of node i under rule l
$\overline{E}_{i,l}$	the normalized value of $E_{i,l}$
O_i	the desirability score of node i ($O_i = \sum_l (\overline{E}_{i,l} \sum_h B_h \overline{M}_{h,l})$)
Decision variables	
$\lambda_{w'w}$ or $\lambda_{ij,pq}$	hopping ratio as the proportion of original demand between O-D pair w or (p,q), which is routed among virtual O-D pair w' or (i,j)
λ	matrix of all hopping ratios, i.e., $\lambda = \{\lambda_{w'w} \mid w' \in W', w \in W\}$; thus we have $\mathbf{d'} = \lambda \mathbf{d}$

*In general, we denote the variables under virtual assignment by adding a prime symbol (′) to the original notations.

References

Angelelli, E., Arsik, I., Morandi, V., Savelsbergh, M., Speranza, M., 2016. Proactive route guidance to avoid congestion. Transport. Res. Part B: Methodol. 94, 1–21.

Bagloee, S.A., Sarvi, M., Patriksson, M., Rajabifard, A., 2017. A mixed user-equilibrium and system-optimal traffic flow for connected vehicles stated as a complementarity problem. Comput.-Aided Civ. Infrastruct. Eng. 32, 562–580.

Bagloee, S.A., Tavana, M., Asadi, M., Oliver, T., 2016. Autonomous vehicles: challenges, opportunities, and future implications for transportation policies. J. Modern Transport. 24, 284–303.

Bahrami, S., Roorda, M.J., 2020. Optimal traffic management policies for mixed human and automated traffic flows. Transport. Res. Part A: Policy Practice 135, 130–143.

Beckmann, M.J., McGuire, C.B., Winsten, C.B., 1956. Studies in the Economics of Transportation. Yale University Press, New Haven, CT.

Bell, M.G.H., Iida, Y., 1997. Transportation network analysis. John Wiley & Sons, Chichester, UK.

Braess, D. 1968. Über ein Paradoxon aus der Verkehrsplanung. Unternehmensforschung, 12, 258–268.

Chen, Z., He, F., Yin, Y., Du, Y., 2017. Optimal design of autonomous vehicle zones in transportation networks. Transport. Res. Part B: Methodol. 99, 44–61.

Chen, Z., He, F., Zhang, L., Yin, Y., 2016. Optimal deployment of autonomous vehicle lanes with endogenous market penetration. Transport. Res. Part C: Emerg. Technol. 72, 143–156.

Chen, Z., Lin, X., Yin, Y., Li, M., 2020. Path controlling of automated vehicles for system optimum on transportation networks with heterogeneous traffic stream. Transport. Res. Part C: Emerg. Technol. 110, 312–329.

Dafermos, S., 1980. Traffic equilibrium and variational inequalities. Transport. Sci. 14, 42–54.

Dempe, S., Zemkoho, A.B., 2012. Bilevel road pricing: theoretical analysis and optimality conditions. Ann. Oper. Res. 196, 223–240.

Di, X., Ma, R., Liu, H.X., Ban, X., 2018. A link-node reformulation of ridesharing user equilibrium with network design. Transport. Res. Part B: Methodol. 112, 230–255.

Du, L., Han, L., Chen, S., 2015. Coordinated online in-vehicle routing balancing user optimality and system optimality through information perturbation. Transport. Res. Part B: Methodol. 79, 121–133.

Faez, K., Khanjary, M., 2008. UTOSPF: A distributed dynamic route guidance system based on Wireless Sensor Networks and Open Shortest Path First protocol. In: 2008 IEEE International Symposium on Wireless Communication Systems, 21-24 Oct. 2008. pp. 558–562.

Farahani, H.R., Rassafi, A.A., Mirbaha, B., 2019. Forced-node route guidance system: incorporating both user equilibrium and system optimal benefits. IET Intel. Transport Syst. 13, 1851–1859.

Friesz, T.L., Tobin, R.L., Cho, H.-J., Mehta, N.J., 1990. Sensitivity analysis based heuristic algorithms for mathematical programs with variational inequality constraints. Math. Program. 48, 265–284.

Gu, Z., Waller, S.T., Saberi, M., 2019. Surrogate-based toll optimization in a large-scale heterogeneously congested network. Comput.-Aided Civ. Infrastruct. Eng. 34, 638–653.

Harker, P.T., Pang, J.-S., 1988. Existence of optimal solutions to mathematical programs with equilibrium constraints. Oper. Res. Lett. 7, 61–64.

Haurie, A., Marcotte, P., 1985. On the relationship between Nash—Cournot and Wardrop equilibria. Networks 15, 295–308.

Jahn, O., Möhring, R.H., Schulz, A.S., Stier-Moses, N.E., 2005. System-optimal routing of traffic flows with user constraints in networks with congestion. Oper. Res. 53, 600–616.

Jayakrishnan, R., Tsai, W.K., Prashker, J.N., Rajadyaksha, S., 1994. Faster Path-based Algorithm for Traffic Assignment. Transp. Res. Rec.

Josefsson, M., Patriksson, M., 2007. Sensitivity analysis of separable traffic equilibrium equilibria with application to bilevel optimization in network design. Transport. Res. Part B: Methodol. 41, 4–31.

