Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Origin-Based Algorithm for the Traffic Assignment Problem

Hillel Bar-Gera,

Origin-Based Algorithm for the Traffic Assignment Problem

Hillel Bar-Gera
Department of Industrial Engineering and Management, Ben-Gurion University, P.O.B. 653, Be'er-Sheva 84105, Israel
bargera@bgumail.bgu.ac.il

We present an origin-based algorithm for the traffic assignment problem, which is similar conceptually to the algorithm proposed by Gallager and Bertsekas for routing in telecommunication networks. Apart from being origin-based, the algorithm is different from other algorithms used so far for the traffic assignment problem by its restriction to acyclic solutions and by the use of approach proportions as solution variables. Projected quasi-Newton search directions are used to shift flows effectively and to eliminate residual flows. Experimental results comparing the proposed algorithm with the state-of-the-practice algorithm of Frank and Wolfe demonstrate the algorithm's excellent convergence performance, especially when highly accurate solutions are needed. Reasonable memory requirements make this algorithm applicable to large-scale networks. The resulting solution has an immediate route flow interpretation, thus providing equivalent detail to route-based solutions.

Introduction

At the heart of most transportation models stands the traffic assignment problem, which is to predict the route choices of travelers given their origins and their destinations, under the assumption that each traveler seeks to minimize the time/cost associated with their chosen route. Most algorithms used in practice solve the traffic assignment problem in terms of the total flow on each link (roadway segment) and discard information about the origin and the destination of travelers. Even though theoretical convergence of such link-based algorithms is guaranteed, these algorithms often fail to achieve highly accurate solutions within reasonable amounts of computation time. An alternative approach that has received increasing academic interest is the route-based approach that records all used routes and their flows. This approach can achieve higher accuracies, but at the expense of large memory requirements often regarded as impractical for large-scale networks.

This paper follows a different approach of representing the solution by origin-based link flows aggregated over all destinations. A key point in the proposed algorithm is that only acyclic solutions are considered through all iterations. The absence of cycles allows most of the computations to be performed in a time that is a linear function of network size. A simple and efficient procedure allows the disaggregation of acyclic origin-based link flows into route flows. In that sense, the detail provided by acyclic origin-based link flows is equivalent to the detail provided by route flows. However, an origin-based solution does not require as much memory as an equivalent route-based solution. Special data structures can reduce memory requirements even further. Computational efficiency in time and memory makes this algorithm highly suitable for large-scale networks.

The algorithm presented in this paper is conceptually similar to the algorithm proposed by Gallager (1977a) for routing in packet-switched telecommunication networks and further developed by Bertsekas (Bertsekas 1979, Bertsekas et al. 1979, Bertsekas et al. 1984). The main difference between the two algorithms is the way restricted subnetworks are updated and new links are introduced. This paper also introduces the concept of "last common node," which is

0041-1655/02/3604/0398$05.00
1526-5447 electronic ISSN

used in the approximation of second-order derivatives. A detailed discussion of the differences between the two algorithms is given in Bar-Gera (1999, §2.10). The presentation of the algorithm in this paper is substantially different from that of Gallager and Bertsekas. The user-equilibrium context simplifies the discussion and provides important intuition. New notation is presented to clarify the derivation and the formulation of the algorithm.

To simplify the discussion, and particularly the proof of convergence, we assume that travelers' route choices respect the deterministic user-equilibrium condition of Wardrop (1952); O-D flows are static in time and fixed with respect to the cost of travel; and link costs are separable monotonic, strictly positive functions of link flows. A proof of convergence for the case where link costs may be zero is given in Bar-Gera (1999). In the definition of the algorithm only link-cost functions and their derivatives are used; therefore, the algorithm can be applied as is to any general cost structure including nonseparable asymmetric costs. Theoretical convergence in the latter case, however, is not necessarily guaranteed.

The rest of the paper is organized as follows. Section 1 provides general definitions, a mathematical statement of the traffic assignment problem, and a discussion of the acyclic nature of its solution. A short review of the literature is given in §2. An overview of the proposed algorithm is given in §3. The algorithm is based on a reformulation of the problem, which is derived in §4, followed by a formal definition of the algorithm in §5. Experimental results are presented in §6. Finally, conclusions and plans for future research are discussed in §7.

