Acta Universitatis Apulensis No. 35/2013
ISSN: 1582-5329 pp. 169-186

AN EXTENDED ORIGIN-BASED METHOD FOR SOLVING CAPACITATED TRAFFIC ASSIGNMENT PROBLEM

S. Morowati-Shalilvand, J. Mehri-Tekmeh

Abstract. In this paper, we have proposed a new algorithm for solving Capacitated Traffic Assignment Problem (CTAP). The proposed method first approximates original problem with a sequence of standard Traffic Assignment Problems (TAP) by an inner penalty strategy and then this subproblems will be solved by recently proposed Origin-Based (OB) algorithm with some modifications. This algorithm will be more useful for large scale problems since all computations in OB algorithm are done under a topological order.

2000 *Mathematics Subject Classification*: 90B20, 90C90, 90C35.

Keywords: Capacitated traffic assignment problem, Origin-Based algorithm, Penalty strategy, Topological order.

1. Introduction

Capacitated traffic assignment problem is the same as standard traffic assignment problem but link capacity constraints are added to it. Generally, without regarding multi-commodity flow models, methods provided for solving capacitated traffic assignment problem (CTAP) are divided into two categories [15]:

- Algorithms that use asymptotical link costs.
- Algorithms that transform original problem to a sequence of uncapacitated traffic assignment problems by a penalty/dual strategy.

Daganzo [5, 6], was first one that considered link capacity constraints implicitly by using asymptotical link cost functions. Although this method always provides feasible solutions but it causes some problems. Results of a research by Boyce et al. [4], shows that using asymptotical link costs functions may cause unrealistical high travel costs, devious rerouting of trips, and numerical difficulty near capacity flows.

In algorithms of second category, link capacities are added to objective function using penalty functions or Lagrange multipliers. For first time, Hearn [7], solved capacitated traffic assignment problem by transforming it to a sequence of standard TAP subproblems. He used an exterior penalty function for transforming CTAP to standard TAP, and solved subproblems by Frank-Wolf (FW) algorithm. Inouye [10] proposed a similar method by using asymptotical penalty functions, and called it inner penalty function (IPF) method, and like Hearn, he used FW algorithm to solve subproblems.

One of the other methods in second category is Augmented Lagrange Multipliers (ALM) method. This method was proposed separately by Powell [14] and Hestenes [9] for solving nonlinear programming problems. Hearn and Ribera [8] used ALM for solving traffic assignment problem with link capacity constraints. They applied FW algorithm to solve uncapacitated subproblems and provided the results for some small size networks. Larsson and Patriksson [11] also presented the results obtained from the application of a similar ALM method for small to medium size networks. In this method, the subproblems are solved using the disaggregate simplicial decomposition (DSD) algorithm. Nie et al. [12] implemented IPF method on small, medium and large scale networks and used Gradient Projection (GP) method for solving uncapacitated subproblems. Except for Larson's DSD method, solutions provided by above mentioned methods have not a simple route flow interpretation, because all of these methods are link based, that is, their main variables are total link flows.

Recently Bar-Gera [2], proposed a new method for solving TAP, and called it Origin-Based (OB) method. In Origin Based algorithm all computations are done under topological order, and as showed in [1], algorithm running time reduces significantly in large scale networks. One main advantage of OB algorithm is that, solutions provided by this algorithm have simple route flow interpretation with modest memory requirement [17].

In this paper we have proposed some modification on original OB algorithm to apply it for solving CTAP, and we called it extended origin based (EOB) algorithm. Further we proposed an iterative method, that transforms CTAP to a sequence of TAPs by an inner penalty strategy, and then solves subproblems by EOB. Solutions provided by proposed method has a simple route flow interpretation, and according to the property that all computations have topological order, we expect it to be useful for large scale networks.

The rest of paper is organized as follows. Next section discusses mathematical formulation of CTAP, and preparing CTAP to apply OB algorithm on it. A short review of OB algorithm is given in sect. 3. Extended origin based algorithm is presented in sect. 4. Convergence proof of EOB algorithm is provided in sect. 5.

2. Mathematical formulation of capacitated traffic assignment problem

2.1. Notation

A	the set of all links
A_p	the restricting subnetwork for origin p
N_o	the set of origin nodes
N_d	the set of destination nodes
R^+	the set of positive real numbers
R_{pq}	the set of all routes from p to q
d_{pq}	O-D flow (demand)
$\bar{B}_j^p$	the set of nonbasic links to node j in A_p
h_{pqr}	the flow on route r from p to q
$\vec{h}$	route flow vector
f_{ij}	total flow on link (i,j)
$\vec{f}$	total link flow vector
C_{ij}	capacity of link (i,j)
η_j^p	the flow on node j from origin p
τ_{ij}^p	proportion of η_j^p which arrives from (i,j)
t_{ij}	the cost of link (i,j)
$\vec{t}$	link cost vector
t'_{ij}	link cost derivative i.e. $t'_{ij} = \frac{\partial t_{ij}}{\partial f_{ij}}$
$\vec{t}'$	vector of link cost derivative
tt_{ij}	penalized cost of link (i,j)
$\vec{tt}$	penalized link cost vector
tt'_{ij}	penalized link cost derivative i.e. $tt'_{ij} = \frac{\partial tt_{ij}}{\partial f_{ij}}$
$\vec{tt}'$	vector of penalized link cost derivative
(b,j)	basic link among all entering links to node j
ϕ_{ij}^p	average cost of link (i,j) in A_p
ϕ_{bj}^p	average cost of basic link to node j i.e. (b,j) in A_p
ω_j^p	average cost to node j in A_p
ψ_{ij}^p	approximated derivative of ϕ_{ij}^p with respect to f_{ij}
ξ_j^p	approximated derivative of ω_j^p with respect to η_j^p
$\Omega_{ib}(j)$	desirable shift from (i,j) to (b,j) in A_p
Θ	algorithmic map of original OB algorithm
Ξ	algorithmic map of EOB algorithm
ζ_j^p	last common node to j in restricting subnetwork A_p

