WORKING PAPER

ITLS-WP-14-05

Two new methods for solving the path-based stochastic user equilibrium problem

By
Bojian Zhou[*], Michiel Bliemer, Michael Bell and Jie He[*]

[*] School of Transportation, Southeast University, Nanjing, China

March 2014

ISSN 1832-570X

INSTITUTE of TRANSPORT and LOGISTICS STUDIES
The Australian Key Centre in
Transport and Logistics Management

The University of Sydney
Established under the Australian Research Council's Key Centre Program.

NUMBER: Working Paper ITLS-WP-14-05

TITLE: **Two new methods for solving the path-based stochastic user equilibrium problem**

ABSTRACT: In this paper, we present two new methods for the path-based logit stochastic user equilibrium problem, and investigate their convergence properties. First, a two level partial linearization method is proposed. Second, a dual method is developed. Both of these two methods use second order approximation of the objective function. Our novel methods are compared to Damberg's partial linearization method (Damberg, 1996), which is known to be one of the best performing methods. Numerical results on the Sioux Falls and Winnipeg networks show that, if properly scaled, our new methods can significantly improve the performance of Damberg’s method.

KEY WORDS: ***Stochastic user equilibrium, two level, partial linearization, Lagrange dual, method of successive averages***

AUTHORS: **Zhou, Bliemer, Bell and He**

CONTACT: INSTITUTE OF TRANSPORT AND LOGISTICS STUDIES (C13)

The Australian Key Centre in Transport and Logistics Management

The University of Sydney NSW 2006 Australia

Telephone: +612 9114 1824

E-mail: business.itlsinfo@sydney.edu.au

Internet: http://sydney.edu.au/business/itls

DATE: March 2014

1. Introduction

The stochastic user equilibrium (SUE) static traffic assignment problem has been extensively studied in the past decades. This problem was defined first by Daganzo and Sheffi (1977), who generalized the user equilibrium (UE) principle of Wardrop (1952). The SUE principle states that no traveler can reduce his or her own perceived travel cost by unilaterally changing routes.

In the literature, two stochastic models are of particular interest: The probit model (Daganzo and Sheffi 1977; Sheffi and Powell 1982) and the logit model (Dial 1971; Fisk 1980; Chen and Alfa 1991;Leurent 1997). The probit model, though behaviorally more appealing, is impractical, because it requires Monte Carlo simulation techniques or complete path enumeration. The logit model, however, has enjoyed much greater attention. The reason is that the logit model can be explicitly formed and solved, and therefore it is very useful not only for theoretical investigation(Meng and Wang, 2008; Szeto et al. 2010; Szeto et al., 2011), but also for practical implementation (Cascetta and Papola, 2008; Castillo et al., 2008; Liu and Luo, 2012). In this study, we will concentrate on the solution algorithm of the logit model.

In general, solution algorithms for the logit-based SUE problem can be divided into two categories. In the first category are link-based algorithms, which do not require explicit path enumeration. It only assumes an implicit path choice set, such as the use of all efficient paths (Dial, 1971; Maher, 1998), or all cyclic and acyclic paths (Bell, 1995; Akamatsu,1996). In the other category are path-based algorithms, which require explicit choice of a subset of feasible paths prior to or during the assignment. Therefore, it makes the path choice set more realistic from a behavioural standpoint. In the literature, a large number of path choice set generation methods are proposed by different authors, see for example Ben-Akiva et al. (1984), Azevedo et al. (1993), De la Barra et al. (1993), Cascetta et al. (1996). Some path set generation methods are constrained deterministic enumeration methods using so-called branch-and-bound decision rules to add paths to the choice set (Hoogendoorn-Lanser, 2005; Prato and Bekhor, 2006), while others are stochastic methods which repeatedly add shortest paths to the choice set using randomized link costs (Fiorenzo-Catalano 2007; Bliemer and Taale, 2006).

This paper focuses on path-based solution algorithms to the logit SUE problem. To our best knowledge, the partial linearization method proposed by Damberg et al. is one of the most efficient algorithms currently existing for the path-based SUE problem (Damberg 1996; Bekhor and Toledo 2005). Damberg's method proceeds as follows. Given a feasible path flow, a subproblem is constructed by partially linearizing the objective function of the logit SUE problem, and an auxiliary path flow is obtained by solving this subproblem. A one dimensional minimization is then performed on the line segment between the current and the auxiliary path flow solutions, resulting in a new feasible solution. The procedure is repeated until the solution is optimal.

As we know, Damberg's partial linearization method generalizes the method of Frank and Wolfe applied to the logit SUE problem, in that the auxiliary problem is the result of an approximation of the objective function by means of a partial linearization only, and has been observed to be more efficient (Boyce 1984; LeBlanc and Farhangian 1981). However, note that both methods are based on the same algorithmic principle, in that they use a first order linear approximation of the objective function. Hence, both of them may suffer from slow convergence. In order to make some improvements of Damberg's method, we propose two second order approximation methods to solve [TAP-SUE]. This is the motivation of this paper. First, a two level partial linearization method is developed. In this method, we propose a second order partial linearization method for the outer level iteration phase, in which a linearly-constrained entropy maximization subproblem with quadratic cost is created, and a first order partial linearization method for the inner level iteration phase, in which the subproblem is approximately solved. Second, a dual method is presented. In this method, we formulate a Lagrange dual problem for the SUE model and use the scaled steepest ascend method to solve it. These methods are compared with Damberg's method on different networks.

In order to have a fair comparison of different methods and concentrate on the main ideas, throughout this paper, a pre-generated path set was used, obtained from a route choice set generation algorithm (Bekhor et al. 2006). Behaviourally, it has the advantage of explicitly identifying those paths that are most likely to be used and also allows greater flexibility to include path-specific attributes that may not be obtainable directly from the link attributes.

The rest of this paper is organized as follows: Section 2 reviews the path-based logit SUE problem and algorithms developed to solve it. Section 3 presents the two-level iteration method with convergence results. Section 4 discusses the basic properties of the Lagrange dual model of the logit SUE problem and presents the dual method. In section 5, a proper measure to compare different methods is given. Section 6 compares the two proposed methods with Damberg's partial linearization method, and discusses the numerical results. Section 7 provides conclusions and suggests some important future research directions.

2. Path based logitstochastic user equilibrium problem

2.1. Basic concepts

Consider a transportation network $G(N,A)$, where N is the set of nodes and A is the set of directed links, respectively. For each link $a \in A$ there is a link cost function $t_a(f_a)$ that depending on link flow f_a. As a result of congestion, the link cost function is assumed to be positive, differentiable and strictly increasing. For certain origin-destination(OD)pairs $(p,q) \in C \subset N \times N$,there is a given positive demand d_{pq} of flow. We define the set of paths from origin p to destination q by R_{pq} and denote the path flow and path travel cost on path $r \in R_{pq}$ by h_{pqr} and c_{pqr}, respectively. We will also use the notations $\boldsymbol{f}$, $\boldsymbol{t}$, $\boldsymbol{h}$, and $\boldsymbol{c}$, for the vectors of link flow, link cost, path flow and path cost, respectively. Further define $\Delta = (\delta_{pqra})$ as the link-path incidence matrix, where $\delta_{pqra} = 1$, if path $r \in R_{pq}$ contains link a, and $\delta_{pqra} = 0$ otherwise. With this incidence matrix Δ , we have the relationship between link-flow and path-flow, as $\boldsymbol{f} = \Delta \boldsymbol{h}$, and the relationship between link-cost and path-cost as $\boldsymbol{c} = \Delta^T \boldsymbol{t}$.

In SUE assignment, the path flow is determined by $h_{pqr} = d_{pq} P_{pqr}$. where P_{pqr} is the probability that traveler traverses through path $r \in R_{pq}$. As shown in Sheffi (1985), when the random term of discrete route choice satisfies a Gumbel or Normal distribution, the route choice probability can be described as multinomial logit or probit, respectively. For the logit model, the route choice probability is defined by:

$$h_{pqr} = d_{pq} \frac{\exp\left(-\theta c_{pqr}\right)}{\sum_{l \in R_{pq}} \exp\left(-\theta c_{pql}\right)} \quad \forall r \in R_{pq} , \quad \forall (p,q) \in C \tag{1}$$

where the parameter θ reflects an aggregate measure of people's perception of travel costs. When θ is large, their route choice becomes more deterministic, while a small value makes travellers more uncertain about the travel costs such that route choice behavior becomes more random. It is well-known that the logit model above cannot handle path overlap, which is why several authors have proposed alternative formulations such as the C-logit or path-size logit model. For an overview of different network route choice models, see Bliemer and Bovy (2008).