1. Definitions and Problem Statement

Consider a *study area* that is divided into *zones*. The activities in each zone are represented in the model as if they all occur at the same point, the *zone centroid*. The transportation network is represented by a strongly connected directed graph $G = \{N, A\}$, where N is the set of nodes, A the set of directed links, and by strongly connected we mean that there is a directed route between every two nodes. A (simple) route segment is a sequence of (distinct) nodes $[v_1, \ldots, v_k]$ such that $[v_l, v_{l+1}] \in A \ \forall 1 \le l \le k-1$. In particular, the route segment $[i, j]$ is the link from node i to node j. (We assume that there is only one link, if any, between every pair of nodes, and that there are no links from a node to itself.) For generality we allow the route segment $[v]$, which is the empty route segment at v, i.e., the route segment that starts from v, ends at v, and does not contain any links. The first node of route segment r is considered its *tail* and denoted by r_t, and the last node is considered the route's *head*, denoted by r_h. In particular, by definition, $a \equiv [a_t, a_h]$ for every link $a \in A$ and $a_t, a_h \in N$.

The set of all simple route segments, that is, route segments that do not contain cycles, from node i to node j is denoted by R_{ij}. The set of all simple routes is denoted by $R = \bigcup_{i,j \in N} R_{ij}$. If route segment $r = [i = v_1, \ldots, v_n = j] \in R_{ij}$ is followed by route segment $s = [j = u_1, \ldots, u_m = k] \in R_{jk}$, then the combination (concatenation) of the two segments is denoted by $(r+s) = [i = v_1, \ldots v_{n-1}, v_n = j = u_1, u_2 \ldots, u_m = k]$. The statement $s \subseteq r$ means that route segment s is part of route segment r. In particular, $a \subseteq r$ means that link a is part of route r; this relationship is also represented by the element of the link-route incidence matrix δ_{ra}, which is equal to one if link a is part of route r and zero otherwise. The inclusion of route segments requires not only that one set of nodes is included in the other set of nodes, but also that the sequential order of the nodes is preserved, for example, $[1, 3] \not\subseteq [1, 2, 3]$.

The set of possible origins is denoted by N_o, and the set of possible destinations for each origin $p \in N_o$ is denoted by $N_d(p)$. The flow of travelers (also called demand) in units of vehicles per hour (vph) from each origin $p \in N_o$ to every destination $q \in N_d(p)$ is denoted by d_{pq}; $\mathbf{d}$ denotes the array of O-D flows. Flows are viewed as averages or expected values; therefore, they can be fractional and do not have to be integers. The flow along route $r \in R_{pq}$ from origin p to destination q is denoted by h_{rpq}, and $\mathbf{h}$ denotes the vector of route flows. Aggregating route flows through a link over all destinations results in origin-based link flows

$f_{ap}(\mathbf{h}) = \sum_{q\in N_d(p)} \sum_{r\in R_{pq}:a\subseteq r} h_{rpq}$. The $|A|$ by $|N_o|$ array of origin-based link flows is denoted by $\mathbf{f}$. Further aggregating those over all origins result in total link flows $f_{a\bullet}(\mathbf{h}) = \sum_{p\in N_o} f_{ap}$. The vector of total link flows is denoted by $\mathbf{f}_\bullet$.

The traffic assignment problem is to allocate the given O-D flows to specific routes according to a certain behavioral hypothesis. The most common assumption, known as Wardrop's user-equilibrium principle, is that travelers seek to minimize the cost associated with their chosen routes. The term *cost* is used here in the most general way, and can be interpreted as travel time, monetary cost, or any other measure of disutility.

The focus of this paper is a relatively naive model which is based on several additional assumptions. O-D flows are assumed to be static in time and fixed with respect to travel costs. All travelers are assumed to be identical in the sense that they value time, monetary cost, and other route attributes in the same way. Travelers are assumed to have perfect information about actual travel conditions. Route cost is assumed to be the sum of link costs along the route, $c_s = \sum_{a\subseteq s} t_a$, where $\mathbf{t} = (t_a)_{a\in A}$ represents the vector of link costs. Travel times, and therefore travel costs, depend upon the level of congestion as reflected by total link flows. For simplicity we assume that link costs are separable; that is, $t_a(\mathbf{f}_\bullet) = t_a(f_{a\bullet})$. We further assume that link costs are strictly positive, monotonically nondecreasing, and continuously differentiable functions of total link flows. For brevity the cost derivative of link a is denoted by $t'_a = \mathrm{d}t_a/\mathrm{d}f_{a\bullet}$.

Simplistic as it is, this basic model, known as the static deterministic symmetric user-equilibrium *traffic assignment problem* (TAP), is widely used in practice and has been thoroughly studied by various researchers. Many results and insights from TAP are also valid for extensions of this model. As indicated in the introduction, the proposed algorithm presented here is likely to be useful for extensions of TAP as well; however, this remains a subject for future research.

