HYBRID ALGORITHM FOR DYNAMIC USER-OPTIMAL ROUTE CHOICE

Huey-Kuo Chen
Professor
Department of Civil Engineering,
National Central University, Taiwan
32054
Tel: +886-3-4227151 ext 4115, Email:
ncutone@cc.ncu.edu.tw

Shu-Yurn Hsiao
Graduate Student
Department of Civil Engineering,
National Central University, Taiwan
32054
Tel: +886-3-4227151 ext 4144, Email:
yurn@trans.cv.ncu.edu.tw

Abstract:The dynamic user-optimal route choice problem has been successfully solved by both link-based and path-based algorithms. The former algorithm embeds the Frank-Wolfe algorithm during the solution procedure whereas the latter contains a path-based algorithm such as Gradient Projection in its solution process. The Frank-Wolfe (FW) algorithm is economic in the use of computer memory but converges very slow at the neighborhood of the optimal solution. The Gradient Projection (GP) method, on the other hand, requires more computer memory but converges faster. For a comparison of computational efficiency between the FW method and GP, see Chen and Chang (1998).

In this paper, a projection method which was originally thought of as a path-based algorithm is revisited. With a hope of economic use of computer memory, the mapping concept of projection method is therefore realized by the (link-based) FW method instead, hereafter referred to as the Hybrid method. Two versions of the Hybrid method are described and demonstrated with numerical examples. The results show that the dynamic equilibrium conditions for dynamic user-optimal route choice can be satisfied, however, the expected benefits of using less computer memory are not readily obtained. To fully exploit the advantages of the Hybrid method, determination of the parameter settings that associated with the so-call *new* travel time function such as the contraction operator needs more research.

1. INTRODUCTION

The dynamic user-optimal (DUO) route choice problem can be regarded as a generalization of its static counterpart by incorporating flow propagation constraints into the model. The DUO route choice problem has been successfully formulated as a discrete time model using the variational inequality (VI) approach and solved by both the link based and path based algorithms (Chen et al., 1998). The former algorithm such as diagonalization method embeds the Frank-Wolfe (FW) algorithm during the solution procedure whereas the latter contains a path-based algorithm such as the gradient projection (GP) method. The FW algorithm is economic in the use of computer memory but converges very slow at the neighborhood of the optimal solution. The GP method, on the other hand, requires more computer memory but converges faster.

A Projection method which was originally thought of as path-based algorithm (Nagurney, 1993) is revisited. With a hope of economic use of computer memory, the mapping concept of projection method is therefore realized by the (link-based) FW method instead, hereafter

referred to as the Hybrid method (HB). Two version of the Hybrid method are proposed and compare with the current two methods for the DUO route choice problem.
In the following, the DUO route choice problem is first described and the corresponding equilibrium conditions are elaborated in Section 2. Two versions of the HB method are presented in Section 3 and demonstrate with a numerical example in Section 4. Intensive comparisons of the four solution algorithms are made in Section 5. Finally, concluding remarks and suggestions are given in Section 6.

2. EQUILIBRIUM CONDITIONS AND MODEL FORMULATION

2.1 DYNAMIC USER-OPTIMAL CONDITIONS

Assuming O-D demands are fixed and time-dependent, the DUO route choice conditions state for each O-D pair that the actual route travel times experienced by travelers departing during the same interval are equal and minimal. In contrast, the actual route travel time of any unused route for each O-D pair is greater than or equal to the actual used route travel times. In other words, at equilibrium, if the flow departing from origin r during interval k over route p toward destination s is positive, i.e., $h_p^{rs*}(k) > 0$, then the corresponding actual route travel time is minimal. On the contrary, if no flow occurs on route p, i.e., $h_p^{rs*}(k) = 0$, then the corresponding actual route travel time is at least as great as the minimal actual route travel time. These equilibrium conditions can be mathematically expressed as follows:

$$c_p^{rs*}(k) \begin{cases} = \pi^{rs}(k) & \text{if } h_p^{rs*}(k) > 0 \\ \geq \pi^{rs}(k) & \text{if } h_p^{rs*}(k) = 0 \end{cases} \quad \forall r,s,p,k \tag{1}$$

where

$$c_p^{rs*}(k) = \sum_a \sum_t c_a^*(t) \delta_{apk}^{rs}(t) \quad \forall r,s,p,k \tag{2}$$

$$\pi^{rs}(k) = \min_{p,k} \{ c_p^{rs*}(k) \} \quad \forall r,s \tag{3}$$

2.2 VARIATIONAL INEQUALITY FORMULATION

The DUO route choice problem is equivalent to finding a solution $\mathbf{u}^* \in \Omega$ such that the following VIP holds.

$$\sum_a \sum_t c_a^*(t) \left[u_a(t) - u_a^*(t) \right] \geq 0 \quad \forall \mathbf{u} \in \Omega^* \tag{4}$$

where Ω^* is a subset of Ω with $\delta_{apk}^{rs}(t)$ being realized at equilibrium, i.e., $\left(\delta_{apk}^{rs}(t) = \delta_{apk}^{rs*}(t) \right)$, $\forall r,s,a,p,k,t$. The symbol Ω denotes the feasible region that is delineated below by flow conservation, flow propagation, nonnegativity, and definitional constraints.