The logit SUE model can be formulated as the following convex optimization problem:

[TAP-SUE]

$$\min \; Z(h) = \sum_{a \in A} \int_0^{f_a} t_a(s)ds + \frac{1}{\theta} \sum_{(p,q)\in C} \sum_{r \in R_{pq}} h_{pqr} \ln h_{pqr} \tag{2}$$

$$\sum_{r \in R_{pq}} h_{pqr} = d_{pq}, \quad \forall (p,q) \in C \tag{3}$$

$$h_{pqr} \geq 0, \; \forall r \in R_{pq}, \;\; \forall (p,q) \in C \tag{4}$$

$$\sum_{(p,q)\in C} \sum_{r \in R_{pq}} \delta_{pqra} h_{pqr} = f_a, \;\; \forall a \in A \tag{5}$$

where the (entropy) expression $h_{pqr} \ln h_{pqr}$ is defined as zero at $h_{pqr} = 0$.This formulation was first proposed by Fisk (1980), who showed that its optimal solution satisfies logit choice formula (1). In this paper we consider the original SUE formulation by Fisk (1980), but we note that it can be extended to include for example the C-logit model by adding the term $\sum_{(p,q)} \sum_r h_{pqr} \Psi_{pqr}$ to objective function (2) (see e.g., see Xu et al., 2012), where Ψ_{pqr} is a path commonality factor described in Cascetta et al. (1996).

Some properties of the [TAP-SUE] are summarized in the following proposition.

Proposition 1. The objective of [TAP-SUE] is a sum of a convex function and a strictly convex (entropy) function over the feasible set. Thus the problem [TAP-SUE] is strictly convex. There is a unique optimal path flow solution h^*.

Proof. See Evans (1973) Theorem 1. ■

2.2 Review of previous path-based solution methods

The Method of Successive Averages (MSA) developed by Sheffi and Powell(1982) was the first algorithm applied to solve the SUE problem. In the MSA process, the link costs are calculated by the current link flows. A stochastic network loading procedure is then performed to produce an auxiliary link flow pattern. The search direction is obtained by the difference between the auxiliary link flow and the current link flow. The step size is predetermined by a descent sequence with respect to the iterations. Although the MSA algorithm is designed to solve the link based SUE problems, it can also be applied to the path based SUE problems if the path choice set is predetermined.

Chen and Alfa (1991) developed an algorithm that uses the same search direction as the MSA algorithm, but the step size is computed by minimizing Fisk's (1980) objective function. However, this computation requires an inverse of a link-path incidence matrix, which makes the algorithm impractical for large networks.

Bell et al. (1993) presented a modified Frank-Wolfe algorithm. In this algorithm, the link costs are augmented to allow for the contribution of link flows to the entropy component of Fisk's objective function, and paths are generated by a shortest path algorithm on the basis of the augmented link costs. An iterative balancing procedure is embedded in the path generation scheme to find dual variables for the entropy maximizing sub-problem, which are added to the link costs.

Damberg et al. (1996) used a combination of the disaggregate simplicial decomposition method and the partial linearization method to solve the SUE problem. The algorithm alternates between two phases:(1) given known subsets of all the paths in the network, a restricted master problem is solved using the partial linearization descent method, and (2) at the solution to this restricted master problem the current path set is augmented by the generation of new paths through a column generation procedure. The algorithm terminates when no new paths are obtained from the column generation procedure. Different strategies of generating paths are also discussed in Damberg's paper.

Bekhor and Toledo (2005) proposed the Gradient Projection (GP) algorithm for the SUE problem. In this algorithm, the working path set is generated prior to the traffic assignment. The search direction is generated by projecting the gradient of the objective function on a linear manifold of the active constraints. The scaling matrix is chosen as the diagonal of the Hessian. A line search is performed along this direction, yielding the next iteration point. Bekhor and Toledo (2005) showed that the method of Damberg et al. (1996) is as efficient or even more efficient than the GP algorithm. Therefore, in this paper we will use the method described in Damberg et al. as the method for comparison.

3. A two level partial linearization method for the logit SUE problem

In this section, we propose a two-level partial linearization method for solving [TAP-SUE]. This method consists of two phases: The outer level iteration phase applies a second order partial linearization method to [TAP-SUE], and creates a linearly-constrained entropy maximization subproblem with quadratic cost. The inner level iteration phase uses a first order partial linearization method to solve the subproblem approximately. We also discuss convergence properties of this method.

3.1 The outer level iteration phase

We first describe the outer level iteration of the two level partial linearization method. Rewrite the objective function of [TAP-SUE] as follows

$$Z(h) = Z_1(h) + Z_2(h), \tag{6}$$

where $Z_1(h) = \sum_a \int_0^{f_a} t_a(s)ds$ and $Z_2(h) = \theta^{-1} \sum_{(p,q)} \sum_r h_{pqr} \ln h_{pqr}..$

At iteration k, let $h^{(k)}$ be a given feasible path flow vector. Partially linearizing the objective function $Z(h)$ by using a second order approximate of the first term $Z_1(h)$, we can obtain

$$Z(h) \square Z_1(h^{(k)}) + C(h^{(k)})(h - h^{(k)}) + \frac{1}{2}(h - h^{(k)})^T B(h^{(k)})(h - h^{(k)}) + Z_2(h), \tag{7}$$

where $C(h^{(k)}) = \nabla Z_1(h^{(k)})$ is the gradient of $Z_1(h)$ at $h^{(k)}$, and $B(h^{(k)})$ is a positive definite matrix.

The gradient of $Z_1(h^{(k)})$ is given by

$$\nabla Z_1(h^{(k)}) = \Delta^T \boldsymbol{t}, \tag{8}$$

where Δ is the link-path incidence matrix, and $\boldsymbol{t} = [t_a(f_a^{(k)})]_{a\in A}^T$ is the link travel time vector.

To discuss what matrix for $B(h^{(k)})$ is most suitable for the two level partial linearization method, we first calculate the Hessian matrix of the objective function $Z(h)$ at $h^{(k)}$

$$\nabla^2 Z(h^{(k)}) = \nabla^2 Z_1(h^{(k)}) + \nabla^2 Z_2(h^{(k)}) = \Delta^T(\nabla \boldsymbol{t})\Delta + \frac{1}{\theta} diag(h_{pqr}^{-1}), \tag{9}$$

where $\nabla \boldsymbol{t}$ is the gradient of the vector valued function $\boldsymbol{t} = [t_a(f_a^{(k)})]_{a\in A}^T$, and $diag(h_{pqr}^{-1})$ is a diagonal matrix whose diagonal elements are h_{pqr}^{-1}. Since the cost of a link only depends on the flow of that link, it follows that $\nabla \boldsymbol{t}$ is a diagonal matrix with elements $t_a'(f_a^{(k)})$.

In practice, choosing $B(h^{(k)})$ as a general non-diagonal positive definite matrix (e.g., the Hessian of the objective function $Z(h)$) is usually not computationally efficient, since the Hessian vector product $\nabla^2 Z(h^{(k)})(h-h^{(k)})$ is expensive to calculate. A more popular choice is to let $B(h^{(k)})$ be a diagonal (strongly) positive definite matrix. For the two-level partial linearization method, three forms of $B(h^{(k)})$ can be potentially used.

Form 1: $B(h^{(k)})$ is the diagonal of the Hessian of $Z_1(h^{(k)})$, i.e.,

$$B(h^{(k)}) = diagExtract[\nabla^2 Z_1(h^{(k)})] = diagExtract[\Delta^T(\nabla \boldsymbol{t})\Delta] \tag{10}$$

Form 2: $B(h^{(k)})$ is the diagonal of the Hessian of $Z_2(h^{(k)})$, i.e.,

$$B(h^{(k)}) = diagExtract[\nabla^2 Z_2(h^{(k)})] = \frac{1}{\theta} diag(h_{pqr}^{-1}) \tag{11}$$

Form 3: $B(h^{(k)})$ is the diagonal of the Hessian of $Z(h^{(k)})$, i.e.,

$$B(h^{(k)}) = diagExtract[\nabla^2 Z(h^{(k)})] = diagExtract[\Delta^T(\nabla \boldsymbol{t})\Delta] + \frac{1}{\theta} diag(h_{pqr}^{-1}), \tag{12}$$

where the function $diagExtract[\;]$ extracts the diagonal of a matrix.