It is well known that under the assumptions stated above, the traffic assignment problem is equivalent to the following convex optimization problem (Patriksson 1994):

$$\begin{aligned} &[\text{TAP}] \quad \min\ T(\mathbf{f}_\bullet(\mathbf{h})) = \sum_{a\in A} \int_0^{f_{a\bullet}(\mathbf{h})} t_a(x)\,\mathrm{d}x \\ &\qquad \text{s.t.} \sum_{r\in R_{pq}} h_{rpq} = d_{pq} \quad \forall p\in N_o \quad \forall q \in N_d(p), \\ &\qquad \mathbf{h} \ge 0. \end{aligned} \tag{1}$$

We denote the set of feasible solutions for TAP by H, and the set of optimal solutions by H^*. In an origin-based framework it is sometimes convenient to replace the O-D flow constraint with an origin-based conservation of flow constraint, $\mathbf{f}^t \cdot \mathbf{E} = \mathbf{e}$, where $\mathbf{e}$ is the expanded O-D flow matrix, $e_{pq} = d_{pq}$ for all $q \in N_d(p) \setminus \{p\}$; $e_{pp} = -\sum_{q\in N_d(p)\setminus\{p\}} d_{pq}$; and $e_{pi} = 0$ for all $i \notin N_d(p)$; $i \ne p$. $\mathbf{E}$ denotes the link-node incidence matrix, $E_{aa_h} = 1$; $E_{aa_t} = -1$; and $E_{ai} = 0$ for $i \ne a_h$; $i \ne a_t$. We consider nonnegative origin-based link flow arrays that satisfy this constraint as *conservative*, and denote by F the set of all conservative origin-based link flow arrays. In formulating TAP it is possible to consider F rather than H as the feasible set. Hagstrom and Tseng (1998) discuss the differences between the resulting formulations.

One of the well-known properties of user-equilibrium solutions is that they need not contain cyclic flows. It is relatively easy to demonstrate that routes that contain cycles need not be used by a user-equilibrium solution. Following Gallager (1977b), we define a feasible solution $\mathbf{h} \in H$ to be *completely acyclic* iff it is minimal feasible, i.e., if for any other feasible solution $\mathbf{h}' \in H$; $\mathbf{f}_\bullet(\mathbf{h}') \le \mathbf{f}_\bullet(\mathbf{h}) \Rightarrow \mathbf{f}_\bullet(\mathbf{h}') = \mathbf{f}_\bullet(\mathbf{h})$. Notice that if $\mathbf{h} \in H$ is not minimal feasible, then there is a minimal feasible solution $\mathbf{h}' \in H$ such that $\mathbf{f}_\bullet(\mathbf{h}') \le \mathbf{f}_\bullet(\mathbf{h})$. The difference in total link flows $\Delta\mathbf{f}_\bullet = \mathbf{f}_\bullet(\mathbf{h}) - \mathbf{f}_\bullet(\mathbf{h}')$ is a nonzero circulation, where a circulation is defined as a flow vector $\mathbf{x}$ s.t. $\mathbf{x} \ge 0$; $\mathbf{x}^t \cdot \mathbf{E} = 0$. It is well known and fairly easy to see that any nonzero circulation can be represented as a sum of cyclic flows. In that sense $\mathbf{h}$ is not completely acyclic.

In this paper we focus on solutions that are *acyclic by origin*, meaning that for every origin p and every directed cycle $s = [v_1, \ldots, v_n = v_1]$ there is at least one unused link $a \subseteq s$: $f_{ap}(\mathbf{h}) = 0$. We say that an origin-based link flow array $\mathbf{f}$ is *minimal conservative* iff for any other conservative origin-based link flow array $\mathbf{f}' \in F$; $\mathbf{f}' \le \mathbf{f} \Rightarrow \mathbf{f}' = \mathbf{f}$.

Lemma 1. *$\mathbf{f} \in F$ is acyclic by origin iff it is minimal conservative.*

Proof. Suppose $\mathbf{f}$ is not acyclic by origin; i.e., there exists an origin p_0 and a cycle $s = [v_1, \ldots, v_n = v_1]$ such that $f_{ap_0} > \epsilon \ \forall a \subseteq s$ for some $\epsilon > 0$. Consider the origin-based link flow array $\mathbf{f}^1$ where $f^1_{ap_0} = \epsilon \ \forall a \subseteq s$; $f^1_{ap_0} = 0 \ \forall a \nsubseteq s$; $f^1_{ap} = 0 \ \forall p \ne p_0 \ \forall a \in A$, and the array $\mathbf{f}^2 = \mathbf{f} - \mathbf{f}^1 \ge 0$. $\mathbf{f}^1$ is an origin-based circulation; therefore ${\mathbf{f}^1}^t \cdot \mathbf{E} = 0$ and therefore ${\mathbf{f}^2}^t \cdot \mathbf{E} = \mathbf{f}^t \cdot \mathbf{E} = \mathbf{e}$, meaning that $\mathbf{f}^2$ is conservative. Clearly $\mathbf{f}^2 \le \mathbf{f}$; $\mathbf{f}^2 \ne \mathbf{f}$; hence $\mathbf{f}$ is not minimal conservative.