Flow conservation constraint:

$$\sum_p h_p^{rs}(k) = \bar{q}^{rs}(k) \quad \forall r,s,k \tag{5}$$

Flow propagation constraints:

$$u_{apk}^{rs}(t) = h_p^{rs}(k) \delta_{apk}^{rs}(t) \quad \forall r,s,a,p,k,t \tag{6}$$

$$\sum_t \delta_{apk}^{rs}(t) = 1 \quad \forall r,s,p,a \in p,k \tag{7}$$

$$\delta_{apk}^{rs}(t) = \{0,1\} \quad \forall r,s,a,p,k,t \tag{8}$$

Nonnegativity constraint:

$$h_p^{rs}(k) \geq 0 \quad \forall r,s,p,k \tag{9}$$

Definitional constraints:

$$u_a(t) = \sum_{rs}\sum_p\sum_k h_p^{rs}(k)\delta_{apk}^{rs}(t) \quad \forall a,t \tag{10}$$

$$c_p^{rs}(k) = \sum_a\sum_t c_a(t)\delta_{apk}^{rs}(t) \quad \forall r,s,p,k \tag{11}$$

3. SOLUTION ALGORITHMS

3.1 THE GENERAL ITERATIVE SCHEME

The iterative scheme seeks to determine $\mathbf{u}^* \in \Omega \subseteq R^n$, such that

$$\mathbf{c}(\mathbf{u}^*)^T(\mathbf{u} - \mathbf{u}^*) \geq 0 \quad \forall \mathbf{u} \in \Omega \tag{12}$$

where $\mathbf{c}$ is a given continuous function from Ω to R^n and Ω is a given closed, convex set. The feasible region Ω is also assumed to be compact and the function $\mathbf{c}(\mathbf{u})$ continuously differentiable.

Assume that there exists a smooth function

$$F(\mathbf{u},\mathbf{u}^m): \Omega \times \Omega \mapsto R^n \tag{13}$$

with the following properties:

(i) $F(\mathbf{u},\mathbf{u}) = \mathbf{c}(\mathbf{u}) \quad \forall \mathbf{u} \in \Omega,$

(ii) for every fixed $\mathbf{u},\mathbf{u}^m \in \Omega$, the $n \times n$ matrix $\nabla_{\mathbf{u}} F(\mathbf{u},\mathbf{u}^m)$ is symmetric and positive definite.

Any function $F(\mathbf{u},\mathbf{u}^m)$ with the above properties generates the following (Dafermos, 1983).

The General Iterative Algorithm

Step 0: Initialization

Start with an $\mathbf{u}^0 \in \Omega$. Set $m = 1$.

Step 1: Construction and Computation

Compute $\mathbf{u}^{m+1}$ by solving the variational inequality problem (VIP):

$$F(\mathbf{u}^{m+1},\mathbf{u}^m)(\mathbf{u} - \mathbf{u}^{m+1}) \geq 0 \quad \forall \mathbf{u} \in \Omega \tag{14}$$

Step 2: Convergence Verification

If $|\mathbf{u}^{m+1} - \mathbf{u}^m| \leq \varepsilon$, for some $\varepsilon > 0$, a prespecified tolerance, then stop; otherwise, set $m = m+1$ and go to Step 1.

Since $\nabla_{\mathbf{u}} F(\mathbf{u},\mathbf{u}^m)$ is assumed to be symmetric and positive definite, the line integral $\oint F(\mathbf{u},\mathbf{u}^m)d\mathbf{x}$ defines a function $z(\mathbf{u},\mathbf{u}^m): \Omega \times \Omega \mapsto R$ such that, for fixed $\mathbf{u}^m \in \Omega$, $z(\bullet,\mathbf{u}^m)$ is strictly convex and

$$F(\mathbf{u},\mathbf{p}) = \nabla_{\mathbf{u}} z(\mathbf{u},\mathbf{p}) \tag{15}$$

Hence, VIP (12) is equivalent to the strictly convex mathematical programming problem

$$\min_{\mathbf{u}\in\Omega} z(\mathbf{u},\mathbf{u}^m) \tag{16}$$

for which a unique solution $\mathbf{u}^{m+1}$ exists. The solution to expression (16) can be computed using any appropriate algorithm. If there is, however, a special-purpose algorithm that takes advantage of the problem's structure, then such an algorithm is usually preferable from an efficiency point of view. Of course, inequality (14) should be constructed in such a manner so that, at each iteration m, this subproblem is easy to solve.

Note that if the sequence $\{\mathbf{u}^m\}$ is convergent, i.e., $\mathbf{u}^m \to \mathbf{u}^*$, as $m \to \infty$, then because of the continuity of $F(\mathbf{u},\mathbf{u}^m)$, inequality (14) yields

$$c(\mathbf{u}^*)^T(\mathbf{u}-\mathbf{u}^*) = F(\mathbf{u}^*,\mathbf{u}^*)(\mathbf{u}-\mathbf{u}^*) \geq \mathbf{0} \quad \forall \mathbf{u}\in\Omega \tag{17}$$

and, consequently, $\mathbf{u}^*$ is a solution to VIP (12). A condition on $F(\mathbf{u},\mathbf{u}^m)$, which guarantees that the sequence $\{\mathbf{u}^m\}$ is convergent is found in Nagurney (1993).

Note that function $F(\mathbf{u},\mathbf{u}^m)$ approximates $\mathbf{c}(\mathbf{u})$ at point $\mathbf{u}^m$ during the iterative process. In the case of nonlinear approximation, the following equation results (and hence the diagonalization method):

$$F(\mathbf{u},\mathbf{u}^m) = (\cdots, F_i(\mathbf{u},\mathbf{u}^m), \cdots)^T : R^n \to R^n \tag{18}$$

where

$$F_i(\mathbf{u},\mathbf{u}^m) = c_i(u_1^m, \cdots, u_{i-1}^m, u_i, u_{i+1}^m, \cdots, u_n^m) \quad \forall i \tag{19}$$

In the case of linear approximation, the following equation results (and hence the projection method):

$$F(\mathbf{u},\mathbf{u}^m) = \mathbf{c}(\mathbf{u}^m) + \frac{1}{\rho}\mathbf{G}(\mathbf{u}-\mathbf{u}^m) \tag{20}$$

where contraction operator $\rho > 0$, matrix $\mathbf{G}$ is fixed, symmetric and positive definite.