2.2. Mathematical Formulation

Some times we have $t_{ij} = t_{ij}(f_{ij})$ for each link (i, j), i.e. the travel time function is separable. In this special case, CTAP is given by the following convex nonlinear programming problem [11]:

$$\text{Min} \qquad Z(\vec{f}) = \sum_{(i,j)\in A} \int_0^{f_{ij}} t_{ij}(x)dx$$

$$\text{subject to :} \qquad \sum_{r\in R_{pq}} h_{pqr} = d_{pq}; \qquad \forall p \in N_o, q \in N_d \tag{1}$$

$$h_{pqr} \geq 0; \qquad \forall r \in R_{pq}, p \in N_o, q \in N_d \tag{2}$$

$$f_{ij} = \sum_{p\in N_o} \sum_{q\in N_d} \sum_{r\in R_{pq}} h_{pqr}\delta_{ij,r}^{pq}; \qquad \forall (i, j) \in A \tag{3}$$

$$f_{ij} \leq C_{ij}; \qquad \forall (i, j) \in A; \tag{4}$$

where

$$\delta_{ij,r}^{pq} = \begin{cases} 1 & (i, j) \subset r \in R_{pq} \\ 0 & \text{o.w. .} \end{cases}$$

Since the problem CTAP is a strictly convex optimization program (with respect to the link flows) subject to linear constraints, it has a unique optimal solution of link flows. Furthermore, the necessary and sufficient conditions of its optimality can be stated explicitly according to Karush-Kuhn-Tucker (KKT) conditions (See [3]). Let u_{rs} and v_{ij} denote the optimal values of multipliers associated with trip demands and link capacities respectively, then the KKT conditions are:

$$\begin{array}{rcll}
h_{pqr}\left(c_{pqr} + \sum_{r\in R_{pq}} v_{ij}\delta_{ij,r}^{pq} - u_{pq}\right) & = & 0 & \forall r \in R_{pq}, p \in N_o, q \in N_d \\
v_{ij}(C_{ij} - f_{ij}) & = & 0 & \forall (i, j) \in A \\
c_{pqr} + \sum_{r\in R_{pq}} v_{ij}\delta_{ij,r}^{pq} - u_{pq} & \geq & 0 & \forall r \in R_{pq}, p \in N_o, q \in N_d \\
(C_{ij} - f_{ij}) & \geq & 0 & \forall (i, j) \in A \\
h_{pqr} & \geq & 0 & \forall r \in R_{pq}, p \in N_o, q \in N_d \\
v_{ij} & \geq & 0 & \forall (i, j) \in A \\
\sum_{r\in R_{pq}} h_{pqr} & = & d_{pq} & \forall p \in N_o, q \in N_d;
\end{array}$$

where $c_{pqr} = \sum_{(i,j)\in A} t_{ij}\delta_{ij,r}^{pq}$ denotes the path travel cost. It can be shown that by introducing a generalized path travel cost which is defined as

$$\hat{c}_{pqr} = \sum_{(i,j)\in A} (t_{ij} + v_{ij})\delta_{ij,r}^{pq},$$

the optimality conditions of CTAP give rise to Wardrop's first principle [16] in terms of generalized travel cost [13], i.e. if $(\vec{h}, \vec{f})$ solves CTAP then

$$\begin{aligned} h_{pqr}(\hat{c}_{pqr} - u_{pq}) &= 0 \qquad \forall r \in R_{pq}, p \in N_o, q \in N_d \qquad (5) \\ \hat{c}_{pqr} - u_{pq} &\geq 0 \qquad \forall r \in R_{pq}, p \in N_o, q \in N_d. \qquad (6) \end{aligned}$$

It is obvious that (5) and (6) state the Wardrop's first principle considering the generalized travel cost of any path r.

2.3. Inner Penalty Function Method

In this method, CTAP has been defined as a sequence of unbounded traffic assignment problems, in which the objective functions include a penalty term to prevent constraint violation. Using inner penalty function (IPF) method, CTAP can be replaced by an uncapacitated TAP as follows, called CTAP-IPF by Nie et al. [12] :

$$\begin{aligned} \text{Min} \qquad T(\vec{f}, \gamma) &= \sum_{(i,j)\in A} \int_0^{f_{ij}} t_{ij}(x)dx + \gamma \sum_{(i,j)\in A} p(f_{ij}) \\ \text{subject to :} \qquad & (1) - (3), \end{aligned}$$

where $\gamma \in R^+$ is penalty parameter and

$$\begin{aligned} \lim_{f_{ij}\to C_{ij}} p(f_{ij}) = +\infty, \qquad & p(f_{ij}) > 0, \\ p(f_{ij}) \in C^m[0, C_{ij}); \qquad & m \geq 2, \forall (i,j) \in A. \end{aligned}$$

As penalty parameter γ tends to 0, solution points of CTAP-IPF problems and $\gamma \nabla p(\vec{f})$ tends to optimum point of original CTAP and optimal Lagrange multipliers vector, respectively [12].

