

Finding the ϵ -user Equilibrium Solution Using an Augmented Frank-Wolfe Algorithm

Hsun-Jung Cho · Yu-Kuang Chen

Published online: 23 June 2009
© Springer Science + Business Media, LLC 2009

Abstract The Frank-Wolfe algorithm has been extensively adopted in recent decades to solve the user-equilibrium problem because of its simple structure and low memory requirements. However, Dial observed that the results obtained by the Frank-Wolfe algorithm differed markedly from the B algorithm of Dial in terms of link flows, and the results obtained via Frank-Wolfe algorithm could be incorrect. This study attempts to provide a clear example showing that the Frank-Wolfe algorithm has difficulty in achieving an ϵ -user equilibrium state when ϵ is sufficiently small. An Augmented Frank-Wolfe algorithm is presented that overcomes the weaknesses of the conventional arc-based Frank-Wolfe algorithm.

Keywords Frank-Wolfe · User-equilibrium

1 Introduction

The traffic assignment problem (denoted as TAP) is a key step in transportation planning. Two alternative criteria based on these journey times can be used to determine the distribution on the routes; namely, the user-equilibrium (denoted as UE) and system optimum (denoted as SO) principles (John Glen Wardrop 1952). As is generally believed, for networks in the UE state, all used routes share the same travel costs for the same origin-destination (denoted as

H.-J. Cho (✉) · Y.-K. Chen
Department of Transportation Technology and Management,
Nation Chiao-Tung University, Hsinchu City 300, Taiwan
e-mail: hjcho@cc.nctu.edu.tw

Y.-K. Chen
e-mail: ykchen.tem85@nctu.edu.tw

OD) pair, while unused routes always have a higher cost than used ones. The UE problem was first formulated as a mathematical programming problem, as shown in Eq. (1) a situation that can be traced back to Beck et al. (1956).

$$\begin{aligned}
 \min Z &= \sum_{a \in A} \int_0^{f_a} C_a(w)dw, \\
 \text{s.t. } f_a &= \sum_r \sum_s \sum_p h_p^{rs} \delta_{a,p}^{rs} \quad \forall a \in A, \\
 \sum_p h_p^{rs} &= q^{rs} \quad \forall r \in R, s \in S, \\
 h_p^{rs} &\geq 0, \\
 \delta_{a,p}^{rs} &= \begin{cases} 1, & \text{if a path } p \text{ passes arc } a, \\ 0, & \text{otherwise,} \end{cases}
 \end{aligned} \tag{1}$$

where

- A, R, S network’s arc, origins, and destinations,
- Z the value of the objective function,
- f_a total flow on arc a ,
- $C_a(f_a)$ cost of traversing arc a when it carries flow f_a ,
- h_p^{rs} flow on path p from $r \in R$ to $s \in S$,
- q^{rs} number of trips going from $r \in R$ to $s \in S$.

Since the combination of cost functions mentioned above has the convex property, the optimal solution can be achieved at x^* such that $\nabla Z(x^*)'(x - x^*) \geq 0, \forall x \in \Omega$, where Ω is the feasible set. The application of the Frank-Wolfe algorithm (Frank and Pinder 1956) to the solution of solving the transportation network equilibrium was first proposed by Bruynooghe et al. (1968) and applied by Murchland (1969). Furthermore, Mguyen (1974) proposed using the convex simplex method to solve the UE problem. LeBlanc et al. (1975) coded and tested the Frank-Wolfe (denoted as FW) algorithm for a small city (Sheffi 1985). The FW algorithm has extensive applications because of its simple structure.

Recently, Dial (2006) observed that the UE solutions of Frank-Wolfe differ from the B algorithm, and claimed that the solutions of Frank-Wolfe fail to satisfy the UE principle when the stopping criteria are met. The work of Dial has forced numerous researchers to reexamine the relationships between the UE and FW algorithm. This study makes the following three observations. First, the arc-based FW algorithm cannot be used for examination whether paths belong to the UE solutions, since the arc-based FW algorithm carries no path information. Second, not all of the assigned paths eventually become UE paths. Third, removing a non-UE path facilitates a significant reduction in computational burden. These observations are illustrated by using a Braess network, as shown below.