Other approximations for $\mathbf{c}(\mathbf{u})$ at point $\mathbf{u}^m$ are possible, such as Newton method, quasi Newton method, symmetric Newton method, linear Jacobi method, the interested reader may refer to Harker and Pang (1990) for details.

3.2 THE PROJECTION METHOD

The optimal solution for a VIP is at the fixed point defined as $\mathbf{u}^* = M_G(\mathbf{u}^*)$, in which $M_G : \Omega \to \Omega$ designates a $\mathbf{G}$ norm continuous mapping function. If the current solution $\mathbf{u}^m \in \Omega$ is not optimal, the contractive mapping of $M_G(\bullet)$ on Ω generates a better solution, i.e., $\mathbf{u}^{m+1} = M_G(\mathbf{u}^m)$. The resulting sequence $\{\mathbf{u}^m\}$ is Cauchy and converges to the unique fixed point characterized by an optimal solution, according to Banach fixed point theorem (Smith, 1979).

It may be convenient to interpret the mapping function $M_G(\bullet)$ as two consecutive stages. In the first stage, the current solution $\mathbf{u}^m \in \Omega$ is mapped into the Euclidean n-dimensional space R^n at $\mathbf{u}^m - \rho\mathbf{G}^{-1}\mathbf{c}(\mathbf{u}^m)$ and then, in the second stage, project back into the feasible region through the **G** norm operation $P_{\Omega,G}\left(\mathbf{u}^m - \rho\mathbf{G}^{-1}\mathbf{c}(\mathbf{u}^m)\right)$, resulting in a better solution $\mathbf{u}^{m+1} \in \Omega$. This procedure may be expressed as follows:

$$\mathbf{u}^{m+1} = M_G\left(\mathbf{u}^m\right) = P_{\Omega,G}\left(\mathbf{u}^m - \rho\mathbf{G}^{-1}\mathbf{c}\left(\mathbf{u}^m\right)\right) \tag{21}$$

where ρ is the contraction operator and $P_{\Omega,G}(\bullet)$ is the projection operator defined by the minimum distance between the current and the improved feasible solution through the **G** norm operation. That is

$$\min_{\mathbf{u}^{m+1}\in\Omega}\left\|\mathbf{u}^{m+1} - \left(\mathbf{u}^m - \rho\mathbf{G}^{-1}\mathbf{c}\left(\mathbf{u}^m\right)\right)\right\|_G \tag{22}$$

By definition $\|\mathbf{x}\|_G = \left(\mathbf{x}^T\mathbf{G}\mathbf{x}\right)^{\frac{1}{2}}$, we have:

$$\min_{\mathbf{u}^{m+1}\in\Omega} \frac{1}{2}\left(\mathbf{u}^{m+1}\right)^T \mathbf{G}\mathbf{u}^{m+1} + \left(\rho\mathbf{c}\left(\mathbf{u}^m\right) - \mathbf{G}\mathbf{u}^m\right)^T \mathbf{u}^{m+1} \tag{23}$$

Since matrix **G** is symmetric and positive definite, then by Green theorem, the optimization problem (23) is identical to the following VIP:

$$\left(\mathbf{c}\left(\mathbf{u}^m\right) + \frac{1}{\rho}\mathbf{G}\left(\mathbf{u}^{m+1} - \mathbf{u}^m\right)\right)^T \left(\mathbf{u} - \mathbf{u}^{m+1}\right) \geq 0 \tag{24}$$

In fact, VIP (24) is essentially equivalent to VIP (14) with *new* travel time function defined by equation (20). The derivation can be derived as follows:

$$\begin{aligned}
&\left(\mathbf{c}\left(\mathbf{u}^m\right) + \frac{1}{\rho}\mathbf{G}\left(\mathbf{u}^{m+1} - \mathbf{u}^m\right)\right)^T \left(\mathbf{u} - \mathbf{u}^{m+1}\right) \\
&= \left(\rho\mathbf{c}\left(\mathbf{u}^m\right) + \mathbf{G}\left(\mathbf{u}^{m+1} - \mathbf{u}^m\right)\right)^T \left(\mathbf{u} - \mathbf{u}^{m+1}\right) \\
&= \left(\mathbf{G}\mathbf{u}^{m+1} + \left(\rho\mathbf{c}\left(\mathbf{u}^m\right) - \mathbf{G}\mathbf{u}^m\right)\right)^T \left(\mathbf{u} - \mathbf{u}^{m+1}\right) \\
&= \tilde{\mathbf{c}}\left(\mathbf{u}^{m+1}\right)\left(\mathbf{u} - \mathbf{u}^{m+1}\right) \geq 0
\end{aligned} \tag{25}$$

where

$$\tilde{\mathbf{c}}\left(\mathbf{u}^{m+1}\right) = \frac{1}{\rho}\mathbf{G}\mathbf{u}^{m+1} + \mathbf{h}_1^m \tag{26}$$

$$\mathbf{h}_1^m = \mathbf{c}\left(\mathbf{u}^m\right) - \frac{1}{\rho}\mathbf{G}\mathbf{u}^m \tag{27}$$

The projection method may be better understood by a graphical illustration shown in Figure 2.