Suppose $\mathbf{f}$ is not minimal conservative; then there is a minimal conservative origin-based link flow array $\mathbf{f}' \le \mathbf{f}$. The difference $\Delta\mathbf{f} = \mathbf{f} - \mathbf{f}'$ is nonzero and consists of origin-based circulations that can be represented as a sum of origin-based cyclic flows. □

Lemma 2. *If $\mathbf{h} \in H$ is completely acyclic, then $\mathbf{h}$ is acyclic by origin.*

Proof. Suppose $\mathbf{h} \in H$ is not acyclic by origin, then $\mathbf{f}(\mathbf{h})$ is not minimal conservative, and therefore there is a minimal conservative solution $\mathbf{f}' \le \mathbf{f}(\mathbf{h})$; $\mathbf{f}' \ne \mathbf{f}(\mathbf{h})$. By Lemma 1, $\mathbf{f}'$ is acyclic by origin. Any conservative acyclic origin-based link flow array can be represented as an aggregation of a feasible route flow solution (Bar-Gera and Boyce 1999); hence $\mathbf{f}' = \mathbf{f}(\mathbf{h}')$ for some $\mathbf{h}' \in H$. It may be possible to choose $\mathbf{h}'$ so that $\mathbf{h}' \le \mathbf{h}$, but this is not necessary. In any case, since $\mathbf{f}(\mathbf{h}') \le \mathbf{f}(\mathbf{h})$; $\mathbf{f}(\mathbf{h}') \ne \mathbf{f}(\mathbf{h})$, then $\mathbf{f}_\bullet(\mathbf{h}') \le \mathbf{f}_\bullet(\mathbf{h})$; $\mathbf{f}_\bullet(\mathbf{h}') \ne \mathbf{f}_\bullet(\mathbf{h})$ and hence $\mathbf{h}$ is not minimal feasible. □

Lemma 3. *There exists an optimal solution for TAP that is minimal feasible.*

Proof. Suppose $\mathbf{h} \in H^*$ is an optimal solution of TAP. Let $\mathbf{h}' \in H$ be a minimal feasible solution such that $\mathbf{f}_\bullet(\mathbf{h}') \le \mathbf{f}_\bullet(\mathbf{h})$. Since link costs are nonnegative, $T(\mathbf{f}_\bullet(\mathbf{h}')) \le T(\mathbf{f}_\bullet(\mathbf{h}))$. Therefore, $\mathbf{h}'$ is also an optimal solution for TAP which is minimal feasible. □

Conclusion. There exists an optimal solution for TAP that is acyclic by origin.

2. Review of Algorithms for TAP

Since the original formulation of the traffic assignment problem by Beckmann et al. (1956), many algorithms for its solution have been presented. In the following review, some of the main algorithms are briefly described, in particular those that are pertinent to this work. There are several excellent extensive reviews on the problem and existing algorithms; see Patriksson (1994) or Florian and Hearn (1995).

The most common algorithm for TAP is based on the general nonlinear optimization method of Frank and Wolfe (1956). In each iteration of the Frank and Wolfe (FW) algorithm, a subproblem of minimizing the linearized objective function is solved by assigning all traffic to minimum cost routes, where link costs are determined by the link flows in the current solution for the main problem. A new solution is obtained by minimizing the original objective function over the line segment connecting the current solution and the subproblem solution. The objective function can be evaluated using total link flows only. An aggregated link-based representation of the current solution is therefore sufficient for this method. As a result, the memory requirement of this method is relatively small, which is its main advantage. The main drawback of FW is its slow convergence rate. See Patriksson (1994, pp. 99–101) for more details, and §6 for computational examples.

Related link-based methods were proposed by Florian and Spiess (1983), Fukushima (1984), LeBlanc et al. (1985), Lupi (1986), and others. In all cases some combination of previous solutions and the subproblem solution is used as a search direction. The Restricted Simplicial Decomposition (RSD) method of Hearn et al. (1987) suggests performing a multidimensional search over the convex hull of all previous subproblem solutions. The nonlinear simplicial decomposition of Larsson et al. (1998) is a similar method in the sense that solutions to the main problem are obtained by solving a similar multidimensional nonlinear problem, except that the subproblems are nonlinear, rather than linear, approximations of the main problem.

