Computational Study of Alternative Methods for Static Traffic Equilibrium Assignment

Zhong Zhou, Alberto Brignone and Michael Clarke
Citilabs Inc.
316 Williams Street
Tallahassee, Florida 32303 USA
Phone: +1 888 770 CUBE Fax: +1 888 771 CUBE
Email: {zzhou, abrignone,mclarke} @citilabs.com

Abstract

Traffic equilibrium problem, also known as the traffic assignment problem, is the core of many important transportation models. The traditional Frank-Wolfe (FW) algorithm has been criticized for its slow convergence speed to approach high precision level. Recently, various traffic equilibrium algorithms have been developed in literature. Basically, they can be classified into three main categories, i.e. link-based algorithms, path-based algorithms and origin-based algorithms, depending on the solution variables adopted in the problem. These new algorithms are expected to fulfill the requirement of the implementations of more advanced and comprehensive transportation models and analysis, which desire rapid convergence speed and highly precise equilibrium solutions.

Nowadays, distributed computing is getting popular in many fields, due to the increasing power of the computer hardware. Multiple machines, machines with multiple processors, or processor with multiple cores become more widely available and more affordable in recent years. Therefore, it is also desired to examine the possibility and efficiency of incorporating the power of distributed computing with the newly developed algorithms in order to further improve the performance of solving traffic equilibrium problems.

In this paper, a new computational study of alternative traffic equilibrium algorithms is conducted. Comparing with the results in literature, our implementation with a proprietary data process enables the applicability of the algorithms on real size networks in planning practice. The numerical study shows that the improved link-based algorithms and path-based algorithms we implemented are able to reach highly precise equilibrium solution in modest computational time. It also reveals that the path-based algorithm begins to outperform the link-based algorithms after reaching certain level of precision. For the 'best in practice' converge level, i.e. relative gap 1e-4, the improved link based and path based algorithms converge much quicker than the original FW algorithm and comparable with the origin based algorithm.

Furthermore, the effects of combining the distributed computing technology with the alternative traffic assignment algorithms are explored. Comparing with the path/origin-based algorithms, whose high convergence speed relies on the sequentially process of origins or OD pairs (i.e. property of order dependency), the test results demonstrate that the improved link-based algorithms are able to reach even higher efficiency by taking advantage of the increased power of distributed computing. This is due to their inherent parallelism of the algorithm structure, where each origin or each OD pair is implicitly treated equally.

Key Words: Traffic equilibrium problem, improved link-based algorithm, path-based algorithm, origin-based algorithm, distributed computation

1. INTRODUCTION

The static user equilibrium (UE) based traffic assignment problem is to find the traffic flow pattern by allocating the O-D demands to the network such that all used routes between each OD pair have equal travel cost, and no unused route has a lower travel cost. Such an equilibrium state is what results if each and every traveler simultaneously attempts to minimize individual travel cost, which is stated by the Wardrop's (1952) first principle. This UE based assignment is the standard approach adopted by Metropolitan Planning Organizations' travel demand forecasting models for planning purpose. Therefore, throughout this paper, the word 'traffic assignment' particularly refers to the static UE based traffic assignment.

It is well known that the classic traffic assignment problem can be formulated as a nonlinear convex program with linear constraints (Beckman et al., 1956). Then, the user equilibrium solution can be obtained by minimizing the objective function of this convex program, which is the base of all the traffic assignment algorithms discussed in the following contents. Over the last a few decades, the Frank-Wolfe (FW) algorithm (Frank and Wolfe, 1956), has been widely adopted as the standard traffic assignment algorithm in real applications, and supported by all major software packages. The FW algorithm is simply and easy to implement. Basically, an all-or-nothing assignment is performed in each iteration, which generates an auxiliary flow pattern. Then, the auxiliary flow is combined with the current one by using weight factor produced by a one dimensional line search, such that the objective value of the convex program is minimized. This process is repeated till it reaches predefined number of iteration or desired convergence level. Another advantage of the FW algorithm is its memory efficiency, because only link variables need to be stored during the iteration process. This memory efficiency is one of the main reasons of the popularity of the FW algorithm in previous decades, due to the limitation of the computers' memory and CPU speed at that time, which are much smaller and slower than present ones, such that it cannot handle the large size of real transportation networks.

To measure the convergence of traffic assignment algorithms, many different criteria have been proposed in practice. The most commonly adopted criterion, which is also regarded as an industry standard right now, is relative gap (RG). For a given iteration, gap is defined as the difference between the total vehicle travel cost (sum of the product of link flows and associated link costs) and the vehicle travel cost given by all-or-nothing (AON) assignment at the current iteration (equals to the sum of OD demand multiply the cost of the shortest path connecting the OD pair). Then, the relative gap is defined as the ratio of the gap to the total vehicle travel cost (Rose et al. 1988), or as the ratio of the gap to the best lower bound (BLB) (Boyce et al., 2004), where BLB is defined as the maximum value of the lower bound (i.e. gap plus the objective value) of all previous iterations.

In general, the FW algorithm reaches modest convergence levels (e.g., RG 0.01) very quickly (around 20 – 30 iterations in many models). However, it suffers from slow convergence to higher precisions (i.e. RG < 0.0001) (Patriksson, 1994). This is because the search direction becomes orthogonal to the steepest decent direction when approaching equilibrium solution.

To get quick convergence speed while achieving high levels of convergence, many different algorithms have been proposed in literature. Depending on the solution variables adopted in the problem, they can be classified into three main categories: link-based algorithms, path-based algorithms and origin-based algorithms. That is, if the solution variables are link flows, then the algorithm is regarded as link-based. If the solution variables are path flows, then the algorithm is regarded as path-based. Similarly, the solution variables for origin-based algorithms are origin-based link flows (or origin-based approach proportions), which is in an intermediate way between links and paths.

The information provided by the different algorithms is also different: link-based algorithms only provide aggregated link flow information, though disaggregate information, such as vehicle routes information, can be retrieved by additional process. Path-based algorithms provide a complete picture of traffics among different routes from origins to destinations and corresponding details of turning movements. Origin-based algorithms are able to produce comparable traffic flow details as path-based algorithms by using approach proportions or additional process.

