DTU Library

Efficient Computation of User Optimal Traffic Assignment via Second-Order Cone and Linear Programming Techniques

Wei, Wei; Hu, Longxian; Wu, Qiuwei; Ding, Tao

Published in:
IEEE Access

Link to article, DOI:
10.1109/ACCESS.2019.2942497

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 5, 2019, accepted September 16, 2019, date of publication September 19, 2019, date of current version October 2, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2942497

Efficient Computation of User Optimal Traffic Assignment via Second-Order Cone and Linear Programming Techniques

WEI WEI[1], (Senior Member, IEEE), LONGXIAN HU[1], QIUWEI WU[2], (Senior Member, IEEE), AND TAO DING[3], (Senior Member, IEEE)

[1]State Key Laboratory of Power Systems, Department of Electrical Engineering, Tsinghua University, Beijing 100084, China
[2]Department of Electrical Engineering, Technical University of Denmark, 2800 Lyngby, Denmark
[3]Department of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710049, China

Corresponding author: Wei Wei (wei-wei04@mails.tsinghua.edu.cn)

This work was supported in part by the Young Elite Scientists Program of CSEE under Grant JLB-2018-95, and in part by the National Natural Science Foundation of China under Grant 51621065 and Grant U1766203.

ABSTRACT Static traffic assignment aims to disclose the spatial distribution of vehicular flow over a transportation network subject to given traffic demands, and plays an essential role in transportation engineering. User-optimal pattern adheres the individual rationality of motorists, in which everyone chooses a route that minimizes his own travel cost, while considering congestion effects influenced by the aggregated movement of vehicles. User optimal traffic assignment, which is also known as the user equilibrium, entails solving an optimization problem with a strictly convex objective function and linear constraints. However, the performances of general-purpose solvers are quite disappointing. This paper proposes two highly efficient computation models for the user equilibrium problem. The first one exploits the second-order cone reformulation of a convex power function, resulting in a second-order cone program, and no approximation is incurred. The second one approximates the convex objective function using a piece-wise linear function, and comes down to a linear program. An adaptive path generation oracle is devised in order to circumvent path enumeration in problem setup. Case studies demonstrate that the proposed method can deal with large-scale transportation systems, and outperforms the most popular iterative algorithm in the literature.

INDEX TERMS Linear program, second-order cone program, traffic assignment, user equilibrium.

I. INTRODUCTION

The traffic assignment problem (TAP) is a fundamental problem in transportation engineering and underlies many practical applications such as network topology design, road capacity expansion planning, and traffic signal control. Provided with the volume of vehicles traveling between each pair of origin-destination (O-D) nodes, TAP discloses the vehicular flow distribution among roadway segments subject to flow conservation constraints. The most widely used criteria that extract a reasonable outcome from infinitely many feasible network flow solutions were proposed by Wardrop in 1952 [1], including the user-equilibrium (UE) principle and system optimum (SO) principle. SO assumes that a central operator determines the routes for every motorist aiming to minimize the total travel time, and all motorists are willing to accept the supervision of the operator. This is an ideal assumption. In practice, every driver is most likely to choose a route that minimizes his own travel time, leading to the UE principle. In this circumstance, a stable distribution of network flow emerges if no one has the incentive to alter his current route. UE has a close relationship with the Nash equilibrium when the number of players in Nash game approaches infinity [2].

The Wardrop UE condition is stated in an if-then form and can be expressed by complementarity constraints, which are non-convex. Beckmann et al. revealed that the UE condition exactly constitutes the Karush-Kuhn-Tucher (KKT) optimality condition of a convex optimization problem [3]. Due to the elegant property of convex optimization, Beckmann's formulation quickly becomes the reference model for solving a user-optimal TAP. A detailed tutorial on UE and SO based

The associate editor coordinating the review of this manuscript and approving it for publication was Md Asaduzzaman.

TAP can be found in [4]. This paper mainly focuses on the former one. Since the pioneering work in [1] and [3], various methods have been developed to solve the UE based TAP, which can be divided into two categories.

The first category directly solves Beckmann's model while leveraging its convexity. The renowned representative in this category is the Frank-Wolfe (F-W) algorithm [5], which was firstly embedded in a descent algorithm to solve the link-flow based Beckmann model in [6]. The procedure in [6] follows the basic steps of an iterative descent algorithm for nonlinear program [7]. It entails solving a linear program (LP) and a one-dimensional search in each step. Due to the nature of UE, the LP always assigns all traffic flow to the shortest path, so that the subproblem can be solved instantly. Several variants were given in [8]–[10], with modifications on the search direction or step length. The Link-flow based Beckmann model does not provide path flow information, which offers insights on how motorists between the same O-D pair choose different routes. The path flow-based Beckmann model was discussed with a simplicial decomposition algorithm in [11], a gradient projection algorithm in [12], and a conjugate gradient projection algorithm in [13]. Likewise, the performance of these algorithms depends on the updating strategies in each iteration and specific problem data.

The second category formulates the Wardrop UE condition as a variational inequality since the work in [14]–[16], and thus any algorithm for the variational inequality problem can be used to solve the UE problem, such as the projection method [17]. More algorithms for the variational inequality problem can be found in [16]. Since variational inequality is a more general modeling framework than traditional mathematical programming, it allows incorporating multi-variate link travel cost functions depending on traffic flows in multiple links. Recently, the variational inequality theory has been shown to be very useful in dynamic TAP, which takes the elapse of time into account [18], [19]. This category mainly pursues the modeling capability of variational inequality rather than the computational efficiency, as the dynamic TAP is extremely challenging and is much less mature than the static TAP.

In the past decade, research efforts are devoted to more dedicated traffic models, considering the macroscopic fundamental diagram [20], limited driving distance and parking choice [21], [22], electric vehicles and charging stations [23]–[25], ride-sharing travellers [26]–[28], and reference-dependent utility [29], users' heterogeneity [30], transit/queue dynamics [31], [32], and traffic signal control [33], [34]. Most of them can be formulated as a convex optimization problem. In addition to the traffic assignment approach, data-driven methods have been developed to estimate O-D travel time [35], [36]. Nonetheless, traffic assignment is still acknowledged as the most fundamental problem that provides the holistic information of the entire transportation network.