Recently, increased attention has been given to route-based methods. These methods assume that all used routes, and the flow on each route, are known for the current solution. Using that information, flows can be shifted from high-cost routes to low-cost routes in order to achieve equilibrium. The first method proposed to solve TAP, in fact, was a route-based method. In this method, for each O-D pair considered sequentially in a cyclic order, flows are shifted from the maximum cost used route to the minimum cost route until both routes have the same cost. This idea was suggested by Dafermos (1968) and Dafermos and Sparrow (1969) and independently implemented by Gibert (1968). Bothner and Lutter (1982) implemented a similar route-based method. When link-cost derivatives are known, they can be used to approximate flow shifts from all routes to the minimum cost route of every O-D pair. The aggregation of flow shifts over all O-D pairs is used as a search direction, and the next solution is chosen as the minimum point of the objective function along that direction. Larsson and Patriksson (1992) refer to this approach as Disaggregated Simplicial Decomposition (DSD); they also provide encouraging experimental results. Jayakrishnan et al. (1994) proposed another route-based method, where shifts are based on Gradient Projection (GP). In general, route-based methods seem to achieve high accuracy levels.

A third category of solution methods is the origin-based approach. The first minimization formulation of the traffic assignment problem proposed by Beckmann et al. (1956) was in fact origin-based. To the best of our knowledge, there have been few attempts to pursue this approach in developing computational methods for TAP. Bruynooghe, Gilbert, and Sakarovitch (1969) made an attempt to develop such a method; however, Gibert (1968) subsequently concluded that the presence of cycles makes origin-based algorithms quite complicated. As we shall see, cycles and the ways to avoid them have a key role in the proposed algorithm.

In the late 1970s and early 1980s, Gallager and Bertsekas developed algorithms for routing in communication networks. Gallager describes his major interest as "distributed algorithms for quasi-static routing." These algorithms are therefore prescriptive, rather than descriptive like transportation models. Their goal is to minimize the expected delay per message on the network; in that sense they seek the system-optimal solution rather than the user equilibrium. Despite the different framework, their mathematical formulation of the routing problem is equivalent to TAP.

Gallager (1977a) proposed a destination-based algorithm, using "routing variables" $\Phi_{i,j}(q)$ that represent the proportion of flow that continues from node i on link $[i, j]$ towards destination q. The algorithm considers only solutions that are loop-free, or acyclic. To prevent cycles, Gallager introduces the concept of *improper link*, which is a used link for which the average cost to its tail is higher than the average cost to its head. Cycles are avoided by prohibiting every link that is downstream from an improper link along a used route. Bertsekas (1979) added the use of second-order derivatives to the algorithm, and a projected piecewise linear search with an Armijo stopping condition. Additional improvements to Hessian approximations and additional computational results can be found in Bertsekas et al. (1979) and Bertsekas et al. (1984).

3. Algorithm Overview

The origin-based algorithm described here considers flows from each origin separately in a sequential fashion. We divide the traffic assignment problem into two separate questions: Which links should be used? How much flow should be assigned to each one of these links? A temporary answer to the first question is described by a subnetwork $A_p \subseteq A$, referred to as the *restricting subnetwork*. By definition the flow on any link outside A_p is zero. The proposed algorithm as described in Figure 1 consists of two main steps: update restricting subnetwork and update origin-based link flows.

A key point in this algorithm is that restricting subnetworks are always acyclic; i.e., they do not contain any directed cycles. As shown in §1, TAP always has an equilibrium solution that is acyclic by origin. The restriction to acyclic solutions has several advantages. It permits a simple route flow interpretation. It enables a definition of maximum cost. It also allows for a definition of topological order, i.e., a one-to-one function o: $N \to \{1, 2, 3, \ldots, |N|\}$ such that $[i, j] \in A_p \Rightarrow$

Initialization:
for p in N_o do
A_p = tree of minimum cost routes from p
$\mathbf{f}_p$ = all-or-nothing assignment using A_p
end for

Main loop:
for $n = 1$ to number of main iterations
for p in N_o do
update restricting subnetwork A_p
update origin-based link flows $\mathbf{f}_p$
end for
for $m = 1$ to number of inner iterations
for p in N_o do
update origin-based link flows $\mathbf{f}_p$
end for
end for
end for

Update restricting subnetwork for origin p:
remove unused links from A_p
compute maximum cost u_i from p to i for all $i \in N$
for $a = [i, j]$ in A
if $u_i < u_j$ add link a to A_p
find topological order for new A_p
find last common nodes in A_p
update data structures

Update origin-based link flows for origin p:
compute average costs and Hessian approximations
for step size $\lambda = 2^{-k}$; $k = 0, 1, 2, 3, \ldots$
compute flow shifts for λ
project and aggregate flow shifts
if social pressure is positive then stop
end for
apply flow shifts
update total link flows and link costs

Figure 1 The Origin-Based Algorithm

$o(i) < o(j)$, and particularly $o(p) = 1$. Most computations in the proposed algorithm are done in a *single pass* over the nodes, either in ascending or descending topological order. The time required by such computations is a linear function of the number of links in the network. This computational efficiency is the main reason for restricting to acyclic solutions.