In each iteration, say n, gradient of asymptotical penalized objective function is

$$g^n(f_{ij}, \gamma) = t_{ij} + \gamma_n \frac{dp(f_{ij})}{df_{ij}},$$

where according to past researches [18], an efficient form of $\frac{dp(f_{ij})}{df_{ij}}$ is

$$\frac{dp(f_{ij})}{df_{ij}} = \frac{1}{C_{ij} - f_{ij}}, \qquad \forall (i,j) \in A.$$

To implement OB algorithm on CTAP, we have made some modification on original CTAP-IPF model. According to selection of $\frac{1}{C_{ij}-f_{ij}}$ as penalty function, it's derivative with respect to total link flow, $\frac{dp(f_{ij})}{df_{ij}}$, is a separable, strictly ascending and convex function in $[0, C_{ij})$. Therefore by defining extended link cost as $tt_{ij} = t_{ij} + \gamma \frac{dp(f_{ij})}{df_{ij}}$, for a given constant γ, we can state CTAP-IPF equivalently as:

$$\begin{array}{llll} \text{Min} & T^{\gamma}(\vec{f}) & = & \sum_{(i,j)\in A} \int_0^{f_{ij}} tt_{ij}(x)dx \\ & \text{subject to :} & & (1) - (3). \end{array} \tag{7}$$

According to above illustrations, tt_{ij} is a separable and strictly ascending function with respect to f_{ij}; that is, $T^{\gamma}(\vec{f})$ is a convex function of $\vec{f}$. Now OB algorithm can be applied for this convex problem.

2.4. Constructing a pseudo-feasible set Our algorithm like other methods first makes an all or nothing assignment, and thus it may causes link capacity violation. According to definition of tt_{ij}, extended link cost is not well-defined when some links are over saturated. To solve this problem, we adapted the method has been used by Nie et al. [12]. They defined a temporary capacity for all saturated links as follows:

$$\widehat{C}_{ij} = \begin{cases} C_{ij} & f_{ij} < C_{ij} \\ f_{ij} + \varepsilon & \text{o.w.}, \end{cases} \tag{8}$$

where ε is a small real number. For sufficiently small ε, a significant increase is obtained on cost of saturated links, so algorithm will attempt to decrease flows on this kinds of links. Since we want to transform a pseudo-feasible solution to a feasible one, we adapt a dynamic ε as Nie et al. [12], and update (decrease) it in every iteration of the proposed method.

$$\varepsilon^n = \frac{\gamma_n}{\ell \max\left(t_{ij} + \frac{\gamma_n}{C_{ij} - f_{ij}}\right)}, \qquad 0 \le \ell \le 100. \tag{9}$$

3. Origin Based algorithm for solving TAP

"Recent researches on the OB algorithm [2] demonstrate that it is one of the state-of-the-art algorithms for solving the traffic assignment problem. There are two key steps in this algorithm:

- Restricting the origin-based subnetworks to be acyclic.
- Shifting flows within the acyclic subnetwork using cost deviations.

Since the restricting subnetwork is always acyclic for a given origin, it permits a simple route flow interpretation, enables us to define cost, and allows for a definition of topological order. Using the link proportions, the memory required to store routes is significantly reduced. The OB algorithm is considered to be suitable for large-scale networks due to its computational efficiency and modest memory requirements"[17].

Origin based algorithm for solving TAP starts to implement from shortest cost trees with all or nothing assignment as initial solutions. Then updates restricted subnetworks for every origin, and adjusts link proportions of current subnetwork. This algorithm computes the amount of flow that must be transferred from a non-basic link (i, j) to a basic link (b, j) by a point to set algorithmic map $\Theta_\lambda^{ij\to bj}$: $[0,1]^{|A|} \times \Re_+^{|A|} \times \Re_+^{|A|} \to 2^{[0,1]}$:

$$\Theta_\lambda^{ij\to bj}(\vec{\tau}, \vec{t}, \vec{t}') = \begin{cases} \left\{\min\left(\tau_{ij}^p, \lambda\frac{\Omega_{ib}(j)(\vec{\tau},\vec{t},\vec{t}')}{\eta_j^p(\vec{\tau})}\right)\right\} & \eta_j^p > 0 \\ \{\tau_{ij}^p\} & \eta_j^p = 0; \phi_{ij}^p > \phi_{bj}^p \\ ([0, \tau_{ij}^p]) & \eta_j^p = 0; \phi_{ij}^p = \phi_{bj}^p \end{cases} \tag{10}$$

where

$$\Omega_{ib}(j)(\vec{\tau}, \vec{t}, \vec{t}') = \frac{\phi_{ij}^p(\vec{\tau}, \vec{t}) - \phi_{bj}^p(\vec{\tau}, \vec{t})}{\max\left(\epsilon_\nu, \psi_{ij}^p(\vec{\tau}, \vec{t}') + \nu_{bp}(\vec{\tau}, \vec{t}') - \rho_{\zeta_j^p}(\vec{\tau}, \vec{t}')\right)}$$

is the desired amount of flow to be shifted ignoring feasibility constraints and λ is the step size in boundary search method proposed by Bar-Gera [1]. In $\Omega_{ib}(j)(\vec{\tau}, \vec{t}, \vec{t}')$, ϵ_ν is a small constant to overcome zero derivative approximation [1].

