Networks and Spatial Economics (2019) 19:869–902
https://doi.org/10.1007/s11067-018-9433-y

Computing Dynamic User Equilibria on Large-Scale Networks with Software Implementation

Ke Han[1] · **Gabriel Eve**[1] · **Terry L. Friesz**[2]

Published online: 11 January 2019
© The Author(s) 2019

Abstract

Dynamic user equilibrium (DUE) is the most widely studied form of dynamic traffic assignment (DTA), in which road travelers engage in a non-cooperative Nash-like game with departure time and route choices. DUE models describe and predict the time-varying traffic flows on a network consistent with traffic flow theory and travel behavior. This paper documents theoretical and numerical advances in synthesizing traffic flow theory and DUE modeling, by presenting a holistic computational theory of DUE, which is numerically implemented in a MATLAB package. In particular, the dynamic network loading (DNL) sub-problem is formulated as a system of differential algebraic equations based on the Lighthill-Whitham-Richards fluid dynamic model, which captures the formation, propagation and dissipation of physical queues as well as vehicle spillback on networks. Then, the fixed-point algorithm is employed to solve the DUE problems with simultaneous route and departure time choices on several large-scale networks. We make openly available the MATLAB package, which can be used to solve DUE problems on user-defined networks, aiming to not only facilitate benchmarking a wide range of DUE algorithms and solutions, but also offer researchers a platform to further develop their own models and applications. The MATLAB package and computational examples are available at https://github.com/DrKeHan/DTA.

Keywords Dynamic traffic assignment · Dynamic user equilibrium · Dynamic network loading · Traffic flow model · Fixed-point algorithm · Software

✉ Ke Han
k.han@imperial.ac.uk

Extended author information available on the last page of the article.

1 Introduction

This paper is concerned with a class of models known as *dynamic user equilibrium* (DUE). DUE problems have been studied within the broader context of *dynamic traffic assignment* (DTA), which is viewed as the modeling of time-varying flows on traffic networks consistent with established travel demand and traffic flow theory.

DTA models, from the early 1990s onward, have been greatly influenced by Wardrop's principles (Wardrop 1952).

- Wardrop's first principle, also known as the *user optimal* principle, views travelers as Nash agents competing on a network for road capacity. Specifically, the travelers selfishly seek to minimize their own travel costs by adjusting route choices. A user equilibrium is envisaged where the travel costs of all travelers in the same origin-destination (O-D) pair are equal, and no traveler can lower his/her cost by unilaterally switching to a different route.
- Wardrop's second principle, known as the *system optimal* principle, assumes that travelers behave cooperatively in making their travel decisions such that the total travel cost on the entire network is minimized. In this case, the travel costs experienced by travelers in the same O-D pair are not necessarily identical.

Since the seminal work of Merchant and Nemhauser (1978a, b) the DTA literature has been focusing on the dynamic extension of Wardrop's principles, which gives rise to the notions of *dynamic user equilibrium* (DUE) and *dynamic system optimal* (DSO) models. The DUE model stipulates that experienced travel cost, including travel time and early/late arrival penalties, is identical for those route and departure time choices selected by travelers between a given O-D pair. The DSO model seeks system-wide minimization of travel costs incurred by all the travelers subject to the constraints of fixed travel demand and network flow dynamics.

For a comprehensive review of DTA models, the reader is referred to Boyce et al. (2001); Peeta and Ziliaskopoulos (2001); Szeto and Lo (2005, 2006); Jeihani (2007) and Bliemer et al. (2017). A discussion of DTA from the perspective of intelligent transportation system can be found in Ran and Boyce (1996a). Chiu et al. (2011) present a primer on simulation-based DTA modeling. Garavello et al. (2016) focus on the traffic flow modeling aspect of DTA, namely the hydrodynamic models for vehicular traffic and their network extensions. Wang et al. (2018) review relevant DTA literature concerning environmental sustainability.

Dynamic user equilibrium (DUE), which is one type of DTA, remains a modern perspective on traffic network modeling that enjoys wide scholarly support. It is conventionally studied as a open-loop, non-atomic Nash-like game (Friesz et al. 1993). The notion of open loop refers to the assumption that the travelers' route choices do not change in response to dynamic network conditions after they leave the origin. The non-atomic nature refers to the prevailing technique of flow-based modeling, instead of treating the traffic as individual vehicles; this is in contrast to agent-based modeling (Balmer et al. 2004; Cetin et al. 2003; Shang et al. 2017). DUE captures two aspects of travel behavior quite well: departure time choice and route choice. Within the DUE model, travel cost for the same trip purpose is identical for all utilized path and departure time choices. The relevant notion of travel cost is a weighted sum of travel time and arrival penalty.

In the last two decades there have been many efforts to develop a theoretically sound formulation of DUE that is also a canonical form acceptable to scholars and practitioners alike. Analytical DUE models tend to be of two varieties:

(1) *route-choice* (RC) DUE (Bliemer and Bovy 2003; Chen and Hsueh 1998; Lo and Szeto 2002; Long et al. 2013; Ran and Boyce 1996b; Tong and Wong 2000; Varia and Dhingra 2004; Zhu and Marcotte 2000); and
(2) *simultaneous route-and-departure-time* (SRDT) choice DUE (Friesz et al. 1993, 2001, 2013, 2011; Han et al. 2013b, 2015a, b; Huang and Lam 2002; Nie and Zhang 2010; Szeto and Lo 2004; Ukkusuri et al. 2012; Wie 2002).

There are two essential components within the RC or SRDT notions of DUE: (1) the mathematical expression of the equilibrium condition; and (2) a network performance model, which is often referred to as *dynamic network loading* (DNL). There are multiple means of expressing the Nash-like notion of dynamic equilibrium, including:

1. variational inequalities (Friesz et al. 1993, 2013; Han et al. 2013b, 2015a, b; Smith and Wisten 1994, 1995);
2. nonlinear complementarity problems (Han et al. 2011; Pang et al. 2011; Ukkusuri et al. 2012; Wie et al. 2002);
3. differential variational inequalities (Friesz and Meimand 2014; Friesz and Mookherjee 2006; Han et al. 2015a);
4. differential complementarity systems (Ban et al. 2012);
5. fixed-point problems in Hilbert spaces (Friesz et al. 2011; Han et al. 2015b); and
6. stationary points of evolutionary dynamics (Mounce 2006; Smith and Wisten 1995).

On the other hand, the DNL sub-problem captures the relationship between dynamic traffic flows and travel delays, by articulating link dynamics, junction interactions, flow propagation, link delay, and path delay. It has been the focus of a significant number of studies in traffic modeling (Ran and Boyce 1996b; Xu et al. 1999; Tong and Wong 2000; Lo and Szeto 2002; Wie et al. 2002; Szeto and Lo 2004; Yperman et al. 2005; Perakis and Roels 2006; Nie and Zhang 2010; Ban et al. 2011; Ukkusuri et al. 2012; Han et al. 2013a). The DNL model aims at describing and predicting the spatial and temporal evolution of traffic flows on a network that is consistent with established route and departure time choices of travelers. This is done by introducing appropriate dynamics to flow propagation, flow conservation, link delay, and path delay on a network level. Any DNL must be consistent with the established path flows, and is usually performed under the *first-in-first-out* (FIFO) principle (Szeto and Lo 2006).

In general, DNL models have the following components:

1. some form of link and/or path dynamics;
2. an analytical relationship between flow/speed/density and link traversal time
3. flow propagation constraints;
4. a model of junction dynamics and delays;
5. a model of path traversal time; and
6. appropriate initial conditions.

DNL gives rise to the path delay operator, which is analogous to the pay-off function in classical Nash games, and plays a pivotal role in DTA and DUE problems. The properties of the delay operator are critical to the existence and computation of DUE models. However, it is widely recognized that the DNL model or the delay operator is not available in closed form; instead, they have to be numerically evaluated via a computational procedure. As a result, the mathematical properties of the delay operator remain largely unknown. This has significantly impacted the computation of DUE problems due to the lack of provable convergence theories, which all require certain form of generalized monotonicity of the delay operators. Table 1 shows some relevant computational algorithms for DUE and their convergence conditions with respect to the continuity and monotonicity of the delay operators. The reader is referred to Han et al. (2015a) for definitions of different types of generalized monotonicity.

This paper documents theoretical and numerical advances in synthesizing traffic flow theory and traffic assignment models, by presenting a computational theory of DUE, which includes algorithms and software implementation. While there have been numerous studies on the modeling and computation of DUEs, including those reviewed in this paper, little agreement exists regarding an appropriate mathematical formulation of DUE or DNL models, as well as the extent to which certain models should/can be applied. This is partially due to the lack of open-source solvers and a set of benchmarking test problems for DUE models. In addition, large-scale computational examples of DUE were rarely reported and, when they were, little detail was provided that allows the results to be *validated*, *reproduced* and *compared*.

This paper aims to bridge the aforementioned gaps by presenting a computable theory for the simultaneous route-and-departure-time (SRDT) choice DUE model along with open-source software packages. In particular, the DNL model is based on the Lighthill-Whitham-Richards fluid dynamic model (Lighthill and Whitham 1955; Richards 1956), and is formulated as a system of differential algebraic equations (DAEs) by invoking the variational theory for kinematic wave models. This technique allows the DAE system to be solved with straightforward time-stepping without the need to solve any partial differential equations. Moreover, this paper presents the fixed-point algorithm for solving the DUE problem, which was derived from the *differential variational inequality* formalism (Friesz and Han 2018). Both the DNL procedure and the fixed-point algorithm are implemented in MATLAB, and

Table 1 Computational algorithms for DUE. The algorithms are arranged in an increasing order in terms of the generality of the relevant notions of monotonicity

	DUE type	Computational	Convergence
Friesz et al. (2011)	SRDT DUE	Fixed-point algorithm	Lipschitz cont. strongly monotone
Lo and Szeto (2002)	RC DUE	Alternating direction algorithm	Co-coercive
Szeto and Lo (2004)	SRDT DUE elastic demand	descent algorithm (projection)	Co-coercive
Szeto and Lo (2006)	RC DUE bounded rationality	Route-swapping algorithm	Continuous monotone
Huang and Lam (2002)	SRDT DUE	Route-swapping algorithm	Continuous monotone
Tian et al. (2012)	SRDT DUE	Route-swapping algorithm	Continuous monotone
Long et al. (2013)	RC DUE	Extragradient/ double projection	Lipschitz cont. pseudo monotone
Han et al. (2015b)	SRDT DUE bounded rationality	Self-adaptive projection	Continuous D-property
Han et al. (2015a)	SRDT DUE elastic demand	Proximal point method	Dual solvable

we present the computational results on several test networks, including the Chicago Sketch network with 250,000 paths. To our knowledge, this is the largest instance of SRDT DUE computation reported in the literature to date.

In addition, we make openly available the MATLAB package, which can be used to solve DUEs on user-defined networks. The package is documented in this paper (Appendix), and the codes and examples are available at https://github.com/DrKeHan/DTA. It is our intention that the open-source package will not only help DTA modelers with benchmarking a wide range of algorithms and solutions, but also offer researchers a platform to further develop their own models and applications.

The rest of the paper is organized as follows. Section 2 introduces some key notions and mathematical formulations of DUE. Section 3 details the dynamic network loading procedure and the DAE system formulation. The fixed-point algorithm for computing DUE is presented in Section 4. The computational results obtained from the proposed DUE solver are presented in Section 5. Section 6 offers some concluding remarks. Finally, the MATLAB software package is documented in the Appendix.

2 Formulations of Dynamic User Equilibrium

We introduce a few notations and terminologies for the ease of presentation below.