Several link-based algorithms have been proposed in literature. One set of them have similar structure as Frank-Wolfe algorithm, thus can be considered as improved Frank-Wolfe algorithms. They try to improve the performance of the Frank-Wolfe algorithm by finding better search directions (e.g., Fukushima, 1984;Arezki,1987;Leblanc et al., 1985; Florian et al., 1987; Arezki and vliet, 1990; Lupi, 1986) or stepsize (e.g., Powell and Sheffi, 1982; Weintraub et al., 1985). The other set of link-based algorithms, including restricted simplicial decomposition (RSD) (Hearn et al., 1985) and nonlinear simplicial decomposition (NSD) (Larrson and Patriksson, 1997), can be consider as extended Frank-Wolfe algorithm, because they utilize more complex structures that include the Frank-Wolfe algorithm as a special case. Basically, they alternatively solve a nonlinear master problem with simple convexity constraints and a linear/nonlinear programming subproblem. The subproblem generates extreme points of the feasible region, while the master problem finds the optimum of the objective within a subset of previous generated points. However, these link based algorithms above and others were not available in commercial software packages. Recently, Daneva (2003) proposed two new improved Frank-Wolfe algorithm, i.e. Conjugate Frank-Wolfe (CFW) and Bi-Conjugate Frank-Wolfe (BiFW) algorithm. In each iteration, the new search directions are constructed to be conjugate to the previous (one or two) search directions. Similar as the original Frank-Wolfe algorithm, the CFW and BiFW algorithms are simple, easy to implement and memory efficient. Though the numerical results are promising, Daneva (2003) coded the CFW and BiFW algorithms in Matlab and only tested on small to middle size networks.

As an intermediate between link-based and path-based algorithm, origin-based algorithms have the mild memory requirement and quick converge speed. The first origin-based algorithm (OBA) in transportation field was proposed by Bar-Gera (2002). The decision variables are origin-based approach proportions for every origin and every link. The key point of the OBA is to decompose the transportation network into a sequence of acyclic restricting subnetworks by origin, where the acyclic means there is no direct cycle contained in the subnetwork, and restricting means that the approach proportions of links that are not included in the subnetwork are restricted to zero. The advantage of the acyclic network is to allow an origin-specific topological order of nodes, such that many computations can be done efficiently following this order. During the iterations, the acyclic restricting subnetworks are updated sequentially, where the traffic flows are shifted from high cost alternatives to low cost alternatives within the subnetwork. Nie (2008) proposed an improved origin-based quasi-newton method (NOB) for traffic assignment problem by implementing some of the subnetwork constructing and flow shifting stepsize strategies. Dial (2006) proposed another origin-based algorithm, called Algorithm B. Similar as Bar-Gera’s OBA, one of the key feature of the Algorithm B is to decompose a traffic assignment problem instance on a generally cyclic network into a sequence of origin-restricted subproblem on acyclic subnetworks (named *bush*). Therefore, the topological order of nodes can be constructed, and the traffic flows are shifted from the highest cost path to the shortest path within the bush. Slavin et al. (2006) implemented a variant of Algorithm B and named it as OUE algorithm. A comprehensive comparison of the above origin based algorithms is conducted by Nie (2009). Gentile (2009) proposed a linear user cost equilibrium (LUCE) algorithm. The LUCE algorithm constructs the bushs based on destination, and solves a local linear equilibria at each node in terms of destination flow. Recently, Bar-Gera (2010) proposed a new assignment algorithm called TAPAS (traffic assignment by paired alternative segments). The TAPAS algorithm uses pairs of alternative segments as the key building block to the UE solution, and is able to acquire a route flow solution that satisfy a condition of proportionality.

The first path-based algorithm for solving traffic assignment problem is proposed by Dafermos (1968). The algorithm sequentially examines each OD pair and shift flow from the maximum cost path to the shortest path. The path-based algorithms have the ability to provide quick convergence speed to reach high precision level, and give the richest information on travel demand patterns. However, due to the requirement of storing and tracing paths and the memory limitation of the computer, they have not gained much attention in practice until recently. The rapid increase of computer powers and memory finally opens the door for incorporating path-based algorithms in real planning applications. Larsson and Patriksson (1992) proposed a disaggregated simplicial decomposition (DSD) algorithm, which has the similar structure as the RSD method expect the extreme points are on path flow space. That is, the master problem aims to solve an auxiliary nonlinear program, and the subproblem is to generate new feasible path set for improving the solution. Jayakrishnan et al. (1994) proposed a gradient projection (GP) algorithm for solving traffic assignment problem. The algorithm utilizes the Goldstein-Levitin-Polyyak gradient projection method formulated by Bertsekas (1976) for general nonlinear multi-commodity problem. In each iteration, the OD flows are shifted from the nonshortest paths to shortest path, where the portions of traffic to be shifted are derived by quasi-newton method and restricted by projection to maintain the feasibility. Here, the projection has the ability to efficiently eliminate the residual flow that the FW algorithm is unable to do. Chen et al. (2002) examined extensive experiments, where GP algorithm was showed more advantages than DSD algorithm in small to medium size networks. Florian et al. (2009) implemented another path-based algorithm, called projected gradient (PG) method. The algorithm was originally proposed by Rosen (1960) for solving nonlinear program. It is similar to the GP algorithm except that the OD flows are shifted from the paths with cost higher than the average to the paths with cost lower than the average, and the portions of shifting are decided by linear search. Jayakrishnan et al. (1994) and Chen et al. (2002) provided extensive analysis of the gradient projection algorithm. However, they only considered the largest network with 4900 nodes due to memory limitation and the adopted relative error is 0.001.

The computational study conducted in this paper is based on our implementations of the Conjugate/Bi-Conjugate Frank-Wolfe algorithms and the gradient projection (GP) algorithm in the CUBE Voyager software. Our implementation with a proprietary data process enables the applicability of the path-based (gradient projection) algorithm on real size networks in planning practice. And the capability of distributed computation enables higher efficiency of the improved link-based (CFW and BiFW) algorithms. The implemented link-based and path-based algorithms are all compared with Bar-Gera's executable code of origin based algorithm (OBA).

The remainder of the paper is organized as follows: the notation and problem statement are given in section 2; section 3 provides a brief review of the basic ideas of the CFW/BiFW and GP algorithm; numerical results are presented in section 4; finally, conclusions are summarized and some future researches are suggested in Section 5.