4. Extended Origin Based Algorithm for Solving CTAP

In the original OB algorithm on CTAP problem, if we specify the amount of shifted flow by original algorithmic map, then it may cause over-saturating on basic links. To overcome this, we define a new algorithmic map

$$\Xi_\lambda^{ij\to bj}(\vec{\tau}, \vec{tt}, \vec{tt}') = \begin{cases} \left\{\min\left(\tau_{ij}^p, \lambda\frac{\Omega_{ib}(j)(\vec{\tau},\vec{tt},\vec{tt}')}{\eta_j^p(\vec{\tau})}, \frac{\hat{C}_{bj}-f_{bj}}{\eta_j^p(\vec{\tau})}\right)\right\} & \eta_j^p > 0 \\ \{\tau_{ij}^p\} & \eta_j^p = 0; \phi_{ij}^p > \phi_{bj}^p \\ ([0, \tau_{ij}^p]) & \eta_j^p = 0; \phi_{ij}^p = \phi_{bj}^p. \end{cases} \tag{11}$$

In the new algorithmic map the term $\frac{\hat{C}_{bj}-f_{bj}}{\eta_j^p(\vec{\tau})}$ prevents over saturating basic link b when we transfer flow from link a to link b; because the amount of shifted flow from nonbasic link a to basic link b is $\Delta\tau_{ij}^{p}\ \eta_j^p(\vec{\tau})$, where $\Delta\vec{\tau}_p \in \Xi_{\lambda}^{ij\to bj}$, that is, $\Delta\tau_{ij}^{p} \le \frac{\hat{C}_{bj}-f_{bj}}{\eta_j^p(\vec{\tau})}$, and therefore the amount of shifted flow is less than $\hat{C}_{bj} - f_{bj}$.

In the next section we prove that the new algorithmic map is a closed map. In EOB algorithm, the sequence of penalty parameters must tend to zero, so we add a penalty parameter update step to the algorithm, and update penalty parameter at the end of every subproblem solution. Since we attempt to make pseudo-feasible solutions in every step, we must also add a link capacity update step to the algorithm and update γ at the same time.

The sequence of penalty parameters, i.e. $\{\gamma_n\}_n$, tends to zero strictly descending. So, there is a subsequence $\{\gamma_{n_k}\}_k$ that after updating penalty parameter in iteration n_k, the new penalty parameter satisfies:

$$T^{\gamma_{n_{k+1}}}(\vec{f}) \le T^{\gamma_{n_k}}(\vec{f}), \qquad k = 1, 2, 3, \ldots \ .$$

For simplicity, the subsequence $\{\gamma_{n_k}\}_k$ is denoted by $\{\gamma_n\}$, unless we define it explicitly. We can state the EOB algorithm formally as follows:

Algorithm 1.
for p in N_o **do**
  A_p :=tree of minimum cost routes from p.
  $\vec{f}_p^{*0}$:= all-or-nothing assignment using A_p.
end for
update link cost by selecting γ_0 as penalty parameter; i.e.
$tt_{ij}(f_{ij}) = t_{ij}(f_{ij}) + \gamma_0 \frac{dp(f_{ij})}{df_{ij}}$.
$n \leftarrow 1$
while convergence criteria is satisfied **do**
  update over-saturated links capacity using (8) and (9).
  update link cost selecting $\gamma_n \in \{\gamma_n\}_n$ as penalty parameter.
  while achieving given accuracy call Algorithm 2, starting from $\vec{f}^{*n-1}$
  $\vec{f}^{*n}$:= The optimal solution of (7) with penalty parameter γ_n,
    obtained from previous step.
  $n \leftarrow n + 1$.
end while

Algorithm 2.
Input a given feasible solution $(\vec{f})$ of (7) with penalty parameter γ.
Main Loop
for n from 1 to the number of main iterations (I_{main}) **do**
  for p in N_o **do**
    Update link costs $\vec{tt}_p = (tt_{ij})_{a\in A_p}$.

Update restricting subnetwork A_p.
Update link proportions vector as follows:
$k \leftarrow 0$
repeat
$\lambda \leftarrow 2^{-k}$.
Compute average costs by following recursive formulas:
$\sigma_{pp}(\vec{\tau}, \vec{tt}) = 0,$
$\phi^p_{ij}(\vec{\tau}, \vec{tt}) = tt_{ij} + \sigma_{a_t p}(\vec{\tau}, \vec{tt})$
$\omega^p_j(\vec{\tau}, \vec{tt}) = \sum_{a \in A_p; a_h = j} \tau^p_{ij} \phi^p_{ij}(\vec{\tau}, \vec{tt}), \qquad j \neq p$
Compute second order derivative approximations:
$\rho_{pp}(\vec{\tau}, \vec{tt}') = 0$
$\psi^p_{ij}(\vec{\tau}, \vec{tt}') = tt'_{ij} + \rho_{a_t p}(\vec{\tau}, \vec{tt}')$
$\xi^p_j(\vec{\tau}, \vec{tt}') = \sum_{a' \in A_p; a_h = j} \alpha^2_{ap} \psi^p_{ij}(\vec{\tau}, \vec{tt}'), \qquad j \neq p$
For all $j \in A_p \setminus \{p\}$ compute the following algorithmic map
$$\Xi^{ij \to bj}_{\lambda}(\vec{\tau}, \vec{tt}, \vec{tt}') = \begin{cases} \left\{ \min \left(\tau^p_{ij}, \lambda \frac{\Omega_{ib}(j)(\vec{\tau}, \vec{tt}, \vec{tt}')}{\eta^p_j(\vec{\tau})}, \frac{\hat{C}_{bj} - f_{bj}}{\eta^p_j(\vec{\tau})} \right) \right\} & \eta^p_j > 0 \\ \{\tau^p_{ij}\} & \eta^p_j = 0; \phi^p_{ij} > \phi^p_{bj} \\ ([0, \tau^p_{ij}]) & \eta^p_j = 0; \phi^p_{ij} = \phi^p_{bj}, \end{cases}$$
where $a_h = b_h = j$.
Aggregate the shift of all links to the same node j
$$\Xi^{j:b} = \left\{ \vec{\Delta}\alpha_p : \begin{array}{ll} \Delta\tau^p_{ij} \in -\Xi^{ij \to bj}_{\lambda}(\vec{\tau}, \vec{tt}, \vec{tt}') & \forall a \in \bar{B}^p_j \\ \Delta\alpha_{bp} = -\sum_{a \in \bar{B}^p_j} \Delta\tau^p_{ij}, & \\ \Delta\alpha_{a'p} = 0, & a'_h \neq j \end{array} \right.$$
$k \leftarrow k + 1$
until $\sum_{a \in A_p} \Delta f_{ij}\, tt(f_{ij} + \lambda \Delta f_{ij}) < 0$.
end for
for m from 1 to the number of inner iterations (I_{inner}) **do**
for p in N_o **do**
Update link costs $\vec{tt}_p = (tt_{ij})_{a \in A_p}$.
Update link proportions vector same as above.
end for
end for
end for