$\mathcal{P}$	set of paths in the network
$\mathcal{W}$	set of O-D pairs in the network
Q_{ij}	fixed O-D demand between $(i,\ j) \in \mathcal{W}$
$\mathcal{P}_{ij}$	subset of paths that connect O-D pair $(i,\ j)$
t	continuous time parameter in a fixed time horizon $[t_0,\ t_f]$
$h_p(t)$	departure rate along path p at time t
$h(t)$	complete vector of departure rates $h(t) = (h_p(t) :\ p \in \mathcal{P})$
$\Psi_p(t,\ h)$	travel cost along path p with departure time t, under departure profile h
$v_{ij}(h)$	minimum travel cost between O-D pair $(i,\ j)$ for all paths and departure times

We stipulate that the path departure rates are square integrable:

$$h_p(\cdot) \in L^2_+[t_0,\ t_f], \qquad h(\cdot) \in \left(L^2_+[t_0,\ t_f]\right)^{|\mathcal{P}|}$$

We define the *effective delay operator* Ψ as follows:

$$\begin{aligned} \Psi :\ & \left(L^2_+[t_0,\ t_f]\right)^{|\mathcal{P}|} \ \rightarrow\ \left(L^2_+[t_0,\ t_f]\right)^{|\mathcal{P}|} \\ & h(\cdot) = \left\{h_p(\cdot),\ p \in \mathcal{P}\right\} \ \mapsto\ \Psi(h) = \left\{\Psi_p(\cdot,\ h),\ p \in \mathcal{P}\right\} \end{aligned} \tag{2.1}$$

Here, the term 'effective delay' (Friesz et al. 1993) is a generalized notion of travel cost that may include not only a linear combination of travel time and arrival penalty, but also other forms of costs such as road pricing. The effective delay operator Ψ is essential to the DUE model as it encapsulates the physics of the traffic network by capturing the dynamics of traffic flows at the link, junction, path, and network levels. We will discuss this operator in greater detail in Section 3.

The travel demand satisfaction constraint is expressed as

$$\sum_{p \in \mathcal{P}_{ij}} \int_{t_0}^{t_f} h_p(t)dt \ =\ Q_{ij} \qquad \forall (i,\ j) \in \mathcal{W} \tag{2.2}$$

Therefore, the set of feasible path departure vector can be expressed as

$$\Lambda = \left\{ h \geq 0 :\ \sum_{p \in \mathcal{P}_{ij}} \int_{t_0}^{t_f} h_p(t)dt = Q_{ij} \quad \forall (i,\ j) \in \mathcal{W} \right\} \subset \left({}^2[t_0,\ t_f]\right)^{|\mathcal{P}|} \tag{2.3}$$

The following definition of dynamic user equilibrium is first proposed by Friesz et al. (1993) in a measure-theoretic context.

Definition 2.1 *(SRDT DUE) A vector of departures $h^* \in \Lambda$ is a dynamic user equilibrium with simultaneous route and departure time (SRDT) choice if*

$$h_p^*(t) > 0,\ p \in \mathcal{P}_{ij} \implies \Psi_p(t,\, h^*) = v_{ij}(h^*) \quad a.e.\ t \in [t_0,\, t_f] \tag{2.4}$$

where 'a.e.', standing for 'for almost every', is a technical term employed by measure-theoretic arguments to indicate that Eq. (2.4) only needs to hold in $[t_0,\, t_f]$ with the exception of any subset that has zero Lebesgue measure.

2.1 Variational Inequality Formulation of DUE

Using measure-theoretic arguments, Friesz et al. (1993) establish that a SRDT DUE is equivalent to the following variational inequality under suitable regularity conditions:

$$\left.\begin{array}{c} \text{find } h^* \in \Lambda \text{ such that} \\ \displaystyle\sum_{p\in\mathcal{P}} \int_{t_0}^{t_f} \Psi_p(t,\, h^*) \left(h_p(t) - h_p^*(t)\right) dt \geq 0 \\ \forall h \in \Lambda \end{array}\right\} \tag{2.5}$$

The VI (2.5) may be written in a more generic form by invoking the inner product $\langle\cdot\rangle$ in the Hilbert space $\left(L^2[t_0,\, t_f]\right)^{|\mathcal{P}|}$:

$$\langle f,\, g\rangle \doteq \sum_{p\in\mathcal{P}} \int_{t_0}^{t_f} f_p(t) g_p(t) dt \qquad \forall f,\, g \in \left(L^2[t_0,\, t_f]\right)^{|\mathcal{P}|}$$

This leads to the following VI representation of DUE:

$$\left\langle \Psi(h^*),\, h - h^* \right\rangle \geq 0 \qquad \forall h \in \Lambda$$

2.2 Nonlinear Complementarity Formulation of Due

The variational inequality formulation (2.5) is equivalent to the following nonlinear complementarity problem (Pang et al. 2011; Ukkusuri et al. 2012): $\forall (i,\, j) \in \mathcal{W}$,

$$0 \leq h_p^*(t) \perp \Psi_p(t,\, h^*) - v_{ij}(h^*) \geq 0 \quad \forall p \in \mathcal{P}_{ij},\ \text{a.e. } t \in [t_0,\, t_f] \tag{2.6}$$

$$0 \leq v_{ij}(h^*) \perp \sum_{p\in\mathcal{P}_{ij}} \int_{t_0}^{t_f} h_p^*(t) dt - Q_{ij} \geq 0 \quad \forall (i,\, j) \in \mathcal{W} \tag{2.7}$$

where $a \perp b$ means that $a \cdot b = 0$. While the equivalence between the VI and complementarity formulations is obvious in a discrete-time setting, the continuous-time version requires a non-trivial measure-theoretic argument. This is yet to be rigorously proven.

2.3 Differential Variational Inequality Formulation of DUE

It is noted in (Friesz et al. 2011) that the VI formulation (2.5) of DUE is equivalent to a differential variational inequality (DVI). This is most easily seen by noting that the demand satisfaction constraints may be re-stated as

$$\left.\begin{array}{l} \frac{d}{dt} y_{ij}(t) = \sum_{p \in \mathcal{P}_{ij}} h_p(t) \qquad \forall (i,\, j) \in \mathcal{P}_{ij} \\ y_{ij}(t_0) = 0 \qquad \forall (i,\, j) \in \mathcal{P}_{ij} \\ y_{ij}(t_f) = Q_{ij} \qquad \forall (i,\, j) \in \mathcal{P}_{ij} \end{array}\right\}, \tag{2.8}$$

which is recognized as a two-point boundary value problem. The DUE may be consequently expressed as a DVI:

$$\left.\begin{array}{c} \text{find } h^* \in \Lambda_0 \text{ such that} \\ \sum_{p \in \mathcal{P}} \int_{t_0}^{t_f} \Psi_p(t,\, h^*) \left(h_p(t) - h_p^*(t)\right) dt \geq 0 \\ \forall h \in \Lambda_0 \end{array}\right\} \tag{2.9}$$

where

$$\Lambda_0 = \left\{ h \geq 0 : \frac{d}{dt} y_{ij}(t) = \sum_{p \in \mathcal{P}_{ij}} h_p(t),\; y_{ij}(t_0) = 0,\; y_{ij}(t_f) = Q_{ij} \;\; \forall (i,\, j) \in \mathcal{W} \right\} \tag{2.10}$$

The equivalence of DUE and DVI (2.9) can be shown with elementary optimal control theory applied to a linear-quadratic problem as demonstrated in Friesz et al. (2011) and Friesz and Han (2018). The DVI formulation is significant because it allows the still emerging theory of differential variational inequalities (Friesz 2010; Pang and Stewart 2008) to be employed for the analysis and computation of solutions of the DUE problem when simultaneous departure time and route choices are within the purview of users (Friesz and Meimand 2014; Friesz and Mookherjee 2006; Han et al. 2015a). However, the emerging literature on abstract differential variational inequalities has not been well exploited for either modeling or computing simultaneous route-and-departure-time choice equilibria; this gap was recently bridged by Friesz and Han (2018).

2.4 Fixed-point Formulation of DUE

Experience with variational inequalities suggests that there exists a fixed-point re-statement of the DUE problem in a proper functional space. The fixed-point formulation of continuous-time DUE is first articulated in Friesz et al. (2011) using optimal control theory. We define $P_{\Lambda_0}[\cdot]$ to be the minimum-norm projection operator in the space $\left(L^2[t_0,\, t_f]\right)^{|\mathcal{P}|}$. Then the following fixed-point problem is equivalent to the DUE problem:

$$h^* = P_\Lambda\left[h^* - \alpha\Psi(h^*)\right] \quad \text{or} \quad h^* = P_{\Lambda_0}\left[h^* - \alpha\Psi(h^*)\right] \tag{2.11}$$

where $\alpha > 0$ is a fixed constant. The equivalence result may be proven by applying the minimum principle and associated optimality conditions to the minimum-norm problem intrinsic to the projection operator $P_{\Lambda_0}[\cdot]$. See Friesz (2010) for the details suppressed here.

3 Dynamic Network Loading

An integral component of the DUE formulation is the effective delay operator Ψ, which is constructed using the *dynamic network loading* (DNL) procedure. This section details one type of DNL models based on the fluid dynamic approximation of traffic flow on networks, known as the Lighthill-Whitham-Richards (LWR) model (Lighthill and Whitham 1955; Richards 1956). The model, including its various forms (Newell 1993a; Daganzo 1994, 1995; Yperman et al. 2005; Osorio et al. 2011), is widely studied in the DTA literature. This section presents a complete DNL procedure based on the LWR model and its variational representation, which lead to a differential algebraic equation (DAE) system.

3.1 The Lighthill-Whitham-Richards Link Model

The LWR model is capable of describing the physics of kinematic waves (e.g. shock waves, rarefaction waves), and allows network extensions that capture the formation and propagation of vehicle queues as well as spillback. It describes the spatial and temporal evolution of vehicle density $\rho(t,\, x)$ on a road link using the following partial differential equation:

$$\partial_t \rho(t,\, x) + \partial_x f\left(\rho(t,\, x)\right) = 0 \qquad x \in [a,\, b],\ t \in [t_0,\, t_f], \tag{3.1}$$

where the link of interest is represented as a spatial interval $[a,\, b]$. The fundamental diagram $f(\cdot)$ is a continuous and concave function of density ρ, and satisfies $f(0) = f(\rho^{\text{jam}}) = 0$ where ρ^{jam} denotes the jam (maximum) density. Furthermore, there exists a unique critical density value ρ^{c} at which $f(\cdot)$ attains its maximum $f(\rho^{\mathrm{c}}) = C$ were C denotes the flow capacity of the link.

A few widely adopted forms of $f(\cdot)$ include the Greenshields (Greenshields 1935), the trapezoidal (Daganzo 1994, 1995), and the triangular (Newell 1993a, b, c) fundamental diagrams. In the remainder of the paper (and also in the MATLAB package), we consider the following triangular fundamental diagram:

$$f(\rho) = \begin{cases} v\rho & \rho \in [0,\ \rho^{\mathrm{c}}] \\ -w(\rho - \rho^{\mathrm{jam}}) & \rho \in (\rho^{\mathrm{c}},\ \rho^{\mathrm{jam}}] \end{cases} \tag{3.2}$$

where $v > 0$ and $w > 0$ denote the forward and backward kinematic wave speeds, respectively.

While (3.1) captures the within-link dynamics, the inter-link propagation of congestion requires a careful treatment of junction dynamics, which is underpinned by the notions of link demand and supply.