2. NOTATION AND FORMULATION OF TRAFFIC ASSIGNMENT PROBLEM

Throughout this study, we assume the origin-destination (*OD*) travel demands are given and fixed. Consider a strongly connected network [*N, A*], where *N* and *A* denote the sets of nodes and links, respectively. Let *R* and *S* denote a subset of *N* for which travel demand q^{rs} is generated from origin $r \in R$ to destination $s \in S$. The assumption of a strongly connected network guarantees that there exists at least one route from every origin-destination (O-D) pair with positive travel demand. Let f_p^{rs} denote the flow on route $p \in P^{rs}$, where P^{rs} is a set of routes from origin *r* to destination *s*. Also, let v_a denote the link flow on link $a \in A$, and let Δ= [δ_{pa}^{rs}] denote the route-link incidence matrix, where $\delta_{pa}^{rs} = 1$ if route *p* from origin *r* to destination *s* uses link *a*, and 0, otherwise. Then, we have the following relationships:

$$q^{rs} = \sum_{p \in P^{rs}} f_p^{rs}, \quad \forall\ r \in R, s \in S, \tag{1}$$

$$v_a = \sum_{r \in R} \sum_{s \in S} \sum_{p \in P^{rs}} f_p^{rs} \delta_{pa}^{rs}, \quad \forall\ a \in A, \tag{2}$$

$$f_p^{rs} \geq 0, \quad \forall\ p \in P^{rs}, r \in R, s \in S, \tag{3}$$

where (1) is the travel demand conservation constraint; (2) is a definitional constraint that sums up all route flows that pass through a given link *a*; and (3) is a non-negativity constraint on the route flows.

Then the user equilibrium traffic assignment problem can be formulated as a convex program as below:

$$Z_{UE} = \sum_{a \in A} \int_0^{x_a} t_a(w)\, dw \tag{4}$$

s.t. (1) – (3)

where $t_a(\cdot)$ represents the cost of link *a*.

Let η denote the vector $\left(\ldots, \eta_p^{rs}, \ldots\right)^T$, where $\eta_p^{rs} = \sum_{a \in A} t_a \delta_{pa}^{rs}$, and the link cost function t_a are assumed to be positive and continuous; π^{rs} denote the minimal cost between O-D pair (*r,s*), and *f* denote the route-flow pattern which is a vector of $\left(\ldots, f_p^{rs}, \ldots\right)^T$. Then, at the equilibrium, all used routes between each O-D pair have equal travel cost, and no unused route has a lower travel cost, i.e. the following conditions hold (Nagurney, 1993):

$$\eta_p^{rs}(f^*) - \pi^{rs} \begin{cases} = 0 & if\ (f_p^{rs})^* > 0 \\ \geq 0 & if\ (f_p^{rs})^* = 0 \end{cases}, \quad \forall\ p \in P^{rs}, r \in R, s \in S. \tag{5}$$

Such an equilibrium state is what results if each and every traveler simultaneously attempts to minimize individual travel costs.

3. BRIEF REVIEWS OF THE IMPROVED FRANK-WOLFE ALGORITHMS GRADIENT PROJECTION METHOD AND ORIGIN-BASED ALGORITHM

Conjugate Frank-Wolfe Algorithm (Daneva, 2003)

The traditional FW algorithm is also called convex combination method or linear approximation method in nonlinear optimization theory. Basically, the descent direction d_{FW}^{i} on the current iteration *i* of the UE program (4) is acquired by solving a linearized subproblem, then line search is used to find a new solution. Since the conjugate gradient methods generally outperform the pure gradient method, the conjugate FW algorithm naturally adapted this idea and tried to get a better search direction. The conjugate FW descent direction d_{CFW}^{i} at the current iteration *i* is computed as:

$$d_{CFW}^{i} = d_{FW}^{i} + \beta_i d_{CFW}^{i-1}, \qquad (6)$$

such that $\left(d_{CFW}^{i}\right)^T H^i d_{CFW}^{i-1} = 0$, where d_{CFW}^{i-1} is the conjugate FW descent direction at the previous iteration *i-1*, β_i is the parameter to be computed, and H^i is the Hessian of the UE objective function in the *i*th iteration. That is, the search direction at current iteration is conjugate with the previous search direction. It is easy to see that the CFW is as simply as the traditional FW algorithm, and inherits its advantage of the memory efficiency. In each iteration, the CFW algorithm only needs to keep 3 vectors in memory to find a new conjugate search direction, and only one line search step has to be performed in order to find the new solution.

Bi-Conjugate Frank-Wolfe Algorithm (Daneva, 2003)

Comparing with the CFW algorithm, the BiFW algorithm goes one step further. It is to make the current search direction conjugate with the last two search directions, i.e.,

$$d_{BFW}^{i} = \beta_0^i d_{FW}^{i} + \beta_1^i d_{BFW}^{i-1} + \beta_2^i d_{BFW}^{i-2}, \qquad (7)$$

such that

$$\left(d_{BFW}^{i}\right)^T H^i d_{BFW}^{i-1} = 0 \text{ and } \left(d_{BFW}^{i}\right)^T H^i d_{BFW}^{i-2} = 0$$

where β_0^i, β_1^i and β_2^i are parameters to be computed during the iteration, and H^i is the Hessian matrix. In each iteration, the BiFW algorithm only needs to keep 4 vectors in memory to find a new conjugate search direction and still only one line search step has to be performed in order to find the new solution (same as FW and CFW). The implementation is slightly more complicated, but it is still much simpler and more memory efficient than other type of algorithms, such as path-based and origin-based algorithms. In addition, the numerical study in the following sections shows that the BiFW has more advantage than CFW when approaching higher converge precisions.

Gradient Projection Algorithm (Jayakrishnan, et al.,1994)

The gradient projection algorithm described here is based on the Goldstein-Levitin-Polyak gradient projection method formulated by Bertsekas (1976) for general nonlinear multi-commodity problems, applied to the traffic assignment problem by Jayakrishnan et al., (1994). The main idea is to eliminate the travel demand constraints (1) by reformulating the path-flow variables in terms of the non-shortest path flows, i.e.

$$f_{\overline{p}^{rs}}^{rs} = q^{rs} - \sum_{\substack{p \in P^{rs} \\ p \neq \overline{p}^{rs}}} f_p^{rs} \;,\; \forall \; p \in P^{rs}, r \in R, s \in S \tag{8}$$

where $\overline{p}^{rs}$ represents the shortest path from origin r to destination s.