5. Convergence Proof

In this section we prove the convergence of the proposed algorithm. Bar-Gera [1] proved that the algorithmic map defined by (10) is closed. Here we recall this result for algorithmic map $\Xi_\lambda^{ij\to bj}(\vec{\tau},\vec{tt},\vec{tt}')$. The only difference between algorithmic maps (10) and (11) is the existence of additional term $\frac{\hat{C}_a-f_{ij}}{\eta_j^p(\vec{\tau})}$ in first case of (11).

Lemma 1. *The algorithmic map $\Xi_\lambda^{ij\to bj}(\vec{\tau},\vec{tt},\vec{tt}')$, defined by (11), is a closed map; That is, if $\vec{\tau}^k \to \vec{\tau}$, $\vec{tt}^k \to \vec{tt}$, $\vec{tt}'^k \to \vec{tt}'$, $\tilde{\vec{\theta}}^k \in \Xi_\lambda^{ij\to bj}(\vec{\tau},\vec{tt},\vec{tt}')$; then*

$$\tilde{\vec{\theta}} \in \Xi_\lambda^{ij\to bj}(\vec{\tau},\vec{tt},\vec{tt}')$$

Proof. There exist three cases:

1. $\eta_j^p(\vec{\tau}) > 0$
2. $\eta_j^p(\vec{\tau}) = 0; \phi_{ij}^p(\vec{\tau},\vec{tt}) > \phi_{bj}^p(\vec{\tau},\vec{tt})$
3. $\eta_j^p(\vec{\tau}) = 0; \phi_{ij}^p(\vec{\tau},\vec{tt}) = \phi_{bj}^p(\vec{\tau},\vec{tt})$

We prove here just case 1; proof of other cases is similar to Lemma 4 of [1]. According to [1], f_{ij} and $\eta_j^p(\vec{\tau})$ are continuous functions of $\vec{\tau}$, and therefore $\frac{\hat{C}_a-f_{ij}}{\eta_j^p(\vec{\tau})}$ is continuous. Also $\Omega_{ib}(j)(\vec{\tau},\vec{tt},\vec{tt}')$ is a continuous function of $(\vec{\tau},\vec{tt},\vec{tt}')$, therefore $\Xi_\lambda^{ij\to bj}(\vec{\tau},\vec{tt},\vec{tt}')$ is continuous in a neighborhood of $(\vec{\tau},\vec{tt},\vec{tt}')$, which completes the proof.

Lemma 2. *There is a subsequence $\{\gamma_n\}_n$ of penalty parameters such that after updating link capacities, γ_{n+1} satisfies:*

$$T^{\gamma_{n+1}}(\vec{f}^{*n}) \le T^{\gamma_n}(\vec{f}^{*n}), \tag{12}$$

where $\vec{f}^{*n}$ is a solution of (7), obtained from Algorithm 1 in iteration n, with penalty parameter γ_n and given accuracy.

Proof. Let $\overline{T}^{\gamma_n}(\vec{f}^{*n})$ be the objective function value of problem (7) after updating link capacities and before updating penalty parameter. We have two cases:

Case 1: If $\overline{T}^{\gamma_n}(\vec{f}^{*n}) \le T^{\gamma_n}(\vec{f}^{*n})$, then according to descending property of penalty parameters, $\gamma_{n+1} \le \gamma_n$, and therefore according to construction of tt_{ij}, the amount of $\overline{T}^{\gamma_n}(\vec{f}^{*n})$ will not increase after updating penalty parameter; that is, $T^{\gamma_{n+1}}(\vec{f}^{*n}) \le T^{\gamma_n}(\vec{f}^{*n})$.

Case 2: Otherwise, because penalty parameters sequence is descending and convergent to zero, according to construction of tt_{ij}, after passing a finite number of penalty parameters, (12) will be satisfied.

Theorem 3. *Algorithm 1 is convergent.*

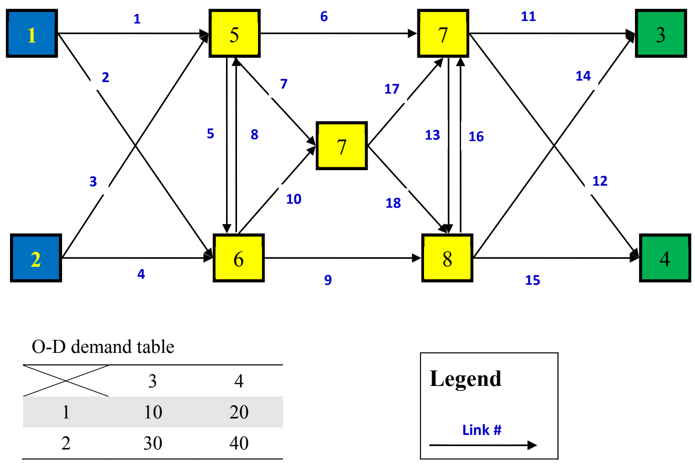
Proof. According to Lemma 1, Algorithm 2 is convergent for subproblems. Also, for a constant penalty parameter, objective function value is a descending sequence during implementation Algorithm 2, and according to Lemma 2, objective function value doesn't increase after updating link capacities and penalty parameter. So, objective function value sequence $\left\{T^{\gamma_n}(\vec{f}^{kn})\right\}_{k,n}$ is always descending. But, this sequence is bounded from below, so is convergent.