3.2 Link Demand and Supply

We consider a road junction with m incoming links and n outgoing links. The dynamic on each of the $m + n$ links is governed by the LWR model (3.1). Understandably these $m + n$ equations are coupled via their relevant boundary conditions. In particular, the following flow conservation constraint must hold:

$$\sum_{i=1}^{m} f_i\left(\rho_i(t,\ b_i)\right) = \sum_{j=1}^{n} f_j\left(\rho_j(t,\ a_j)\right) \qquad \forall t \in [t_0,\ t_f], \tag{3.3}$$

where, without causing any confusion, we always use subscript i or j to indicate the association with link i or j. Equation (3.3) simply means that the total flow through the junction is conserved. However, this condition alone does not guarantee a unique flow profile on these $m + n$ links, and additional conditions need to be imposed (Garavello et al. 2016). To this end, we define the link demand and supply (Lebacque and Khoshyaran 1999), where the demand (supply) is viewed as a function of the density near the exit (entrance) of the link:

$$D(\rho(t,\ b-)) = \begin{cases} f(\rho(t,\ b-)) & \text{if } \rho(t,\ b-) < \rho^c \\ C & \text{if } \rho(t,\ b-) \geq \rho^c \end{cases} \tag{3.4}$$

$$S(\rho(t,\ a+)) = \begin{cases} C & \text{if } \rho(t,\ a+) < \rho^c \\ f(\rho(t,\ a+)) & \text{if } \rho(t,\ a+) \geq \rho^c \end{cases} \tag{3.5}$$

Intuitively, the demand (supply) indicates the maximum flow that can exit (enter) the link. That is,

$$f_i(\rho_i(t,\ b_i)) \leq D_i(\rho_i(t,\ b_i-)), \quad f_j(\rho_j(t,\ a_j)) \leq S_j(\rho_j(t,\ a_j+)) \tag{3.6}$$

for $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$. Similar to Eq. (3.3), Eq. (3.6) ensures the physical validity of the flows through the junction. Nevertheless, additional conditions are still needed to isolate a unique flow profile at this junction; these conditions are often derived based on driving behavior or traffic management measures, such as flow distribution (Daganzo 1995), right of way (Daganzo 1995; Coclite et al. 2005; Jin 2010), and traffic signal control (Han et al. 2014; Han and Gayah 2015). A review of these different models is provided in Garavello et al. (2016).

3.3 The Variational Representation of Link Dynamics

The variational solution representation of Hamilton-Jacobi equations has been widely applied to the LWR-type traffic modeling (Newell 1993a, b, c; Daganzo 2005, 2006; Claudel and Bayen 2010; Laval and Leclercq 2013; Costeseque and Lebacque 2014; Han et al. 2017). Our paper follows this approach. In particular, we consider the Moskowitz function (Moskowitz 1965), $N(t, x)$, which measures the cumulative number of vehicles that have passed location x along a link by time t. The following identities hold:

$$\partial_t N(t, x) = f(\rho(t, x)), \quad \partial_x N(t, x) = -\rho(t, x) \qquad \text{a.e. } x \in [a, b]$$

where 'almost everywhere' indicates possible spatial discontinuities in the density profile, i.e. shockwaves. It is straightforward to show that $N(t, x)$ satisfies the following Hamilton-Jacobi equation:

$$\partial_t N(t, x) - f(-\partial_x N(t, x)) = 0 \qquad x \in [a, b],\ t \in [t_0, t_f]. \tag{3.7}$$

Next, we denote by $f^{\text{in}}(t)$ and $f^{\text{out}}(t)$ the link inflow and outflow, respectively. The cumulative link entering and exiting vehicle counts are defined as

$$N^{\text{up}}(t) = \int_{t_0}^{t} f^{\text{in}}(s)ds, \qquad N^{\text{dn}}(t) = \int_{t_0}^{t} f^{\text{out}}(s)ds,$$

where the superscripts 'up' and 'dn' represent the *upstream* and *downstream* boundaries of the link, respectively. Han et al. (2016b) derive explicit formulae for the link demand and supply based on a variational formulation known as the Lax-Hopf formula (Aubin et al. 2008; Claudel and Bayen 2010), as follows:

$$D(t) = \begin{cases} f^{\text{in}}\left(t - \frac{L}{v}\right) & \text{if } N^{\text{up}}\left(t - \frac{L}{v}\right) = N^{\text{dn}}(t) \\ C & \text{if } N^{\text{up}}\left(t - \frac{L}{v}\right) > N^{\text{dn}}(t) \end{cases} \tag{3.8}$$

$$S(t) = \begin{cases} f^{\text{out}}\left(t - \frac{L}{w}\right) & \text{if } N^{\text{up}}(t) = N^{\text{dn}}\left(t - \frac{L}{w}\right) + \rho^{\text{jam}} L \\ C & \text{if } N^{\text{up}}(t) < N^{\text{dn}}\left(t - \frac{L}{w}\right) + \rho^{\text{jam}} L \end{cases} \tag{3.9}$$

where $L = b - a$ denotes the link length. Note that Eqs. (3.8) and (3.9) express the link demand and supply, which are inputs of the junction model, in terms of $N^{\text{up}}(\cdot)$, $N^{\text{dn}}(\cdot)$, $f^{\text{in}}(\cdot)$ and $f^{\text{out}}(\cdot)$. This means that one no longer needs to compute the dynamics within the link, but to focus instead on the flows or cumulative counts at the two boundaries of the link. This tremendously simplifies the link dynamics, and gives rise to the link-based formulation (Jin 2015; Han et al. 2016b) or, in its discrete form, the link transmission model (Yperman et al. 2005).

3.4 Junction Dynamics that Incorporate Route Information

Essential to the network extension of the LWR model are the junction dynamics. Unlike many existing junction models such as those reviewed in Section 3.2, in a path-based DNL formulation, one must incorporate established routing information into the junction model. Such information is manifested in an endogenous *flow distribution matrix*, which specifies the proportion of exit flow from a certain incoming link that advances into a given outgoing link. This can be done by explicitly tracking the route composition in every unit of flow along the link.

We begin by defining the link entry time function $\tau(t)$ where t denotes the corresponding exit time. Such a function can be obtained by evaluating the horizontal difference between the cumulative curves $N^{\text{up}}(\cdot)$ and $N^{\text{dn}}(\cdot)$ (Friesz et al. 2013). Next, for link i and path p such that $i \in p$, we define $\mu_i^p(t, x)$ to be the percentage of flow on link i that belongs to path p at any point (t, x). The first-in-first-out principle yields the following identity:

$$\mu_i^p(t, b_i) = \mu_i^p(\tau_i(t), a_i) \tag{3.10}$$

Now we consider a junction J with incoming links labeled as $i \in \{1, \ldots, m\}$ and outgoing links labeled as $j \in \{1, \ldots, n\}$. The distribution matrix $A^J(t)$ can be expressed as

$$A^J(t) = \{\alpha_{ij}(t)\}, \qquad \alpha_{ij}(t) = \sum_{p \ni i, j} \mu_i^p(t, b_i) = \sum_{p \ni i, j} \mu_i^p(\tau_i(t), a_i)$$

Here, the scalar $\alpha_{ij}(t)$ represents the proportion, among traffic exiting link i, that advances to link j. There exist a number of choices for the junction models, all of which need to satisfy the flow conservation constraint (3.3), the demand-supply constraints (3.6), and depend on the flow distribution matrix $A^J(t)$. Any such model can be conceptually expressed as:

$$\left(\left[f_i^{\text{out}}(t)\right]_{i=1,\ldots,m}, \left[f_j^{\text{in}}(t)\right]_{j=1,\ldots,n}\right) = \Theta\left(\left[D_i(t)\right]_{i=1,\ldots,m}, \left[S_j(t)\right]_{j=1,\ldots,n}; A^J(t)\right) \tag{3.11}$$

where $D_i(t)$, $S_j(t)$ and $A^J(t)$ are treated as input arguments of the junction model. The output, shown as the left hand side of (3.11), include the outflows (inflows) of the incoming (outgoing) links. The mapping Θ for simple merge or diverge junction is illustrated by Daganzo (1995). In the conservation law literature Θ is sometimes referred to as the *Riemann Solver*, and we refer the reader to Lebacque and Khoshyaran (1999), Coclite et al. (2005), Jin (2010), Han and Gayah (2015), Han et al. (2016b), and Garavello et al. (2016) for more discussion.

3.5 Dynamics at the Origin Nodes

A model at the origin (source) nodes is needed since the path flows $h_p(\cdot)$, defined by Eq. (2.3), are not bounded from above. In this case, a queuing model is needed at the origin node to accommodate departure flow exceeding the supply of relevant downstream link.

We employ a simple point-queue type dynamic for the origin node o. Denote by $q_o(t)$ the size of the point queue, and let link j be the link connected to the origin node. We have that

$$\frac{d}{dt}q_o(t) = \sum_{p\in\mathcal{P}^o} h_p(t) - \min\{D_o(t),\ S_j(t)\} \tag{3.12}$$

where $\mathcal{P}^o$ denotes the set of paths originating from o. The first term on the right hand side of Eq. (3.12) represents flow into the point queue, while the second term represents flow leaving the queue, where the demand at the origin is defined as

$$D_o(t) = \begin{cases} \mathcal{M} & q_o(t) > 0 \\ \sum_{p\in\mathcal{P}^o} h_p(t) & q_o(t) = 0 \end{cases},$$

and $\mathcal{M}$ is a sufficiently large number, e.g. larger than the flow capacity of link j. We note that the proposed queuing dynamics differ from the classical point-queue model (Vickrey 1969; Ban et al. 2011; Han et al. 2013a) in that our model assumes varying queue exit capacity, as constrained by the downstream supply $S_j(t)$, instead of a fixed flow capacity.

3.6 Calculating Path Travel Times

The DNL procedure calculates the path travel times for a given set of path departure rates. The path travel time consists of link travel times plus possible queuing time at the origin. We define the link exit time function $\lambda(t)$ by measuring the horizontal difference between the cumulative entering and exiting counts:

$$N^{\text{up}}(t) = N^{\text{dn}}(\lambda(t)) \tag{3.13}$$

We note that Eq. (3.13) is also applied to calculate the queue exit time function at the origin following the dynamics presented in Section 3.5. For a path expressed as $p = \{1, 2, \ldots, K\}$, the path exit time for a given departure time t is calculated as

$$\lambda_o \circ \lambda_1 \circ \lambda_2 \ldots \circ \lambda_K(t) \tag{3.14}$$

where $f \circ g(t) \doteq g(f(t))$ denotes the composition of two functions. $\lambda_o(\cdot)$ is the exit time function for the potential queuing at the origin o.

3.7 The Differential Algebraic Equation System Formulation of DNL

As a summary of the individual sections presented so far, we formulate the complete system of differential algebraic equations (DAEs). We begin with the following list of key notations.

$\mathcal{P}$	set of all paths
$\mathcal{S}$	set of origins
$\mathcal{P}^o$	set of paths originating from $o \in \mathcal{S}$
$\mathcal{I}^J$	set of incoming links of a junction J
$\mathcal{O}^J$	set of outgoing links of a junction J
A^J	flow distribution matrix of junction J
$h_p(t)$	departure rate along path $p \in \mathcal{P}$
$f_i^{\text{in}}(t)$	inflow of link i
$f_i^{\text{out}}(t)$	outflow of link i
$N_i^{\text{up}}(t)$	cumulative link entering count
$N_i^{\text{dn}}(t)$	cumulative link exiting count
$D_i(t)$	demand of link i
$S_i(t)$	supply of link i
$\mu_i^p(t, x)$	percentage of flow on link i that belongs to path p
$q_o(t)$	point queue at the origin node $o \in \mathcal{S}$
$\tau_i(t)$	entry time of link i corresponding to exit time t
$\lambda_i(t)$	exit time of link i corresponding to entry time t