The initial solution is based on trees of minimum cost routes under free-flow travel conditions as restricting subnetworks, leading to an "all-or-nothing" assignment. To update the restricting subnetwork, first unused links are removed, then the maximum cost u_i among all used routes from the origin p to node i is computed for all nodes, and finally every link $[i, j] \in A$ that satisfies $u_i < u_j$ is added to the restricting subnetwork. Comment: The term *unused* links and *maximum* cost were used here in an intuitive imprecise fashion. Precise definitions are given in Equation (45).

This condition may seem less intuitive than Dial's minimum cost condition for *efficient links* (Dial 1971); however, efficient links by current costs are likely to create cycles with currently used links. It will be shown that the new restricting subnetwork generated by the proposed maximum cost condition is always acyclic.

Once a new restricting subnetwork is found, several computationally intensive steps are needed, including reorganization of the data structure. However, restricting subnetworks tend to stabilize fairly quickly. Therefore, it is useful to update origin-based link flows while keeping the restricting subnetworks fixed. This is done by introducing "inner iterations" as described in Figure 1.

To update origin-based link flows within a given restricting subnetwork we use a quasi-Newtonian search direction. The search direction is based on shifting flow from high-cost alternatives to low-cost alternatives. One can show that when two independent routes are considered, objective function second-order approximation suggests that the amount of flow to be shifted equals the difference between route costs divided by the sum of route cost derivatives. In the origin-based algorithm we consider flow shifts between alternative approaches. For example, in Figure 2 we consider the alternative approaches to node q, $[4, q]$ and $[5, q]$. We assume that the flow shifted to or from approach $[5, q]$ will be divided among the different routes using approach $[5, q]$ by the same proportions as the current flows, that is, 70% percent on route $[p, 1, 3, 5, q]$ and 30% on route $[p, 1, 2, 5, q]$. The average cost incurred by travelers shifted to or from approach $[5, q]$ is therefore $0.3 \cdot 80 + 0.7 \cdot 110 =$

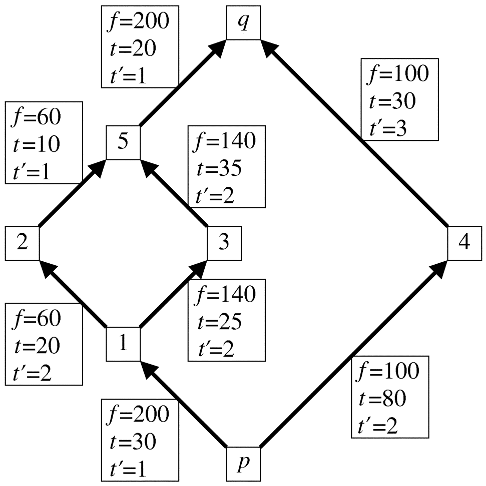


Figure 2 **Flow Shift Between Composite Alternatives**
Note. f-flow; t-cost; t'-cost derivative.

$101_{[\text{minutes}]}$. It will be shown that using approach proportions as solution variables, and using the conservation of flow to define basic and nonbasic variables, the first-order derivative of the objective function equals the difference in average approach cost.

From a dimensional analysis perspective, the gradient is in units of cost, say minutes. To obtain a search direction in units of flow, vehicles per hour, we must multiply the gradient by the inverse of the Hessian, or by its approximation. As in most nonlinear programming problems, computing the Hessian is quite time consuming, if possible at all. Typically, even in quasi-Newtonian algorithms a decent portion of computation time is devoted to approximating the inverse Hessian. This is perhaps one of the reasons why second-order algorithms are in many cases inferior to first-order algorithms, especially in the initial stages of the optimization, even though they are proven to provide better convergence rate in terms of accuracy improvement per iteration in the neighborhood of the optimum (equilibrium) point.

We use a fairly crude approximation of the diagonal elements of the Hessian. The main advantage of this approximation is that it is easy to compute. By inspection of the code, we estimate that in the proposed algorithm the time required to compute the approximation of the inverse Hessian is similar to the time required to compute first-order derivatives. A similar amount of time is required for each line search iteration, and on average, the algorithm performs about two line search iterations for each origin. We conclude that the consideration of second-order derivatives increases the computation time per iteration by 20–30%. We believe that this is a reasonable trade-off between computational overhead and search direction effectiveness.