Embedding constraints (8) into the objective function, we obtain a new formulation with just the non-negativity constraints on the non-shortest path flows as the decision variables. Therefore, this reformulation is a convex program with only non-negativity constraints. The flow update in each iteration can be described as below:

$$f_p^{rs}(i) = \max\left\{ f_p^{rs}(i-1) - \frac{\alpha(i)}{s_p^{rs}(i)}\left(d_p^{rs}(i) - d_{\overline{p}^{rs}(i)}^{rs}\right), 0 \right\},\; \forall \; p \in P^{rs}, r \in R, s \in S$$

And

$$f_{\overline{p}^{rs}(i)}^{rs}(i) = q^{rs} - \sum_{\substack{p \in P^{rs}(i) \\ p \neq \overline{p}^{rs}(i)}} f_p^{rs}(i-1), \qquad \forall \; p \in P^{rs}, r \in R, s \in S$$

where $d_p^{rs}(i)$ and $d_{\overline{p}^{rs}(i)}^{rs}$ are the first derivative of the objective function along path p and shortest path $\overline{p}$ respectively; $\alpha(i)$ is the stepsize and $s_p^{rs}(i)$ is a diagonal, positive definite scaling based on the second derivative Hessian.

In contrast to the traditional FW algorithm, which finds auxiliary solutions that are vertices (extreme points) of the feasible region, GP algorithm can be regarded as a Quasi-Newton method, which makes successive moves in the direction of negative gradient, scaled by the approximation of the second derivative Hessian. In addition, a projection is made back to the nonnegative orthogonal whenever the move results in an infeasible solution. This boundary search makes the flow shifting more efficiently.

4. Numerical Results

In this section, numerical results are presented to examine and compare the convergence and computational performance of the CFW/BiFW algorithms, GP algorithm and origin-based algorithm (OBA). The executable OBA code developed by Bar-Gera was obtained from the Open Channel Foundation. It is used as a benchmark for numerical comparisons of different algorithms as used in literature (e.g., Florian et al, 2009, Slavin et al., 2006). The CFW/BiFW algorithms and GP algorithm were implemented in the CUBE Voyager planning software package by the C programming language. The implementation of the GP algorithm took advantage of a proprietary data process, which enables the path-based algorithm to be applied on any size of large scale networks. The OD pairs are processed sequentially, that is, the process of the current OD pair utilizes the latest information from the flow updating of the previous OD pair. The route information can also be explicitly generated during the iteration. The implementation of the CFW and BiFW algorithms kept consistent with existing practice. It allows full functionality as what is provided by the traditional FW assignment procedure, such as multiple user classes, turning penalties, junction modeling, select link analysis and other similar analysis, with no demand to modify any existing inputs. Furthermore, it allows the distributed computing to utilize the increasing computing powers.

The network adopted in this study is the Chicago Regional Network (Figure 1), which has been widely used in various empirical studies (e.g., Bar-Gera, 2002, Nie, 2009, Slavin, 2006, Florian et al., 2009), and made available to researchers (http://www.bgu.ac.il/~bargera/tntp/). The network has 1790 zones, 12982 nodes, 39018 links and total demand 1360427.88. The test environment is a 64bit Intel platform with two Xeon E5335 2 GHz Quad Cord processor and 8G RAM. The operation system is Windows Vista 64.

To evaluate the convergence and performance of various algorithms, a critical question for transportation planners and modelers is: "how much convergence is enough for traffic assignments?" According to Boyce et al. (2004), there are two criteria for determining the desired level of convergence: one criterion is the stability of the solution. That is, the link flows should not fluctuate as the solution converges further, which is a requirement in the scenario comparison analysis. The other criterion is that the computation time required to achieve this stable solution should be reasonable. Based on their empirical experiments (a build and no-build scenarios comparison of two proposed ramps between I-295 and SR-42 in the New Jersey part of the Delaware Valley region), they suggested relative gap (RG) of 0.0001 to be desired precision for planning practice, such as scenario comparison, rather than academic interests. At this convergence level, the traffic assignments are ensured to be sufficiently converged to achieve link flow stability. This conclusion has also been validated and stressed by other empirical studies (e.g. Boyce t al., 2001, Ralevic-Dekic, 2000).Therefore, our numerical comparison and analysis below will mainly focus on the converge level up to relative gap 1e-5.