The subproblem solved in iteration k then is:

[PL-SUB$^{(k)}$]

$$Z^{(k)}(h) = \sum_{(p,q)\in C}\sum_{r\in R_{pq}} c_{pqr}^{(k)} h_{pqr} + \frac{1}{2}\sum_{(p,q)\in C}\sum_{r\in R_{pq}} b_{pqr}^{(k)}(h_{pqr}-h_{pqr}^{(k)})^2 + \frac{1}{\theta}\sum_{(p,q)\in C}\sum_{r\in R_{pq}} h_{pqr}\ln h_{pqr}, \tag{13}$$

s.t. (3) (4) (5)

where $c_{pqr}^{(k)} = \sum_a \delta_{pqra} t_a(f_a^{(k)})$ and $b_{pqr}^{(k)}$ equals $\sum_a \delta_{pqra} t'_a(f_a^{(k)})$ (Form 1), or $(\theta h_{pqr})^{-1}$ (Form 2), or $\sum_a \delta_{pqra} t'_a(f_a^{(k)}) + (\theta h_{pqr})^{-1}$ (Form 3).

Similar to the conclusion of Proposition 1, problem [PL-SUB$^{(k)}$] is a strictly convex problem, so there is a unique optimal solution to [PL-SUB$^{(k)}$].

From a practical point of view, [PL-SUB$^{(k)}$] cannot be solved exactly. Proposition 2 below shows that if an approximate solution $\bar{h}^{(k)}$ of [PL-SUB$^{(k)}$] makes the objective value of [PL-SUB$^{(k)}$] smaller, then the vector $\bar{h}^{(k)} - h^{(k)}$ defines a feasible descent direction with respect to the objective function $Z(h)$.

Proposition 2. (Descent property of the outer level iteration method) Let $h^{(k)}$ be a feasible point of [TAP-SUE], and $\bar{h}^{(k)}$ be an inexact solution of [PL-SUB$^{(k)}$] such that

$$Z^{(k)}(\bar{h}^{(k)}) < Z^{(k)}(h^{(k)}), \tag{14}$$

then $\nabla Z(h^{(k)})^T(\bar{h}^{(k)} - h^{(k)}) < 0$.

Proof. Follows from Patriksson (1999) Proposition 2.14(b). (a similar proof can be found in Patriksson (1998) Lemma 3.2(b)). ■

For the outer level iteration, we propose to use the Armijo rule to compute the step size. The Armijo rule needs to evaluate the objective function, so the computational cost is larger than the predetermined step size rule (see below) at each iteration. However, since the Armijo rule can ensure a sufficient decrease in the objective function at each iteration, its convergence rate is faster than the predetermined step size rule, especially in later iterations. As a result, by using the Armijo rule, we can obtain any level of accuracy we desire in reasonable iteration counts.

The Armijo step size is defined by:

$$\lambda^{(k)} = \beta^{m_k}, \tag{15}$$

where m_k is the first integer, $m \geq 0$, which satisfies:

$$Z(h^{(k)} + \beta^m(\bar{h}^{(k)} - h^{(k)})) - Z(h^{(k)}) \leq \sigma\beta^m \nabla Z(h^{(k)})^T(\bar{h}^{(k)} - h^{(k)}), \tag{16}$$

where $0 < \beta < 1$ and $0 < \sigma < 1$ are parameters.

The new iteration point of the outer loop is:

$$h^{(k+1)} = h^{(k)} + \lambda^{(k)}(\bar{h}^{(k)} - h^{(k)}), \tag{17}$$

We next give the convergence result for the outer level iteration method as applied to [TAP-SUE].

Proposition 3. (Convergence of the outer level iteration method) Assume that the inexact solution $\bar{h}^{(k)}$ of [PL-SUB$^{(k)}$] is obtained by performing a descent algorithm with a closed algorithm map with $l^{(k)}$ ($l^{(k)} \geq 1$) iterations. Let h^* be the unique optimal solution of [TAP-SUE], and $\{h^{(k)}\}$ be the sequence of points generated by the outer level iteration method, using the Armijo step size rule. Then $\{h^{(k)}\} \to h^*$.

Proof. Follows from Patriksson (1999) Theorem 5.19. (a similar proof can be found in Patriksson (1998) Theorem 4.2). ■

3.2 The inner level iteration phase

Now we study how to solve [PL-SUB$^{(k)}$] approximately. This is the inner level iteration phase of the two level partial linearization method. Sub-problem [PL-SUB$^{(k)}$] is in essence a linearly-constrained entropy maximization problem with quadratic cost. A typical method for this problem can be found in Fang and Tsao (1995), in which an unconstrained dual approach with a curved search method is proposed.

As is discussed above, to find an inexact solution $\bar{h}^{(k)}$ to [PL-SUB$^{(k)}$], we only need to perform a descent algorithm with a closed algorithm map with $l^{(k)}$ ($l^{(k)} \geq 1$) iterations, such that equation (14) is satisfied. In this case, we propose still use the partial linearization method to solve [PL-SUB$^{(k)}$]. The reason is that if we do so, the solution of [PL-SUB$^{(k)}$]'s subproblem (i.e., [PL-SUB$^{(k)}$-SUB$^{(l)}$]) can be given in a closed form.

At iteration *l* of [PL-SUB$^{(k)}$], Let $h^{(l)}$ be a given feasible path flow. By using a first order approximation to the first two terms in [PL-SUB$^{(k)}$], we obtain:

[PL-SUB$^{(k)}$-SUB$^{(l)}$]

$$\min\ Z^{(k)(l)}(h) = \sum_{(p,q)\in c}\sum_{r\in R_{pq}} g_{pqr}^{(l)} h_{pqr} + \frac{1}{\theta}\sum_{(p,q)\in c}\sum_{r\in R_{pq}} h_{pqr}\ln h_{pqr}, \tag{18}$$

s.t. (3) (4) (5)

where

$$g_{pqr}^{(l)} = c_{pqr}^{(k)} + b_{pqr}^{(k)}(h_{pqr}^{(l)} - h_{pqr}^{(k)}). \tag{19}$$

Similar to the conclusion of Proposition 1, problem [PL-SUB$^{(k)}$-SUB$^{(l)}$] is strictly convex. There is a unique optimal solution to [PL-SUB$^{(k)}$-SUB$^{(l)}$].

Different from the outer level iteration phase, the solution to [PL-SUB$^{(k)}$-SUB$^{(l)}$] can be given in closed (exact) form,

$$\bar{h}_{pqr}^{(l)} = d_{pq} \frac{e^{-\theta g_{pqr}^{(l)}}}{\sum_{r \in R_{pq}} e^{-\theta g_{pqr}^{(l)}}}. \tag{20}$$

The following proposition shows that if the vector $\bar{h}^{(l)} - h^{(l)}$ is non-zero, it defines a descent direction with respect to the objective function $Z^{(k)}(h)$.

Proposition 4. (Descent properties of the inner level iteration method) Let $h^{(l)}$ be a feasible point of [PL-SUB$^{(k)}$], and $\bar{h}^{(l)}$ be the unique exact solution of [PL-SUB$^{(k)}$-SUB$^{(l)}$]. If $\bar{h}^{(l)} - h^{(l)}$ is non-zero, then $\nabla Z^{(k)}(h^{(l)})^T(\bar{h}^{(l)} - h^{(l)}) < 0$.

Proof. Follows from Patriksson (1999) Proposition 2.14 (a). (a similar proof can be found in Patriksson (1998) Lemma 3.2(a)). ■

For the inner level iteration, we propose to use the predetermined step size rule, as in the MSA method. Using a predetermined step size typically implies a slow asymptotic convergence rate in the late iterations, not only because it cannot guarantee descent at each iteration, but also because the step size is diminishing for convergence (i.e., $\lambda^{(l)} \to 0$). However, in the early iterations, a predetermined step size often achieves a relatively fast convergence rate. Another advantage of using a predetermined step size is that there is no need to evaluate the objective function, making it computationally efficient. Therefore, a predetermined step size is often favored for large problems where great solution accuracy is not of paramount importance. As a result, it is suitable for the inner iteration of the two level partial linearization method.

Another point to be noted is that, if we use the predetermined step size rule for the partial linearization method, the algorithmic map is not assured to be a descent map (it is only asymptotically descent, as is the MSA method). So the conditions for convergence in Proposition 3 may not be satisfied. However, we can overcome this by introducing a composite map. Let A be the algorithmic map for the partial linearization method with predetermined step size. We perform n iterations of this method, and define a new map G to represent the n composite map A○A○... ○A. As long as G makes the objective value of [PL-SUB$^{(k)}$] smaller, we can conclude that G is a descent map. Therefore, after introducing the new map G, conditions in Proposition 3 can be satisfied. Then we can conclude that the outer level iteration phrase is convergent.