6. Numerical Results

In this section we will examine EOB algorithm on some real world networks and an small size network. The EOB algorithm will also be compared with two experimented algorithms in literature. These two algorithms are IPF(Inner Penalty Function using GP subproblem solver) and ALM(Augmented Lagrange Method) (see [12]). We have employed BPR performance function in all of our test problems. The EOB algorithm was coded in MATLAB and tested on a PC with a 1.8 GHz P-IV CPU and 1GB SDRAM. Since [12] have coded IPF and ALM algorithms in Fortran while EOB algorithm is coded in MATLAB, we just compare EOB with IPF and ALM algorithms using number of iterations. In all of our experiments we have used penalty parameter sequences $\gamma_n = q^n$, where $q \in (0, 1)$. We have selected $\ell = q$ in our experiments.

Our first test problem is an small size well known network namely Hearn's nine node network. Figure 1 shows the Hearn's network. Nodes 1 and 2 are origin nodes and nodes 3 and 4 are destination nodes.

Table 1 represents the results of implementing algorithms IPF, ALM and EOB on Hearn's network. The results due to IPF and ALM algorithms are adopted from [12]. Nie et al. [12] reported that IPF algorithm achieves an objective of 1572.31 after 19 main iterations and 156 GP iterations, while ALM takes an objective value 1572.36 after 65 main iterations and 84 GP iterations. However EOB algorithm reaches an objective value 1572.287 after 18 iterations of Algorithm 1 and 93 iterations of Algorithm 1. It will be seems that rapid convergence and accurate solutions of EOB algorithm depends on accuracy and rapid convergence of OB algorithm(Algorithm 2). As showed in Table 1 the solution obtained by EOB algorithm



	3	4
1	10	20
2	30	40

Figure 1: Hearn's nine node network

has less queuing delay than that of both IPF and ALM algorithms. Note that each of these algorithms have provided a solution with six active constraints.

Our experiments demonstrated that for medium and big size networks EOB algorithm is an advantageous algorithm, since it can achieve very accurate solutions in subsequent cpu time and number of iterations. Another result is that EOB algorithm was stable with respect to penalty parameter sequence selection.

In Figure 2 and Figure 3 we have examined EOB algorithm on well known Sioux Falls network with 24 nodes, 76 links and 528 O-D pairs. Figure 2 represents the application of EOB algorithm on Sioux Falls network with three different sequences of penalty parameters (i.e. $\gamma_n = (0.5)^n, \bar{\gamma}_n = (0.1)^n, \tilde{\gamma}_n = (0.01)^n$). As is obvious from Figure 2, EOB algorithm is stable with respect to selection of penalty parameters. Our experiments declared that flow infeasibility drops down to zero in several beginning iterations of EOB algorithm. Figure 3 and Figure 6 confirms this claim. With a penalty parameter $\gamma_n = (0.1)^n$ after forth iteration flow infeasibility dropped to zero in Sioux Falls network.

Nie et al. [12] reported that for Sioux Falls network, IPF reaches an objective value 33.31 implementing 15 main iterations and 141 inner GP iterations while ALM achieves a same objective function value after 26 main iterations and 127 GP iterations. However our experiments showed that EOB algorithm reaches an objective value 33.2897 after 13 iterations of Algorithm 1 and 133 iterations of Algorithm 2. As a result EOB algorithm is superior to both IPF algorithm and

Table 1: The flow patterns by different solution methods for Hearn's network.

			IPF		ALM		EOB	
Link♯	Original Capacity	Modified Capacity	Flow	Delay	Flow	Delay	Flow	Delay
1	10	12.02	**12.02**	0.14	**12.02**	0.22	**12.02**	0.135
2	16	18.02	17.98	0.00	17.98	0.00	17.98	0.000
3	35	43.59	**43.59**	5.92	**43.59**	5.94	**43.59**	5.908
4	18	26.59	26.42	0.00	26.42	0.00	26.42	0.000
5	50	50.00	0.14	0.00	1.21	0.00	0.10	0.000
6	25	25.00	20.46	0.00	19.40	0.00	20.38	0.000
7	35	35.00	**35.00**	0.77	**35.00**	0.68	**35.00**	0.775
8	50	50.00	0.00	0.00	0.00	0.00	0.00	0.000
9	25	25.00	20.46	0.00	20.66	0.00	20.41	0.000
10	35	35.00	24.07	0.00	24.95	0.00	24.00	0.000
11	25	25.00	**25.00**	5.76	**25.00**	5.59	**25.00**	5.765
12	24	24.00	**24.00**	0.54	**24.00**	0.54	**24.00**	0.540
13	50	50.00	5.54	0.00	5.34	0.00	5.58	0.000
14	39	39.00	15.00	0.00	15.00	0.00	15.00	0.000
15	43	43.00	36.00	0.00	36.00	0.00	36.00	0.000
16	50	50.00	0.00	0.00	0.00	0.00	0.00	0.000
17	35	35.00	34.07	0.00	34.95	0.00	34.00	0.000
18	25	25.00	**25.00**	0.99	**25.00**	0.99	**25.00**	0.987