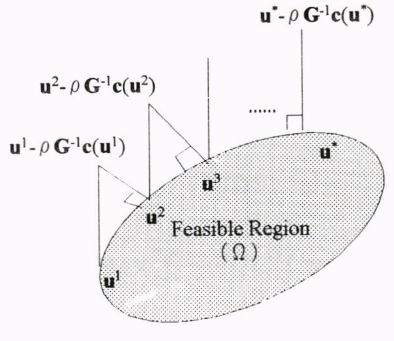

Figure 1: Solution Procedure of Projection Method

We start from an initial solution $\mathbf{u}^1 \in \Omega$ from which the point $\mathbf{u}^1 - \rho\mathbf{G}^{-1}\mathbf{c}(\mathbf{u}^1) \in R^n$ in the Euclidean space is mapped. This point then projects back to the feasible region through the **G** norm operation, resulting in $\mathbf{u}^2 \in \Omega$. Similarly, the current solution $\mathbf{u}^2$ is mapped into the Euclidean space at $\mathbf{u}^2 - \rho\mathbf{G}^{-1}\mathbf{c}(\mathbf{u}^2) \in R^n$ from which a projection onto the feasible region is performed through **G** norm operation, resulting in $\mathbf{u}^3 \in \Omega$. This process continues until the optimal solution **u*** is obtained.

3.3 HYBRID METHOD

The Hybrid method (HB) basically adopts the mapping concept of the Projection method but accommodate the link-based FW method, in contrast with a path-based method as it originally attempted, for the network equilibrium subproblem of the DUO rout choice problem. The proposed HB method is expected to be advantageous of using less memory at the price of longer execution time. The solution procedure of the HB method may be described as follows:

Hybrid Method

Step 0: Initialization.

Step 0.1: Let m=0. Set $\tau_a^0(t) = NINT\left[c_{a_0}(t)\right], \forall a, t$.

Step 0.2: Let n=1. Find an initial feasible solution $\{u_a^1(t)\}$. Compute the associated link travel times $\{c_a^1(t)\}$.

Step 1: *First Loop* Operation.

Let m=m+1. Update the estimated actual link travel times by

$$\tau_a^m(t) = NINT\left[(1-\gamma)\tau_a^{m-1}(t) + \gamma c_a^n(t)\right] \quad \forall a, t \tag{28}$$

where $0 < \gamma \leq 1$. Construct the corresponding feasible time-space network based on the estimated actual link travel times.

Step 2: *Second Loop* Operation.

Step 2.1: Let n=1. Compute and reset the initial feasible solution $\{u_a^n(t)\}$, based on the time-space network constructed by the estimated actual link travel times $\{\tau_a^m(t)\}$.

Step 2.2: Fix the inflows for each physical link other than on the subject time-space link at the current level, yielding the following optimization problem:

$$\min\ z(\mathbf{u}) = \sum_a \sum_t \int_0^{u_a^{n+1}(t)} c_a\left(u_a^n(1), u_a^n(2), \cdots, u_a^n(t-1), \omega\right) d\omega \tag{29}$$

or,

$$\min_{\mathbf{u}^{m+1}\in\Omega} \frac{1}{2}\left(\mathbf{u}^{m+1}\right)^T \mathbf{G}\mathbf{u}^{m+1} + \left(\rho\mathbf{c}(\mathbf{u}^m) - \mathbf{G}\mathbf{u}^m\right)^T \mathbf{u}^{m+1}$$
$$= \sum_b \sum_t \sum_j u_b^2(t) g_{bj} + \rho \sum_a \left(c_a^m(t) - \sum_b g_{ab} u_b^m(t)\right) u_a(t) \tag{30}$$

Flow conservation constraint:

$$\sum_p h_p^{rs}(k) = \bar{q}^{rs}(k) \quad \forall r,s,k \tag{31}$$

Nonnegativity constraint:

$$h_p^{rs}(k) \geq 0 \quad \forall r,s,p,k \tag{32}$$

Definitional Constraints:

$$u_{apk}^{rs}(t) = h_p^{rs}(k)\bar{\delta}_{apk}^{rs}(t) \quad \forall r,s,a,p,k,t \tag{33}$$

$$\bar{\delta}_{apk}^{rs}(t) = \{0,1\} \quad \forall r,s,a,p,k,t \tag{34}$$

$$u_a(t) = \sum_{rs}\sum_p\sum_k h_p^{rs}(k)\bar{\delta}_{apk}^{rs}(t) \quad \forall a,t \tag{35}$$

$$c_p^{rs}(k) = \sum_a\sum_t c_a(t)\bar{\delta}_{apt}^{rs}(t) \quad \forall r,s,p,k \tag{36}$$

where

$$h_a^n(t) = c_a^n(t) - \frac{1}{\rho}\sum_b g_{ab}u_b^n(t) \quad \forall a,t \tag{37}$$

where $0 < \rho < 2\nu/\mu$. μ is any lower bound over u of the minimum eigenvalue of the symmetric part of the Jocobian $\left[\frac{\partial c}{\partial u}\right]$ and ν is any upper bound over u of the maximum eigenvalue of the positive definite symmetric matrix $\left[\frac{\partial c}{\partial u}\right]^T \mathbf{G}^{-1}\left[\frac{\partial c}{\partial u}\right]$. g_{ab} is the element positioned in the ath row and bth column of matrix **G**.

$$\tilde{c}_a^l(t) = \frac{1}{\rho}\sum_b g_{ab}u_b^l(t) + h_a^n(t) \quad \forall a,t \tag{38}$$

Step 3: *Third Loop* Operation.