The predetermined step size $\lambda^{(l)}$ satisfies:

$$0 < \lambda^{(l)} \le 1, \ \sum_{l=1}^{\infty} \lambda^{(l)} = \infty, \text{ and } \sum_{l=1}^{\infty} (\lambda^{(l)})^2 < \infty. \tag{21}$$

The new iteration point of the inner loop is:

$$h^{(l+1)} = h^{(l)} + \lambda^{(l)}(\bar{h}^{(l)} - h^{(l)}). \tag{22}$$

We can get the following convergence result for the inner level iteration method as applied to [PL-SUB$^{(k)}$].

Proposition 5. (Convergence of the inner level iteration method) Let h^{**} be the unique optimal solution of [PL-SUB$^{(k)}$], and $\left\{h^{(l)}\right\}$ be the sequence of points generated by the inner level iteration method, using the predetermined step size. Then $\left\{h^{(l)}\right\} \to h^{**}$.

Proof. The inner level iteration phase is in essence the same as the MSA method proposed by Powell and Sheffi. The convergence theorem of the MSA method can be found in Powell and Sheffi (1982). ■

4. A dual method for the logit SUE problem

In this section, we establish some properties of the Lagrange dual problem of [TAP-SUE], and present a scaled steep ascend method to solve this dual problem.

4.1 A Lagrange dual problem for [TAP-SUE]

Adding explicit non-negativity constraints (27) for link flows to [TAP-SUE], we can reformulate [TAP-SUE] as follows:

[TAP-SUE]*

$$\min \; Z(h,f) = \sum_{a \in A} \int_0^{f_a} t_a(s)ds + \frac{1}{\theta} \sum_{(p,q)\in C} \sum_{r \in R_{pq}} h_{pqr} \ln h_{pqr} \tag{23}$$

$$\sum_{r \in R_{pq}} h_{pqr} = d_{pq}, \;\; \forall (p,q) \in C \tag{24}$$

$$h_{pqr} \ge 0, \;\; \forall r \in R_{pq}, \;\; \forall (p,q) \in C \tag{25}$$

$$\sum_{(p,q)\in C} \sum_{r \in R_{pq}} \delta_{pqra} h_{pqr} = f_a, \;\; \forall a \in A \tag{26}$$

$$f_a \ge 0, \; \forall a \in A \tag{27}$$

We note that constraints (27) are redundant in this primal formulation, but are not redundant in the Lagrangean dualized problem (Patriksson, 1994).

Duralizing constraints (26) and letting $\mu = (\mu_a)_{a \in A}$ be the dual variables for constraint (26), we obtain the following dual stochastic user equilibrium problem

[DTAP-SUE]

$$\max \; \varphi(\mu) \tag{28}$$

If we define

$$L(h,f,\mu) = \sum_{a \in A} \int_0^{f_a} t_a(s)ds + \frac{1}{\theta} \sum_{(p,q)\in C} \sum_{r \in R_{pq}} h_{pqr} \ln h_{pqr} + \sum_{a \in A} \mu_a \left(\sum_{(p,q)\in C} \sum_{r \in R_{pq}} h_{pqr} \delta_{pqra} - f_a \right) \tag{29}$$

then the dual objective function can be expressed as

$$\varphi(\mu) = \min L(h,f,\mu) \tag{30}$$

s.t. (24) (25) (27)

If we view h (the path flow variable) and f (the link flow variable) as two independent variables, $\varphi(\mu)$ can be separated into two independent problems

[P1]

$$\varphi_1(\mu) \square \min \sum_{a\in A} \int_0^{f_a} t_a(x)dx - \sum_{a\in A} \mu_a f_a \tag{31}$$

$$\text{s.t. } f_a \geq 0, \quad \forall a \in A \tag{32}$$

[P2]

$$\varphi_1(\mu) \square \min \sum_{(p,q)\in C} \left[\frac{1}{\theta} \sum_{r\in R_{pq}} h_{pqr} \ln h_{pqr} + \sum_{r\in R_{pq}} h_{pqr} \sum_{a\in A} \delta_{pqra} \mu_a \right] \tag{33}$$

$$\text{s.t. } \sum_{r\in R_{pq}} h_{pqr} = d_{pq}, \quad \forall (p,q) \in C \tag{34}$$

$$h_{pqr} \geq 0, \quad \forall r \in R_{pq}, \ \forall (p,q) \in C \tag{35}$$

The solution of [P1] is (see Larsson et al., 1997)

$$f_a(\mu_a) = \begin{cases} t_a^{-1}(\mu_a), & \text{if } \mu_a \geq t_a(0), \\ 0, & \text{otherwise,} \end{cases} \quad \forall a \in A \tag{36}$$

The solution of [P2] is (see Damberg et al., 1996)

$$h_{pqr}(\mu) = d_{pq} \frac{e^{-\theta \sum_{a\in A} \delta_{pqra}\mu_a}}{\sum_{r\in R_{pq}} e^{-\theta \sum_{a\in A} \delta_{pqra}\mu_a}}, \quad \forall r \in R_{pq}, \ \forall (p,q) \in C \tag{37}$$

Therefore, the dual objective function $\varphi(\mu)$ can be given in closed form. Properties of the [DTAP-SUE] are given in the following proposition.

Proposition 6. (Properties of [DTAP-SUE]) The Lagrange dual program [DTAP-SUE] is an *unconstrained* differentiable concave maximization problem. There is a unique optimal solution μ^* to this problem.

Proof. For a given μ, $L(h,f,\mu)$ is strictly convex with respect to (h,f). So there is a unique point $(h(\mu), f(\mu))$ that minimize $L(h,f,\mu)$. According to Theorem 6.3.3 in Bazaraa et al. (1993), $\varphi(\mu)$ is differentiable. The concavity of $\varphi(\mu)$ follows from Theorem 6.3.1 in Bazaraa et al. (1993). The uniqueness of μ^* is proved as follows. Since the path choice set is determined in advance, as is the case in this paper, we can delete links that are not contained by any path, and only consider each *used* link. One key property of the SUE problem is that at equilibrium, each *used* link has strictly positive link flows. Hence the actual link travel time for each *used* link is strictly larger than its free flow travel time at the SUE equilibrium point. As is well known, the dual optimum μ^* can be interpreted as the actual link travel times (Larsson et al. 1997). Hence at equilibrium, $\mu_a^* > t_a(0)$ holds for all *used* links. For the SUE problem, the equilibrium link flow f^* is unique. By (36), $f_a = t_a^{-1}(\mu_a)$ is strictly increasing on $\mu_a > t_a(0)$. Therefore, μ^* is unique. ■

The following proposition relates the optimal solutions to [TAP-SUE] and [DTAP-SUE].

Proposition 7. (Relationships between [TAP-SUE] and [DTAP-SUE]) Strong duality holds, that is $Z(h^*, f^*) = \varphi(\mu^*)$. Furthermore, $f^* = f(\mu^*)$ and $h^* = h(\mu^*)$.

Proof. The constraints of [TAP-SUE] are linear. By Lemma 5.1.4 in Bazaraa et al. (1993), the Abadie constraint qualification is satisfied. Hence, by Theorem 6.2.4 in Bazaraa et al. (1993), the strong duality holds from the convexity of [TAP-SUE] and the fact that Abadie constraint qualification holds. Applying Theorem 6.5.2 in Bazaraa et al. (1993), the optimal solution of the primal problem can be characterized by the optimal solution of the dual problem. Hence, $f^* = f(\mu^*)$ and $h^* = h(\mu^*)$ hold. ■

4.2 Solving the dual problem

We propose to use the scaled steepest ascend method (Bertsekas, 1999) to solve the dual problem. The iteration for the scaled steepest ascend method is

$$\mu^{(k+1)} = \mu^{(k)} + \alpha^{(k)} B(\mu^{(k)})^{-1} \nabla\varphi(\mu^{(k)}), \tag{38}$$

where $\alpha^{(k)}$ is the step size and $B(\mu^{(k)})$ is a symmetric positive definite matrix. The gradient of $\varphi(\mu^{(k)})$ is (see Bertsekas,1999, Chapter 6)

$$\nabla\varphi(\mu^{(k)}) = \Delta\boldsymbol{h} - \boldsymbol{f}, \tag{39}$$

where $\boldsymbol{h} = \left[h_{pqr}(\mu^{(k)})\right]^T_{(p,q)\in c, r\in R_{pq}}$ is the path flow vector, and $\boldsymbol{f} = \left[f_a(\mu^{(k)})\right]^T_{a\in A}$ is the link flow vector.