The DAE system reads:

$$\frac{d}{dt}q_o(t) = \sum_{p\in\mathcal{P}^o} h_p(t) - \min\{D_o(t),\ S_j(t)\},$$

$$D_o(t) = \begin{cases} \mathcal{M} & q_o(t) > 0 \\ \sum_{p\in\mathcal{P}^o} h_p(t) & q_o(t) = 0 \end{cases} \tag{3.15}$$

$$D_i(t) = \begin{cases} f_i^{\text{in}}\left(t - \frac{L_i}{v_i}\right) & \text{if } N_i^{\text{up}}\left(t - \frac{L_i}{v_i}\right) = N_i^{\text{dn}}(t) \\ C_i & \text{if } N_i^{\text{up}}\left(t - \frac{L_i}{v_i}\right) > N_i^{\text{dn}}(t) \end{cases} \tag{3.16}$$

$$S_j(t) = \begin{cases} f_j^{\text{out}}\left(t - \frac{L_j}{w_j}\right) & \text{if } N_j^{\text{up}}(t) = N_j^{\text{dn}}\left(t - \frac{L_j}{w_j}\right) + \rho_j^{\text{jam}} L_j \\ C_j & \text{if } N_j^{\text{up}}(t) < N_j^{\text{dn}}\left(t - \frac{L_j}{w_j}\right) + \rho_j^{\text{jam}} L_j \end{cases} \tag{3.17}$$

$$N_i^{\text{dn}}(t) = N_i^{\text{up}}(\tau_i(t)), \qquad N_i^{\text{up}}(t) = N_i^{\text{dn}}(\lambda_i(t)) \tag{3.18}$$

$$\mu_j^p(t,\ a_j) = \frac{f_i^{\text{out}}(t)\mu_i^p(\tau_i(t),\ a_i)}{f_j^{\text{in}}(t)} \qquad \forall p \text{ s.t. } \{i,\ j\} \subset p \tag{3.19}$$

$$A^J(t) = \{\alpha_{ij}(t)\}, \quad \alpha_{ij}(t) = \sum_{p\ni i,j} \mu_i^p(\tau_i(t),\ a_i) \tag{3.20}$$

$$\begin{aligned} &\left(\left[f_i^{\text{out}}(t+)\right]_{i=1,\ldots,m},\ \left[f_j^{\text{in}}(t+)\right]_{j=1,\ldots,n}\right) \\ &= \Theta\left([D_i(t)]_{i=1,\ldots,m},\ [S_j(t)]_{j=1,\ldots,n};\ A^J(t)\right) \end{aligned} \tag{3.21}$$

$$\frac{d}{dt}N_i^{\text{up}}(t) = f_i^{\text{in}}(t), \qquad \frac{d}{dt}N_i^{\text{dn}}(t) = f_i^{\text{out}}(t) \tag{3.22}$$

$$D_p(t,\ h) = \lambda_o \circ \lambda_1 \circ \lambda_2 \ldots \circ \lambda_K(t) - t \qquad p = \{1,\ 2,\ \ldots,\ K\} \tag{3.23}$$

Equations (3.15)–(3.23) form the DAE system for the DNL procedure. Compared to the partial differential algebraic equation (PDAE) system presented in Han et al. (2016a), the DAE system does not involve any spatial derivative as one would expect from the LWR-type equations, by virtue of the variational formulation.

The proposed DAE system may be time-discretized and solved in a forward fashion. Figure 1 explains the time-stepping logic, which is implemented in the Matlab package alongside this publication.

4 The Fixed-Point Algorithm for Computing DUE

The computation of DUE is facilitated by the equivalent mathematical formulations such as variational inequality, differential variational inequality, fixed-point problem, and nonlinear complementarity problem. Some solution algorithms and their convergence conditions are mentioned in Table 1. In this section, we present the following algorithm based on the fixed-point formulation (2.11):

$$h^{k+1} = P_{\Lambda_0}\left[h^k - \alpha\Psi(h^k)\right] \tag{4.1}$$

where $\alpha > 0$ is a constant representing the step size, h^{k+1} and h^k respectively represent the path departure rate vector at the $(k+1)$-th and k-th iterations; $\Psi(h^k)$ denotes the effective path delay vector. Recalling the definition of Λ_0 from Eq. (2.10), the right hand side of Eq. (4.1), which involves projection, amounts to a linear-quadratic optimal control problem whose dual variables can be explicitly found following Pontryagin's minimum principle. The reader is referred to Friesz et al. (2011) for detailed discussion.

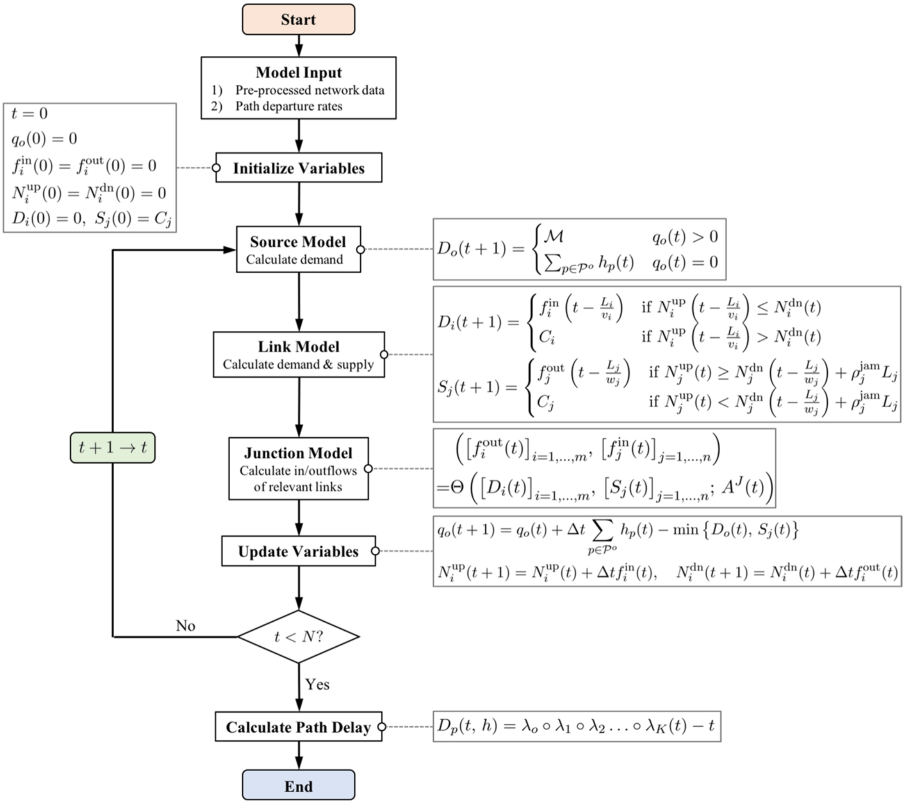


Fig. 1 Time-stepping logic of the discretized DAE system (3.15)-(3.23). Here, $t = 0, 1, 2 \cdots N$, represents discrete time steps with step size denoted Δt

We outline the main steps of the fixed-point algorithm below.

Fixed-Point Algorithm for Solving SRDT DUE

Step 0. Initialization. Set $k = 0$ and select an initial departure rate vector $h^0 \in \Lambda$. Fix a suitable constant $\alpha > 0$ for use in all iterations.

Step 1. Dynamic Network Loading. Perform the dynamic network loading with departure rate vector $h^k \in \Lambda$ to compute the effective path delays $\Psi_p(t, h^k)$ for all $t \in [t_0, t_f]$ and $p \in \mathcal{P}$.

Step 2. Fixed-Point Update. For every origin-destination pair $(i, j) \in \mathcal{W}$, solve the following algebraic equation for the dual variable v_{ij} (where $[x]_+ \doteq \max\{0, x\}$ assures non-negativity):

$$\sum_{p\in\mathcal{P}_{ij}} \int_{t_0}^{t_f} \left[h_p^k(t) - \alpha\Psi_p(t, h^k) + v_{ij}\right]_+ dt = Q_{ij} \tag{4.2}$$

For all $t \in [t_0, t_f]$ and $p \in \mathcal{P}_{ij}$, compute

$$h_p^{k+1}(t) = \left[h_p^k(t) - \alpha\Psi_p(t, h^k) + v_{ij}\right]_+$$

Step 3. Stopping Test. For a predetermined threshold $\epsilon > 0$, if

$$\frac{\|h^{k+1} - h^k\|^2}{\|h^k\|^2} \leq \epsilon$$

stop and declare h^{k+1} a DUE solution. Otherwise set $k = k + 1$ and go to **Step 1**.

Remark 4.1 *In the fixed-point algorithm, the critical step is to find the dual variable v_{ij} in (4.2). Note that this amounts to finding x such that $G(x) = 0$, where*

$$G(x) \doteq \sum_{p\in\mathcal{P}_{ij}} \int_{t_0}^{t_f} [h_p^k(t) - \alpha\Psi_p(t, h^k) + x]_+ dt - Q_{ij}$$

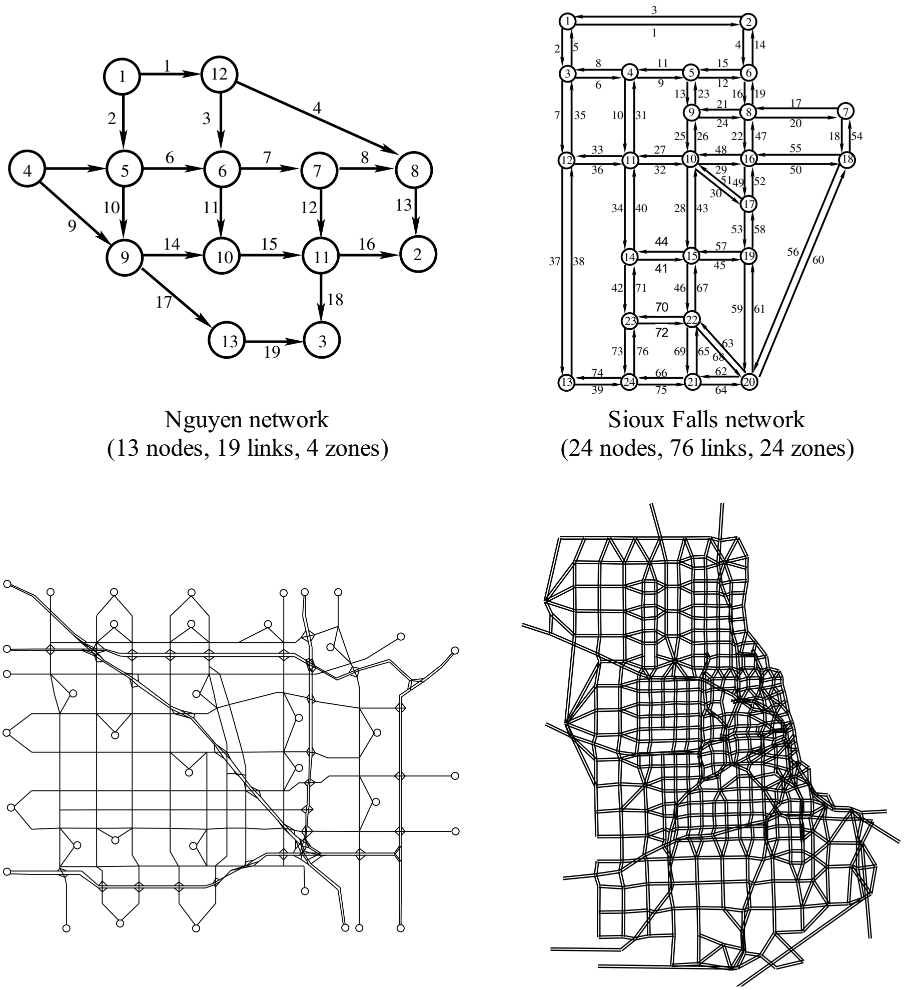
is a continuous function with a single argument x. Therefore, v_{ij} can be found via standard root-finding algorithms such as Bracketing or Bisection methods.

5 Computational Examples of DNL and DUE

We present several computational examples of the simultaneous route-and-departure-time choice dynamic user equilibria on four networks of varying sizes and shapes, as illustrated in Table 2 and Fig. 2. In particular, the Nguyen network was initially studied by Nguyen (1984), and the other three networks are based on real-world cities in the US, although different levels of network aggregation and simplification

Table 2 Key attributes of the test networks

	Nguyen network	Sioux Falls	Anaheim	Chicago sketch
No. of links	19	76	914	2,950
No. of nodes	13	24	416	933
No. of zones	4	24	38	387
No. of O-D pairs	4	528	1,406	86,179
No. of paths	24	6,180	30,719	250,000



Nguyen network
(13 nodes, 19 links, 4 zones)

Sioux Falls network
(24 nodes, 76 links, 24 zones)

Anaheim network
(416 nodes, 914 links, 38 zones)

Chicago sketch network
(933 nodes, 2950 links, 387 zones)

Fig. 2 The four test networks

have been applied. Detailed network parameters, including coordinates of nodes and link attributes, are sourced and adapted from Transportation Networks for Research Core Team (2018). Given that the DUE and DNL formulations in this paper are path-based, enumeration of paths was applied to generate path sets using the Frank-Wolfe algorithm.