The second-order search direction described above is viewed as desirable flow shifts, which may be infeasible. Projecting the search direction by truncating negative desirable flow shifts leads to a feasible search direction that usually has to be scaled by a step size to guarantee descent of the objective function. We refer to this technique as the convex line search procedure. In the scaled gradient projection technique (Bertsekas 1999, p. 229) one step size is applied prior to projection, and another step size is applied after the projection. We use a special case of scaled gradient projection, where the step size after projection is always set to one. We refer to this technique as the boundary search procedure, as it tends to choose solutions along the boundary, although it does consider interior points as well. In the context of origin-based assignment, convex line search is particularly problematic because it tends to leave small residual flows on suboptimal routes. These residual flows reduce the effectiveness of the restrictions update procedure, and may in fact prevent convergence. For example, the search direction in the case described in Figure 3 suggests a desirable flow shift of 60 vph from route $[p, 1, 2, 3, q']$ to route $[p, 3, q']$. In the convex line search this is first truncated to 10 vph, then multiplied by a step size which is usually less than one, leaving some flow on route $[p, 1, 2, 3, q']$. The residual flow on link $[2, 3]$ prevents adding the dashed link $[3, 2]$ which is part of the shortest route to q. In the boundary search this residual flow is eliminated with any step size above 0.16. We show that the boundary search is effective in eliminating residual flows in general, and prove that the incorporation of boundary search in the proposed algorithm guarantees global convergence.

The stopping condition is based on the concept of *social pressure* introduced by Kupiszewska and Van Vliet (1999). The basic idea is that every traveler

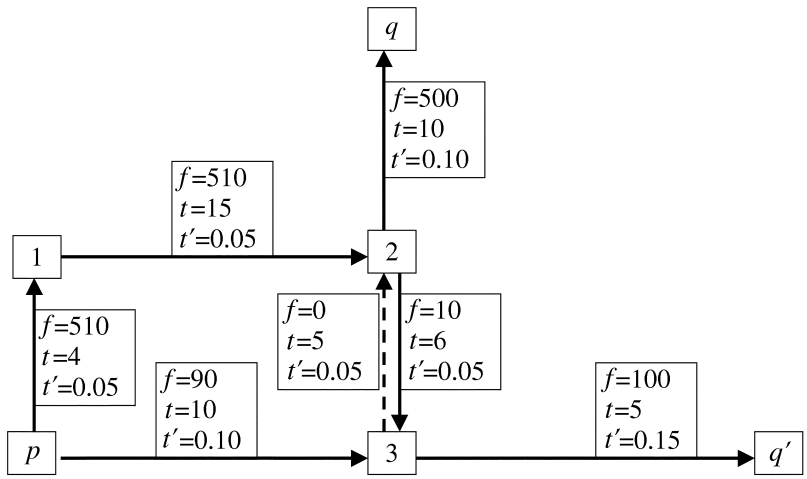


Figure 3 Residual Flow
Note. f-flow; t-cost; t'-cost derivative.

shifted from route r_1 to r_2 puts pressure (positive or negative) that is equal to $c_{r_1} - c_{r_2}$. The total social pressure is the sum of the pressure from all the travelers, which is equal to the negative of the directional derivative of the objective function along the vector of route flow changes. When searching along such a vector it is reasonable to continue as long as the social pressure is positive. This is equivalent to optimizing the objective function, which is the common stopping condition in the Frank-Wolfe algorithm. Any change to route flows that enjoys positive social pressure after the change yields descent of the objective function. In our boundary search solutions resulting from step sizes of $\lambda = 2^{-k};\ k = 0, 1, 2, \ldots$ are tested sequentially, and the first solution that satisfies the condition of positive social pressure is chosen.

4. Restricted Problem Reformulation

In the next two sections we restrict the discussion to the case where there is only one origin, denoted by p. This allows us to simplify the notation by omitting the origin index whenever possible. In particular, let f_a denote the origin-based link flow, and d_j denote the O-D flow from origin p to every node $j \in N$ where $d_j = 0\ \forall j \in N \setminus N_d(p)$.

Any multiple-origin problem with link cost functions $\hat{\mathbf{t}}$ can be temporarily converted to a single-origin problem for a specific origin p by fixing the flows from all other origins at some current values. The flows from origin p remain solution variables, while all other flows are viewed as background flows $\hat{f}_a = \sum_{p' \neq p} f_{ap'}$, and link costs are viewed as functions of flows from the current origin only, $t_a(f_a) = t_a(f_{ap}) = \hat{t}_a(f_{ap} + \hat{f}_a)$.

In this section we further assume that a restricting, spanning, acyclic subnetwork $A_p \subseteq A$ is given. By acyclic we mean that A_p does not contain any directed cycles, and by spanning we mean that A_p contains at least one route from p to every node $j \in N$. The set of route segments from node i to node j that are included in the subnetwork A_p is denoted by $R_{ij}[A_p] = \{r \in R_{ij}\colon a \subseteq r \Rightarrow a \in A_p\}$.