The Hessian of $\varphi(\mu^{(k)})$ is (details of the derivation of the Hessian are in the appendix)

$$\begin{aligned}\nabla^2\varphi(\mu^{(k)}) &= -\left[\Delta diag(\theta h_{pqr})\Delta^T + diag(t_a'(f_a)^{-1})\right] \\ &\square -\left[H_1(\mu^{(k)}) + H_2(\mu^{(k)})\right]\end{aligned} \tag{40}$$

where $diag(\theta h_{pqr})$ is a diagonal matrix whose diagonal elements are θh_{pqr}, and $diag(t_a'(f_a)^{-1})$ is a diagonal matrix whose diagonal elements are $t_a'(f_a)^{-1}$.

Similar to the discussion above, $B(\mu^{(k)})$ is usually chosen as a diagonal matrix. For the dual method, three forms of $B(\mu^{(k)})$ can be potentially used:

Form 1: $B(\mu^{(k)})$ is the diagonal of the hessian of $H_1(\mu^{(k)})$, i.e.,

$$B(h^{(k)}) = diagExtract[H_1(\mu^{(k)})] = diagExtract[\Delta diag(\theta h_{pqr})\Delta^T] \tag{41}$$

Form 2: $B(\mu^{(k)})$ is the diagonal of the hessian of $H_2(\mu^{(k)})$, i.e.,

$$B(h^{(k)}) = diagExtract[H_2(\mu^{(k)})] = diag(t_a'(f_a)^{-1}) \tag{42}$$

Form 3: $B(\mu^{(k)})$ is the diagonal of the hessian of $\nabla^2\varphi(\mu^{(k)})$, i.e.,

$$B(h^{(k)}) = diagExtract[\nabla^2\varphi(\mu^{(k)})] = diagExtract[\Delta diag(\theta h_{pqr})\Delta^T + diag(t_a'(f_a)^{-1})], \tag{43}$$

Utilizing the closed form solution of [P1] and [P2], the dual objective function can be easily evaluated. Therefore, for the dual method, we still use the Armijo rule to compute the step size.

Although not explicitly required, it may be useful for the implementation of the algorithm to calculate an upper bound on the step size.

Define

$$p = B(\mu^{(k)})^{-1}\nabla\varphi(\mu^{(k)}), \tag{44}$$

as the search direction of the dual method. A maximum step size to ensure strict feasible link travel times is given by:

$$\alpha_{\max}^{(k)} = \begin{cases} \infty, & \text{if } p \geq 0, \\ \min_{a\in A}\left\{-\dfrac{\mu_a - t_a(0)}{p_a} \mid \forall p_a < 0\right\}, & \text{otherwise.} \end{cases} \tag{45}$$

From the above, all the ingredients of the dual method are obtained. We can iteratively implement equation (38), and finally obtain the equilibrium link travel times (i.e., the optimal dual solution).

For the dual method, the following convergence result is valid.

Proportion 8. (Convergence of the dual method) Let μ^* be the unique optimal solution of[DTAP-SUE], and $\{\mu^{(k)}\}$ be the sequence of points generated by the dual method, using the scaled steepest ascend direction with Armijo step size rule. Then $\{\mu^{(l)}\} \to \mu^*$.

Proof. Follows from Bertsekas (1999) Proposition 1.2.1 and Ortega and Rheinboldt (1970) Theorem 14.1.4. ■

5. A measure to compare the performance of different algorithms

Define Z^* as the optimal objective value of [TAP-SUE]. As we know, Damberg's partial linearization method and the two level partial linearization method are primal feasible methods (i.e., each iteration point $h^{(k)}$ is primal feasible). The objective function value $Z(h^{(k)})$ defines an upper bound of Z^*, and $Z(h^{(k)})$will converge to Z^* from above. However, the dual method is a primal infeasible method. Given a dual iteration point $\mu^{(k)}$, we cannot obtain a primal feasible solution from the dual subproblem [P1] and [P2] unless $\mu^{(k)}$ reaches its limit μ^*. If we want to yield a primal feasible solution of [TAP-SUE], some heuristic algorithm (Larsson etal.,1997) should be used. The dual objective function value $\varphi(\mu^{(k)})$ defines a lower bound of Z^*(weak duality theorem). Since there is no duality gap, $\varphi(\mu^{(k)})$ will convergence to Z^* from below. In order to evaluate the performance of the three different methods, the following convergence measure can be used (Leurent, 1997):

$$\ln\left|\frac{M^k}{Z^*} - 1\right| \tag{46}$$

wherein Damberg's partial linearization method or the two level partial linearization method $M^k = Z(h^{(k)})$; and in the dual method, $M^k = \varphi(\mu^{(k)})$. $Z^*$is the optimal value of [TAP-SUE].

Obviously, equation (46) only measures the proximity of the *k*th objective function value to the optimal value of [TAP-SUE], regardless of how M^k approaches to Z^* (i.e., regardless of M^k converges to Z^* from above or below). Therefore, it is a suitable measure to compare the three algorithms.

6. Numerical results

In this section, we will make comparisons between Damberg's partial linearization (Damberg's PL) method, the two level partial linearization (Two Level PL) method and the Dual method on the Sioux Falls network and Winnipeg network. Both of the two networks are taken from Bar-Gera (2013). The Sioux Falls network consists of 76 links, 24 nodes and 528 OD pairs. It is a small size network. The Winnipeg network consists of 2836 links, 1052 nodes and 4344 OD pairs. It is a real size network.

The paths for these two networks are generated prior to the traffic assignment. We use a combination of the link elimination method (Azevedo et al., 1993) and link penalty method (De La Barra et al., 1993) to generate the working path set. For the Sioux Falls network, the average number of generated paths is 7.3 per OD pair, and the maximum number of generated paths is 11 for any OD pair. For the Winnipeg network, the average number of generated paths is 20.3 per OD pair, and the maximum number of generated paths is 29 for any OD pair.

As is discussed above, depending on different choice of the scaling matrix, three forms of Two Level PL method can be obtained. For abbreviation, we denote them by Two Level PL-1, Two Level PL-2, and Two Level PL-3-method. Similarly, we denote the three forms of the Dual method by Dual-1, Dual-2, and Dual-3 method. For the Two Level PL method, the step size of the inner level iteration phase is set to $1/n+1$, and the inner iteration number is set to 12. Our computer programs are coded in MATLAB and executed on a notebook computer.

6.1 Algorithms performance

We first compare the three methods for most common case. In this case, the demands of the two networks are assumed to be the same as their original source. The dispersion parameter θ is assumed to be 0.5, which means that given a 5-minute difference between two paths, about 8% of the drivers will choose the route with the higher cost. Note that this case is most likely to occur in practice, so we give a detailed comparison of the three algorithms.

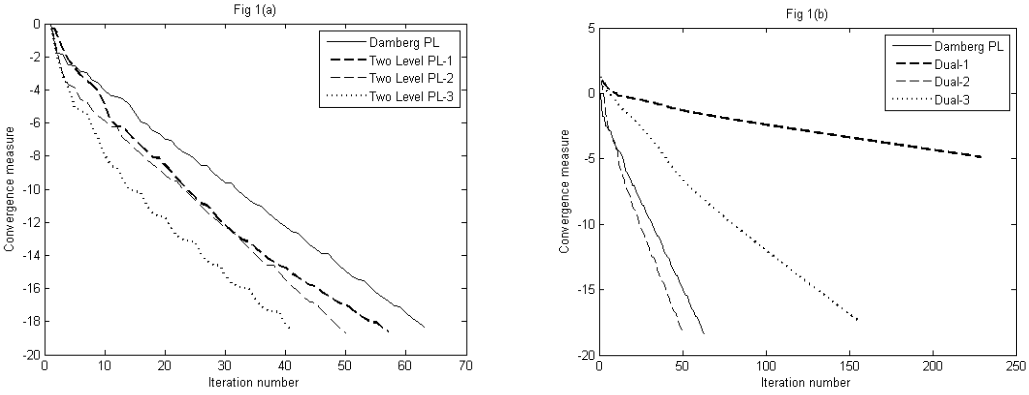

Figure 1 Convergence performance in terms of iteration numbers-Sioux Falls network