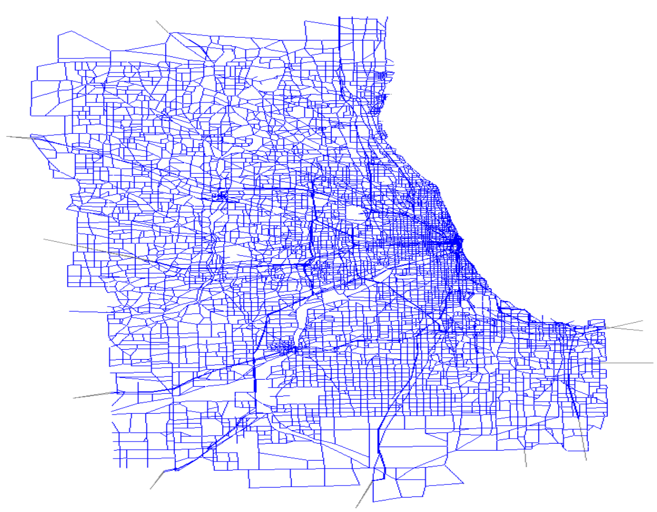

Figure 1 Chicago Regional Network

Comparisons of Convergence Performance

Firstly, we compare the convergence performance of the different algorithms. Figure 2 shows the convergence curve, where x axis represents CPU time and y axis is the logarithm (10 base) of the relative gap. Figure 3 shows the curve of the UE objective value, i.e. Z_{UE} in Eq (4), along the iteration. From the figures, we can see: (1) the FW algorithm implemented in CUBE Voyager actually has better convergence performance than OBA when the relative gap greater than 1e-4. When the required precision is higher (RG< 1e-4), the OBA begins to outperform the FW algorithm; (2) both improved link-based algorithms (CFW&BiFW) outperform the FW algorithm and OBA. They only take around 1/3 CPU time to reach RG 1e-4, and show the ability to reach higher precision level. In addition, the BiFW begins to show better convergence than CFW when converge level is higher (RG<1e-4 in this test). Therefore, people can choose either CFW or BiFW for practical converge level (RG>=1e-4), which gives similar performance. For people who pursue higher precision level, the BiFW will be the best choice here; (3) path-based algorithm (GP) starts slower than the link-based algorithms under low level of convergence (RG>1e-2). But it shows much more advantages over the link-based algorithms when the level of convergence increases. This is because in each iteration the gradient projection algorithm takes more CPU times to manipulate path flows than the link-based algorithms, but it requires much less number of iteration to reach the same converge level than the link-based algorithms do. We can see that the CFW and BiFW algorithm are still comparable with the path-based algorithm at lease at converge level 1e-3 in this test, and path-based algorithm shows better performance at higher precisions. Furthermore, the path based algorithms are always outperforms the OBA in the practical precision levels; (4) in Figure 3, the improved link-based algorithms have the steepest curve that means they achieve the vicinity of the optimal objective value much quicker than other algorithms. The gradient projection algorithm starts slow, but approaching the optimal value quicker than FW after a few iterations. Figure 3 also shows that the path-based algorithm achieves better objective value than link-based algorithms after certain iterations. Though the OBA shows better convergence performance than link-based algorithms at higher precision level (RG<1e-4), it appears to be the slowest among all tested algorithms to approach the optimal UE objective value.

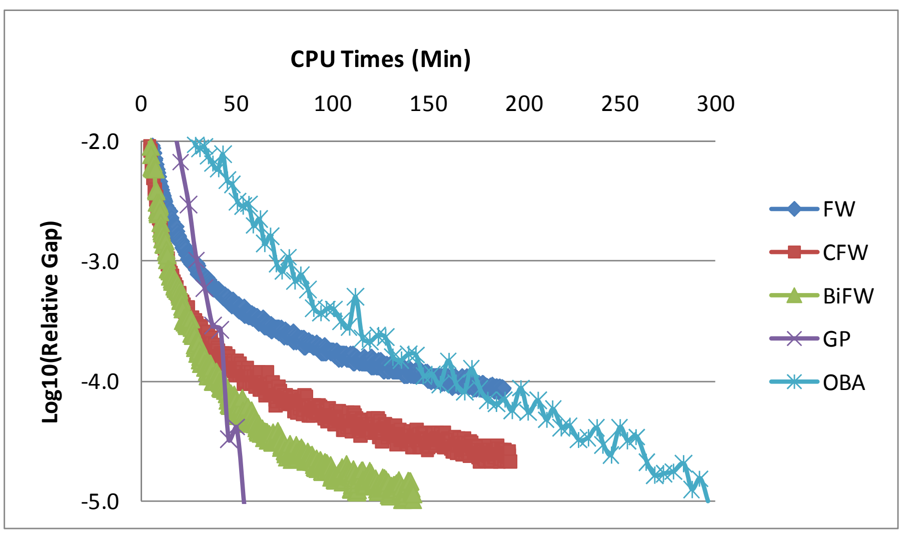

Figure 2 Convergence Performances of Different Algorithms
(Relative Gap vs. CPU Time)

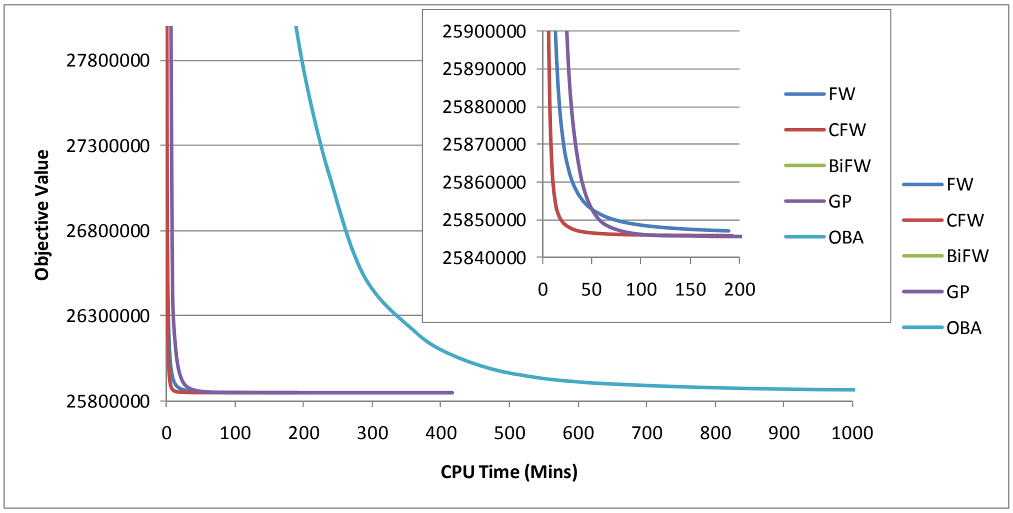

Figure 3 Convergence Performances of Different Algorithms
(UE Objective Value vs. CPU Time)

Secondly, we look at the detailed comparison of the algorithms based on CPU times and run time savings at different level of convergence. The results are shown in Table 1 below, where the run times of the FW algorithm are adopted as base value in the comparison. From the table, we can see that the improved link-based algorithms are obviously the best choice from lower to best practice level of convergence, i.e. up to relative gap 1e-4, and gradient projection algorithm dominates all others when higher precision are required. These observations are also illustrated in Figure 4.

Table 1 CPU Time & Run Time Savings at Different Level of Convergence

		Run Time Saving			Run Time Saving			Run Time Saving		
Relative Gap	0.01			0.001			0.0001			0.00001
FW	6.47			25.85			151.82			/
CFW	4.80	26%		14.03	46%		53.05	65%		/
BiFW	5.02	22%		13.78	47%		37.53	75%		142.62
GP	20.80	-222%		33.28	-29%		45.77	70%		54.09
OBA	28.82	-346%		71.02	-175%		156.15	-3%		296.00

*** '/' means the required precision is not reached after 1000 iterations**

It should be noted that more efficient implementation of OBA than the one adopted here could be available. Therefore, the results of comparisons with OBA algorithm should not be interpreted in a strict way. The results here do demonstrate that the improved link-based algorithms (CFW&BiFW) and path-based algorithm (GP) are computationally efficient to solve the real size problems in practice.