Solve for the solution, $\{u_a^{n+1}(t)\}$, in optimization problem (30)~(36) by the FW method. Compute the resulting link travel times $\{c_a^{n+1}(t)\}$.

Step 3.1: Let $u_a^l(t) = u_a^n(t), \forall a,t$, and set l=1. Compute the *new* link travel times $\{\tilde{c}_a^l(t)\}$.

Step 3.2: Search for the shortest path for each time-dependent O-D pair based on the *new* link travel times $\{\tilde{c}_a^l(t)\}$ and perform all-or-nothing (AON) assignment, resulting auxiliary flow pattern $\{p_a^l(t)\}$. The search direction becomes:

$$\mathbf{d}^l = \mathbf{p}^l - \mathbf{u}^l \tag{39}$$

Step 3.3: Determine the step size α^n using the bisection method.

$$\min_{0\leq\alpha^n\leq 1} z(\alpha^n) = \sum_a\sum_t \int_0^{u_a^n(t)+\alpha^n d_a^n} \tilde{c}_a\left(u_a^n(1), u_a^n(2), \cdots u_a^n(t-1), \omega\right)d\omega \tag{40}$$

or,

$$\min_{0\leq\alpha^l\leq 1} \frac{1}{2}\left(\mathbf{u}^{l+1}\right)^T \mathbf{G}\mathbf{u}^{l+1} + \left(\rho\mathbf{c}(\mathbf{u}^l) - \mathbf{G}\mathbf{u}^l\right)^T \mathbf{u}^{l+1}$$

$$= \frac{1}{2}\sum_b\sum_t u_b^{l+1}(t)\left(\sum_j u_j^{l+1}(t)g_{bj}\right) + \sum_a\sum_t u_a^{l+1}(t)\left(\rho c_a^l(t) - \sum_b g_{ab}u_b^l(t)\right) \tag{41}$$

Step 3.4: Update the link inflows and calculate the corresponding link travel times $\{c_a^{l+1}(t)\}$:

$$u_a^{l+1}(t) = u_a^l(t) + \lambda^l\left(p_a^l(t) - u_a^l(t)\right) \quad \forall a,t \tag{42}$$

Step 3.5: Covergence check for the *third loop* operations.

If $\frac{u_a^{l+1}(t) - u_a^l(t)}{u_a^{l+1}(t)} \le \varepsilon_3, \forall a,t$, let $u_a^{n+1}(t) = u_a^{l+1}(t), \forall a,t$ and $c_a^{n+1}(t) = c_a^{l+1}(t), \forall a,t$, continue; otherwise. let $l = l+1$, go to Step 3.2.

Step 4: Convergence Check for the *Second Loop* Operation.

If $u_a^{n+1}(t) \approx u_a^n(t), \forall a,t$, $\frac{u_a^{n+1}(t) - u_a^n(t)}{u_a^{n+1}(t)} \le \varepsilon_2, \forall a,t$ go to Step 5; otherwise, set n=n+1, go to Step 2.2.

Step 5: Convergence Check for the *First Loop* Operation.

If $\tau_a^m(t) = NINT\left[c_a^{n+1}(t)\right], \forall a,t$ stop; the current solution is optimal. Otherwise, set n=n+1, and go to Step 1.

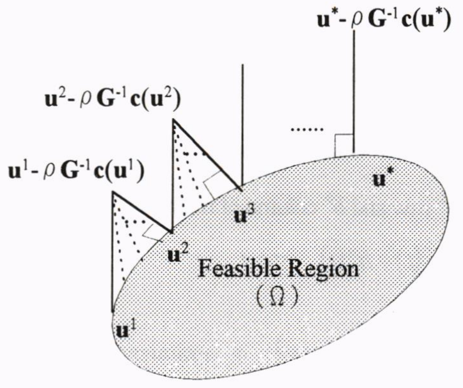

Figure 2: Solution Procedure of Hybrid Method

The $NK \times NK$ symmetric positive definite matrix **G** in equation (30) (and throughout the paper) is the Jacobian matrix of the travel time function. For simplicity, it is set as a multiple of an unity matrix, i.e., **G**=$a \times$[**I**], $a > 0$. Symbol K is the maximum number of time intervals by which all the departures must have reached at their destinations. N is the total number of physical links. The contraction operator ρ in equation (30) (and throughout the paper) must be set in the range of $\left[0, \frac{2\mu}{v}\right]$ so as to optimally converge (Defamos, 1980). Note that μ is any lower bound over u of the minimum eigenvalue of the symmetric part of the $NK \times NK$ Jocobian $\left[\frac{\partial c}{\partial u}\right]$ and υ is any upper bound over u of the maximum eigenvalue of the $NK \times NK$ positive definite symmetric matrix $\left[\frac{\partial c}{\partial u}\right]^T \mathbf{G}^{-1}\left[\frac{\partial c}{\partial u}\right]$. Except for the respective diagonal elements, $\left[\frac{\partial c_a}{\partial u_a}\right]^T \mathbf{G}^{-1}\left[\frac{\partial c_a}{\partial u_a}\right]$ and $\left[\frac{\partial c_a}{\partial u_a}\right]$, all the off-diagonal elements are zerovalued. To reduce computation effort, the two matrices with dimensions $NK \times NK$ matrix are first decomposed into N smaller matrices with dimensions $K \times K$ matrix and then calculate eigenvalues independently. We first chose $\rho = \frac{\mu}{\nu}$ for the first version of HB (hence named HB1) to guarantee convergence, though not necessarily converging fast. An alternative approach, hereafter referred to as HB2, is simply set the contraction operator as a constant, for example, $\rho = 0.5$.