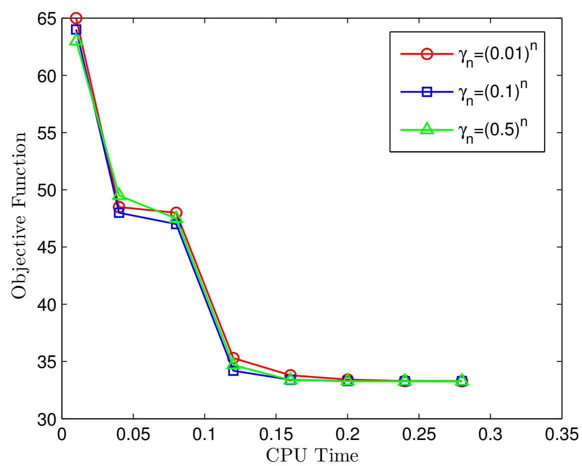


Figure 2: EOB Algorithm on Sioux-falls network: CPU time vs. objective function value for penalty parameter sequences $\gamma_n = (0.5)^n, \gamma_n = (0.1)^n$ and $\gamma_n = (0.01)^n$.

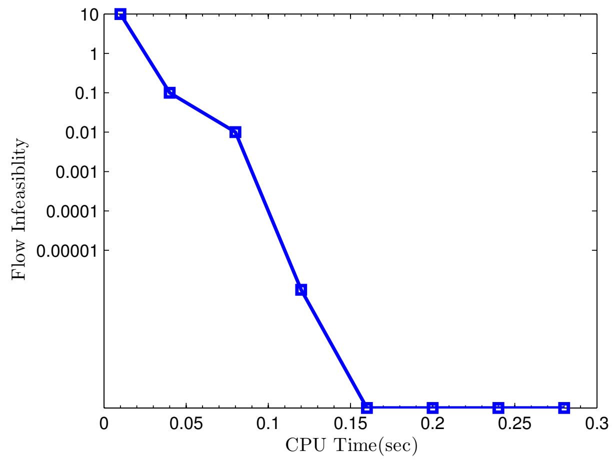


Figure 3: EOB Algorithm on Sioux-falls network: CPU time vs. flow infeasibility.

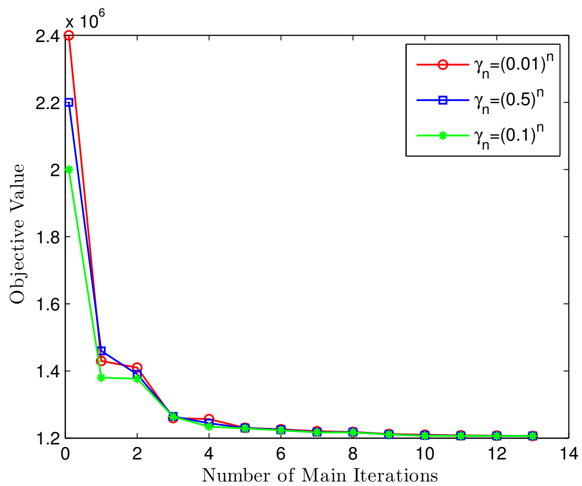


Figure 4: EOB Algorithm on Anaheim network: Number of main iterations (Algorithm 1) vs. objective function value for penalty parameter sequences $\gamma_n = (0.1)^n, \gamma_n = (0.05)^n$ and $\gamma_n = (0.01)^n$.

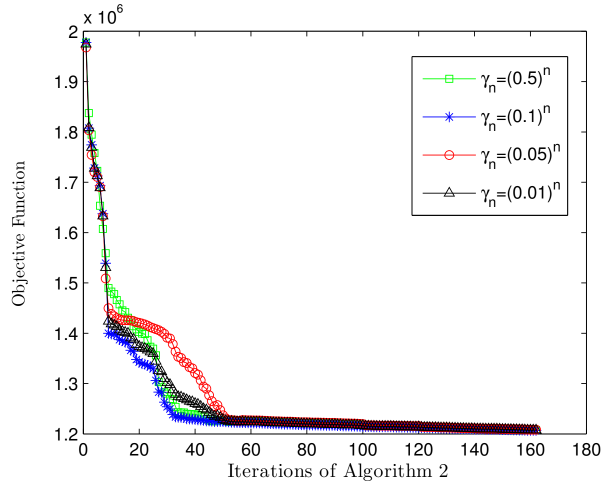


Figure 5: EOB Algorithm on Anaheim network: Number of inner iterations (Algorithm 2) vs. objective function value for several penalty parameter sequences.

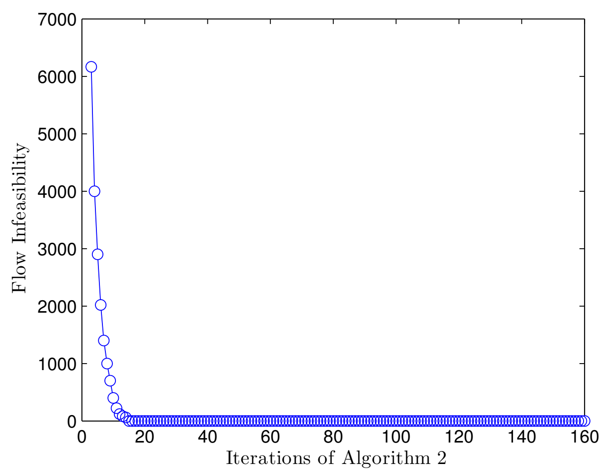


Figure 6: EOB Algorithm on Anaheim network: Number of inner iterations (Algorithm 2) vs. flow infeasibility.

ALM.