The remainder of this paper is organized as follows. Section 2 presents stopping criteria that are applied to the the arc-based FW algorithm, and attempts to demonstrate that the solutions obtained using the FW algorithm could not achieve an $\epsilon - \text{UE}$ state when stopping criteria for terminating this iterative procedure are met. The convergence criteria, $\epsilon - \text{UE}$, means that the costs of all used paths of each OD pair lie within a given ϵ gap of the cheapest, provided ϵ is sufficiently small. Section 3 demonstrates that path information is hidden in an all-or-nothing (AON) assignment step. Furthermore, such path information could assist the arc-based FW algorithm in examining whether a path belongs to the UE solutions. Section 4 introduces the well-known example presented by Braess (1968) to illustrate the difficulty of achieving an $\epsilon - \text{UE}$ state, and Section 5 demonstrates the flow of the proposed Augmented Frank-Wolfe (denoted as AFW) algorithm. The AFW algorithm is designed to eliminate non-UE path flows. Section 6 compares the AFW with the arc-based FW algorithm and details the relevant experimental results. Concluding remarks are finally made in Section 7, along with recommendations for future research.

2 FW algorithm stopping criteria

Notably, most path-based algorithms adopt the same stopping criterion, $\epsilon - \text{UE}$, which is used to examine whether a network achieves an UE status, since $\epsilon - \text{UE}$ is an intuitive rule for examining whether a network approaches the UE state. However, owing to the arc-based FW algorithm not carrying any path information, numerous stopping criteria are introduced to terminate an iterative procedure. The most common stopping criterion, relative gap (denoted as RG), aims to determine the improvement in the objective function (Boyce et al. 2004), and is defined as

$$RG^{(n)} = \frac{-gap^{(n)}}{|BLB|} \geq 0, \quad RG^{(n)} \leq \epsilon, \tag{2}$$

where

$$gap^{(n)} = \sum_a C_a(f_a^{(n)}) \times (y_a^{(n)} - f_a^{(n)}) \leq 0,$$

$$LB^{(n)} = Z^{(n)} + gap^{(n)} = \sum_a \int_0^{f_a^{(n)}} C_a(w)dw + \sum_a C_a(f_a^{(n)}) \times (y_a^{(n)} - f_a^{(n)}),$$

$y^{(n)}$ arc flow resulting from an AON assignment at the n -th iteration, or an auxiliary feasible direction,

BLB is the best lower bound.

The second stopping criterion is based on the percentage change in the objective function, and can be described as

$$\left(\frac{Z^{(n)} - Z^{(n-1)}}{Z^{(n)}} \right) \times 100\% \leq \epsilon. \tag{3}$$

The third equation is the maximum percentage change in link flows, and is described as follows.

$$\max_n \left(\frac{f_a^{(n)} - f_a^{(n-1)}}{f_a^{(n)}} \right) \times 100\% \leq \epsilon. \tag{4}$$

The fourth stopping criterion calculates root mean square, and is presented as follows

$$\text{RMS}^{(n)} = \sqrt{\frac{1}{N} \sum_{n=1}^N \frac{(f_a^{(n)} - y_a^{(n)})^2}{0.5(f_a^{(n)} + y_a^{(n)})}} \leq \epsilon. \tag{5}$$

The stopping criteria mentioned above essentially share similar features, including objective function or link flows, and the change in the features is expected to be below a certain threshold at successive iterations. Notably, none of these stopping criteria indicate the closeness of an obtained solution to the UE state. Restated, the stopping criteria provide a termination condition for an infinite step algorithm, but whether the paths belong to the UE solutions is unclear.