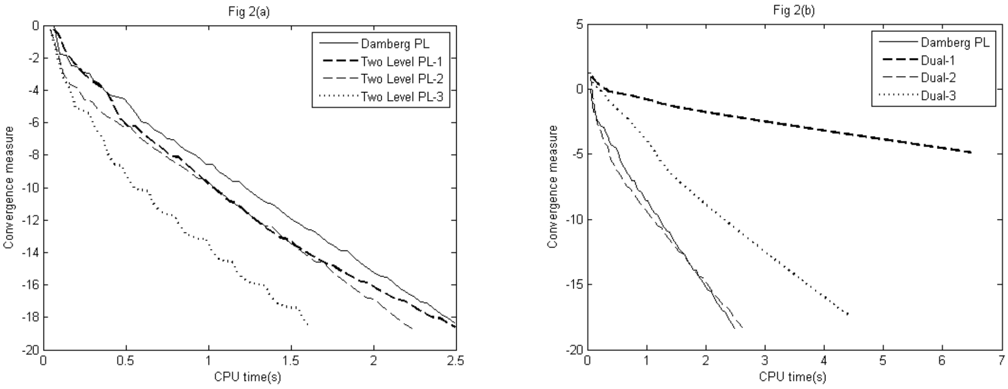

Figure 2 Convergence performance in terms of CPU times-Sioux Falls network

Fig 1 illustrates the convergence performance of the three methods in terms of iteration numbers in the Sioux Falls case. This figure in essence indicates the convergence rate of the three algorithms. From Fig1(a), we see that in case of the Sioux Falls network, all three forms of the Two Level PL method are faster than Damberg's PL method, with the Two Level PL-3 method performs best. From Fig1(b), we observe that Damberg's PL method is slightly slower than the Dual-2 method, faster than the Dual-3 method, and much faster than the Dual-1 method.

Fig 2 shows the convergence performance of the three methods in terms of CPU times for the Sioux Falls network. This figure in essence indicates the computational efficiency of the three algorithms. From Fig 2(a), we can see that the computational efficiency of the Two Level PL method is higher than that of Damberg's PL method, no matter what form is used. The reason for more efficiency of the Two Level PL method is two-fold. Firstly, the number of iterations required by the Two Level PL method is smaller than Damberg's PL method. Secondly, in each iteration, evaluating the objective function in the Armijo rule accounts for most of the CPU times (the Armijo rule requires to successively evaluate the objective function until Armijo's inequality is satisfied). The Two Level PL method requires less function evaluations than Damberg's PL method, which means it spends less CPU times in each iteration. We can also observe that the Two Level PL-3 method is most efficient. It improves Damberg's PL method by roughly 30%-50%, which is a significant decrease in CPU time. From Fig 2(b), we can observe that the computation efficiency of Damberg's PL method is slightly lower than Dual-2 method, higher than Dual-3 method, and much higher than Dual-1 method. The reason why Dual-1 and Dual-3 method are less efficiency is mainly because the number of iterations required by these two methods is much larger than Damberg's PL method.

Fig 3 illustrates the convergence performance of the three methods in terms of iteration numbers for the Winnipeg network. From Fig 3(a), we see that the convergence rate of Damberg's PL method is slightly slower than the Two-Level PL-1 method, and is slower than both of the Two-Level PL-2 and Two-Level PL-3 method. From Fig 3(b), we observe that Damberg's PL method and Dual-1 method have similar convergence rate, and both are slower than Dual-2 method, but much faster than the Dual-3 method.

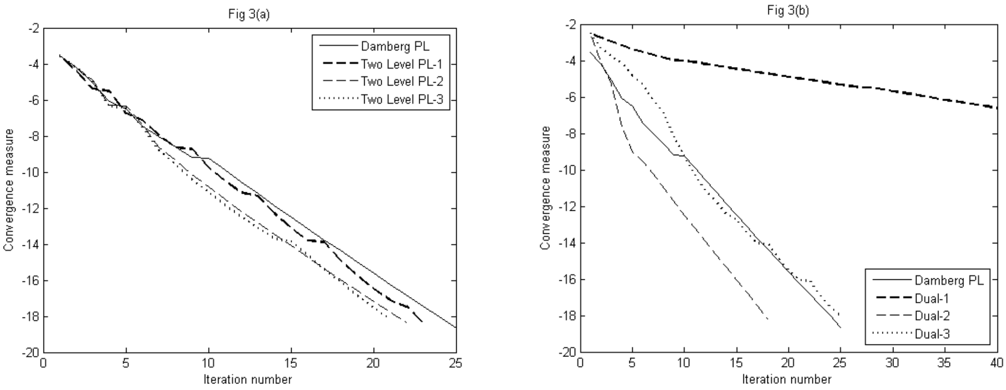

Figure 3 Convergence performance in terms of iteration numbers-Winnipeg network

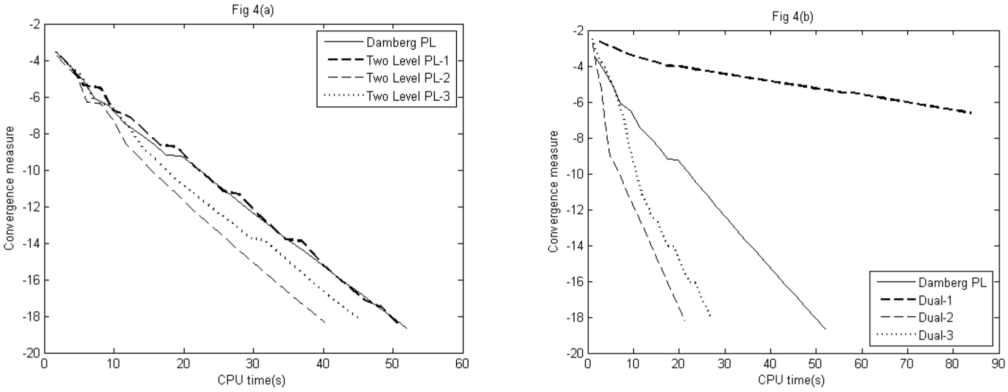

Figure 4 Convergence performance in terms of CPU times-Winnipeg network

Fig 4 shows the convergence performance of the three methods in terms of CPU times for the Winnipeg network. From Fig 4(a), we find that Damberg's PL method and the Two Level PL-1 method have similar computational efficiency, and both are less efficient than the Two Level PL-2 and Two Level PL-3 method. From Fig 4(b), we know that the computational efficiency of Damberg's PL method is lower than the Dual-2 and Dual-3 method, but much higher than Dual-1 method. Note that the convergence rate of the Dual-2 method is similar to Damberg's PL method, but the efficiency of Dual-2 method is much higher. The reason is that in each iteration, the Dual-2 method requires the less function evaluations than Damberg's PL method, so its computational efficiency is higher per iteration. Compared to Damberg’s PL method, the Dual-2 method reduces the CPU time by almost 50%.

6.2 Sensitivity Analysis

Now we compare the three methods under different cases, i.e., we perform sensitivity analysis for the three methods. By varying the value of θ or multiplying the model demand with different factors, we can examine the impact of the dispersion parameter or demand factor on the performance of different methods. Due to space limitation, we only choose the most efficient forms of the two proposed methods that was found in Section 6.1 to conduce the sensitivity analysis, i.e. we choose the Two-Level PL-3 method and the Dual-2 method for the sensitivity analysis. We only report the performance of the three algorithms achieve a solution within 0.01% of the equilibrium objective function value.