Effects of Distributed Computing

Nowadays, distributed computing is getting popular in many fields, including transportation, due to the increasing power of the computer hardware. Multiple machines, machines with multiple processors, or processor with multiple cores become more widely available and more affordable in recent years. The Frank-Wolfe type link-based traffic assignment algorithms are able to take advantage of this increased computing power easily due to their inherent parallelism of the algorithm structure. That is, the traffic flows are shifted from nonshortest paths to shortest path at the same time with same portion. Therefore, each origin or each OD pair is implicitly treated equally. Even the order of OD or origin changed, the equilibrium solutions are still the same. Therefore, it is essential to examine the effects of distributed computing in the traffic assignment.

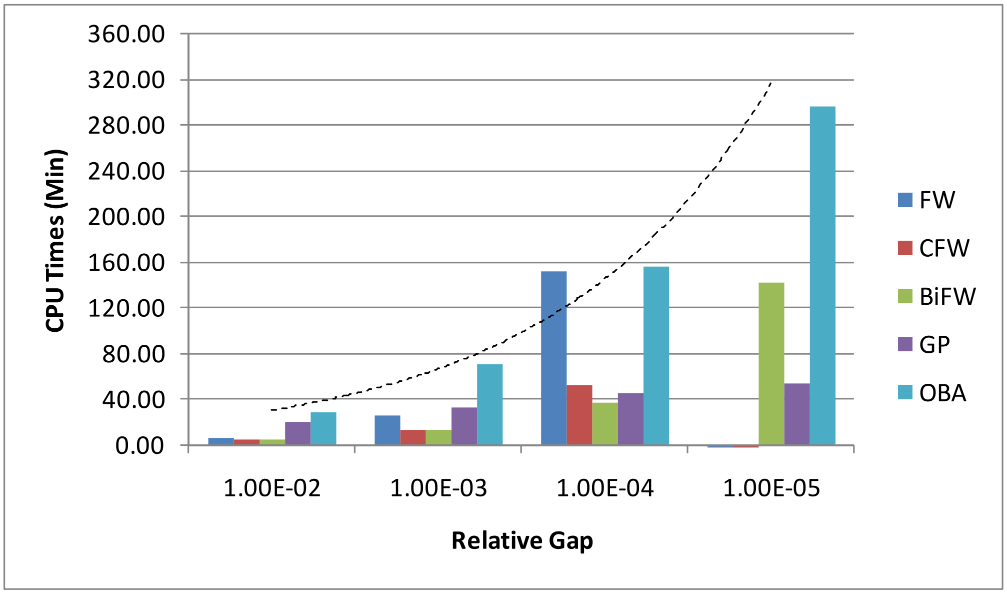

Figure 4 Run Time at Different Level of Convergence

In the following tests, we compare the effects of distributed computing of the three link-based algorithms. The CPU times under different level of convergence and various numbers of cores are depicted in Figure 5 to Figure 7 below.

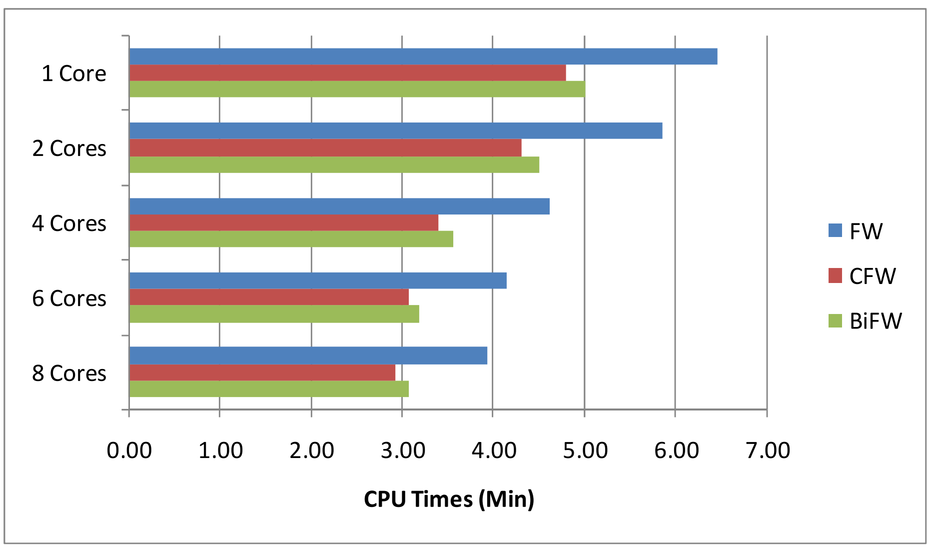

Figure 5 Run Time to Reach Relative Gap 0.01(Link-based Algorithms)

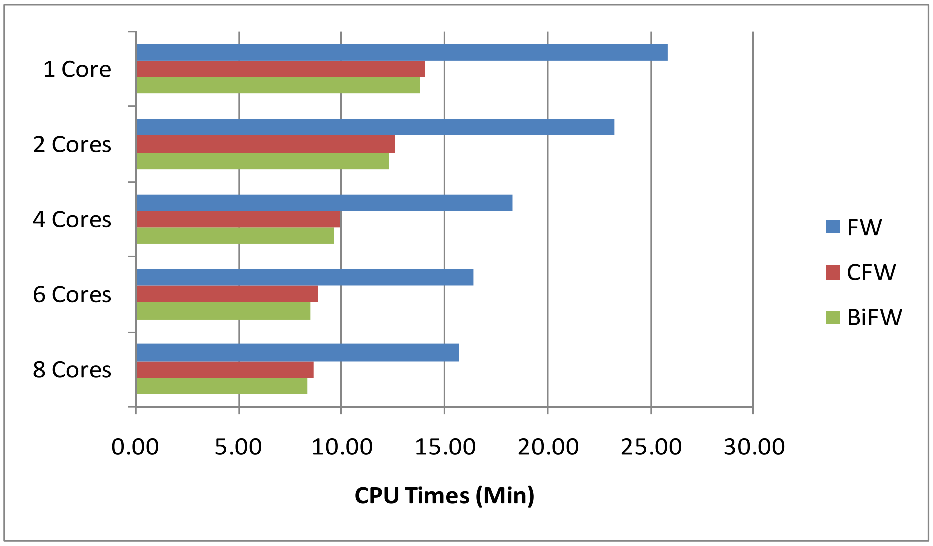

Figure 6 Run Time to Reach Relative Gap 0.001(Link-based Algorithms)