With the development of effective algorithms for convex programs, e.g., the interior-point algorithm which has worst-case polynomial-time complexity [37], problems with moderate sizes can be solved by off-the-shelf solvers. This paper focuses on the fundamental UE problem, and aims to fill the gap between the UE computation and highly efficient algorithms for conic and linear programs. On the one hand, the convergence rates of the procedures in [6] and its variants heavily rely on the updating strategy in each iteration. Although the problem in each step can be solved fast, we find in experiments that this algorithm converges slowly around the optimal solution, and the computation time grows quickly with the reduction in error tolerance. On the other hand, although TAP is formulated as a convex optimization problem, we find in experiments that general-purpose solvers, no matter interior-point algorithm based ones or sequential quadratic programming based ones, can only solve small-scale instances, unlike the state-of-the-art LP solver which can cope with problems with hundreds of thousands of variables. In fact, between two extremes of convex programs, i.e., an LP and a general convex program, there are many other important and useful instances; one of them is the second-order cone program (SOCP). Although nonlinear, SOCP still possesses nice structured properties which can be utilized to develop more dedicated interior-point algorithms. These algorithms for SOCP turn out to be considerably faster than those exploiting only general convexity. Finally, as a common difficulty, in the path-based TAP model, enumerating all available paths between a given O-D pair not only consumes a lot of time but also introduces a large number of unnecessary decision variables in the optimization model, because only a few paths will be actually used. However, the subset of paths that carry traffic flow is not known in advance. In the link-based TAP model, the total link flow is expressed via the sum of subcomponents contributed by the traffic demand between each O-D pair. Hence, the link-based TAP model is also not scalable with the number of O-D pairs.

This paper proposes two highly efficient computation models for the user-optimal TAP which overcome the aforementioned difficulties. The first one exactly reformulates the Beckmann's model as a SOCP, as long as road latency can be expressed by a convex power function, which is a common paradigm in transportation engineering. The second one approximates the convex objective function via a piece-wise linear function and comes down to a single LP. It is eligible for any increasing road latency function and is slightly more tractable than the SOCP model. The approximation error is discussed and tested in the case study. An adaptive path generation oracle is devised in order to circumvent path enumeration.

The rest of this paper is organized as follows. The Beckmann model is introduced in Section II. The SOCP formulation, the LP formulation, and the adaptive path generation oracle are presented in Section III. Case studies are conducted in Section IV. Finally, conclusions are drawn in Section V. Throughout the paper, UE and user-optimal TAP are used interchangeably.

II. MATHEMATICAL FORMULATION

A. TRANSPORTATION NETWORK MODEL

A transportation network can be abstractly modeled by a graph $G = [N, A]$, where N stands for the set of nodes such as origin and destination areas, as well as intersection points of roadways; A denotes the set of links representing roadways. The connection topology is described by node-link incidence matrix Λ with a dimension of $|N| \times |A|$, where $|\cdot|$ is the cardinality (number of elements) of a set. Each column of Λ corresponds to a link with $1(-1)$ at the entry associated with the entrance (exit) node, i.e.:

$$\Lambda_{ij} = \begin{cases} +1, & \text{if node } i \text{ is the entrance of link } j; \\ -1, & \text{if node } i \text{ is the exit of link } j; \\ 0, & \text{if there is no connection;} \end{cases}$$

Let (r, s) be an O-D pair. The volume of vehicles traveling from an origin node $r \in N$ to a destination node $s \in N$ is called the traffic demand q^{rs}. Matrix Q with a dimension of $|N| \times |N|$ is the full-dimensional O-D matrix, and is generally non-symmetric. Let $D_S^R = \{(r, s)|q^{rs} > 0\}$ be the set of active O-D pairs. It should be mentioned that traffic flow and demand in TAP are real numbers, indicating that the system impact of a single vehicle is negligible.

A path is a chain of connected links between an O-D pair (r, s). All available paths connecting O-D pair $(r, s) \in D_S^R$ are labeled by index $k \in K^{rs}$. f_k^{rs} is the traffic flow carried by path k between (r, s). For any O-D pair, the sum of path flows must meet the traffic demand

$$f_k^{rs} \geq 0, \ \sum_k f_k^{rs} = q^{rs}, \quad \forall (r, s) \tag{1}$$

Topological relations between paths and links are portrayed by the link-path incidence matrix $\Delta = [\Delta^{rs}], \forall (r, s) \in D_S^R$, where the sub-matrix Δ^{rs} with a dimension of $|A| \times |K^{rs}|$ corresponds to a particular O-D pair (r, s), and its elements are defined as

$$\delta_{ak}^{rs} = \begin{cases} 1, & \text{if path } k \text{ passes link } a \\ 0, & \text{otherwise;} \end{cases}$$

With the definition of matrix Δ, the traffic flow x_a on link a can be expressed as a linear function of path flow f_k^{rs}

$$x_a = \sum_{rs} \sum_k f_k^{rs} \delta_{ak}^{rs}, \quad \forall a \tag{2}$$

To formulate equation (2), we should enumerate all available paths between any O-D pair, which is a challenging task. In the next section, we discuss an adaptive path generation oracle so that path enumeration can be circumvented.

B. USER OPTIMAL TRAFFIC ASSIGNMENT MODEL

In the user-optimal TAP, motorists decides their paths based on the travel cost. Travel time is often the primary concern. Let t_a be the time spent on link a. In the classic model, t_a is a function of x_a and independent of the flows in other links, while x_a depends on the decision of all motorists in the system as in (2). The following power function is widely used to quantify road travel time [4]

$$t_a(x_a) = t_a^0 \left[1 + \alpha \left(\frac{x_a}{c_a} \right)^\beta \right] \tag{3}$$

where t_a^0 is the zero-flow travel time; c_a is sometimes referred to as the capacity of link a. In fact, it is not a mandatory upper bound. The travel time t_a would grow quickly if x_a exceeds c_a, placing a penalty for further usage of this road. The remaining two parameters recommended by the Bureau of Public Road (BPR) are $\alpha = 0.15$ and $\beta = 4.0$ [38].

Given link travel time, the path travel time c_k^{rs} perceived by a single traveller between O-D pair (r, s) is calculated by the sum of link travel times

$$c_k^{rs} = \sum_a t_a(x_a) \delta_{ak}^{rs}, \quad \forall k, \ \forall (r, s) \tag{4}$$

TAP aims to identify a reasonable outcome of traffic flow distributions from its feasible set (1)-(2) taking into account the individual rationality of motorists. The following Wardrop UE principle is used [4]:

At a user equilibrium state, for each O-D pair, the travel time on all used paths is equal, and no greater than the travel time that would be experienced by a single vehicle on any unused path.

The intuition is clear. If the travel time on used paths is not equal, or travelling on an unused path could save travel time, then some vehicles will have the incentive to switch to that path, leading to a change in the distribution of traffic flow. Above UE principle can be interpreted by a logic condition

$$\begin{aligned} &\text{if } f_k^{rs} > 0, \quad \text{then } c_k^{rs} = u^{rs}, \ k \in K^{rs}, \ \forall (r, s) \\ &\text{if } f_k^{rs} = 0, \quad \text{then } c_k^{rs} \geq u^{rs}, \ k \in K^{rs}, \ \forall (r, s) \end{aligned}$$

where the minimal travel time u^{rs} between (r, s) is also a variable to be determined. These logical conditions can be expressed by complementarity and slackness constraints

$$0 \leq f_k^{rs} \perp c_k^{rs} - u^{rs} \geq 0, \quad k \in K^{rs}, \ \forall (r, s) \tag{5}$$

where $0 \leq x \perp y \geq 0$ stands for $x \geq 0$, $y \geq 0$, and $xy = 0$, preventing a and b from being strictly positive at the same time. (5) should be jointly solved with network flow model (1)-(2) and travel time model (3)-(4), which is computationally difficult, because complementarity constraints are non-convex and violate Mangasarian-Fromovitz constraint qualification at any feasible point [39].