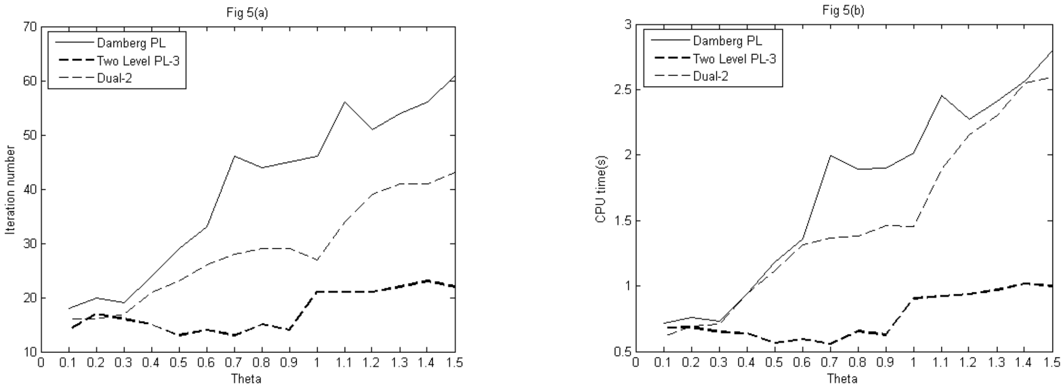

Figure 5 Sensitivity of the algorithm performance to the dispersion parameter-Sioux Falls network

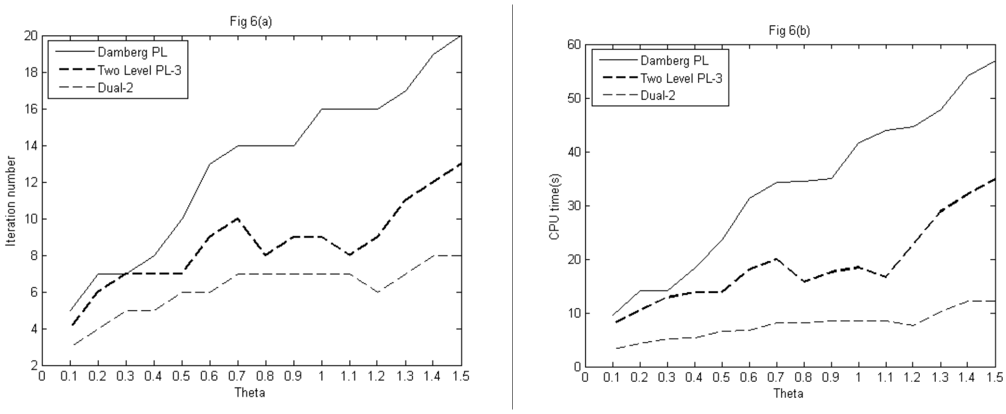

Figure 6 Sensitivity of the algorithm performance to the level of demand-Sioux Falls network

Figures 5–6 show the number of iterations and CPU times required by the three methods using the Sioux Falls and Winnipeg networks as a function of the dispersion parameter. As is presented in these two figures, both the Two Level PL-3 method and the Dual-2 method outperform Damberg's PL method, with the Two Level PL-3 method performs best for the Sioux Falls network, and the Dual-2 method performs best on the Winnipeg network.

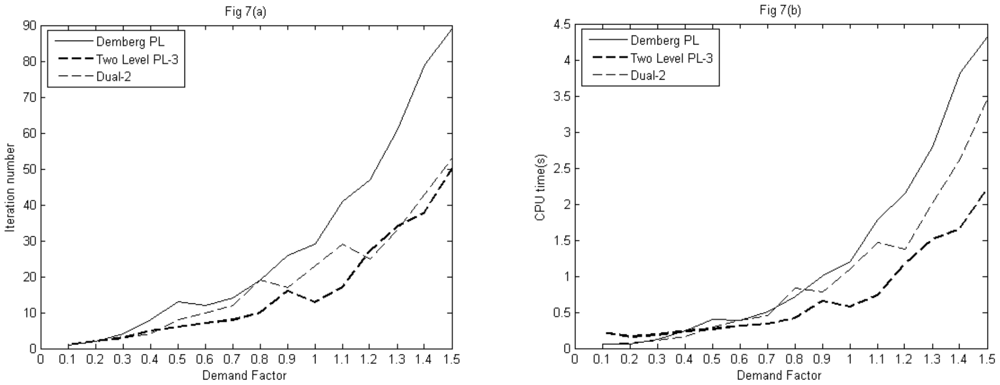

Figure 7 Sensitivity of the algorithm performance to the dispersion parameter-Winnipeg network

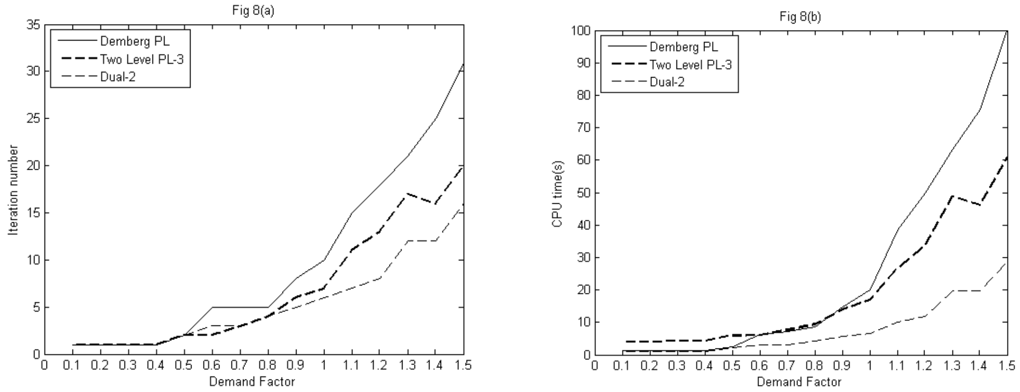

Figure 8 Sensitivity of the algorithm performance to the level of demand-Winnipeg network

Figures 7–8 show the number of iterations and CPU times required by the three methods using the Sioux Falls and Winnipeg networks as a function of the demand factor. As is illustrated in Fig 7(a) and Fig 8(a), the number of iterations for the Two Level PL-3 method and the Dual-2 method are smaller than or equal to Damberg's PL method. This indicates that the convergence rate of these two methods is at least as fast as Damberg's PL method. As is illustrated in Fig 7(b) and Fig 8(b), when the demand factor is small, the CPU times for Damberg's PL method and the Dual-2 method are similar, and both are less than the Two Level PL-3 method; when the demand factor is large, the CPU times for the Two Level PL-3 method and the Dual-2 method are less than Damberg's PL method, with the Two Level PL-3 method consumes the least CPU times for the Sioux Falls network, and the Dual-2 method consumes the least CPU times for the Winnipeg network. This means that only for the case when the demand is very small, the Two Level PL-3 method is less efficient than Damberg's PL method. Note that cases with very small total demand are rare, so we conclude that the Two Level PL-3 method and the Dual-2 method perform better than Damberg's PL method for most practical cases.

7. Conclusions

This study investigated solution methods for the path-based stochastic user equilibrium problem. Two new methods–a two level partial linearization method and a dual method were proposed. We compared these two methods with Damberg's partial linearization method on the Sioux Falls network and Winnipeg network. Numerical results showed that both of them can be implemented on real-size networks in practice. If properly scaled, they are more faster and efficient than Damberg's partial linearization method for most test cases.

In this research we limited the application to the logit SUE model. Several possible extensions of this research are suggested here. First, we can adapt the two methods to the more general combined distribution and assignment model. Second, we can apply the two methods to the more general route choice models, such as the Cross-Nested Logit model (Bekhor et al., 2009) or other route choice models that can take path overlap into account (Bliemer and Bovy, 2008). Third, the two methods can also work with a column generation scheme, which generates the path set during the assignment.

Acknowledgements

We would like to thank Prof. Michael Patriksson for his help and valuable suggestions and comments on an earlier draft of this paper.

Appendix: Derivation of the hessian of $\varphi(\mu)$

The Hessian of $\varphi(\mu)$ can be computed in the following form (Bertsekas, 1999, Chapter 6)

$$\nabla^2\varphi(\mu) = -\nabla g(h,f)^T \nabla^2 L(h,f,\mu)^{-1} \nabla g(h,f), \tag{A1}$$

where $L(h,f,\mu)$ is given by equation (31), and $g(h,f)$ is the vector function of constraints (26), defined as

$$g(h,f) = \sum_{(p,q)\in C}\sum_{r\in R_{pq}} \delta_{pqra} h_{pqr} - f_a, \qquad \forall a \in A. \tag{A2}$$

Therefore, we can derive that

$$\nabla^2 L(h,f,\mu)^{-1} = \begin{bmatrix} diag\left[\theta h_{pqr}\right] & \\ & diag\left[t_a'(f_a)^{-1}\right] \end{bmatrix}, \tag{A3}$$

and

$$\nabla g(h,f) = \begin{bmatrix} \Delta^T \\ -I \end{bmatrix}, \tag{A4}$$

where I is an $|A|\times|A|$ identity matrix. (i.e., the dimension of I equals to the number of links of the network)

Therefore,

$$\begin{aligned}\nabla^2\varphi(\mu) &= -[\Delta \quad -I]\begin{bmatrix} diag\left[\theta h_{pqr}\right] & \\ & diag\left[t_a'(f_a)^{-1}\right] \end{bmatrix}\begin{bmatrix} \Delta^T \\ -I \end{bmatrix} \\ &= -\left[\Delta diag\left[\theta h_{pqr}\right] \quad -Idiag\left[t_a'(f_a)^{-1}\right]\right]\begin{bmatrix} \Delta^T \\ -I \end{bmatrix} \\ &= -\left[\Delta diag\left[\theta h_{pqr}\right]\Delta^T + diag\left[t_a'(f_a)^{-1}\right]\right].\end{aligned} \tag{A5}$$

References

Akamatsu, T. (1996), Cyclic flows, Markov process and stochastic traffic assignment, Transportation Research Part B: Methodological, 30(5), 369-386.