3 Path information hidden in the all-or-nothing assignment

The FW algorithm updates network flows using an arc perspective, and lacks carrying path information. However, to examine whether a network could achieve an $\epsilon - \text{UE}$ state, path information is required for examining the definition of UE; restated, journey times on all routes actually used are equal, and less than those experienced by a single vehicle on any unused route (John Glen Wardrop 1952). That is, the arc-based FW algorithm cannot determine whether a path belongs to the UE solutions. Consequently, path information is introduced to examine whether a network can achieve an $\epsilon - \text{UE}$ state. In fact, path information is hidden in an all-or-nothing step. Equation (6) shows the arc flow update scheme in the FW algorithm, and updates arc flows according to an appropriate step size α .

$$f_a^{(n+1)} = f_a^{(n)} + \alpha^{(n)} \times (y_a^{(n)} - f_a^{(n)}), \quad \forall a \in A, \tag{6}$$

where $y_a^{(n)} - x_a^{(n)}$ denotes the searching direction at the n -th iteration, and y_a is represents assigned arc flow using an AON assignment step. In fact, Eqs. (6) and (7) yield the same results, with the latter being formulated from the path perspective.

$$h_p^{rs(n+1)} = h_p^{rs(n)} + \alpha^{(n)} \times (q^{rs} \eta_p^{rs(n)} - h_p^{rs(n)}), \tag{7}$$

where p is a shortest path, and η_p is an indicate function. If $p \in P_{rs}$, $\eta_p^{rs(n)} = 1$; otherwise, $\eta_p^{rs(n)} = 0$. Moreover, $q^{rs} \eta_p^{rs(n)}$ indicates an improving direction via an

AON assignment, and updates flows in the path perspective. Equation (6) is derived from Eq. (7). The deduced procedure is detailed as follows.

$$\begin{aligned}
 h_p^{rs(n+1)} &= h_p^{rs(n)} + \alpha^{(n)} \times (q^{rs} \eta_p^{rs(n)} - h_p^{rs(n)}), \\
 \Rightarrow \sum_{rs} \sum_p h_p^{rs(n+1)} \delta_{a,p}^{rs} &= \sum_{rs} \sum_p h_p^{rs(n)} \delta_{a,p}^{rs} + \sum_{rs} \sum_p \alpha^{(n)} (q^{rs} \eta_p^{rs(n)} - h_p^{rs(n)}) \delta_{a,p}^{rs}, \\
 f_a &= \sum_{rs} \sum_p h_p^{rs} \delta_{a,p}^{rs} \text{ and } \sum_{rs} \sum_p q^{rs} \eta_p \delta_{a,p}^{rs} = y_a \\
 \Rightarrow f_a^{(n+1)} &= f_a^{(n)} + \alpha^{(n)} \times (y_a^{(n)} - f_a^{(n)}), \forall a \in A
 \end{aligned}$$

Equations (6) and (7) illustrate an AON assignment step from different perspectives. Equation (7) demonstrates that these non-shortest path flows in the same OD pair are partially switched to a shortest path. Consequently, once a path bears a flow, that path is selected as the shortest. To examine whether a path belongs to the UE solutions, path information is gathered to determine whether a path belongs the UE solutions. Chen and Jayakrishnan (1996) pointed out that paths obtained by the FW algorithm had worse quality than those of other origins; Restated, an obtained shortest path may not ultimately belong to the UE solutions. A Braess network is presented to demonstrate the insight of a non-UE path in the FW algorithm, and demonstrates that the mentioned stopping criterion, for example RG, are not meaningfully associated with UE state.

4 Cause of a non-UE path

The example of Braess network, as shown in Fig. 1, is employed to demonstrate the insight of a non-UE path, and contains one OD pair with 20 trip units, and five one-way links associated with linear travel costs. Additionally, Fig. 2 demonstrates that a maximum of three paths exist in the Braess network, and

Fig. 1 Braess network

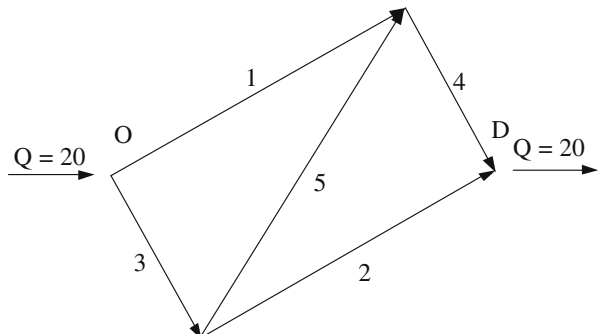
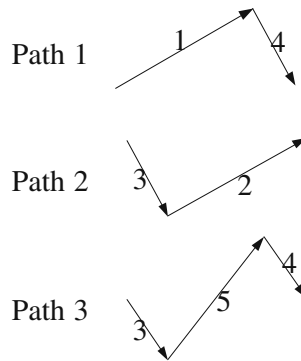


Fig. 2 Path set in a Braess network



the analytical UE solution to Braess network exists because of links' special structure and monotone increasing property shown in Table 1.