We also examined EOB algorithm on Anaheim network with 416 nodes, 914 links and 1406 OD pairs. Figure 4 represents behavior of objective function value versus number of main iterations of EOB algorithm(i.e. Algorithm 1) for several penalty parameter sequences. Figure 5 illustrates the stability of EOB algorithm with respect to penalty parameter sequence selection. As is said previously, our experiments showed that flow infeasibility vanishes in several initial iterations of EOB algorithm. For Anaheim network, Figure 6 shows that after 20 iterations of algorithm 2, flow infeasibility drops down to zero.

In our tests on Anaheim network, EOB algorithm achieved an objective value 1,206,461 after 13 iterations of Algorithm 1 and 150 iterations of Algorithm 2. The best objective value reported in [12] for Anaheim network is bigger than the value achieved by EOB algorithm. Nie et al. [12] reported that IPF obtains a feasible solution with an objective value of 1,206,474.50 after 15 main iterations, 181 inner GP iterations and ALM method achieves objective value 1,206,910.37 after 99 main iterations, 130 GP iterations. As a comparison between EOB algorithm with IPF and ALM algorithms it can be claimed that to achieve accurate solutions in big networks, EOB algorithm acts better than IPF and ALM.

7. Conclusion

We proposed an algorithm to solve capacitated traffic assignment problem. In this algorithm, all computations are done under a topological order, so it does not need more computational times. Another advantage of this algorithm is simple route flow interpretation of its solution, in fact this is one of the main advantages of original origin based algorithm to other algorithms, because transforming a link based solution to a route based solution is not trivial task.

Like original OB algorithm, according to using link proportions as main variables in extended algorithm, memory requirement is significantly reduced. Our examinations Sioux-falls and Anaheim networks showed that EOB algorithm achieves accurate solutions in subsequent time. In this sense EOB algorithm was superior to both ALM and IPF with GP subproblem solver.

Acknowledgements. The authors would like to give special thanks to Dr. K. Mirnia for his useful comments and editing the paper.

References

[1] Bar-Gera H., *Origin-Based Algorithms For Transportation Network Modeling*, Technical Report #103, National Institute of Statistical Science (1999) available at http://www.niss.org/technicalreports/tr103.pdf

[2] Bar-Gera H., *Origin-based algorithm for the traffic assignment problem*, Transportation Science 36 (4), pp. 398-417 (2002)

[3] Bazaraa M.S., Sherali H.D. and Shetty C.M., *Nonlinear Programming: Theory and Algorithms* (2nd ed.), John Wiley and Sons, New York (1993)

[4] Boyce D., Janson B.N. and Eash R.W., *The effect on equilibrium trip assignment of different link congestion functions*, Transportation Research 15A, pp. 223-232 (1981)

[5] Daganzo C.F., *On the traffic assignment problem with flow dependent costs-I.* Transportation Research 11, pp. 433-437 (1977)

[6] Daganzo C.F., *On the traffic assignment problem with flow dependent costs–II.* Transportation Research 11, pp. 439-441 (1977)

[7] Hearn D.W., *Bounding flows in traffic assignment models.* Research Report 80-4 Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL (1980)

[8] Hearn D.W. and Ribera J., *Bounded flow equilibrium problems by penalty methods.* In: Proceedings of IEEE International Conference on Circuits and Computers, pp. 162-166 (1980)

[9] Hestenes M.R., *Multipliers and gradient methods.* Journal of Optimization Theories and Applications 4, pp. 303-320 (1969)

[10] Inouye H., *Traffic equilibria and its solution in congested road networks.* in: R. Genser, (Ed.), Proceedings of IFAC Conference on Control in Transportation Systems, pp. 267-272 (1987)

[11] Larsson T. and Patriksson M., *An augmented Lagrangian dual algorithm link capacity side constrained traffic assignment problems.* Transportation Research 29B, pp. 433-455 (1995)

[12] Nie Y., Zhang H.M. and Lee D.H., *Models and algorithms for the traffic assignment problem with link capacity constraints*, Transportation Research 38B, pp. 285-312 (2004)

[13] Patriksson M., *The Traffic Assignment Problem-Models and Methods*, VSP, BV, Utrecht, The Netherlands, (1994)

[14] Powell M.J.D., *A method for nonlinear constraints in minimization problems.* In: Fletcher R., Editor, Optimization, Academic Press, New York, pp. 283-298 (1969)

[15] Shahpar Amir H., Ashtiani Hedayat Z., Babazadeh A., *Dynamic penalty function method for the side constrained traffic assignment problem.* Applied Mathematics and Computation 206, pp. 332-345 (2008)

[16] Wardrop J. G., *Some theoretical aspects of road traffic research*, in: proceedings of the institute of Civil Engineers,Part II, vol. 1, pp. 325-378 (1952)

[17] Xu M., Chen A. and Gao Z., *An improved origin-based algorithm for solving the combined distribution and assignment problem*, European Journal of Operational Research 188, pp. 354-369 (2008)

[18] Yang H. and Yagar S., *Traffic assignment and traffic control in genral freeway-arterial corridor system.* Transportation Research 28B, pp. 463-486 (1994)

[19] R. Bala, B. Ram, *Trigonometric series with semi-convex coefficients*, Tamang J. Math. 18, 1 (1987), 75-84.

[20] B. Ram, *Convergence of certain cosine sums in the metric space L*, Proc. Amer. Math. Soc. 66 (1977), 258-260.

Shahram Morowati-Shalilvand
Department of Applied Mathematics, Faculty of Mathematical Sciences,
University of Tabriz,
Tabriz, Iran
email: *ShMorowati@live.com*

Javad Mehri-Tekmeh
Department of Applied Mathematics, Faculty of Mathematical Sciences,
University of Tabriz,
Tabriz, Iran
email: *JMehri@tabrizu.ac.ir*