Fortunately, complementarity constraint (5) combining with (1)-(4) happen to constitute the KKT optimality condition of the following optimization problem [3], [4],

$$\begin{aligned} &\min \sum_a \int_0^{x_a} t_a(\theta) \mathrm{d}\theta \\ &\text{s.t. Net-flow} \end{aligned} \tag{6}$$

where the feasible set Net-flow is short for (1)-(2). Problem (6) is the path flow based Beckmann model. Its feasible set is a polyhedron. The objective function is the sum of univariate function. For each term, calculate its derivative

$$\frac{\mathrm{d}^2}{\mathrm{d}x_a^2}\left(\int_0^{x_a} t_a(\theta)\mathrm{d}\theta\right) = \frac{\mathrm{d}t_a(x_a)}{\mathrm{d}x_a}$$

Because t_a is a increasing function in x_a, the second-order derivative is always positive, and thus the Hessian of the objective function is positive definite. Therefore, problem (6) is strictly convex in x_a and has a unique link flow solution. There could be multiple path flow solutions satisfying (2).

III. EFFICIENT COMPUTATION MODELS

Although problem (6) is convex, the worst-case complexity of the interior-point algorithm for finding a ε-optimal solution of a general convex program turns out to be [37]

$$O(1)n\left(n^3 + M\right)\ln\left(\frac{1}{\varepsilon}\right)$$

where n is the number of variables; at any given point, the values of functions in the objective and constraints together with their derivatives can be computed using M arithmetic operations. This bound is already unacceptable for n with an order of magnitude of 10^3. In contrast, an LP with hundreds of thousands of variables can be solved very efficiently. Because a mature LP solver never evaluates function values. All structured information is contained in the constant coefficient matrix. SOCP is almost as tractable as LP [40]. In this section, we will present SOCP and LP models for problem (6). The proposed model can be easily generalized to tackle various recently proposed UE models.

A. SOCP FORMULATION

We have already shown that the objective function of (6) is strictly convex. For BPR recommended parameters, problem (6) evolves minimizing

$$\sum_a t_a^0\left[x_a + \frac{0.03\,x_a^5}{c_a^4}\right]$$

Replace the nonlinear term $(x_a)^5$ with a new variable y_a and let $\gamma_a = 0.03c_a^{-4}$, problem (6) gives rise to

$$\begin{aligned} \min\ & \sum_a t_a^0\,(x_a + \gamma_a y_a) \\ \text{s.t. } & \text{Net-flow}, \quad y_a \geq (x_a)^5,\ \forall a \end{aligned} \tag{7}$$

Because the objective is to be minimized, $y_a = (x_a)^5$ naturally holds at the optimal solution. This is known as the epigraph transformation [41]. The convex region $\mathbb{C}_{\mathbb{P}} = \{(x_a, y_a)|x_a \geq 0, y_a \geq (x_a)^5\}$ ($x_a \geq 0$ is included in (1)) is the epigraph of function $y_a = (x_a)^5$ over $\mathbb{R}_+$. However, a solver is unable to recognize such convexity. We need to represent the epigraph via the intersection of second-order cone constraints. For the sake of exposition, we omit the subscript a. Consider the following set of constraints

$$x^2 \leq v_{21}v_{22} \tag{8a}$$
$$v_{21}^2 \leq v_{11}v_{12} \tag{8b}$$
$$v_{22}^2 \leq v_{13}v_{14} \tag{8c}$$
$$0 \leq v_{11} \leq x \tag{8d}$$
$$v_{12}^2 \leq x \tag{8e}$$
$$v_{13}^2 \leq y \tag{8f}$$
$$0 \leq v_{14} \leq 1 \tag{8g}$$

where $v_{21}, v_{22}, v_{11}, v_{12}, v_{13}, v_{14}$ are auxiliary variables. To see the equivalence between (8) and $y \geq x^5$, squaring both sides of (8a) and substituting (8b) and (8c) we have

$$x^4 \leq v_{11}v_{12}v_{13}v_{14} \tag{9}$$

For any point satisfying $y \geq x^5$, we can find $v_{11} = x$, $v_{12} = \sqrt{x}$, $v_{13} = \sqrt{y}$ and $v_{14} = 1$, such that (9) is met; conversely, suppose v_{11}, v_{12} and v_{14} are not equal to 0 (otherwise, $x = 0$ is the unique feasible solution), because $v_{13} \geq x^4/(v_{11}v_{12}v_{14})$, (8d)-(8g), we have $y \geq v_{13}^2 \geq x^8/(v_{11}^2\,v_{12}^2\,v_{14}^2) \geq x^5$ is met.

In (8), (8a)-(8c) are rotated second-order cones; take (8a) for example, its canonical form is

$$\left\| \begin{matrix} 2x \\ v_{21} - v_{22} \end{matrix} \right\| \leq v_{21} + v_{22}, \quad v_{21} \geq 0,\ v_{22} \geq 0 \tag{10}$$

(8e) and (8f) are convex quadratic constraints, and remaining ones are linear. They do not harm computational efficiency.

We now consider a more general case in which β in function (3) is a rational number, i.e., $\beta = p/q > 1$ where p and q are positive integers. This entails a conic representation of region $\mathbb{C}_{\mathbb{P}}^{\beta} = \{(x, y)|y \geq x^{p/q}, x \geq 0\}$. We utilize the technique in [37]. Let l be the smallest integer satisfying $p \leq 2^l$ and $r = 2^l - p$. Consider

$$s^{2^l} \leq y_1 y_2 \cdots y_{2^l} \tag{11}$$

where all variables are non-negative. This inequality can be represented by second-order cones through introducing l generation tower of variables. More precisely, we call the original 2^l y-variables the variable of generation 0 denoted by $y_{0,i}$ instead of y_i. For every pair of 0-generation variables, add a new variable of generation 1 denoted by $y_{1,i}$. There are totally 2^{l-1} 1-generation variables. Repeat above procedure, we get 2^{l-2} 2-generation variables, 2^{l-3} 3-generation variables, until l generation with a single variable $y_{l,1}$ is built. Construct the following system of constraints

$$\begin{aligned} &\text{Layer } 1: (y_{1,i})^2 \leq y_{0,2i-1}y_{0,2i}, \quad i = 1, \cdots, 2^{l-1} \\ &\text{Layer } 2: (y_{2,i})^2 \leq y_{1,2i-1}y_{1,2i}, \quad i = 1, \cdots, 2^{l-2} \\ &\qquad \cdots\cdots\cdots\cdots \\ &\text{Layer } l: (y_{l,1})^2 \leq y_{l-1,1}y_{l-1,2} \\ &\text{Layer } l+1: s \leq y_{l,1} \\ &\qquad\qquad \text{all variables are non-negative} \end{aligned}$$

By squaring inequalities in each generation, we arrive at (11). The standard form of rotated second-order cones in each layer is similar to (10).