The time horizon of analysis is [0, 5] (hrs), and time is in hr. The effective path delays (in hr) are defined for all four networks as follows:

$$\Psi_p(t,\,h) = D_p(t,\,h) + \begin{cases} 0.8\left(t + D_p(t,\,h) - \mathrm{TA}_{ij}\right)^2 & \text{if } t + D_p(t,\,h) < \mathrm{TA}_{ij} \\ 1.2\left(t + D_p(t,\,h) - \mathrm{TA}_{ij}\right)^2 & \text{if } t + D_p(t,\,h) \geq \mathrm{TA}_{ij} \end{cases} \qquad p \in \mathcal{P}_{ij}$$

where we employ asymmetric quadratic arrival penalties with TA_{ij} being the target arrival time of O-D pair $(i,\,j)$.

We apply the fixed-point algorithm (Section 4) with the embedded DNL procedure (Section 3). The fixed-point algorithm is chosen among many other alternatives in the literature, as our extensive experience with DUE computation suggests that this method tends to exhibit satisfactory empirical convergence within limited number of iterations, despite that its theoretical convergence requires strong monotonicity of the delay operator. The DNL sub-model is solved as a DAE system (3.15)–(3.23), following the time stepping logic in Fig. 1. All the computations reported in this section were performed using the MATLAB (R2017b) package on a standard desktop with Intel i5 processor and 8 GB of RAM.

5.1 Performance of the Fixed-Point Algorithm

The termination criterion for the fixed-point algorithm is set as follows:

$$\frac{\left\| h^{k+1} - h^k \right\|^2}{\left\| h^k \right\|^2} \leq \epsilon \tag{5.1}$$

where h^k denotes the path departure vector in the k-th iteration. The threshold ϵ is set to be 10^{-4} for the Nguyen and Sioux Falls networks, and 10^{-3} for the Anaheim and Chicago Sketch networks. These different thresholds were chosen to accommodate the varying convergence performances of the algorithm on different networks (also see Fig. 3).

Table 3 Performance of the fixed-point algorithm on different networks

	Nguyen network	Sioux Falls	Anaheim	Chicago Sketch
No. of iterations	54	73	45	69
Computational time	5.9 s	6.1 min	23.3 min	4.8 hr
Avg. time per DNL	0.1 s	4.3 s	25.7 s	163.9 s
Avg. time per FP update	0.007 s	0.6 s	3.1 s	81.2 s

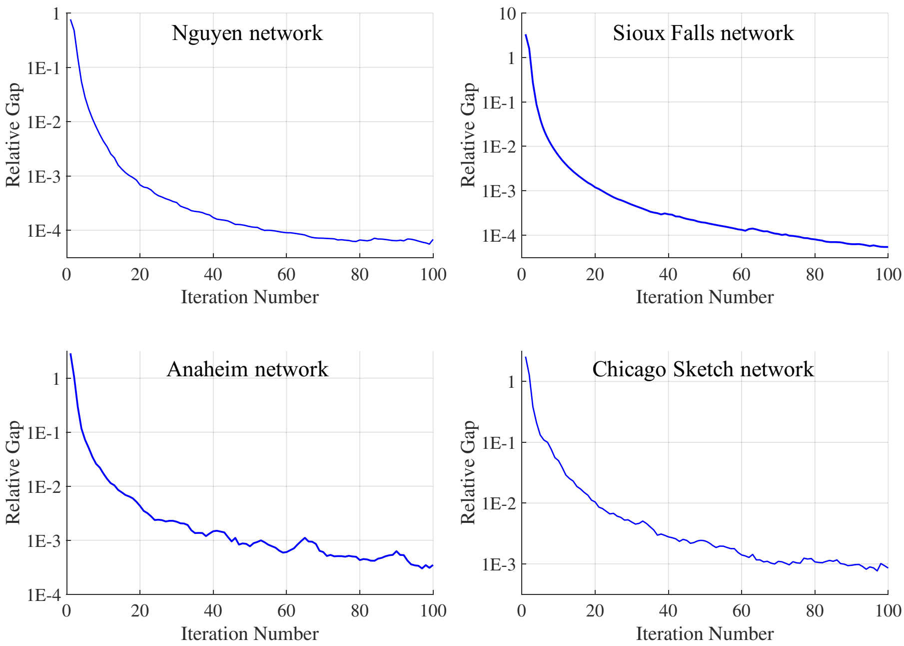


Fig. 3 The relative gaps (base 10 logarithmic scale) within 100 fixed-point iterations. Here, 1E-x means 10^{-x}

Table 3 summarizes the computational performance of the fixed-point algorithm and DNL procedure on different networks based on the termination criterion (5.1). It is shown that the number of fixed-point iterations is not significantly impacted by the varying sizes of the test networks, which suggests the scalability of the algorithms.

Figure 3 shows the relative gaps, i.e. left hand side of Eq. (5.1), for a total of 100 fixed-point iterations on the four networks. It can be seen that for relatively small networks (Nguyen and Sioux Falls), the convergence can be achieved relatively quickly and to a satisfactory degree; and the corresponding curves are monotonically decreasing and smooth. For Anaheim and Chicago Sketch networks, the decreasing trend of the gap can sometimes stall and experience fluctuations locally.

5.2 DUE Solutions

In this section we examine the DUE solutions obtained upon convergence of the fixed-point algorithm. We begin by selecting four arbitrary paths per network to illustrate the properties of the solutions. Figure 4 shows the path departure rates as well as the corresponding effective path delays. We observe that the departure rates are nonzero only when the corresponding effective delays are equal and minimum, which conforms to the notion of DUE. Note that the bottoms of the effective delay curves should theoretically be flat, indicating equal travel costs. This is not exactly the case in the figures since we can only obtain approximate DUE solutions in a numerical sense, given the finite number of fixed-point iterations performed to reach those solutions.

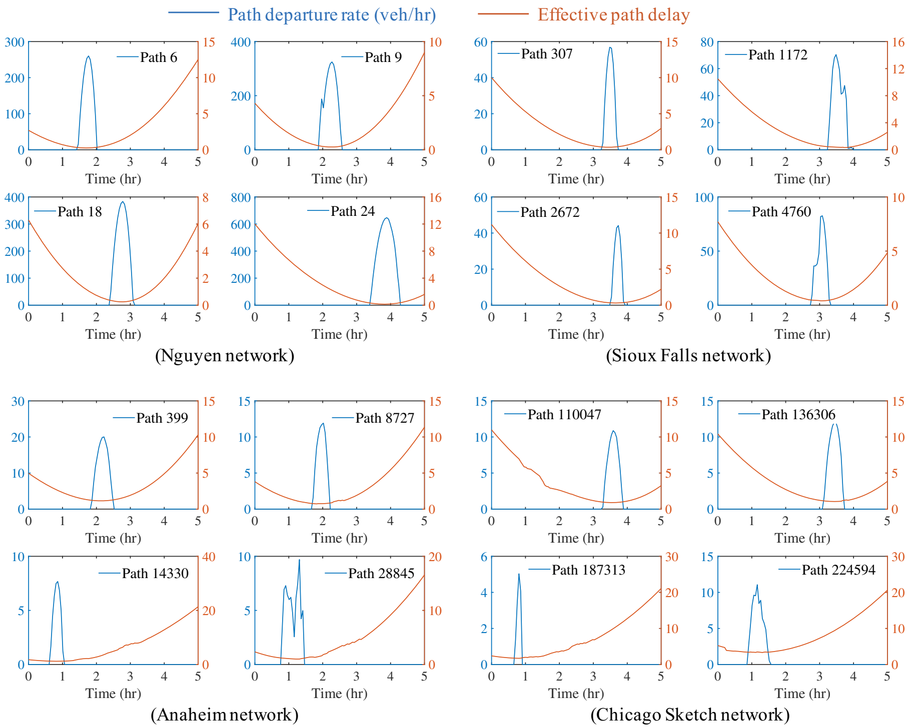


Fig. 4 Path departure rates and corresponding effective path delays (travel costs) of selected paths in the DUE solutions on the four test networks

To rigorously assess the quality of the DUE solutions, we define the gap function between each O-D pair $(i,\ j) \in \mathcal{W}$ as

$$\begin{aligned} \text{GAP}_{ij} \doteq \max &\left\{ \Psi_p(t,\ h^*) :\ t \in [t_0,\ t_f],\ p \in \mathcal{P}_{ij} \text{ such that } h_p^*(t) > 0 \right\} \\ - \min &\left\{ \Psi_p(t,\ h^*) :\ t \in [t_0,\ t_f],\ p \in \mathcal{P}_{ij} \text{ such that } h_p^*(t) > 0 \right\} \end{aligned} \quad (5.2)$$

Here, GAP_{ij} represents the range of travel costs experienced by travelers in the O-D pair $(i,\ j)$. In an exact DUE solution, the gap should be zero for all O-D pairs.

Figure 5 summarizes all the O-D gaps of the DUE solutions on the four test networks. It can be seen that the majority of the O-D gaps are within 0.2 (hrs) across all four networks. Even for large-scale networks (Anaheim and Chicago Sketch), the 75th percentiles are within 0.16, and the whiskers extend to 0.3. A comparison between Anaheim and Chicago Sketch also reveals that the latter yields better solution quality in terms of O-D gaps, despite the obviously larger size of the problem. This suggests that the solution quality is not necessarily compromised by the size of the network.

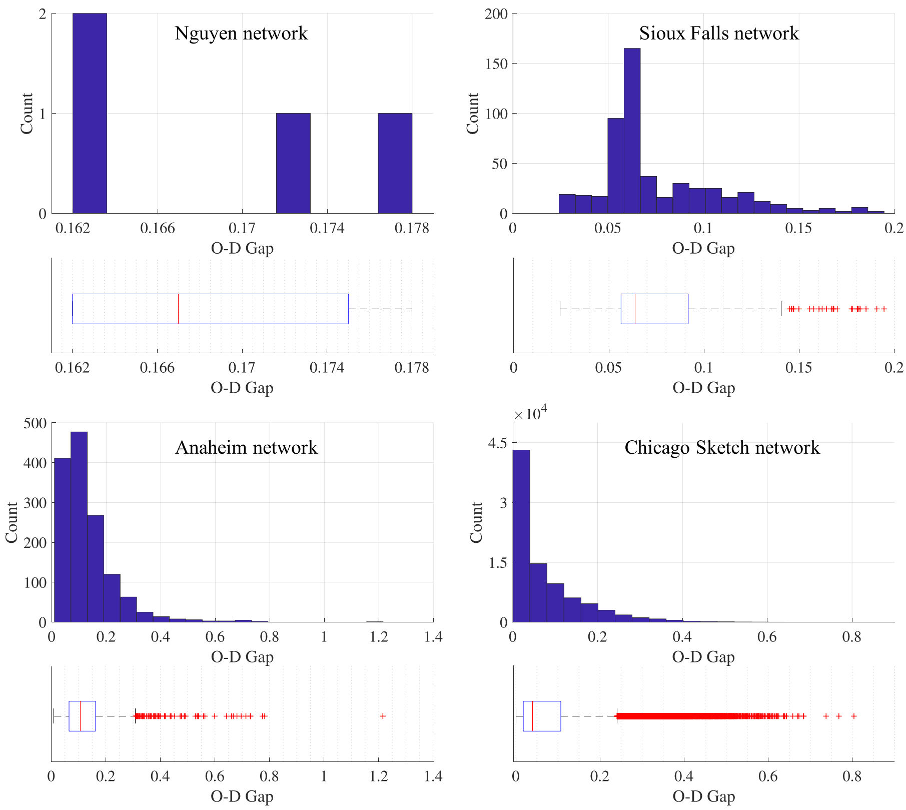


Fig. 5 Distributions of O-D gaps corresponding to the DUE solutions. The O-D gap is calculated according to Eq. 5.2

6 Conclusion

This paper presents a computational theory for dynamic user equilibrium (DUE) on large-scale networks. We begin by presenting a complete and generic dynamic network loading (DNL) procedure based on the network extension of the LWR model and the variational theory, which allows us to formulate the DNL problem as a system of differential algebraic equations (DAEs). The DNL model is capable of capturing the formation, propagation and dissipation of physical queues as well as vehicle spillback. The DAE system can be discretized and solved in a time-forward fashion. In addition, the fixed-point algorithm for solving DUE problems is presented.