Lawphongpanich, S., Yin, Y., 2010. Solving the Pareto-improving toll problem via manifold suboptimization. Transport. Res. Part C: Emerg. Technol. 18, 234–246.

Li, R., Liu, X., Nie, Y., 2018. Managing partially automated network traffic flow: Efficiency vs. stability. Transport. Res. Part B: Methodol. 114, 300–324.

Liu, Z., Song, Z., 2019. Strategic planning of dedicated autonomous vehicle lanes and autonomous vehicle/toll lanes in transportation networks. Transport. Res. Part C: Emerg. Technol. 106, 381–403.

López-Ramos, F., Nasini, S., Guarnaschelli, A., 2019. Road network pricing and design for ordinary and hazmat vehicles: Integrated model and specialized local search. Comput. Oper. Res. 109, 170–187.

Lou, Y., Yin, Y., Lawphongpanich, S., 2010. Robust congestion pricing under boundedly rational user equilibrium. Transport. Res. Part B: Methodol. 44, 15–28.

Lu, S., 2008. Sensitivity of static traffic user equilibria with perturbations in arc cost function and travel demand. Transport. Sci. 42, 105–123.

Lu, S., Nie, Y.M., 2010. Stability of user-equilibrium route flow solutions for the traffic assignment problem. Transport. Res. Part B: Methodol 44, 609–617.

Mahmassani, H.S., Chang, G.-L., 1987. On Boundedly Rational User Equilibrium in Transportation Systems. Transport. Sci. 21, 89–99.

Patriksson, M., Rockafellar, R.T., 2002. A mathematical model and descent algorithm for bilevel traffic management. Transport. Sci. 36, 271–291.

Qiu, Y., Magnanti, T.L., 1989. Sensitivity analysis for variational inequalities defined on polyhedral sets. Mathe. Oper. Res. 14, 410–432.

Roughgarden, T., 2002. How unfair is optimal routing? In: Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms, 2002 San Francisco, California. 545406: Society for Industrial and Applied Mathematics, pp. 203–204.

Schrank, D., Eisele, B., Lomax, T., Bak, J. 2015. 2015 urban mobility scorecard.

Sheffi, Y., 1985. Urban transportation networks. U.S., Prentice-Hall, Englewood Cliffs, NJ.

Smith, M.J., 1979. The existence, uniqueness and stability of traffic equilibria. Transport. Res. Part B: Methodol. 13, 295–304.

Steck, F., Kolarova, V., Bahamonde-Birke, F., Trommer, S., Lenz, B., 2018. How Autonomous Driving May Affect the Value of Travel Time Savings for Commuting. Transport. Res. Re. 036119811875798.

Tobin, R.L., Friesz, T.L., 1988. Sensitivity analysis for equilibrium network flow. Transport. Sci. 22, 242–250.

Van Vuren, T., Van Vliet, D., Smith, M.J., 1990. Combined equalibrium in a network with partial route guidance. Traffic control methods. In: Proceedings of the Fifth NG Foundation Conference, Santa Barbara, Californi.

Wang, W., Wang, D.Z.W., Sun, H., Feng, Z., Wu, J., 2016. Braess Paradox of traffic networks with mixed equilibrium behaviors. Transport. Res. Part E: Logistics Transport. Rev. 93, 95–114.

Wardrop, J.G. 1952. Some Theoretical Aspects of road traffic research. Institution of Civil Engineers, Part 2.

Xie, J., Nie, Y., 2019. A New Algorithm for Achieving Proportionality in User Equilibrium Traffic Assignment. Transport. Sci. 53, 566–584.

Yang, H., Bell, M.G., 1997. Traffic restraint, road pricing and network equilibrium. Transport. Res. Part B: Methodol. 31, 303–314.

Yang, H., Bell, M.G., 1998. Models and algorithms for road network design: a review and some new developments. Transport Rev. 18, 257–278.

Yang, H., Bell, M.G., 2007. Sensitivity analysis of network traffic equilibrium revisited: the corrected approach. 4th IMA International Conference on Mathematics in TransportInstitute of Mathematics and its Applications.

Yang, H., Yagar, S., 1995. Traffic assignment and signal control in saturated road networks. Transport. Res. Part A: Policy Practice 29, 125–139.

Yang, H., Yagar, S., Iida, Y., Asakura, Y., 1994. An algorithm for the inflow control problem on urban freeway networks with user-optimal flows. Transport. Res. Part B: Methodol. 28, 123–139.

Yang, H., Zhang, X., Meng, Q., 2007. Stackelberg games and multiple equilibrium behaviors on networks. Transport. Res. Part B: Methodol. 41, 841–861.

Yang, X., Ban, X.J., Ma, R., 2017. Mixed Equilibria with Common Constraints on Transportation Networks. Networks Spatial Econ. 17, 547–579.

Zhang, K., Nie, Y.M., 2018. Mitigating the impact of selfish routing: An optimal-ratio control scheme (ORCS) inspired by autonomous driving. Transport. Res. Part C: Emerg. Technol. 87, 75–90.

Zhao, T., Zhang, Y., 2020. Transportation infrastructure restoration optimization considering mobility and accessibility in resilience measures. Transport. Res. Part C: Emerg. Technol. 117, 102700.