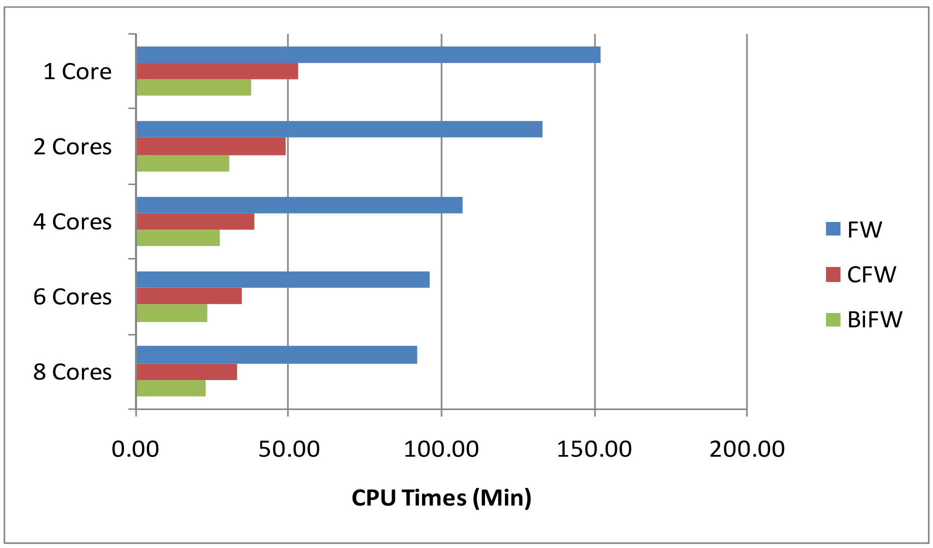

Figure 7 Run Time to Reach Relative Gap 0.0001(Link-based Algorithms)

From the Figures above, we can see that the distributed computing is able to highly enhance the computational efficiency. The benefit, i.e., run time saving, of using distributed computing becomes more significant when higher convergence levels are required. Moreover, the improved link-based algorithms show much better performance than traditional FW algorithm with distributed computing. For instance, to achieve relative gap 0.0001, comparing with the run time of the FW algorithm using only one core, the FW algorithm using 8 cores can save about 39.44% CPU Time, and the CFW and BiFW algorithms can save around 78.06% and 84.81% CPU times, respectively. We should note that, the increasing benefit of using more cores may deteriorate, such as adding number of cores from 6 to 8, due to the rising extra costs of communication and coordination. Therefore, there is a tradeoff between the hardware costs (using more CPUs/cores) and the efficiency (saving more computational time) in practice.

Unfortunately, this benefit from the advanced computing power may not be taken by the origin/path-based algorithms, which is a natural consequence of order dependence. It is well known that the design of efficient origin-based and path based algorithms requires sequential process of the origin or OD pairs (Chen et al. 2002; Bar-Gera, 2002; Slavin et al., 2006). In this way, the flow shifting of the next origin or OD pair can utilize the updated information from the shifting of the previous origin or OD pair. The flow shifting inside each origin may also benefit from this sequential structure, i.e. the portion of the flow to be shifted could be different. This sequential process is one of the key features of the origin/path-based algorithms to make them converge faster than FW algorithm. As long as the sequential structure maintains, the problem of order dependence is unavoidable. Based on our best knowledge so far, there are no origin/path-based algorithms incorporated with the power of distributed computing available in commercial software and applied in practice.

To further demonstrate the advantages of using distributed computing, we compare the run times of different algorithms. The results are illustrated in Figure 8. From the results, we can see that the GP algorithm shows almost identical performance when running with different number of cores. This is also true for OBA algorithm. It is because for the path-based (GP) algorithm and origin-based (OBA) algorithm, there is no distributed computing technology associated. From the figure, we can see that the GP algorithm is still comparable with the improved link-based algorithms when the number of cores is 1 or 2. However, the improved link-based algorithm shows more and more advantages while the number of cores increases.

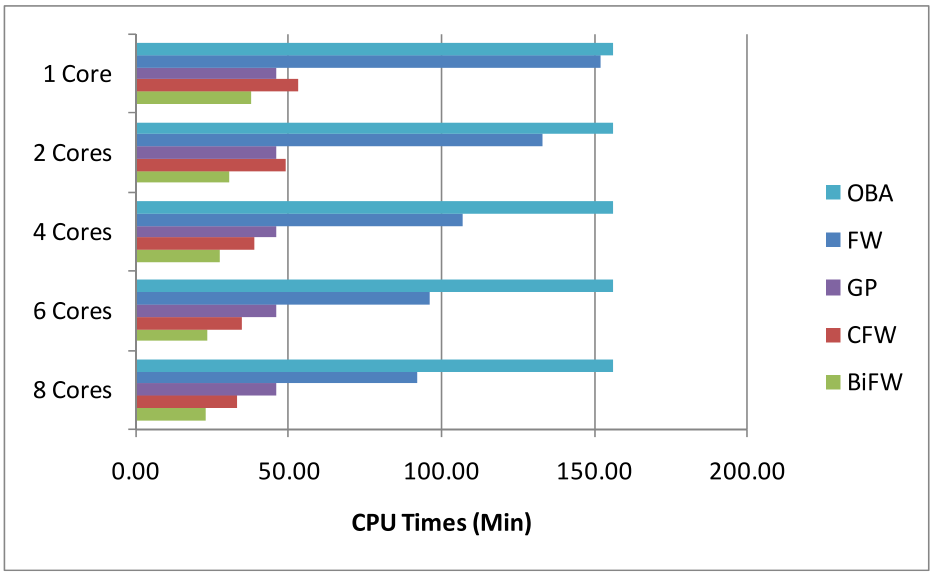

Figure 8 Run Times of Different Algorithms to Reach Relative Gap 0.0001

5. CONCLUSION

In this paper, two improved link-based algorithms (conjugated Frank-Wolfe and bi-conjugate Frank-Wolfe algorithm, and a path-based algorithms (gradient projection algorithm) implemented with CUBE Voyager software are examined. The numerical study conducted on real size networks demonstrates the ability of the algorithms to reach highly convergence levels and efficiency of the implementations. The numerical tests also illustrate that promising enhancement of the computational performance of the link-based algorithms can be achieved by using the distributed computing technology.