Both the DAE system and the fixed-point algorithm are implemented in MATLAB, and the programs are developed in such a way that they can be applied to solve DUE and DNL problems on any user-defined networks. The MATLAB package is documented in the Appendix of this paper, which provides detailed instructions for using the solvers.

The MATLAB program is applied to solve DUE problems on several test networks of varying sizes. The largest one is the Chicago Sketch network with 86,179 O-D pairs and 250,000 paths. To the authors' knowledge this is by far the largest instance of SRDT DUE solution reported in the literature. Hopefully, our efforts in making these codes and data openly accessible could facilitate the testing and benchmarking of dynamic traffic assignment algorithms, and promote synergies between model development and applications.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix: Documentation of the MATLAB Package

A1. Dynamic Network Loading Solver

The MATLAB solver for the dynamic network loading problem is based on the time-discretized DAE system following the procedure in Fig. 1. The individual scripts (.m files) and input data files are illustrated in Fig. 6.

The DNL solver contains three main script files:

1. **Pre-processor: 'CREATE_NETWORK_PROPERTIES.m'**
   The purpose of this script is to prepare all the variables required for the main dynamic network loading script and save them to file for reuse to save computational time. Follow the steps below to run this script.

   1.1. Set user-defined inputs (at the top of file):

      a) Location of the network data file (e.g. '.../**Braess_dat.mat**')
      b) Location of path data file (e.g. '.../**Braess_paths.mat**')
      c) Value of source signal priority[1]

   1.2. Run script

   1.3. The output file will be saved to the working directory with '**_pp.mat**' suffix for use in the main DNL script.

2. **Main program: 'DYNAMIC_NETWORK_LOADING.m'**

   2.1. Set user inputs (at the top of file):

      a) Location of pre-processed data file (e.g. '.../**Braess_pp.mat**')

[1]The DNL code employs junction models that incorporate the continuous signal model proposed by Han et al. (2014) and Han and Gayah (2015). For any given junction (node), each of its incoming links, including the source if the node happens to be an origin, will be assigned a priority parameter between 0 and 1, such that the sum of relevant priority parameters equals 1. Once the priority of the source is specified, the priorities of the remaining incoming links are set proportional to the links' capacities. These parameters may be changed dynamically to accommodate different signal control scenarios or strategies.

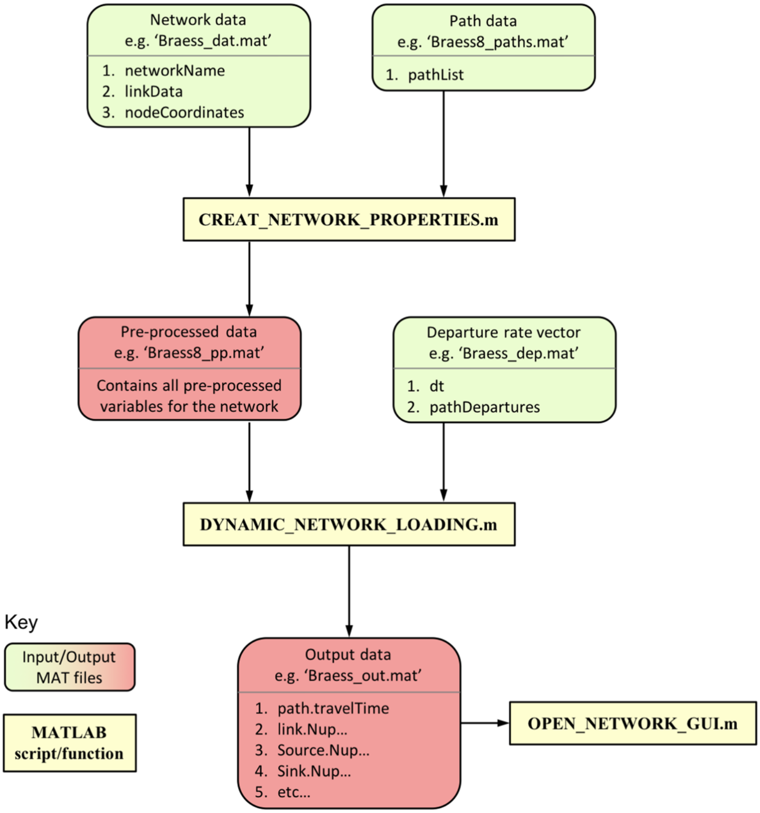


Fig. 6 Structure and interdependencies of all input and output files within the DNL solver. Boxes in green are user-defined data files

b) Location of path departure vector data file (e.g. '.../**Braess_dep.mat**'). The path departure vector should be a matrix of dimension $|\mathcal{P}| \times N$ where $|\mathcal{P}|$ is the number of paths and N is the number of time steps.

c) Number of time steps N, which should match the dimension of the path departure vector.

2.2. Run the script '**DYNAMIC_NETWORK_LOADING.m**'.

2.3. The output file will be saved to the working directory with '**_out.mat**' suffix, containing data required for graphical display.

3. Graphical display: 'OPEN_NETWORK_GUI.m'

3.1. Run script (can be run after loading any saved output files into the workspace)

Remark A.1 *While the main DNL program is written as a script file here, it can be easily modified into a function file. This is the case in our DUE solver presented later.*

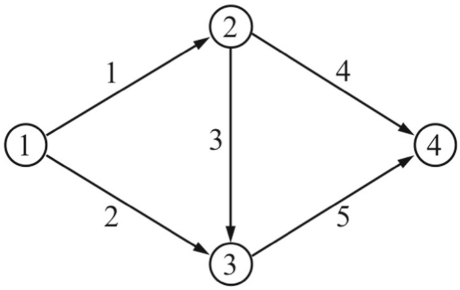


Fig. 7 The Braess network (example)

Variable Name	Description						
networkName	Name of network e.g. 'Braess'						
linkData (rows represent link number)	#	Tail Node	Head Node	Capacity (veh/s)	Length (m)	Free Flow Time (s)	Other Info (optional)
	1	1	2	0.5	7200	360	1
	2	1	3	0.5	7200	360	1
	3	2	3	0.5	7200	360	1
	4	2	4	0.5	7200	360	1
	5	3	4	0.5	7200	360	1
nodeCoordinates (optional – rows represent node number)	#	X-Coordinate	Y-Coordinate				
	1	-2.6	0				
	2	0	1.5				
	3	0	-1.5				
	4	2.6	0				

Fig. 8 Network data for the Braess network (example). Note that, for simplicity, the MATLAB program assumes that $v = 3w$ where v and w are respectively the speeds of the forward and backward kinematic waves; see Eq. 3.2. Therefore, the link attributes listed in this figure will identify a unique triangular fundamental diagram

Variable Name	Description			
pathList a) rows represent path number b) shorter paths must be **padded with zeroes**	#	1st link	2nd link	3rd link
	1	1	3	0
	2	2	0	0
	3	3	0	0
	4	1	4	0
	5	1	3	5
	6	2	5	0
	7	4	0	0
	8	3	5	0

Fig. 9 Path data for the Braess network (example)

Variable Name	Description						
dt	Time step in seconds, e.g. 5						
pathDepartures a) rows represent path number b) units are **veh/s**	#	0 s	1 dt s	2 dt s	3 dt s	4 dt s	etc...
	1	0.3	0.3	0.3	0.3	0.3	...
	2	0	0	0	0	0	...
	3	0.3	0.3	0.3	0.3	0.3	...
	4	0.3	0.3	0.3	0.3	0.3	...
	5	0.3	0.3	0.3	0.3	0.3	...
	6	0	0	0	0	0	...
	7	0.3	0.3	0.3	0.3	0.3	...
	8	0.3	0.3	0.3	0.3	0.3	...

Fig. 10 Path departure rate vector for the Braess network (example)

A2. Example of the DNL Solver

We consider the Braess network shown in Fig. 7 as an illustrative example. This network has four O-D pairs: (1, 3), (2, 3), (1, 4) and (2, 4), and eight paths:

- O-D (1, 3): $p_1 = \{1, 3\}$, $p_2 = \{2\}$;
- O-D (2, 3): $p_3 = \{3\}$;
- O-D (1, 4): $p_4 = \{1, 4\}$, $p_5 = \{1, 3, 5\}$, $p_6 = \{2, 5\}$; and
- O-D (2, 4): $p_7 = \{4\}$, $p_8 = \{3, 5\}$.

Example network data, path data, and path departure vector are illustrated in Figs. 8, 9 and 10, respectively.

To run the dynamic network loading program for the Braess network, we follow the diagram in Fig. 6:

1. Generate two data files, namely '**Braess_dat.mat**' and '**Braess8_paths.mat**' according to Figs. 8 and 9;[2]
2. Run the script '**CREATE_NETWORK_PROPERTIES.m**', which outputs and saves the file '**Braess8_pp.mat**' containing all the pre-processed network variables and parameters.
3. Generate the path departure rate vector file '**Braess_dep.mat**' according to Fig. 10.
4. Run the script '**DYNAMIC_NETWORK_LOADING.m**' in the same directory, which then outputs and saves the file '**Braess_out.mat**'. This file includes path travel times, which is required by the fixed-point algorithm for computing DUE, and several link variables that are required for the graphical display.
5. (optional) Run '**OPEN_NETWORK_GUI.m**' to open and operate the Graphical User Interface.

Figure 11 illustrates the Graphical User Interface. Four display options are available from the drop-down list:

- *Density* (veh/m): average link density computed as the number of vehicles traveling on the link at any time over the link length.
- *Relative density* (%): the aforementioned density over the jam density of the link.
- *Relative inflow* (%): the inflow of the link at any time over the link capacity.
- *Relative outflow* (%): the outflow of the link at any time over the link capacity.

On the lower-left corner of the user interface, one can specify the path number and press the "Path travel time" button. The interface then shows the path on the network and the corresponding path travel time function; see Fig. 12 for an example.

[2]For large networks, the path set can be generated by using the k-shortest path or Frank-Wolfe algorithms.