Pas and Principio (1997) suggested the analytical solution to Braess network. The solution demonstrates that Path 1 and 2 bear equal flows when trip demand is greater than 8.89 or less than 2.58. Therefore, based on the analytic solution, Paths 1 and 2 bear ten trip units, respectively, when OD demand is twenty trip units.

For simplicity, a stopping criterion, RG, is adopted in place of the other stopping criteria, and Table 2 lists numerical results obtained by the FW algorithm. The numerical results are displayed at every 1/10 ratio of RG, and demonstrate that the numerical objective value closely approaches the exact solution 2100 following 10404 iterations. A small movement in step size and *RG* indicates only a slight permissible improvement, as well as an intuitive belief that the obtained solution using the FW algorithm is close to the $\epsilon - UE$ state. However, an approximate solution using the FW algorithm is far from the actual $\epsilon - UE$ solution, as listed in the last column in Table 2. The maximal cost difference of the cheapest path indicates the cost gap between the max-cost and min-cost paths of the same OD pair. Table 2 reveals that a cost gap in the Braess example always exceeds 50 units when the FW algorithm stopped within a finite number of steps. Furthermore, Table 3 reveals that the cost gap primarily arises from Path 3, where Path 3 is known as a non-UE path based on the analytical solution. Notably, Table 3 displays not only path costs, but also path flows. Such path information enables the observation of a non-UE path property; that is, Path 3 always has higher past path cost and lower path flow.

Table 1 Link costs

Link Id	Cost structure
1	$50 + f_1$
2	$50 + f_2$
3	$10 f_3$
4	$10 f_4$
5	$10 + f_5$

Table 2 The numerical results of the FW algorithm

Iteration n	Step size α	RG	Objective value	Maximal cost difference of the cheapest path
8	0.07552433	0.090864293	2174.94881	62.89366009
96	0.00934124	0.009930888	2110.12115	51.80946324
1031	0.00095272	0.000999039	2101.04641	50.18827261
10404	9.63211E-05	9.99917E-05	2100.10544	50.01907595

Boyce et al. had ever posited that a value of RG below 0.01% would be sufficient to achieve an $\epsilon - UE$ state, but the numerical results above indicate that this viewpoint may be incorrect. Since an $\epsilon - UE$ state cannot be achieved within a finite number of steps in the network of Braess. The FW algorithm suffered a certain weakness with regard to in obtaining the $\epsilon - UE$ state solution.

Based on the update scheme illustrated in Eqs. (6) or (7), the FW algorithm transfers higher-cost path flows to the cheapest path associated with an appropriate step size α . A step size is derived by calculating $\nabla Z = 0$. A sequence of step size converges to zero, and a step size does not suddenly equal one. However, the best method for the FW algorithm to remove a flow on the non-UE Path 3 is to permit the step size to equal one, such that a non-UE path flow can be entirely obviated. Restated, provided the existence of a flow in a non-UE path, indicating that the FW algorithm cannot achieve an $\epsilon - UE$ state.

Definition 1 Assume that for each path $p_i, i = 1, \dots, n$, there associate with a positive path flow $h_i > 0$, and a positive cost function $\pi_i \in R^+$. Suppose that there exists a particular k such that $\pi_k(h_1, \dots, h_n) > \pi_j(h_1, \dots, h_n)$ for all $j \neq k$. Let $\delta_j \geq 0$ for $j \neq k$ such that $\sum_{j \neq k} \delta_j = h_k, h_i^* = h_i + \delta_i$ for $j \neq k$ and $h_k^* = 0$.

If $\pi_k(h_1^*, \dots, h_n^*) > \pi_j(h_1^*, \dots, h_n^*)$ for any such combination of δ_j , then path p_k is a non-UE path associated with a non-UE flow h_k .

A path p_k transfers an entire path flow h_k to the other cheaper paths p_j for all $j \neq k$, and still has a higher cost than the other paths p_j . Based on the UE principle, the amount h_k ought to switch to the other cheaper paths.