Future tasks include arranging more tests and comparisons on various real size networks to further examine the performance of the different algorithms and effects of using distributed computing. Also, the assignment tests with multiple classes merit further exploring. In addition, it would be interesting to investigate the properties of non-uniqueness of the equilibrium solution, such as the condition of proportionality of the UE flows (Bar-Gera 2010).

ACKNOWLEDGMENT

We would like to thank Professor David Boyce, Hillel Bar-Gera and Yu Nie for their research and helpful discussions on origin-based algorithms. We'd also like to thank Professor Anthony Chen and Shlomo Bekhor for their helpful comments on path-based algorithms.

REFERENCE

Arezki, Y. 1987. Comparison of some algorithms for equilibrium traffic assignment. In Proceedings of the 3rd mini European Conference for Transportation Problems.

Arezki, Y., and D. Van Vliet. 1990. A full analytical implementation of the PARTAN/Frank-Wolfe algorithm for equilibrium assignment. Transportation Science 24: 58-62.

Bar-Gera, H. 2002. Origin-based algorithm for the traffic assignment problem. Transportation Science 36: 398--417.

Bar-Gera, H., and D. Boyce. 2005. User-equilibrium route set analysis of a large road network. In Proceedings of the 16th International Symposium on Transportation and Traffic Theory: 673-692.

Bar-Gera, H., and A. Luzon. 2007. Non-unique solutions of user-equilibrium assignments and their practical implications. In Proceedings of the TRB 86th Annual Meeting, Washington, DC.

Bar-Gera, H. 2010. Traffic assignment by paired alternative segments. Transportation Research Part B, In press, available online doi:10.1016/j.trb.2009.11.004.

Beckmann, M. J., C. B. McGuire, and C. B. Winsten. 1956. Studies in the Economics of Transportation, Yale University Press, New Haven, CT.

Bertsekas, D. R. 1976. On the Goldstein-Levitin-Polyak Gradient Projection Method. IEEE Transactions on automatic control AC-21(2): 174-184.

Boyce, D., and H. Bar-Gera. 2001. Network Equilibrium Models of Travel Choices with Multiple Classes, p. 85-98. In M. L. Lahr and R. E. Miller (Ed.). Regional Science in Economic Analysis. Elsevier Science, Oxford, UK.

Boyce, D., B. Ralevic-Dekic, and H. Bar-Gera. 2004. Convergence of Traffic Assignments: How Much is Enough? Journal of Transportation Engineering 130(1): 49-55.

Chen, A., D. H. Lee, and R. Jayakrishnan. 2002. Computational study of state-of-the-art path-based traffic assignment algorithms. Mathematics and Computers in Simulation 59: 509-518.

Daneva, M. 2003. Improved Frank-Wolfe directions with applications to traffic problems, Thesis, Linkopings University, Linkoping, Sweden.
Dafermos, S. C. 1968. Traffic assignment and resource allocation in transportation networks, Dissertation, Johns Hopkins University, Baltimore, MD.
Dial, R. B. 2006. A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. Transportation Research Part B 40: 917-936.
Florian, M., J. Guelat, and H. Spiess. 1987. An efficient implementation of the 'PARTAN' variant of the linear approximation method for the network equilibrium problem. Networks 17: 319-339.
Florian, M., I. Constantin, and D. Florian. 2009. A new look at the projected gradient method for equilibrium assignment. In Proceedings of the Transportation Research Board Annual Meeting, Washington, D.C.
Frank, M., and P. Wolfe. 1956. An algorithm for quadratic programming. Naval Research Logistics Quarterly 3(1-2): 95-110.
Fukushima, M. 1984. A modified Frank-Wolfe algorithm for Solving the traffic assignment problem. Transportation Research Part B 18: 169-177.
Gentile, G. 2009. Linear User Cost Equilibrium: a new algorithm for traffic assignment. submitted to Transportation Research Part B.
Hearn, D., S. Lawphongpanich, and J. A. Ventura. 1985. Finiteness in Restricted Simplicial Decomposition. Operations Research Letters 4(3): 125-130.
Jayakrishnan, R., W. K. Tsai, J. N. Prashker, and S. Rajadhyaksha. 1994. A faster path-based algorithm for traffic assignment. In Proceedings of the TRB, National Research Council, Washington D.C.: 75-83.
Larsson, T., and M. Patriksson. 1992. Simplicial decomposition with disaggregated representation for the traffic assignment problem. Transportation Science 26: 4-17.
Larsson, T., M. Partriksson, and C. Rydergren. 1997. Applications of simplicial decomposition with nonlinear column generation to nonlinear network flows, p. 346-373. In P. M. Pardalos, D. W. Hearn, and W. W. Hager (Ed.). Network Optimization, Springer, Berlin.
LeBlanc, L. J., R. V. Helgason, and D. E. Boyce. 1985. Improved efficiency of the Frank-Wolfe algorithm for convex network problems. Transportation Science 19: 445-462.
Lupi, M. 1986. Convergence of the Frank-Wolfe algorithm in transportation network. Civil Engineering Systems 3: 7-15.
Nie, Y. 2008. Improved origin-based quasi-newton method for the traffic assignment problem. In Proceedings of the TRB 2008 Annual Meeting, Washington, DC.
Nie, Y. 2009. A class of bush-based algorithms for the traffic assignment problem. transportation research Part B doi:10.1016/j.trb.2009.06.005.
Patriksson, M. 1994. The Traffic Assignment Problem-Models and Methods, VSP, Utrecht, The Netherlands.
Powell, W. B., and Y. Sheffi. 1982. The convergence of equilibrium algorithms with predetermined step sizes. Transportation Science 16(1): 45-55.

Slavin, H., J. Brandon, and A. Rabinowicz. 2006. An empirical test of alternative traffic assignment methods. In Proceedings of the TRB 2006 Annual Meeting, Washington, DC.

Rose, G., M. S. Daskin, and F. S. Koppelman. 1988. An examination of convergence error in equilibrium traffic assignment models. Transportation Research Part B 22(4): 261-274.

Wardrop, J. 1952. Some theoretical aspects of road traffic research. Proceedings of the Institute of Civil Engineerings Part II, Volumn 1: 325-378.

Weintraub, A., C. Ortiz, and J. Gonzalez. 1985. Accelerating convergence of the Frank-Wolfe algorithm. Transportation Research Part B 19: 113-122.