To retrieve a conic representation of $C_{\mathbb{P}}^{\beta}$, based on (11), let $s = y_1 = y_2 = \cdots = y_r = x$, $y_{r+1}, \cdots, y_{r+q} = y$, and all remaining variables, if there is any, are equal to 1. In this way, (11) evolves $x^{2^l - r} \leq y^q$, which gives $y \geq x^{p/q}$ (recall $r = 2^l - p$).

In summary, TAP (6) is equivalent to the following SOCP

$$\begin{aligned} \min \ & \sum_a t_a^0 \left(x_a + \gamma_a y_a \right) \\ \text{s.t. } & \text{Net-flow, Pow-fun-SOC} \end{aligned} \tag{12}$$

where Pow-fun-SOC collects all constraints of conic representation of power functions. Although auxiliary variables are introduced in (8) and the more general case, such an attempt is worthwhile because solving SOCP is almost as efficient as solving LP, which will be demonstrated in case study.

B. LP APPROXIMATION

Another way to approximate convex function $y_a = (x_a)^{\beta+1}$, $\beta > 1$ in interval $[0, x_{am}]$ is to use piecewise linear function. Suppose x_a^k and $y_a^k = (x_a^k)^{\beta+1}$, $k = 1, \cdots, n$ denote the collection of break points. Introduce new decision variables λ_a^k, $k = 1, \cdots, n$, which represent the weights associated with the break points, then a point on function $y_a = (x_a)^{\beta+1}$ can be approximated by the convex combination of the break points:

$$\begin{aligned} x_a &= \lambda_a^1 x_a^1 + \lambda_a^2 x_a^2 + \cdots + \lambda_a^n x_a^n \\ y_a &= \lambda_a^1 y_a^1 + \lambda_a^2 y_a^2 + \cdots + \lambda_a^n y_a^n \\ \lambda_a &= [\lambda_a^1, \cdots, \lambda_a^n] \in \Delta_n \end{aligned} \tag{13}$$

where vector λ_a is decision variable; $\Delta_n = \{\lambda \in \mathbb{R}_+^n | 1^T \lambda = 1\}$ is the unit probability simplex. So TAP (6) can be approximated by the following LP

$$\begin{aligned} \min \ & \sum_a t_a^0 \left(x_a + \gamma_a y_a \right) \\ \text{s.t. } & \text{Net-flow-Path} \\ & x_a = \sum_k \lambda_a^k x_a^k, \quad \forall a \\ & y_a = \sum_k \lambda_a^k y_a^k, \quad \forall a \\ & \lambda_a \in \Delta_n, \quad \forall a \end{aligned} \tag{14}$$

where $\gamma_a = \alpha(\beta+1)^{-1}/(c_a)^\beta$, which originates from the integral of road latency function (3) following the objective function of problem (6)

$$\int_0^{x_a} t_a(\theta) \mathrm{d}\theta = \sum_a t_a^0 \left[x_a + \frac{\alpha \, (x_a)^{\beta+1}}{(\beta+1) \, (c_a)^\beta} \right]$$

Because of convexity, at the optimal solution, only two adjacent weights can be activated (strictly positive), while the remaining ones are equal to 0.

Then we discuss how to select the break points x_a^k, $k = 1, \cdots, n$. The error introduced by approximating a nonlinear function using piecewise linear function is thoroughly studied in [42]. In our problem, function $y_a = (x_a)^\beta$ is sufficiently smooth, and the maximum of second-order derivative on interval $[0, x_{am}]$ is $y''_{am} = \beta(\beta-1)(x_{am})^{\beta-2}$. According to

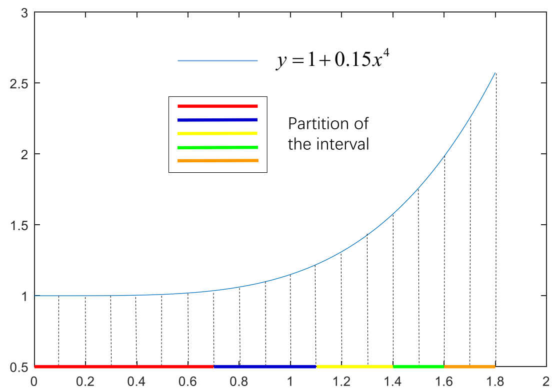


FIGURE 1. Non-uniform partition of interval.

Theorem 2 in [42], if we divide $[0, x_{am}]$ into

$$n \approx \frac{\sqrt{\beta(\beta-1)}(x_{am})^{\beta/2}}{4\sqrt{\varepsilon}} \tag{15}$$

segments with identical length, then the maximum absolute approximation error will be no greater than ε.

Because the typical value of β in the power function is around 4, the function value changes smoothly when x is close to 0, and the error in the rightmost segment beside x_{am} is the largest. To reduce the number of breakpoints and the dimension of λ_a, we resort to non-uniform partition. This is implemented by merging segments and checking the approximation error. The procedure is summarized in Algorithm 1. In view of the shape of the function, we start from the left and move to the right. In the end, the segments become shorter with the growth of x_a, as illustrated in Fig. 2

Algorithm 1 Adaptive Partition

1: Select an error tolerance $\varepsilon > 0$. Uniformly divide traffic flow interval $[0, x_{am}]$ into n segments labeled by $1, \cdots, n$, where n is given in (15). Iteration counter $k = 1$.

2: Find segment k and its endpoints x_l and x_r. Check the maximum approximation error

$$\epsilon = \max_{x \in [x_l, x_r]} \{L(x) - t_a(x)\}$$

where $L(x)$ is the linear function connecting $(x_l, t_a(x_l))$ and $(x_r, t_a(x_r))$. The maximum must be found in an interior point because the error is 0 at two endpoints.

3: If $\epsilon < 0.9\varepsilon$, merge the current interval with the next one; decrease the index numbers of remaining segments by 1, and go to step 2. If segment k is the rightmost one, terminate; otherwise update $k \leftarrow k+1$ and go to step 2.

The piecewise linear approximation approach does not rely on any specific structure of the latency function $t_a(x_a)$. As long as it is increasing, such as the Davidson function in [43] which considers a mandatory upper bound of traffic flow,

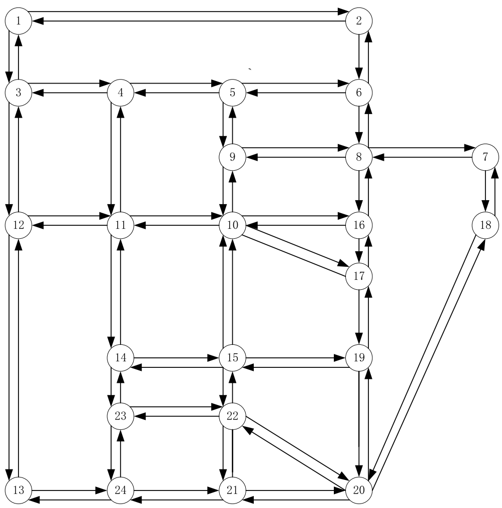


FIGURE 2. Sioux Falls network.

its integral is convex in x_a, and thus this approach remains valid.