Theorem 1 *If we start with a solution containing a positive flow on a non-UE path, then any finite step termination will be unable to remove an entire non-UE path flow unless step size α equal to one.*

Table 3 Path information of the FW algorithm after 10400 iterations

Path	Path flow	Path cost
1	9.9985525	160.0040249
2	9.9994528	160.0139276
3	0.0019947	210.0219419

Proof Based on Eq. (7), a path flow update scheme can be illustrated as follows.

$$h_p^{(n+1)} = h_p^{(n)} + \alpha^{(n)} \times (q\eta_p^{(n)} - h_p^{(n)})$$

Supposing that a particular path p is a non-UE path, then an indicate function $\eta_p^{(n)}$ will equal zero; otherwise, $\eta_p^{(n)}$ will equal one. Since a non-UE path is always more expensive, and it is not selected as the shortest path. Consequently, a path flow update scheme for a non-UE path can be rewritten as

$$h_p^{(n+1)} = h_p^{(n)} + \alpha^{(n)} \times (-h_p^{(n)}) \tag{8}$$

Based on Eq. (8), the best way for the FW algorithm to remove a non-UE path flow is obviously to permit step size $\alpha^{(n)}$ to equal one. Restated, a non-UE path flow h_p can be completely transferred to the other paths unless step size α equal to one. □

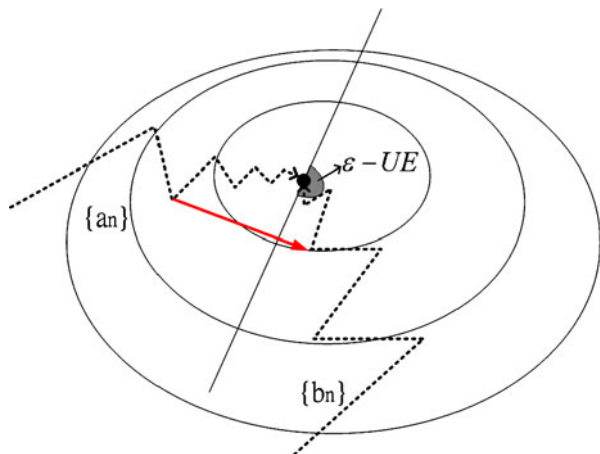
Corollary 1 *Given the existence of a non-UE path flow, the Frank-Wolfe algorithm can not reach an $\epsilon - UE$ state, provided ϵ is sufficiently small.*

5 Augmented Frank-Wolfe algorithm

To overcome the weakness of the FW algorithm, an Augmented Frank-Wolfe (denoted as AFW) algorithm, is proposed. The design of the AFW algorithm aims to eliminate the non-UE path flow. Such a movement can be represented by a red line in Fig. 3, and the convergent sequence switches from $\{a_n\}$ to $\{b_n\}$.

The AFW algorithm exhibits certain similarities in path work to the OD-based Frank-Wolfe (denoted as ODBFW) algorithm proposed by Chen et al. (2002), but utilizes path information differently. For example, the AFW algorithm can switch a non-UE path flow to a cheaper one based on path

Fig. 3 A convergent sequence $\{a_n\}$ represents that a non-UE path bears a path flow, and cannot achieve an $\epsilon - UE$ state within a finite number of steps. In contrast, a convergent sequence $\{b_n\}$ means that a non-UE path does not bear any flow, so obtained solutions eventually yield an $\epsilon - UE$ state



information, but the ODBFW algorithm simply adopts path information to create a heuristic update direction to accelerate the convergence speed. However, a non-UE path flow still exists in the network. Consequently, the ODBFW algorithm cannot achieve an $\epsilon - \text{UE}$ state.

5.1 The flow of the AFW algorithm

The AFW algorithm utilizes path information to switch a non-UE path flow, and is described as follows.

First Stage

- Step 1.1 Initialization. Set iteration counter $n = 0$, and $\forall a \in A$, use free-flow travel cost C_a^0 .
- Step 1.2 Direction finding. Find each OD pair's shortest path $p^{rs(n)}$, and update arc flow $y_a^{(n)}$ via an AON assignment.
- Step 1.3 Update arc and path information. Record these found shortest paths $p^{rs(n)}$ to the path set $P_{rs}^{(n)}$ and AON path flow $h_p^{rs(n)}$, and set $f_a^{(n)} = y_a^{(n)}$.