4. NUMERICAL EXAMPLE

A simple network shown in Figure 3 is used for testing. This test network contains of 6 links and 5 nodes, in which nodes 1 and 3 are the origins, node 5 is the destinations, and nodes 2 and 4 are intermediate nodes.

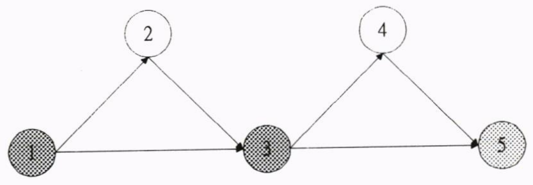

Figure 3: Test Network 1

The assumed time-dependent origin-destination (O-D) demands are shown in Table 1.

Table 1: time-dependent origin-destination (O-D) demands for Test Network 1

O-D	Departure Interval			
pair	k=1	k=2	k=3	k=4
1-5	15	5	0	0
3-5	0	0	10	20

The adopted dynamic travel time function is arbitrarily constructed as follows.

$$c_a(t) = 1 + 0.01(u_a(t))^2 + 0.01(x_a(t))^2 \quad \forall a,t \tag{43}$$

where $u_a(t)$ denotes inflows on link a during interval t, and $x_a(t)$ indicates the number of vehicles on link a during interval t.

The convergence criteria for the three loops are summarized in Table 2. In the first loop, the stopping criterion requires that the actual link travel times must be equal to the corresponding *discretized* link travel times. If the number of iterations exceeds 30, the algorithmic procedure is terminated and no convergent solution is available. In the second loop, the stopping criteria requires for each time-space link that the inflow difference between two consecutive iteration must be less than a tolerance of 0.00003. Otherwise, the solution obtained at iteration 1300 is deemed as optimal. In the third loop, the stopping criteria requires for each time-space link that the inflow difference between two consecutive iteration must be less than a tolerance of 0.003. If the optimal solution cannot be obtained within 100 iterations, the last solution is taken for the subsequent use.

Table 2: Convergence Criteria for the Three Loops

	First Loop	Second Loop	Third Loop
Convergence Criterion	$\tau_a(t) = NINT[c_a(t)]$	$\frac{u_a^{n+1}(t)-u_a^n(t)}{u_a^n(t)} \leq 0.00003$	$\frac{u_a^{l+1}(t)-u_a^l(t)}{u_a^l(t)} \leq 0.003$
Maximum Number of Iterations	30	1300	100

A computer program coded with *Turbo* c++ 3.0, compiled with *small* mode, was executed on *Pentium*-100 with 32M *RAM*. The route travel times for the test network 1 by the four algorithms are summarized in Tables 3~6. The equal and minimum route travel times imply that the dynamic equilibrium conditions are completely satisfied. Slight differences are attributed to round-off errors.

Table 3: Route Travel Times for Test Network 1 by Diagonalization Method

Route	Departure Interval			
	$k=1$	$k=2$	$k=3$	$k=4$
$1\to2\to3\to4\to5$	5.8632	8.0928	NA	NA
$1\to3\to4\to5$	5.8631	8.0928	NA	NA
$1\to2\to3\to5$	5.8635	8.0946	NA	NA
$1\to3\to5$	5.8634	8.0946	NA	NA
$3\to4\to5$	NA	NA	3.5882	5.7320
$3\to5$	NA	NA	3.5885	5.7338

Table 4: Route Travel Times for Test Network 1 by Hybrid Method (HB1)

Route	Departure Interval			
	$k=1$	$k=2$	$k=3$	$k=4$
$1\to2\to3\to4\to5$	5.8628	8.0923	NA	NA
$1\to3\to4\to5$	5.8624	8.0921	NA	NA
$1\to2\to3\to5$	5.8642	8.0951	NA	NA
$1\to3\to5$	5.8638	8.095	NA	NA
$3\to4\to5$	NA	NA	3.5876	5.7314
$3\to5$	NA	NA	3.5890	5.7343

Table 5: Route Travel Times for Test Network 1 by Hybrid Method (HB2)

Route	Departure Interval			
	k=1	k=2	k=3	k=4
1→2→3→4→5	5.8646	8.0900	NA	NA
1→3→4→5	5.8641	8.0988	NA	NA
1→2→3→5	5.8626	8.0875	NA	NA
1→3→5	5.8621	8.0963	NA	NA
3→4→5	NA	NA	3.5894	5.7344
3→5	NA	NA	3.5874	5.7319

Table 6: Route Travel Times for Test Network 1 by Gradient Projection Method

Route	Departure Interval			
	k=1	k=2	k=3	k=4
1→2→3→4→5	5.8634	8.0938	NA	NA
1→3→4→5	5.8634	8.0938	NA	NA
1→2→3→5	5.8634	8.0938	NA	NA
1→3→5	5.8634	8.0938	NA	NA
3→4→5	NA	NA	3.5884	5.7330
3→5	NA	NA	3.5884	5.7330

5. COMPUTATIONAL EFFICIENCY

To compare computational efficiency among the four algorithms ,four more test networks, as plotted in Figure 4~7, were used. The assumed corresponding time-dependent O-D demands are tabulated in Table 7. The other parameters settings are identical to those given in section 4. The used performance measures are threefold: total execution time, computer memory requirement, and quality of solution.