C. AN ADAPTIVE PATH GENERATION ORACLE

Path information is needed during the setup of problem (6). Although the number of feasible paths connecting each O-D pair could be large, only a small fraction will be used. So we can incorporate only a subset of paths which are most likely to be activated. To this end, we start with an arbitrary subset of K^{rs} which includes one or only a few paths and solve a restricted TAP, where "restricted" means that the feasible region of TAP is smaller compared with the original TAP, because the majority of feasible paths are neglected. At the obtained UE pattern, if the travel time can be further reduced by switching to a new path, then the path is added to K^{rs}, and the TAP is solved again, until no one is willing to change their route. In such circumstance, the UE pattern also solves the original TAP.

This section streamlines the mathematical format for above procedure. For an O-D pair (r, s), vector $I^{rs} \in \mathbb{R}^{|N|}$ has two non-zero elements: 1 and -1 at the entries corresponding to the origin node r and the destination node s. Vector v consists of $|A|$ 0-1 variables. By the definition of node-link incidence matrix Λ, suppose $\Lambda v = I^{rs}$ holds, then those links corresponding to non-zero elements in v constitute a chain connecting r and s, i.e, an available path.

To identify the fastest path at a given traffic flow patten x^*, the travel time t_a^* for each link can be calculated via equation (3). The minimal travel time u_b^{rs} between (r, s) on present paths is readily available by comparison. To find a potentially better path, we solve the following MILP

$$
\begin{aligned}
u_c^{rs} = \min_{v} \quad & \sum_a t_a^* v_a \\
\text{s.t.} \quad & \Lambda v = I^{rs} \\
& v \in \mathbb{B}^{|A|}
\end{aligned} \tag{16}
$$

If $u_c^{rs} < u_b^{rs}$, the optimal solution v^* indicates a better path, which can reduce travel time and will be added in the path set K^{rs}. According to above discussions, a procedure for solving problem (6) without path enumeration is summarized in Algorithm 2.

Algorithm 2 Adaptive Path Generation

1: Find the initial fastest path connecting every O-D pair using free travel time t_a^0, $\forall a$, and then construct link-path incidence matrices Δ.
2: Solve the restricted TAP (6) with the current Δ; the UE is x^*, update the link travel time t_a^*, $\forall a$ and the minimal travel time u_b^{rs}, $\forall (r, s)$.
3: solve path generation subproblem (16) for each O-D pair. The optimal solution is v^* and the optimal value is u_c^{rs}. If $u_c^{rs} \geq u_b^{rs}$, $\forall (r, s)$, then there is no better path. Return the current UE solution x^*. Otherwise, if $\exists (r, s) : u_c^{rs} < u_b^{rs}$, then update $\Delta^{rs} \leftarrow [\Delta^{rs}, v^*]$ and Δ; go to step 2.

Because the number of paths is finite, Algorithm 2 must terminate in a finite number of iterations. In practice, it often converges very quickly because only a few paths are actually used. Therefore, the problem size of path-based formulation (6) is actually very small. To accelerate Algorithm 2, multiple paths can be used in the initiation step. Algorithm 2 implements the same function as the Dijkstra algorithm of shortest path. In a large network, when the origin and the destination are distant from each other, Algorithm 2 is more efficient.

IV. CASE STUDIES

We present case studies on systems with different scales. In function (3), $\alpha = 0.15$ and $\beta = 4.0$ are used for all roadway segments if there is no particular declaration, while t_0 depends on real system data. All simulations are implemented on a laptop with Intel i5-3210M CPU and 4 GB memory. MILP and SOCP are solved by CPLEX 12.8. The source code is available in [44].

To circumvent explicit enumeration of paths, SOCP model (12) and LP model (14) are embedded in the framework of Algorithm 2, while in the algorithm framework proposed in [6] (F-W method for short), the convergence criterion requires the change of objective values in two successive iteration should be less than 1% (in [6], this threshold is 5% and also tested, but the accuracy is not satisfactory). Among the SOCP model, the LP model and the F-W method, the first one offers the exact solution which is also unique due to the strict convexity; the second one approximates the objective function with an adjustable error tolerance; the last one tackles the exact traffic assignment model; however, the convergence criterion does not provide clear information on the optimality gap, which is defined as

$$\text{optimality gap} = \frac{|v - v^*|}{v^*}$$

where v^* is the optimal value of SOCP (12), and v is the optimal value of LP (14) or that offered by the F-W method.

TABLE 1. Results of Sioux Falls network.

Normal case	Number of iterations	Optimal value	Solver time (s) [1]	Maximum relative error	Standard deviation
SOCP	5	51.69	4.77	0	0
LP	5	51.69	3.96	0.97%	0.039
F-W (1%)	88	51.81	59.96	5.17%	0.074
F-W (5%)	16	52.74	9.51	21.2%	0.397
Peak case	**Number of iterations**	**Optimal value**	**Solver time (s)**	**Maximum relative error**	**Standard deviation**
SOCP	6	150.07	5.33	0	0
LP	6	150.09	4.58	1.54%	0.054
F-W (1%)	101	151.07	54.24	3.09%	0.171
F-W (5%)	34	156.24	18.1	17.51%	0.898
Off-peak case	**Number of iterations**	**Optimal value**	**Solver time (s)**	**Maximum relative error**	**Standard deviation**
SOCP	5	31.99	3.91	0	0
LP	5	31.99	3.50	1.45%	0.032
F-W (1%)	55	32.02	29.48	3.53%	0.056
F-W (5%)	17	32.09	9.61	6.55%	0.136

To compare solution quality, we use indicators

$$\text{maximum relative error} = \max_a \left\{ \frac{|x_a - x_a^*|}{x_a^*} \right\}$$

and

$$\text{standard deviation} = \sqrt{\frac{1}{|A|} \sum_a \left(x_a - x_a^*\right)^2}$$

where x_a^* is the optimal solution of SOCP (12), and x_a is the optimal solution of LP (14) or that offered by the F-W method.

A. SIOUX FALLS NETWORK

In this section, the renowned Sioux Falls network shown in Fig. 2 which consists of 24 nodes and 76 links is tested. The O-D demand matrix has a dimension of 24×24. In the normal case, system data are identical to those in [6], and $x_{am} = 2c_a$ is adopted in equation (15). In peak (off-peak) case, traffic demand q^{rs} is multiplied by 1.5 (0.75), and x_{am} in (15) is adjusted accordingly. Results are compared in Table 1.