Second Stage

- Step 2.1 Update counter. Set iteration counter $n = n+1$, $f_a^{(n)} = f_a^{(n-1)}$, and $h_p^{rs(n)} = h_p^{rs(n-1)}$.
- Step 2.2 Direction finding. Set $C_a = C_a(f_a^{(n)})$, and find each OD pair's shortest path $p^{rs(n)}$ to update a arc flow $y_a^{(n)}$ using an AON assignment.
- Step 2.3 Step-size determination. Find a step size $\alpha^{(n)} = \arg \min_{0 \leq \alpha \leq 1} Z(f_a^{(n)} + \alpha \cdot (y_a^{(n)} - f_a^{(n)}))$.
- Step 2.4 Update all path sets. If a shortest path $p^{rs(n)} \notin P_{rs}^{(n-1)}$, update the path set $P_{rs}^{(n)} = P_{rs}^{(n-1)} \cup p^{rs(n)}$; otherwise, $P_{rs}^{(n)} = P_{rs}^{(n-1)}$.
- Step 2.5 Update path flag. $\eta_p^{rs(n)} = 1$, if $p = p^{rs(n)}$; otherwise, $\eta_p^{rs(n)} = 0, \forall p \in \{P_{rs}^{(n)} \setminus p^{rs(n)}\}$.
- Step 2.6 Update arc flow. $f_a^{(n)} = f_a^{(n)} + \alpha^{(n)} \cdot (y_a^{(n)} - f_a^{(n)})$.
- Step 2.7 Update path flow. $h_p^{rs(n+1)} = h_p^{rs(n)} + \alpha^{(n)} \cdot (q^{rs} \eta_p^{rs(n)} - h_p^{rs(n)})$
- Step 2.8 Check whether a non-UE path flow exists. If a non-UE path flow exists, switch a non-UE path flow to a cheapest one; otherwise, go to the next step.
- Step 2.9 Convergence test. If $\epsilon - \text{UE}$ is met, then stop; otherwise go to Step 2.1.

The differences between the AFW and FW algorithms are path information and examination of non-UE paths. Path information mainly provides max-cost and min-cost paths. Furthermore, a max-cost path is used to examine whether a descent property is satisfied by switching a max-cost path flow to a cheapest

cost one. Besides the observed differences, the AFW and FW algorithms share the same structures.

Theorem 2 *For any given $\epsilon > 0$, the AFW algorithm converges to the $\epsilon - \text{UE}$ state.*

Proof Let X be a nonempty closed set in R^n and $\Omega \subseteq X$ be the $\epsilon - \text{UE}$ solution set. Given an initial arc flow $f_a^{(0)} \in X$ formed by shortest path flows, the sequence $\{f_a^{(n)}\}$ is generated iteratively as follows. If $f_a^{(n)} \in \Omega$, then stop; otherwise, let $h_p^{(n+1)} = A(h_p^{(n)}) = h_p^{(n)} + \alpha^{(n)} \times (q \cdot \eta_p^{(n)} - h_p^{(n)})$ for $0 \leq \alpha^{(n)} \leq 1$, replace n by $n + 1$, and repeat.

Since the objective function of UE Problem has a convex property, and a step size $\alpha^{(n)}$ is obtained using $\nabla Z = 0$. Assume $f_a = T(h_p) = \sum_{rs} \sum_p h_p^{rs} \delta_{a,p}^{rs}$, then

$$\begin{aligned} Z(T(h_p^{(n+1)})) &= Z(T(A(h_p^{(n)}))) \\ &= Z(T((1 - \alpha^{(n)}) \cdot h_p^{(n)} + \alpha^{(n)} \cdot \eta_p^{(n)} \cdot q)) \\ &\leq (1 - \alpha^{(n)})Z(T(h_p^{(n)})) + \alpha^{(n)} Z(T(q \cdot \eta_p^{(n)})) \end{aligned}$$

is satisfied for $0 \leq \alpha^{(n)} \leq 1$. In other words, an update scheme $h_p^{(n+1)} = A(h_p^{(n)})$ is able to reduce the objective function value of UE Problem.