The total execution time is recorded by accumulating the CPU times for all modules, but the I/O times are not included. The computer memory requirement is calculated by adding up the memory usage for the main module and the subroutine that consumed the computer memory units most. The quality of solution is measured by the maximum difference, $\varepsilon_{\max}$, between the largest and the smallest used route travel times for all time-dependent O-D pairs. In general, the larger the value of $\varepsilon_{\max}$, the better quality of the solution is.

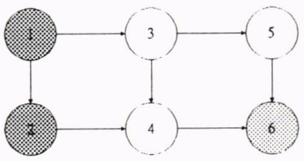

Figure 4: Test Network 2

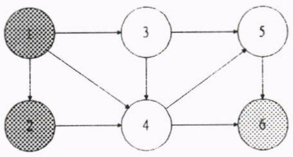

Figure 5: Test Network 3

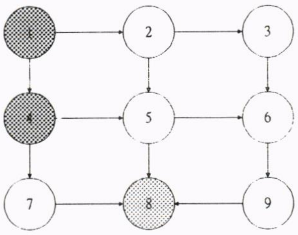

Figure 6: Test Network 4

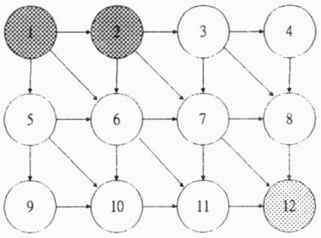

Figure 7: Test Network 5

Table 7: Time-Dependent O-D Demands for Test Networks 2~4

Test Network	O-D Pair	Departure Interval			
		k=1	*k*=2	*k*=3	*k*=4
Network 2	1→6	20.0	NA	15.0	NA
	2→6	NA	20.0	NA	15.0
Network 3	1→6	25.0	NA	NA	NA
	2→6	NA	15.0	NA	NA
Network 4	1→8	15.0	NA	20.0	NA
	4→8	NA	NA	NA	10.0
Network 5	1→12	10.0	NA	NA	NA
	2→12	NA	NA	15.0	NA

The computational efficiency of the four algorithms is compared with the five test networks in terms of three performance measures, which are summarized in Table 8.

Table 8: Computational Efficiency among the Four Methods

Test Network	Measure	Algorithm			
		DG	HB1	HB2	GP
Network 1	Total Execution Time (sec)	136.98	140.33	147.42	4.34
	Maximum Difference of Used Route Travel Times	0.0018	0.0029	0.0025	0
	Memory Requirement (byte)	11120	29858	13472	14120
	Number of Time Intervals	20	20	20	20
	Contraction Operator	-	Varied	0.7	-
	Iterations of the First Loop (Time per Iteration, sec)	7 (19.57)	7 (20.05)	7 (21.06)	7 (0.62)
	Iterations of the First Loop (Time per Iteration, sec)	1300 (0.015)	1300 (0.015)	1300 (0.016)	42 (0.015)
Network 2	Total Execution Time (sec)	334.73	415.72	350.33	1.15
	Maximum Difference of Used Route Travel Times	0.0013	0.0097	0.0029	0.0001
	Memory Requirement (byte)	13760	36258	16704	17044
	Number of Time Intervals	22	22	22	22
	Contraction Operator	-	Varied	0.3	-
	Iterations of the First Loop (Time per Iteration, sec)	13 (25.75)	15 (27.71)	13 (26.95)	13 (0.08)
	Iterations of the First Loop (Time per Iteration, sec)	1300 (0.020)	1300 (0.021)	1300 (0.021)	5 (0.017)
Network 3	Total Execution Time (sec)	67.14	69.56	72.53	0.16
	Maximum Difference of Used Route Travel Times	0.0209	0.001	0.001	0.0001
	Memory Requirement (byte)	8120	13974	9880	9676
	Number of Time Intervals	10	10	10	10
	Contraction Operator	-	Varied	1	-
	Iterations of the First Loop (Time per Iteration, sec)	6 (11.19)	6 (11.59)	6 (12.09)	6 (0.027)
	Iterations of the First Loop (Time per Iteration, sec)	1300 (0.009)	1300 (0.009)	1300 (0.009)	3 (0.009)
Network 4	Total Execution Time (sec)	197.64	198.30	206.65	31.43
	Maximum Difference of Used Route Travel Times	0.0011	0.002	0.0009	0.0001
	Memory Requirement (byte)	15000	27004	18328	17264
	Number of Time Intervals	15	15	15	15
	Contraction Operator	-	Varied	0.3	-
	Iterations of the First Loop (Time per Iteration, sec)	7 (28.23)	7 (28.33)	7 (29.52)	7 (4.49)
	Iterations of the First Loop (Time per Iteration, sec)	1300 (0.022)	1300 (0.022)	1300 (0.023)	42 (0.011)

Network 5	Total Execution Time (sec)	115.99	105.00	118.68	1.10
	Maximum Difference of Used Route Travel Times	0.0006	0.002	0.0002	0.0001
	Memory Requirement (byte)	21296	31566	26288	23116
	Number of Time Intervals	12	12	12	12
	Contraction Operator	-	Varied	1.0	-
	Iterations of the First Loop (Time per Iteration, sec)	4 (29.00)	4 (26.25)	4 (29.67)	4 (0.28)
	Iterations of the First Loop (Time per Iteration, sec)	1000 (0.029)	1000 (0.026)	1000 (0.030)	9 (0.032)

The results show that:

1. Total execution time: The GP method outperforms the other three methods. The DG, HB1, and HB2 algorithms are ranked the second, third and fourth respectively, but about the same order of magnitude. The DG method embeds the FW method during its solution procedure. Since FW method is known to be slow at the neighborhood of the optimal solution, therefore, a longer execution time results. The HB method, either version 1 or version 2, requires a computation for the *new* travel time functions at each third loop iteration, which inevitably involves lots of $K \times K$ matrix operations. For the HB1, more arithmetic operations are needed for computing the value of the contraction operator. For the HB2, the contraction operator is set as a constant by taking one of 0.3, 0.5, 0.7, and 1 values to save the computational time for calculating eigenvalues of the matrices. However, the constant contraction operator may deviate from the optimal value largely and hence offset the benefits of fewer arithmetic operations.
2. Memory requirement: The most economic use of computer memory is attributed to the DG method. Nevertheless, its magnitude is about the same order as for the GP and HB2 methods. The most memory-demanding method is attributed to the HB1 in which computations for the values of the *new* travel time functions and the contraction operator involve many $K \times K$ matrix operations. The lower the value of K, the less computational effort is needed in calculating the corresponding eigenvalues.
3. Quality of solution: Of the four methods the GP results in the most accurate solution, and the method of HB2, DG, and HB1 are ranked as the second, third, and fourth, respectively.

6. CONCLUDING REMARKS

In this paper, the two versions of the HB method were proposed to solve the DUO route choice problem. Though the HB method had performed better for the static user-optimal route choice problem, unfortunately, this result cannot readily apply to its dynamic counterpart. None of the three performance measures, i.e., total execution time, memory requirement, and quality of solution is preferable to the DG and GP methods. The slow convergence of the HB method may be attributed to the improper settings for the contraction operator $\rho \in \left[0, \frac{2\mu}{\nu}\right]$ and matrix $\mathbf{G}$ (an approximation of Jacobian matrix of the travel time function), with which an intensive computation effort is needed. Higher memory requirement is mainly due to the matrix operations for calculating the maximum and minimum eigenvalues.

Lower degree of precision for the final solution is common to any solution procedure that embeds the FW method. Before a conclusion assertion can be made about the computational efficiency of the HB method, the following issues need to be further explored:

1. How to determine the optimal settings for the contraction operator ρ and matrix **G,** considering their combined effect in the second term of the *new* travel time function $\tilde{\mathbf{c}}_a\left(u_a^m\right) = \mathbf{c}_a\left(u_a^{m-1}\right) + \frac{1}{\rho}\mathbf{G}\left(\mathbf{u}_a^m - \mathbf{u}_a^{m-1}\right)$.
2. How to compute the eigenvalues of a large matrix efficiently? Is there any approach other than that proposed in this study?

Moreover, the computational requirements of different types of travel time functions (Harker and Pang, 1990) need to be compared for real size transportation networks.

ACKNOWLEDGMENT

This work was supported in part by the National Science Council, Taiwan (R.O.C.), through grant NSC-87-2211E-008-011 awarded to the first author.

REFERENCES

1. Chen, H.K., 1998. **Dynamic Travel Choice Models : A Variational Inequality Approach**. Springer. Verlag. Berlin.
2. Harker, P. T. and Pang J.S., 1990. Finite-Dimensional Variational Inequality and Nonlinear Complementarity Problem: A Survey of Theory, Algorithms and Applications. **Mathematical Programming**, **48**,161-220.
3. Nagurney, A., 1983. **Stability, Sensitivity Analysis, and Computation of Competitive Network Equilibria**, Ph.D Dissertation, Div. of Applied Mathematics, Brown University, 1983.
4. Nagurney, A., 1993. **Network Economics: A Variational Inequality Approach**, Kluwer Academic Publishers, Massachusetts.
5. Dafermos, S., 1980, Traffic Equilibrium and Variational Inequalities, **Transportation Science**, **14**, 42-54.
6. Dafermos, S., 1983, An Iterative Scheme for Variational Inequalities, **Mathematical Programming**, **26**,40-47.
7. Smith, M., 1979, The Existence Uniqueness and Stability of Traffic Equilibria, **Transportation Research-B**, **13**B, 295-304.
8. Fukushima,M., 1986, A Relaxed Projection Method for Variational Inequalities, **Mathematical Programming**, **35**,58-70.

APPENDIX

Symbols used in this paper are summarized as follows:

a	: link designation
$c_a(t)$	: travel time for link a during time interval t
$c_p^{rs}(k)$	: travel time for route p between O-D pair rs during time interval k
$\mathbf{G}$	: symmetric, positve definite matrix
$h_p^{rs}(k)$	: departure flow rate on route p from origin r toward destination s during time interval k
j	: node designation
k	: time interval designation which usually denotes the departure time interval for a route
K	: total number of time intervals
p	: route designation
r	: origin designation
s	: destination designation
$\mathbf{u}$	: vector of link inflow rates
$u_a(t)$	: inflow rate into link a during time interval t
$u_{apk}^{rs}(t)$	: part of inflow rate for link a during time interval t that is departing origin r over route p toward destination s during time interval k
$v_a(t)$	: exit flow rate from link a during time interval t
$x_a(t)$	: number of vehicles on link a at the beginning of time interval t
$\delta_{apk}^{rs}(t)$	: 1, if inflow rate on link a during time interval t departs from origin r over route p toward destination s during time interval k; otherwise, 0
α	: move size
$\tau_a(t)$	: actual travel time for link a during time interval t
$\pi^{rs}(k)$	: minimal route travel time between O-D pair rs during time interval k
Ω	: feasible region
ρ	: contraction operator