In all three cases, Algorithm 2 converges in 5 or 6 iterations, indicating that its performance is robust against load conditions. Computation times of SOCP and LP do not differ significantly. Since the piecewise linear function (13) overestimates the original convex function (3), the optimal value of LP (14) should be always no less than that of (12), which is the true optimum. Nevertheless, even in the case of peak hour, the optimality gap is smaller than 0.02%. At the optimal solution, the maximum relative error of link flow is kept below 2.0%, which is satisfactory in practical usage.

[1] Optimization problem setup time is not counted. The same applies for all "solver time" throughout the paper.

For the F-W method, from Table 1 we can observe that when the convergence tolerance is set to 1%, the optimality gap and approximation error are generally small. However, dozens (even one hundred) of iterations are needed before the F-W algorithm could converge, and the CPU time is one order of magnitude longer than those of SOCP and LP. When the convergence tolerance is set to 5%, the F-W algorithm converges quickly, but link flow solutions still exhibit errors as large as 17% and 21% in the peak and normal cases. These observations imply that F-W algorithm converges slowly around the optimal solution, and it is usually difficult to improve efficiency and accuracy at the same time, especially when the demand is high.

We also test some general-purpose NLP solvers including the interior-point algorithm based IPOPT [45], the sequential quadratic programming based SNOPT [46], the KNITRO solver [47] that integrates interior-point algorithm, active set algorithm, and sequential quadratic programming algorithm, and a global solver SCIP [48]. Results are given in Table 2.

TABLE 2. Comparison with NLP solvers.

Number of O-D pair		10	20	30	38
Solver time	IPOPT	0.041	0.715	18.94	failed
	SNOPT	0.125	0.751	25.56	failed
	SCIP	1.096	6.503	811.92	6130.8
	KNITRO	0.121	0.776	27.39	failed
	SOCP	0.224	0.257	0.276	0.332
	LP	0.064	0.078	0.101	0.105

All the above solvers failed to solve TAP (6) with the complete set of O-D pairs. We reduce the number of active O-D pairs $|D_S^R|$ and change it from 10 to 40. As we can see from Table 2, IPOPT is the most efficient among the four candidates. If there are only 10 O-D pairs, all methods have comparable efficiency. However, general-purpose NLP solvers are not scalable; their computation times grow quickly with the increase of D_S^R. When $D_S^R = 38$, only SCIP could return a solution in more than 100 minutes. In contrast, SOCP and LP methods successfully solve all instances within less than a second. On this account, directly solving problem (6) as an NLP is not wise although it is convex.

B. BEIJING NETWORK

In this section, the transportation network in the city of Beijing is considered. Ring expressways and arterial roads within the fifth ring are taken into account. The free travel time is set as $t_0 = L/v$, where L is the length of the arc, and v is the speed limit, which is 80km/h for ring expressways and 60km/h for arterial roads. Network topology is built in [49] and shown in Fig. 3. It consists of 222 nodes, 943 links and 2000 O-D pairs. We consider the evening peak from 5:30 to 7:30 p.m. Complete system data can be found in [44]. Similar experiments are conducted. Results are listed in Table 3.

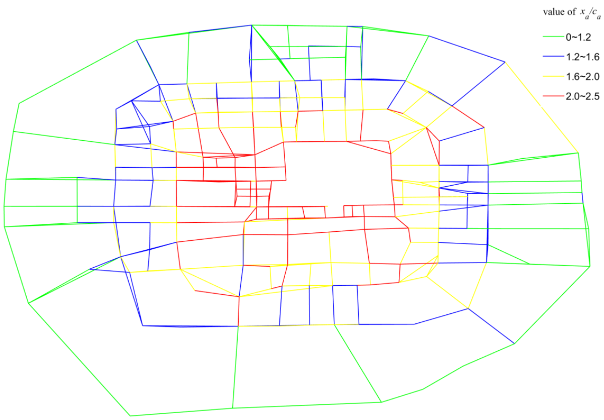


FIGURE 3. Beijing network.

TABLE 3. Results of Beijing network.

Normal case	Number of iterations	Optimal value	Solver time (s)	Maximum relative error	Standard deviation
SOCP	14	1446.5	775	0	0
LP	12	1446.9	644	4.12%	0.181
F-W (1%)	174	1449.4	21494	6.44%	0.159
F-W (5%)	53	1459.5	6517	18.33%	0.518
Peak case	Number of iterations	Optimal value	Solver time	Maximum relative error	Standard deviation
SOCP	16	3788.6	924	0	0
LP	15	3789.4	776	3.78%	0.232
F-W (1%)	192	3808.7	21938	5.26%	0.236
F-W (5%)	48	3920.9	5419	21.53%	0.934
Off-peak case	Number of iterations	Optimal value	Solver time	Maximum relative error	Standard deviation
SOCP	11	937.93	657	0	0
LP	10	937.98	570	4.05%	0.152
F-W (1%)	155	938.74	18346	5.31%	0.091
F-W (5%)	48	941.29	5354	21.22%	0.318
1500 O-D pairs	Number of iterations	Optimal value	Solver time	Maximum relative error	Standard deviation
SOCP	11	957.37	557	0	0
LP	11	957.58	481	4.71%	0.124
F-W (1%)	203	958.22	16832	6.22%	0.076
F-W (5%)	47	961.11	3838	26.33%	0.263
1000 O-D pairs	Number of iterations	Optimal value	Solver time	Maximum relative error	Standard deviation
SOCP	10	600.42	302	0	0
LP	9	600.51	252	3.65%	0.112
F-W (1%)	268	600.58	14994	5.33%	0.138
F-W (5%)	56	601.13	3081	29.52%	0.190

The same trends are observed in this table. LP and F-W (1% tolerance) offer similar link flow solutions with a maximum relative error less than 5% and 7%, respectively, compared with the exact one provided by the SOCP model. LP is slightly faster than SOCP, and one order of magnitude faster than F-W (1% tolerance). F-W (5% tolerance) is not accurate enough and less efficient than SOCP and LP. The maximum relative error of link flow is larger than 20%. We further change the number of O-D pairs in the system. F-W (1% tolerance) takes more iterations when $|D_S^R|$ is reduced, while the convergence rates of SOCP, LP and F-W (5% tolerance) are not significantly influenced. The maximum relative error of F-W (5% tolerance) shows an increasing trend with the decrease of $|D_S^R|$.

TABLE 4. Performances on Other networks.