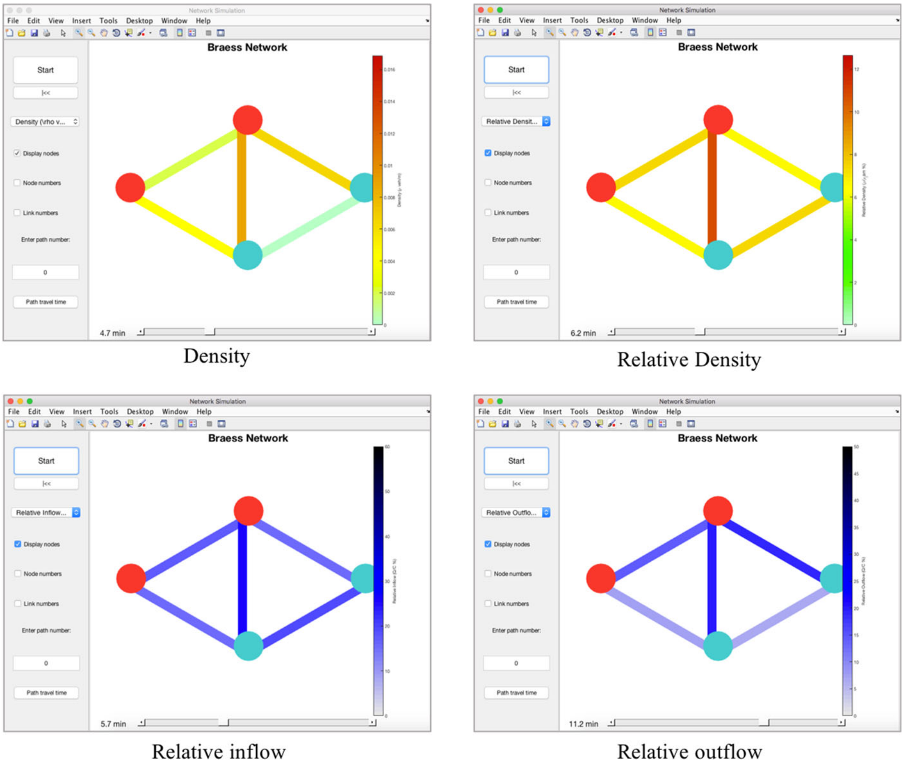


Fig. 11 Snapshots of the graphical user interface for the Braess network

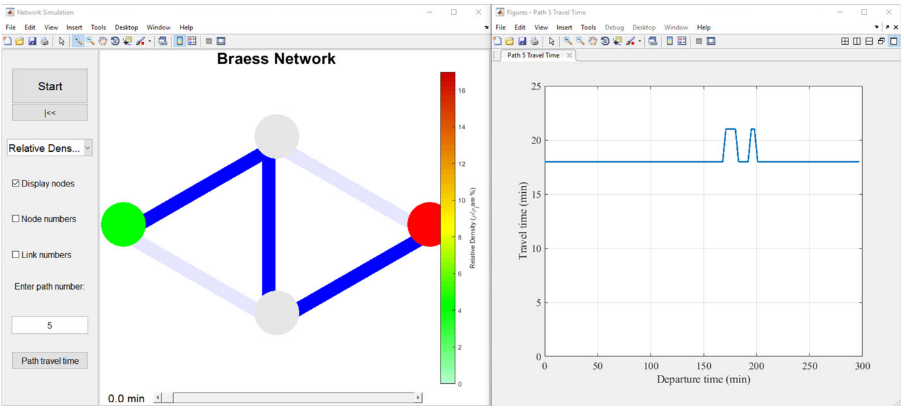


Fig. 12 Snapshots of the graphical user interface for the Braess network, which shows the travel time along path 5

A3. Dynamic User Equilibrium Solver

The dynamic user equilibrium solver is based on the fixed-point algorithm presented in Section 4. The individual scripts or functions (.m files) and input data files are illustrated in Fig. 13.

The MATLAB implementation of the DUE solution procedure consists of three main steps:

1. **Pre-processing the O-D information: 'PROCESS_OD_INFO.m'**
   The purpose of this script is to process the network data (e.g. '**Braess8_pp.mat**') in order to extract information about all the O-D pairs and their association with the paths. Follow the steps below to run this script.

   1.1. Prepare the pre-processed network data (e.g. '**Braess8_pp.mat**') and place it in the same working directory as '**PROCESS_OD_INFO.m**'.

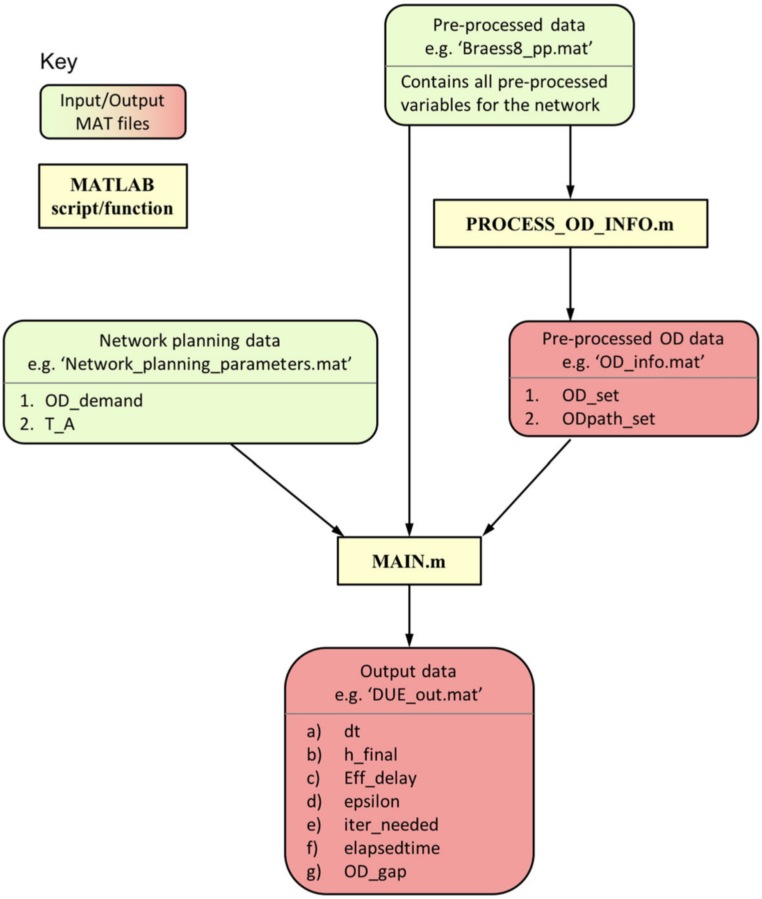


Fig. 13 Structure and interdependencies of all input and output files within the DUE program. Files in green are user-defined data

1.2. Run the script ‘**PROCESS_OD_INFO.m**’.
1.3. The output file, named ‘**OD_info.mat**’, will be saved to the working directory for use by the DUE solver.

2. Specify input file ‘Network_planning_parameters.mat’
This step mainly specifies the O-D demand and target arrival times. The O-D demand data ‘**OD_demand**’ is a vector of positive numbers, whose dimension is equal to the number of O-D pairs in the network. The target arrival time data ‘**T_A**’ is a vector of target arrival times, whose dimension is also equal to the number of O-D pairs. This means that each O-D pair is associated with one target arrival time.

3. Main program: ‘MAIN.m’

3.1. Set user inputs (at the top of file):

a) Time step (in second) of the discretized problem. Note that ideally the time step should be no less than the minimum link free-flow time in the network to ensure numerical stability. However, the code is conditioned to accommodate violation of this rule, in which case a warning message will be displayed.
b) Threshold that serves as the convergence criterion of the fixed-point algorithm; see Eq. 5.1.
c) Tolerance (indifference band) if bounded rationality (BR) is to be considered (Han et al. 2015b); the default value is set to be 0, which means no BR is considered. Empirically, and understandably, a positive tolerance tends to facilitate convergence of the fixed-point algorithm.
d) The maximum number of fixed-point iterations, upon which the algorithm will be terminated regardless of convergence.
e) The step size α of the fixed-point algorithm; see Eq. (4.2). Note that α needs to be adjusted for different networks and demand levels, as it has a major impact on convergence with respect to the criterion (5.1). Indeed, a very small step size α tends to terminate the fixed-point iterations prematurely while compromising the quality of the approximate DUE solution. On the other hand, a very large α causes undesirable oscillations among the iterates; and it normally takes more iterations to reach convergence. Another consideration, arising from Eq. (4.2), is that $\alpha\Psi_p(t,\, h^k)$ should be numerically comparable to $h_p^k(t)$.
f) Set the file path for the pre-processed network data (e.g. ‘.../**Braess8_pp.mat**’).

3.2. Run the script ‘**MAIN.m**’.
3.3. The output file, named ‘**DUE_out.mat**’, will be saved to the working directory. The output includes the following variables:

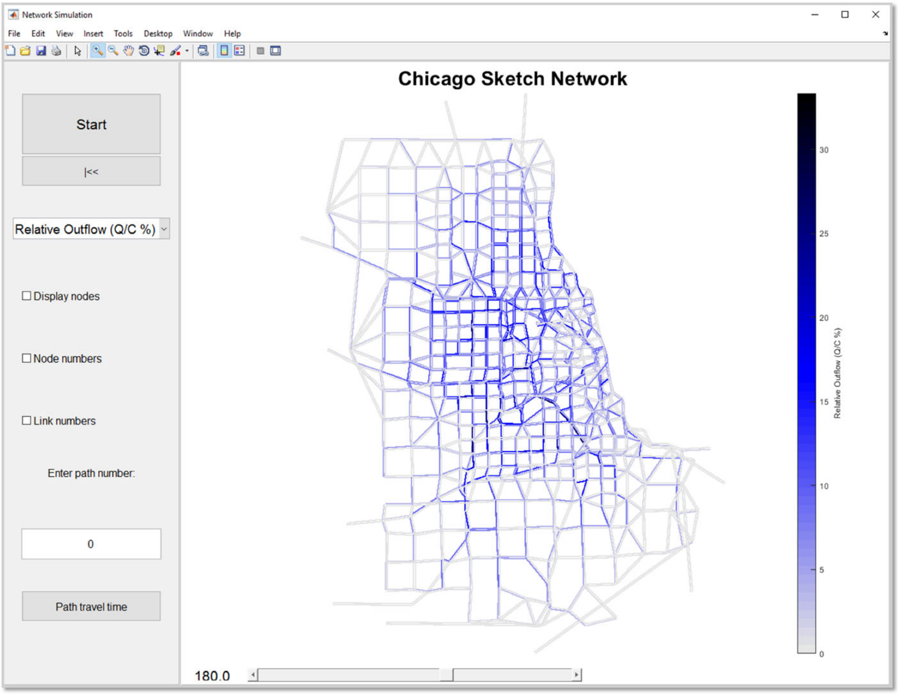


Fig. 14 Snapshot of the GUI that visualizes the DUE solution on the Chicago Sketch network. The color scale indicates the relative link outflow

a) ‘**dt**’: the discrete time step size.
b) ‘**h_final**’: the path flow vector upon convergence or forced termination of the fixed-point algorithm.
c) ‘**Eff_delay**’: the effective path delay vector, whose dimension matches that of ‘**h_final**’.
d) ‘**epsilon**’: the relative gap between two consecutive iterates of the fixed-point algorithm.
e) ‘**iter_needed**’: the number of iterations performed upon termination of the algorithm (either when the convergence criterion is met or the maximum iteration number is reached).
f) ‘**elapsedtime**’: time taken to run the DUE solver.
g) ‘**OD_gap**’: the travel cost gap for all the O-D pairs; see Eq. (5.2).