If at any iteration a point $f_a^{(n)}$ in Ω is generated, then the algorithm stops. Now suppose that an infinite sequence $\{f_a^{(n)}\}$ is generated. Let $\{f_a^{(n)}\}_\kappa$ be any convergent subsequence with limit $f_a^* \in X$. Since Z is continuous, then for $n \in \kappa$, $Z(f_a^{(n)}) \rightarrow Z(f_a^*)$. Thus, for a given $\epsilon > 0$, there is a $K \in \kappa$ such that

$$Z(f_a^{(n)}) - Z(f_a^*) < \epsilon \text{ for } n \geq K \text{ with } n \in \kappa.$$

In particular, for $n = K$ we get

$$Z(f_a^{(K)}) - Z(f_a^*) < \epsilon \tag{9}$$

Now let $n > K$. Since Z is a descent function, $Z(f_a^{(n)}) < Z(f_a^{(K)})$, and from Eq. (9), we get

$$Z(f_a^{(n)}) - Z(f_a^*) = Z(f_a^{(n)}) - Z(f_a^{(K)}) + Z(f_a^{(K)}) - Z(f_a^*) < 0 + \epsilon = \epsilon,$$

$$\lim_{k \rightarrow \infty} Z(f_a^{(n)}) = Z(f_a^*).$$

Table 4 The $\epsilon - \text{UE}$ results obtained by the AFW algorithm

Iteration n	Step size α	RG	Objective value	Maximal cost difference of the cheapest path
3	0.04166698	0.049900453	2100.00000	0.00012543
4	0.00000095	0.000000597	2100.00000	0.00008438

Table 5 The $\epsilon - \text{UE}$ path information obtained by the AFW algorithm

Path	Path flow	Path cost
Path 1	10.0000038	160.0000422
Path 2	9.9999962	159.9999578

If $Z(f_a^*)$ does not satisfy the $\epsilon - \text{UE}$ state, a non-UE positive flow must exist. Since the AFW algorithm can remove a non-UE positive flow, $Z(f_a^*)$ obtained via the AFW algorithm can attain the $\epsilon - \text{UE}$ state. \square

The AFW algorithm is still an arc-based algorithm, but equips with path information for checking whether these non-UE paths exist. It must, however, be noted that the AFW algorithm do destroy some of the simplicity of the Frank-Wolf algorithm, because the proposed algorithm that keeps path information is guaranteed to achieve the $\epsilon - \text{UE}$ state.

6 Numerical results

The numerical results of the AFW algorithm are listed in Tables 4 and 5. Table 4 reveals that the AFW algorithm converges in a few iterations within the example of Braess’s network, as illustrated in Fig. 1, and Table 5 displays the numerical results of the AFW algorithm following 10 iterations. Furthermore, Table 4 illustrates that the value of the maximal cost difference among the paths obtained by the AFW algorithm is significantly smaller than that listed in Table 2, and the numerical result obtained by the AFW algorithm can achieve an $\epsilon - \text{UE}$ state.

The path set listed in Table 5 comprises accurate UE solutions, since the AFW algorithm utilizes path information to permit a non-UE path flow to switch effectively to the shortest path. This numerical example presented here indicates the additional memory requirements can significantly reduce

Fig. 4 Modified Braess’s network

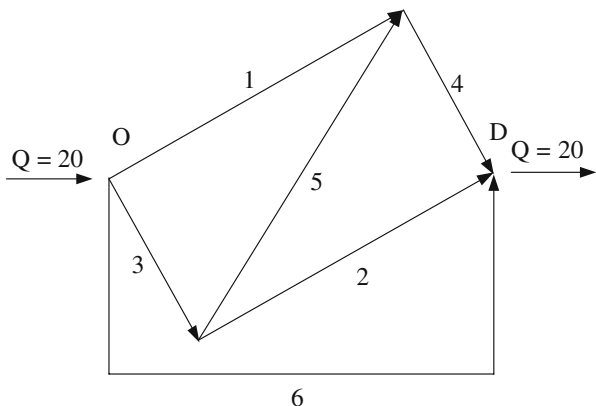


Table 6 Illustrative example of small path flow

Path	Path flow	Path cost
Path 1	9.9946703	159.9615070
Path 2	9.9937368	159.9512386
Path 3	0.0020134	209.9263519
Path 4	0.0095795	159.9579532

computational burdens. Removing a non-UE path flow not only accelerates convergence, but also provides accurate UE solutions.