Network		Anaheim	BMC	BPC	BT
Information on system scales	N_D	38×38	36×36	38×38	26×26
	$\|L\|$	914	871	749	766
	$\|N\|$	416	398	352	361
Solver time (s)	SOCP	1213	1014	967	806
	LP	1036	871	834	691
	F-W (1%)	22652	21018	19261	16394
	F-W (5%)	4757	4657	4007	3426

C. OTHER SYSTEMS

Above methods are tested on Anaheim network in the U.S., and also in Berlin-Mitte-Center (BMC) network, Berlin-Prenzlauerberg-Center (BPC) network, as well as Berlin-Tiergarten (BT) network in Germany. System data are available in [50]. Information on the scale of each system and the computation time of each method are summarized in Table 4. Results show that SOCP and LP have similar efficiency, which is one order of magnitude faster than F-W (1% tolerance) and almost 4 times faster than F-W (5% tolerance).

V. CONCLUSION

This paper studies efficient computation of the user-optimal traffic assignment problem. Experiments demonstrate that solving such a problem using general-purpose nonlinear programming solvers is not scalable. The popular F-W method has some advantages. The solution sequence generated by the algorithm is decreasing, and any solution in the sequence is feasible. However, it is difficult to meet the requirements on accuracy and efficiency simultaneously. The proposed SOCP and LP methods outperform F-W method in both aspects. The key idea is to represent the nonlinear but convex objective function via second-order cones or a piecewise linear function, so as to utilize state-of-the-art conic and linear programming solvers, in which the structured property of cones and polyhedra are better exploited. The acceleration is more than one order of magnitude compared to F-W method with a similar level of accuracy.

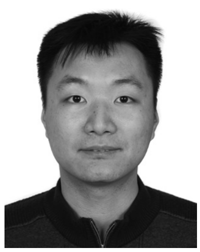
REFERENCES

[1] J. G. Wardrop, "Some theoretical aspects of road traffic research," *Proc. Inst. Civil Eng.*, vol. 1, no. 3, pp. 325–362, May 1952.

[2] A. Haurie and P. Marcotte, "On the relationship between Nash—Cournot and Wardrop equilibria," *Networks*, vol. 15, no. 3, pp. 295–308, 1985.

[3] M. Beckmann, C. McGuire, and C. Winsten, *Studies in the Economics of Transportation*. New Haven, CT, USA: Yale Univ. Press, 1956.

[4] Y. Sheffi, *Urban Transportation Network: Equilibrium Analysis With Mathematical Programming Methods*. Upper Saddle River, NJ, USA: Prentice-Hall, 1985.
[5] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Res. Logistics Quart.*, vol. 3, nos. 1–2, pp. 95–110, 1956.
[6] L. J. LeBlanc, E. K. Morlok, and W. P. Pierskalla, "An efficient approach to solving the road network equilibrium traffic assignment problem," *Transp. Res.*, vol. 9, no. 5, pp. 309–318, Oct. 1975.
[7] W. I. Zangwill, *Nonlinear Programming: A Unified Approach*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1969.
[8] M. Fukushima, "A modified Frank–Wolfe algorithm for solving the traffic assignment problem," *Transp. Res. B, Methodol.*, vol. 18, no. 2, pp. 169–177, Apr. 1984.
[9] L. J. LeBlanc, R. V. Helgason, and D. E. Boyce, "Improved efficiency of the Frank–Wolfe algorithm for convex network programs.," *Transp. Sci.*, vol. 19, no. 4, pp. 445–462, Nov. 1985.
[10] A. Weintraub, C. Ortiz, and J. González, "Accelerating convergence of the Frank–Wolfe algorithm," *Transp. Res. B, Methodol.*, vol. 19, no. 2, pp. 113–122, Apr. 1985.
[11] T. Larsson and M. Patriksson, "Simplicial decomposition with disaggregated representation for the traffic assignment problem," *Transp. Sci.*, vol. 26, no. 1, pp. 4–17, Feb. 1992.
[12] R. Jayakrishnan, W. T. Tsai, J. N. Prashker, and S. Rajadhyaksha, "A faster path-based algorithm for traffic assignment," *Transp. Res. Rec.*, vol. 1443, pp. 75–83, Oct. 1994.
[13] D.-H. Lee, Y. Nie, and A. Chen, "A conjugate gradient projection algorithm for the traffic assignment problem," *Math. Comput. Model.*, vol. 37, nos. 7–8, pp. 863–878, Apr. 2003.
[14] S. Dafermos, "Traffic equilibrium and variational inequalities," *Transp. Sci.*, vol. 14, no. 1, pp. 42–54, Feb. 1980.
[15] C. S. Fisk and D. E. Boyce, "Alternative variational inequality formulations of the network equilibrium-travel choice problem," *Transp. Sci.*, vol. 17, no. 4, pp. 454–463, Nov. 1983.
[16] A. Nagurney, *Network Economics: A Variational Inequality Approach*. Dordrecht, The Netherlands: Kluwer, 1993.
[17] D. P. Bertsekas and E. M. Gafni, "Projection methods for variational inequalities with application to the traffic assignment problem," in *Non-differential and Variational Techniques in Optimization*. Berlin, Germany: Springer, 1982, pp. 139–159.
[18] T. L. Friesz and K. Han, "The mathematical foundations of dynamic user equilibrium," *Transp. Res. B, Methodol.*, vol. 126, pp. 309–328, Aug. 2019.
[19] K. Han, G. Eve, and T. L. Friesz, "Computing dynamic user equilibria on large-scale networks with software implementation," *Netw. Spatial Econ.*, vol. 19, no. 3, pp. 869–902, 2019.
[20] A. Zhang, B. Yue, C. Wang, D. Shao, and X. Zhang, "Modeling the enveloping macroscopic fundamental diagram based on the traffic assignment with deterministic user equilibrium," *IEEE Access*, vol. 7, pp. 69776–69794, 2019.
[21] T.-G. Wang, C. Xie, J. Xie, and "Path-constrained traffic assignment: A trip chain analysis under range anxiety," *Transp. Res. C, Emerg. Technol.*, vol. 68, pp. 447–461, Jul. 2016.
[22] X. T. R. Kong, S. X. Xu, M. Cheng, and G. Q. Huang, "IoT-enabled parking space sharing and allocation mechanisms," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1654–1664, Oct. 2018.
[23] F. He, Y. Yin, and S. Lawphongpanich, "Network equilibrium models with battery electric vehicles," *Transp. Res. B, Methodol.*, vol. 67, pp. 306–319, Sep. 2014.
[24] H. S. Ramadan, M. Becherif, and F. Claude, "Energy management improvement of hybrid electric vehicles via combined GPS/rule-based methodology," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 586–597, Apr. 2017.
[25] W. Wei, L. Wu, J. Wang, and S. Mei, "Network equilibrium of coupled transportation and power distribution systems," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6764–6779, Nov. 2018.
[26] H. Xu, J.-S. Pang, F. Ordóñez, and M. Dessouky, "Complementarity models for traffic equilibrium with ridesharing," *Transp. Res. B, Methodol.*, vol. 81, pp. 161–182, Nov. 2015.
[27] O. Bahat and S. Bekhor, "Incorporating ridesharing in the static traffic assignment model," *Netw. Spatial Econ.*, vol. 16, no. 4, pp. 1125–1149, Dec. 2016.
[28] J. Chen, K. H. Low, Y. Yao, and P. Jaillet, "Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 901–921, Jul. 2015.
[29] W. Wang, X. Liu, L. Ding, G. Gao, and H. Zhang, "Stochastic network user equilibrium and traffic system evolution based on reference-dependent utility theory," *IEEE Access*, vol. 7, pp. 19866–19880, 2019.
[30] J. Liu, Z. He, and S. Ma, "An N-path logit-based stochastic user equilibrium model," *IEEE Access*, vol. 6, pp. 20971–20986, 2018.
[31] L. Zhang, J. Liu, B. Yu, and G. Chen, "A dynamic traffic assignment method based on connected transportation system," *IEEE Access*, vol. 7, pp. 65679–65692, 2019.
[32] T. Lu, Z. Yang, D. Ma, and S. Jin, "Bi-level programming model for dynamic reversible lane assignment," *IEEE Access*, vol. 6, pp. 71592–71601, 2018.
[33] B. L. Ye, W. Wu, L. Li, and W. Mao, "A Hierarchical model predictive control approach for signal splits optimization in large-scale urban road networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2182–2192, Aug. 2016.
[34] B.-L. Ye, W. Wu, K. Ruan, L. Li, T. Chen, H. Gao, and Y. Chen, "A survey of model predictive control methods for traffic signal control," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 3, pp. 623–640, May 2019.
[35] J. Balsa-Barreiro, L. Ambuühl, M. Menéndez, and A. Pentland, "Mapping time-varying accessibility and territorial cohesion with time-distorted maps," *IEEE Access*, vol. 7, pp. 41702–41714, 2019.
[36] J. Zhang, S. Pourazarm, C. G. Cassandras, and I. C. Paschalidis, "Data-driven estimation of origin-destination demand and user cost functions for the optimization of transportation networks," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9680–9685, Jul. 2017.
[37] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. Philadelphia, PA, USA: SIAM, 2001.
[38] *Traffic Assignment Manual*, Bur. Public Roads, U.S. Dept. Commerce, Washington, DC, USA, 1964.
[39] Z. Luo, J. Pang, and D. Ralph, *Mathematical Programs With Equilibrium Constraints*. Cambridge, U.K.: Cambridge Univ. Press, 1996.
[40] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra Appl.*, vol. 284, nos. 1–3, pp. 193–228, Nov. 1998.
[41] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[42] C. L. Frenzen, T. Sasao, and J. T. Butler, "On the number of segments needed in a piecewise linear approximation," *J. Comput. Appl. Math.*, vol. 234, no. 2, pp. 437–446, May 2010.
[43] K. B. Davidson, "A flow travel time relationship for use in transportation planning," in *Proc. Austral. Road Res. Board*, Melbourne, VIC, Australia, 1966, pp. 183–194.
[44] (Mar. 2019). *System Data of Beijing Network*. [Online]. Available: https://github.com/hlx17/Beijing_TransportationNetworks
[45] A. Wächter, and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, Mar. 2006.
[46] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Rev.*, vol. 47, no. 1, pp. 99–131, 2005.
[47] R. H. Byrd, J. Nocedal, and R. A. Waltz, "KNITRO: An integrated package for nonlinear optimization," in *Large-Scale Nonlinear Optimization*. Boston, MA, USA: Springer, 2006, pp. 35–60.
[48] T. Achterberg, "SCIP: Solving constraint integer programs," *Math. Programm. Comput.*, vol. 1, no. 1, pp. 1–41, Jul. 2009.
[49] S. Shao, W. Guan, and J. Bi, "Electric vehicle-routing problem with charging demands and energy consumption," *IET Intell. Transp. Syst.*, vol. 12, no. 3, pp. 202–212, Apr. 2018.
[50] (Mar. 2019). *System Data of Additional Testing Networks*. [Online]. Available: https://github.com/bstabler/TransportationNetworks