Azevedo, J., Santos Costa, M. E. O., Silvestre Madeira, J. J. E., & Vieira Martins, E. Q. (1993), An algorithm for the ranking of shortest paths, European Journal of Operational Research, 69(1), 97-106.

Bar-Gera, H., Transportation network test problems, http://www.bgu.ac.il/~bargera/tntp/, accessed September 10, 2013.

Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (1993), Nonlinear programming: theory and algorithms, John Wiley & Sons, New York.

Bekhor, S., Ben-Akiva, M. E., & Ramming, M. S. (2006), Evaluation of choice set generation algorithms for route choice models, Annals of Operations Research, 144(1), 235-247.

Bekhor, S., & Toledo, T. (2005), Investigating path-based solution algorithms to the stochastic user equilibrium problem, Transportation Research Part B: Methodological, 39(3), 279-295.

Bekhor, S., Toledo, T., & Reznikova, L. (2009), A Path-Based Algorithm for the Cross-Nested Logit Stochastic User Equilibrium Traffic Assignment, Computer-Aided Civil and Infrastructure Engineering, 24(1), 15-25.

Bell, M. G. (1995), Alternatives to Dial's logit assignment algorithm, Transportation Research Part B: Methodological, 29(4), 287-295.

Bell, M. G. H., Lam, W. H., Ploss, G., and Inaudi, D. (1993), Stochastic user equilibrium assignment and iterative balancing, in: Daganzo, C.F. (ed.), Proceedings of the 12th International Symposium on Transportation and Traffic Theory, Elsevier, New York, pp. 427–439.

Ben-Akiva, M., Bergman, M. J., Daly, A. J., & Ramaswamy, R. (1984), Modeling inter-urban route choice behaviour. in Volmuller, J., Hamerslag, R. (eds.), Proceedings of the 9th International Symposium on Transportation and Traffic Theory, VNU Press, Utrecht, pp. 299-330.

Bertsekas, D. P. (1999), Nonlinear programming, Athena Scientific, Belmont, MA.

Bliemer, M. C., & Bovy, P. H. (2008), Impact of route choice set on route choice probabilities, Transportation Research Record: Journal of the Transportation Research Board, 2076(1), 10-19.

Bliemer, M. C., Taale, H. (2006), Route generation and dynamic traffic assignment for large networks, Conference proceedings 1st DTA conference, Leeds, UK.

Boyce, D. E. (1984), Urban transportation network-equilibrium and design models: recent achievements and future prospects, Environment and Planning A, 16(11), 1445-1474.

Cascetta, E., Nuzzolo, A., Russo, F., & Vitetta, A. (1996), A modified logit route choice model overcoming path overlapping problems: Specification and some calibration results for interurban networks, in Lesort, J. B. (ed.), Proceedings of the International Symposium on Transportation and Traffic Theory. Elsevier, New York, pp. 697–711.

Cascetta, E., & Papola, A. (2008), A trip distribution model involving spatial and dominance attributes, Computer-Aided Civil and Infrastructure Engineering, 23(2), 116-124.

Castillo, E., Menéndez, J. M., & Sánchez-Cambronero, S. (2008), Traffic estimation and optimal counting location without path enumeration using Bayesian networks, Computer-Aided Civil and Infrastructure Engineering, 23(3), 189-207.

Chen, M., & Alfa, A. S. (1991), Algorithms for solving Fisk's stochastic traffic assignment model, Transportation Research Part B: Methodological, 25(6), 405-412.

Daganzo, C. F., & Sheffi, Y. (1977), On stochastic models of traffic assignment. Transportation Science, 11(3), 253-274.

Damberg, O., Lundgren, J. T., & Patriksson, M. (1996), An algorithm for the stochastic user equilibrium problem, Transportation Research Part B: Methodological, 30(2), 115-131.

De la Barra, T., Perez, B., & Anez, J. (1993), Multidimensional path search and assignment, In PTRC Summer Annual Meeting, 21st, 1993, University of Manchester, United Kingdom.

Dial, R. B. (1971), A probabilistic multipath traffic assignment model which obviates path enumeration, Transportation research, 5(2), 83-111.

Evans, S. P. (1973), A relationship between the gravity model for trip distribution and the transportation problem in linear programming, Transportation Research, 7(1), 39-61.

Fang, S. C., & Tsao, H. S. (1995), Linearly-constrained entropy maximization problem with quadratic cost and its applications to transportation planning problems, Transportation science, 29(4), 353-365.

Fiorenzo-Catalano, M. S. (2007), Choice set generation in multi-modal transport networks, PhD Thesis, Delft University of Technology, The Netherlands.

Fisk, C. (1980), Some developments in equilibrium traffic assignment, Transportation Research Part B: Methodological, 14(3), 243-255.

Hoogendoorn-Lanser, S. (2005), Modelling travel behaviour in multi-modal networks, PhD Thesis, Delft University of Technology, Delft, The Netherlands.

Larsson, T., Liu, Z., & Patriksson, M. (1997), A dual scheme for traffic assignment problems, Optimization, 42(4), 323-358.

LeBlanc, L. J., & Farhangian, K. (1981), Efficient algorithms for solving elastic demand traffic assignment problems and mode split-assignment problems, Transportation Science, 15(4), 306-317.

Leurent, F. M. (1997), Curbing the computational difficulty of the logit equilibrium assignment model, Transportation Research Part B: Methodological, 31(4), 315-326.

Liu, Y., & Luo, Z. (2012), A Bi-Level Model for Planning Signalized and Uninterrupted Flow Intersections in an Evacuation Network, Computer-Aided Civil and Infrastructure Engineering, 27(10), 731-747.

Maher, M. (1998), Algorithms for logit-based stochastic user equilibrium assignment, Transportation Research Part B: Methodological, 32(8), 539-549.

Meng, Q., & Wang, X. (2008), Sensitivity Analysis of Logit-Based Stochastic User Equilibrium Network Flows with Entry–Exit Toll Schemes, Computer-Aided Civil and Infrastructure Engineering, 23(2), 138-156.

Ortega, J. M., & Rheinboldt, W. C. (1970), Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York.

Patriksson, M. (1994), The traffic assignment problem: models and methods, VSP, Utrecht, the Netherlands.

Patriksson, M. (1998), Cost approximation: a unified framework of descent algorithms for nonlinear programs, SIAM Journal on Optimization, 8(2), 561-582.

Patriksson, M. (1999), Nonlinear Programming and Variational Inequality Problems: A Unified Approach, Kluwer Academic Publishers, Dordrecht.

Powell, W. B., & Sheffi, Y. (1982), The convergence of equilibrium algorithms with predetermined step sizes, Transportation Science, 16(1), 45-55.

Prato, C. G., & Bekhor, S. (2006), Applying branch-and-bound technique to route choice set generation, Transportation Research Record: Journal of the Transportation Research Board, 1985(1), 19-28.

Sheffi, Y. (1985), Urban transportation networks: equilibrium analysis with mathematical programming methods, Prentice-Hall, Englewood Cliffs.

Sheffi, Y., & Powell, W. B. (1982), An algorithm for the equilibrium assignment problem with random link times, Networks, 12(2), 191-207.

Szeto, W. Y., Jaber, X., & O'Mahony, M. (2010), Time-Dependent Discrete Network Design Frameworks Considering Land Use, Computer-Aided Civil and Infrastructure Engineering, 25(6), 411-426.

Szeto, W. Y., Jiang, Y., & Sumalee, A. (2011), A Cell-Based Model for Multi-class Doubly Stochastic Dynamic Traffic Assignment, Computer-Aided Civil and Infrastructure Engineering, 26(8), 595-611.

Wardrop, J. G. (1952), Some theoretical aspects of road traffic research. Proceedings of the Institution of Civil Engineers, Part II, 1(2), pp. 325-362.

Xu, X., Chen, A., Zhou, Z., & Bekhor, S. (2012), Path-Based Algorithms to Solve C-Logit Stochastic User Equilibrium Assignment Problem, Transportation Research Record: Journal of the Transportation Research Board, 2279(1), 21-30.