Figure 14 illustrates the DUE solution on the Chicago Sketch network obtained from the MATLAB solver. This is done by loading the DUE path flow vector into the DNL solver (script file) before running ‘OPEN_NETWORK_GUI.m’.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

Aubin JP, Bayen AM, Saint-Pierre P (2008) Dirichlet problems for some Hamilton-Jacobi equations with inequality constraints. SIAM J Control Optim 47(5):2348–2380

Balmer M, Cetin N, Nagel K, Raney B (2004) Towards truly agent-based traffic and mobility simulations. In: Proceedings of the 3rd international joint conference on autonomous agents and multiagent systems, vol 1, pp 60–67

Ban X, Pang J, Liu HX, Ma R (2011) Continuous-time point-queue models in dynamic network loading. Transp Res B 46(3):360–380

Ban X, Pang J, Liu HX, Ma R (2012) Modeling and solving continuous-time instantaneous dynamic user equilibria: a differential complementarity systems approach. Transp Res B Methodol 46:389–408

Transportation Networks for Research Core Team (2018) Transportation Networks for Research. https://github.com/bstabler/TransportationNetworks. Accessed 2 Jan 2018

Bliemer MCJ, Bovy PHL (2003) Quasi-variational inequality formulation of the multiclass dynamic traffic assignment problem. Transp Res B 37(6):501–519

Bliemer MCJ, Raadsen MPH, Brederode LJN, Bell MGH, Wismans LJJ, Smith MJ (2017) Genetics of traffic assignment models for strategic transport planning. Transp Rev 37(1):56–78

Boyce D, Lee DH, Ran B (2001) Analytical models of the dynamic traffic assignment problem. Networks and Spatial Economics 1:377–390

Cetin N, Burri A, Nagel K (2003) A large-scale agent-based traffic microsimulation based on queue model. In: Proceedings of swiss transport research conference (STRC), Monte Verita, Ch

Chen HK, Hsueh CF (1998) Discrete-time dynamic user-optimal departure time route choice model. Journal of Transportation Engineering - ASCE 124(3):246–254

Chiu YC, Bottom J, Mahut M, Paz A, Balakrishna R, Waller T, Hicks J (2011) Dynamic traffic assignment: a primer. Transportation Research E-Circular, (E-C153)

Claudel CG, Bayen AM (2010) Lax-Hopf Based incorporation of internal boundary conditions into Hamilton-Jacobi equation. Part I: Theory IEEE Trans Autom Control 55(5):1142–1157

Coclite GM, Garavello M, Piccoli B (2005) Traffic flow on a road network. SIAM J Math Anal 36:1862–1886

Costeseque G, Lebacque J (2014) A variational formulation for higher order macroscopic traffic flow models: numerical investigation. Transp Res B 70:112–133

Daganzo CF (1994) The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory. Transp Res B 28(4):269–287

Daganzo CF (1995) The cell transmission model, part ii: network traffic. Transp Res B 29(2):79–93

Daganzo CF (2005) A variational formulation of kinematic waves: basic theory and complex boundary conditions. Transp Res B 39:187–196

Daganzo CF (2006) On the variational theory of traffic flow: well-posedness, duality and application. Network and Heterogeneous Media 1(4):601–619

Friesz TL (2010) Dynamic optimization and differential games. Springer, New York

Friesz TL, Bernstein D, Smith T, Tobin R, Wie B (1993) A variational inequality formulation of the dynamic network user equilibrium problem. Oper Res 41(1):80–91

Friesz TL, Bernstein D, Suo Z, Tobin R (2001) Dynamic network user equilibrium with state-dependent time lags. Network and Spatial Economics 1(3-4):319–347

Friesz TL, Han K (2018) The mathematical foundations of dynamic user equilibrium. Transp Res B. https://doi.org/10.1016/j.trb.2018.08.015

Friesz TL, Han K, Neto PA, Meimand A, Yao T (2013) Dynamic user equilibrium based on a hydrodynamic model. Transp Res B 47(1):102–126

Friesz TL, Kim T, Kwon C, Rigdon MA (2011) Approximate network loading and dual-time-scale dynamic user equilibrium. Transp Res B 45(1):176–207

Friesz TL, Meimand A (2014) Dynamic user equilibria with elastic demand. Transportmetrica A: Transport Science 10(7):661–668

Friesz TL, Mookherjee R (2006) Solving the dynamic network user equilibrium with state-dependent time shifts. Transp Res B 40(3):207–229

Garavello M, Han K, Piccoli B (2016) Models for vehicular traffic on networks. American Institute of Mathematical Sciences, 2016

Greenshields BD (1935) A study of traffic capacity. In: Proceedings of the 13th annual meeting of the highway research board, vol 14, pp 448–477

Han K, Friesz TL, Szeto WY, Liu H (2015a) Elastic demand dynamic network user equilibrium: formulation, existence and computation. Transp Res B 81:183–209
Han K, Friesz TL, Yao T (2013a) A partial differential equation formulation of Vickrey's bottleneck model, part I: methodology and theoretical analysis. Transp Res B 49:55–74
Han K, Friesz TL, Yao T (2013b) Existence of simultaneous route and departure choice dynamic user equilibrium. Transp Res B 53:17–30
Han K, Gayah VV (2015) Continuum signalized junction model for dynamic traffic networks: offset, spillback, and multiple signal phases. Transp Res B 77:213–239
Han K, Gayah VV, Piccoli B, Friesz TL, Yao T (2014) On the continuum approximation of the on-and-off signal control on dynamic traffic networks. Transp Res B 61:73–97
Han K, Piccoli B, Friesz TL (2016a) Continuity of the path delay operator for dynamic network loading with spillback. Transp Res B 92(B):211–233
Han K, Piccoli B, Szeto WY (2016b) Continuous-time link-based kinematic wave model: formulation, solution existence, and well-posedness. Transportmetrica B: Transport Dynamics 4(3):187–222
Han K, Szeto WY, Friesz TL (2015b) Formulation, existence, and computation of boundedly rational dynamic user equilibrium with fixed or endogenous user tolerance. Transp Res B 79:16–49
Han K, Yao T, Jiang C, Friesz TL (2017) Lagrangian-based hydrodynamic model for traffic data fusion on freeways. Netw Spatial Econ 17(4):1071–1094
Han L, Ukkusuri S, Doan K (2011) Complementarity formulation for the cell transmission model based dynamic user equilibrium with departure time choice, elastic demand and user heterogeneity. Transp Res B 45(10):1749–1767
Huang HJ, Lam WHK (2002) Modeling and solving the dynamic user equilibrium route and departure time choice problem in network with queues. Transp Res B 36(3):253–273
Jeihani M (2007) A review of dynamic traffic assignment computer packages. J Transp Res Forum 46(2):35–46
Jin WL (2010) Continuous kinematic wave models of merging traffic flow. Transp Res B 44:1084–1103
Jin WL (2015) Continuous formulations and analytical properties of the link transmission model. Transp Res B 74:88–103
Kuwahara M, Akamatsu T (1997) Decomposition of the reactive dynamic assignments with queues for a many-to-many origin-destination pattern. Transp Res B 31(1):1–10
Laval J, Leclercq L (2013) The Hamilton-Jacobi partial differential equation and the three representations of traffic flow. Transp Res B 52:17–30
Lebacque J, Khoshyaran M (1999). In: Vilsmeier R, Benkhaldoun R, Hanel D (eds) Modeling vehicular traffic flow on networks using macroscopic models. Finite volumes for complex applications II. Hermes Science Publication; 1999, Paris, pp 551–558
Lighthill M, Whitham G (1955) 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(1178):317–345
Lo HK, Szeto WY (2002) A cell-based variational inequality formulation of the dynamic user optimal assignment problem. Transp Res B 36(5):421–443
Long JC, Huang HJ, Gao ZY, Szeto WY (2013) An intersection-movement-based dynamic user optimal route choice problem. Oper Res 61(5):1134–1147
Merchant DK, Nemhauser GL (1978a) A model and an algorithm for the dynamic traffic assignment problem. Transp Sci 12(3):183–199
Merchant DK, Nemhauser GL (1978b) Optimality conditions for a dynamic traffic assignment model. Transp Sci 12(3):200–207
Moskowitz K (1965) Discussion of freeway level of service as influenced by volume and capacity characteristics. In: Drew DR, Keese CJ (eds) Highway research record, vol 99, pp 43–44
Mounce R (2006) Convergence in a continuous dynamic queuing model for traffic networks. Transp Res B 40(9):779–791
Newell G (1993a) A simplified theory of kinematic waves in highway traffic, part i: general theory. Transp Res B 27(4):281–287
Newell G (1993b) A simplified theory of kinematic waves in highway traffic, part ii: queueing at freeway bottlenecks. Transp Res B 27(4):289–303
Newell G (1993c) A simplified theory of kinematic waves in highway traffic, part iii: multi-destination flows. Transp Res B 27(4):305–313

Nie Y, Zhang HM (2010) Solving the dynamic user optimal assignment problem considering queue spillback. Netw Spatial Econ 10(1):49–71
Nguyen S (1984) Estimating origin-destination matrices from observed flows. In: Florian M (ed) Transportation planning models. Elsevier Science Publishers, Amsterdam, pp 363–380
Osorio C, Flotterod G, Bierlaire M (2011) Dynamic network loading: a stochastic differentiable model that derives link state distributions. Transp Res B 45(9):1410–1423
Pang J, Han L, Ramadurai G, Ukkusuri S (2011) A continuous-time linear complementarity system for dynamic user equilibria in single bottleneck traffic flows. Mathematical Programming, Series A 133(1-2):437–460
Pang J, Stewart DE (2008) Differential variational inequalities. Mathematical Programming, Series A 113(2):345–424
Peeta S, Ziliaskopoulos AK (2001) Foundations of dynamic traffic assignment: the past, the present and the future. Netw Spatial Econ 1:233–265
Perakis G, Roels G (2006) An analytical model for traffic delays and the dynamic user equilibrium problem. Oper Res 54(6):1151–1171
Ran B, Boyce D (1996a) Modeling dynamic transportation networks: an intelligent transportation system oriented approach. Springer-Verlag, New York
Ran B, Boyce DE (1996b) A link-based variational inequality formulation of ideal dynamic user-optimal route choice problem. Transp Res C 4(1):1–12
Richards PI (1956) Shockwaves on the highway. Oper Res 4(1):42–51
Szeto WY, Lo HK (2004) A cell-based simultaneous route and departure time choice model with elastic demand. Transp Res B 38(7):593–612
Szeto WY, Lo HK (2005) Dynamic traffic assignment: review and future research directions. J Transp Syst Eng Inf Technol 5(5):85–100
Szeto WY, Lo HK (2006) Dynamic traffic assignment: properties and extensions. Transportmetrica 2(1):31–52
Shang W, Han K, Ochieng WY, Angeloudis P (2017) Agent-based day-to-day traffic network model with information percolation. Transportmetrica A: Transport Science 13(1):38–66
Smith MJ, Wisten MB (1994) Lyapunov Methods for dynamic equilibrium traffic assignment. In: Proceedings of the second meeting of the EURO working group on urban traffic and transportations. INRETS, Paris, pp 223–245
Smith MJ, Wisten MB (1995) A continuous day-to-day traffic assignment model and the existence of a continuous dynamic user equilibrium. Ann Oper Res 60(1):59–79
Tian LJ, Huang HJ, Gao ZY (2012) A cumulative perceived value-based dynamic user equilibrium model considering the travelers' risk evaluation on arrival time. Netw Spatial Econ 12(4):589–608
Tong CO, Wong SC (2000) A predictive dynamic traffic assignment model in congested capacity-constrained road networks. Transp Res B 34(8):625–644
Ukkusuri S, Han L, Doan K (2012) Dynamic user equilibrium with a path based cell transmission model for general traffic networks. Transp Res B 46(10):1657–1684
Varia HR, Dhingra SL (2004) Dynamic user equilibrium traffic assignment on congested multidestination network. Journal of Transportation Engineering - ASCE 130(2):211–221
Vickrey WS (1969) Congestion theory and transport investment. Am Econ Rev 59(2):251–261
Wang Y, Szeto WY, Han K, Friesz TL (2018) Dynamic traffic assignment: methodological ad- vances for environmentally sustainable road transportation applications. Transp Res B 111:370–394
Wardrop J (1952) Some theoretical aspects of road traffic research. Proceedings: Part II Engineering Divisions 1:325–362
Wie BW (2002) A diagonalization algorithm for solving the dynamic network user equilibrium traffic assignment model. Asia-Pacific J Oper Res 19(1):107–130
Wie BW, Tobin RL, Carey M (2002) The existence, uniqueness and computation of an arc-based dynamic network user equilibrium formulation. Transp Res B 36(10):897–918
Xu Y, Wu JH, Florian M, Marcotte P, Zhu D (1999) Advances in the continuous dynamic network loading problem. Transp Sci 33(4):341–353
Yperman I, Logghe S, Immers L (2005) The link transmission model an efficient implementation of the kinematic wave theory in traffic networks. In: Proceedings of the 10th EWGT meeting and 16th mini-EURO conference advanced OR and AI methods in transportation. Publishing House of Poznan University of Technology, Poznan, pp 122–127
Zhu DL, Marcotte P (2000) On the existence of solutions to the dynamic user equilibrium problem. Transp Sci 34(4):402–414

Affiliations

Ke Han[1] · **Gabriel Eve**[1] · **Terry L. Friesz**[2]

Gabriel Eve
gabrielfeve@gmail.com

Terry L. Friesz
tfriesz@psu.edu

1 Department of Civil and Environmental Engineering, Imperial College London, London, SW7 2BU, UK

2 Department of Industrial and Manufacturing Engineering, Pennsylvania State University, Pennsylvania, PA 16803, USA