6.1 Issue of small path flow

From Tables 2 and 3, a non-UE path frequently has higher cost and lower flow provided the stopping criteria are met. Some may think that switching a small path flow to a cheapest path without considering path cost may be intuitively workable. In fact, this view is incorrect, as illustrated in Fig. 4.

Figure 4 adds an extra non-overlapping path 4 (expressed as link 6) between the OD pair illustrated in Fig. 1. Path 4 is set to have larger fixed and marginal costs, such that it belongs to the path information established by the FW algorithm.

Consider

$$\text{Link6: } t_6(f_6) = 159 + 100 f_6$$

Assume that path flows less than 0.01 are switched to the other paths, namely Paths 3 and 4, as shown in Table 6. Nevertheless, Path 4 belongs to the UE solutions despite having very small path flow. Consequently, considering path flow alone without considering path cost is insufficient for determining whether a path belongs to the UE solutions.

7 Conclusions

The three observations are summarized above and detailed as follows. First, the arc-based FW algorithm cannot determine whether a path belongs to the UE solution, since the arc-based FW algorithm carries no path information. Second, not all of the assigned paths ultimately become UE paths. Third, removing a flow on a non-UE path significantly facilitates the reduction of computational burden. Consequently, a modified algorithm, namely AFW algorithm, is proposed, and applied to solve non-UE path flow problems based on the above observations. Furthermore, maintaining path information intuitively increases the memory requirements and computational effort. However, contrary to expectations, utilizing path information can facilitate the convergence speed of the FW algorithm, and provide accurate UE solutions.

Acknowledgements This research was partially supported by MOE ATU Program of the Ministry of Education, Taiwan, R.O.C. and the National Science Council, Taiwan, R.O.C. under grants numbers NSC 96-2221-E-009-118-MY3.

References

- Beck MJ, McGuire CB, Winston CB (1956) *Studies in the economics of transportation*. Yale University Press, New Haven
- Boyce D, Ralevic-Dekic B, Bar-Gera H (2004) Convergence of traffic assignments: how much is enough? *J Transp Eng ASCE* 130(1):49–55
- Braess D (1968) Über ein paradoxen der verkehrsplanung. *Unternehmensforschung* 12:258–268
- Bruynooghe A, Gibert A, Sakarovitch M (1968) Une méthode d'affectation du traffic. In: *Proceedings of the fourth symposium on the theory of traffic flow*, Karlsruhe
- Chen A, Jayakrishnan R (1996) Path and link flow based assignment algorithms: finding and storing path-flow solutions and computational performance. ITS Working Paper, University of California, Irvine
- Chen A, Jayakrishnan R, Tsai, WK (2002) Faster Frank-Wolfe traffic assignment with new flow update scheme. *J Transp Eng* 128:31–39
- Dial RB (2006) A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. *Transp Res Part B* 40:917–936
- Frank M, Pinder KL (1956) An algorithm for quadratic programming. *Nav Res Logist Q* 3:95–110
- John Glen Wardrop BA (1952) Some theoretical aspects of road traffic research. In: *Proceedings of institution of civil engineers*, vol 2, pp 325–378
- LeBlanc LJ, Morlok EK, Pierskalla WP (1975) An efficient approach to solving the road network traffic assignment problem. *Transp Sci* 9:309–318
- Mguyen S (1974) An algorithm for the traffic assignment problem. *Transp Sci* 8:203–216
- Murchland JD (1969) Road network traffic distribution in equilibrium. In: *Proceedings of the conference mathematical methods in economic sciences*. Oberwolfach W. Germany, Mathematisches Forschungsinstitut
- Pas EI, Principio SL (1997) Braess' paradox: some new insights. *Transp Res Part B* 31(3):265–276
- Sheffi Y (1985) *Urban transportation networks: equilibrium analysis with mathematical programming methods*. Prentice-Hall, Englewood Cliffs, 07632