WEI WEI (SM'18) received the B.Sc. and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China, in 2008 and 2013, respectively, where he was a Postdoctoral Research Associate, from 2013 to 2015.

He was a Visiting Scholar with Cornell University, Ithaca, NY, USA, in 2014, and with Harvard University, Cambridge, MA, USA, in 2015. He is currently an Associate Professor with Tsinghua University. His research interests include applied optimization and energy system economics.

LONGXIAN HU received the B.S. degree in electrical engineering from North China Electric Power University, Beijing, China, in 2014. He is currently pursuing the M.S. degree with Tsinghua University, Beijing.

His research interests include optimization theory and intelligent transportation systems.

QIUWEI WU (M'08–SM'15) received the B.Eng. and M.Eng. degrees in power system and its automation from the Nanjing University of Science and Technology, Nanjing, China, in 2000 and 2003, respectively, and the Ph.D. degree in power system engineering from Nanyang Technological University, Singapore, in 2009.

He was a Senior Research and Development Engineer with VESTAS Technology R&D Singapore Pte. Ltd., from March 2008 to October 2009. He has been with the Department of Electrical Engineering, Technical University of Denmark (DTU), since November 2009, held a postdoctoral position, from November 2009 to October 2010, an Assistant Professor, from November 2010 to August 2013, has been an Associate Professor, since September 2013. He was a Visiting Scholar with the Department of Industrial Engineering and Operations Research (IEOR), University of California at Berkeley, from February 2012 to May 2012, funded by Danish Agency for Science, Technology, and Innovation (DASTI), Denmark. He was a Visiting Professor named by Y. Xue, an Academician of the Chinese Academy of Engineering, Shandong University, China, from November 2015 to October 2017. He was a Visiting Scholar with the School of Engineering and Applied Science, Harvard University, in 2018. His research interests include power system operation and control with high renewables, including wind power modeling and control, active distribution networks, and integrated energy systems.

Dr. Wu is an Editor of the IEEE Transactions on Smart Grid and the IEEE Power Engineering Letters. He is also an Associate Editor of the *International Journal of Electrical Power and Energy Systems*, *Journal of Modern Power Systems and Clean Energy*, *IET Renewable Power Generation*, and *IET Generation, Transmission & Distribution*.

TAO DING (SM'19) received the B.S.E.E. and M.S.E.E. degrees from Southeast University, Nanjing, China, in 2009 and 2012, respectively, and the Ph.D. degree from Tsinghua University, Beijing, China, in 2015.

He was a Visiting Scholar with the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN, USA. He is currently an Associate Professor with the State Key Laboratory of Electrical Insulation and Power Equipment, School of Electrical Engineering, Xi'an Jiaotong University, Xi'an, China. He has authored or coauthored more than 60 technical articles and authored by "Springer Theses" recognizing outstanding Ph.D. research around the world and across the physical sciences—Power System Operation With Large Scale Stochastic Wind Power Integration. His current research interests include electricity markets, power system economics and optimization methods, and power system planning and reliability evaluation.