The Restricted Single-Origin Traffic Assignment Problem (RSOTAP) is to assign the demand onto routes in the restricting subnetwork, so that the cost of every used route does not exceed the cost of any alternative route in the restricting subnetwork. This problem can be formulated as a minimization problem

$$\text{[RSOTAP]} \quad \min T(\mathbf{f}(\mathbf{h})) = \sum_{a \in A_p} \int_0^{f_a(\mathbf{h})} t_a(x)\, dx$$

$$\text{s.t.} \sum_{r \in R_{pj}[A_p]} h_{rpj} = d_j \quad \forall j \in N,$$

$$\mathbf{h} \geq 0. \tag{2}$$

We define the *origin-based segment flow* g_s as the total amount of flow from origin p that utilizes route segment s; that is,

$$g_s = \sum_{q \in N} \sum_{r \in R_{pq}[A_p]: s \subseteq r} h_{rpq}. \tag{3}$$

Special cases of origin-based segment flow are the *origin-based link flow* $g_a = f_a$, and the *origin-based node flow* $g_{[j]} = g_j$. By definition, for every node $j \neq p$,

$$\begin{aligned} g_j &= \sum_{q \in N} \sum_{r \in R_{pq}[A_p]: j \in r} h_{rpq} \\ &= \sum_{a \in A: a_h = j} \sum_{q \in N} \sum_{r \in R_{pq}[A_p]: a \subseteq r} h_{rpq} = \sum_{a \in A: a_h = j} f_a, \end{aligned} \tag{4}$$

$$\begin{aligned} g_j &= \sum_{r \in R_{pj}[A_p]} h_{rpq} + \sum_{a \in A: a_t = j} \sum_{q \in N} \sum_{r \in R_{pq}[A_p]: a \subseteq r} h_{rpq} \\ &= d_j + \sum_{a \in A: a_t = j} f_a. \end{aligned} \tag{5}$$

Viewing link a as an *approach* to node $a_h = j$ we define the *origin-based approach proportion* α_a as the proportion of flow that arrives at node j through approach a. When the node flow g_{a_h} is not zero, the approach proportion is determined by $\alpha_a = f_a / g_{a_h}$; otherwise, approach proportions may be chosen arbitrarily. In any case the condition

$$f_a = \alpha_a \cdot g_{a_h} \tag{6}$$

must hold. Bar-Gera and Boyce (1999) provide intuitive behavioral assumptions suggesting that given a vector of acyclic origin-based link flows $\mathbf{f}$ and the associated approach proportions $\boldsymbol{\alpha}$, the most likely route flow interpretation is given by

$$h_{rpq}(\boldsymbol{\alpha}) = d_q \cdot \prod_{a \subseteq r} \alpha_a \quad \forall r \in R_{pq}. \tag{7}$$

We will show here that if $\boldsymbol{\alpha} = (\alpha_a)_{a \in A}$ is any vector that satisfies

$$\sum_{a \in A_p: a_h = j} \alpha_a = 1 \quad \forall j \in N; j \neq p \tag{8a}$$

$$\alpha_a = 0 \quad \forall a \notin A_p, \tag{8b}$$

$$\boldsymbol{\alpha} \geq 0, \tag{8c}$$

then $\mathbf{h}(\boldsymbol{\alpha})$ given by (7) is a feasible solution for RSOTAP, and Condition (6) holds, meaning that $\boldsymbol{\alpha}$ may indeed be viewed as the vector of approach proportions. From here on we consider the approach proportions as the main solution variables, and view route flows and origin-based link flows as functions of $\boldsymbol{\alpha}$. Since A_p is spanning, there is at least one link terminating at every node other than the origin, and therefore Conditions (8) can be satisfied.

To show that $\mathbf{h}(\boldsymbol{\alpha})$ is a feasible solution for RSOTAP, first notice that if route r contains a link $a \notin A_p$ then by (8b), $\alpha_a = 0$; hence $h_{rpq}(\boldsymbol{\alpha}) = 0$. Summing route flows over R_{pq} or over $R_{pq}[A_p]$ is therefore the same. To demonstrate conservation of flow, consider the following definition

$$\chi_{i \to j} = \sum_{s \in R_{ij}[A_p]} \left(\prod_{a \subseteq s} \alpha_a \right). \tag{9}$$

$\chi_{i \to j}$ can be interpreted as the proportion of flow that arrives at node j through node i. (Bar-Gera 1999, Equation (2.16)).

Lemma 4. *If $\boldsymbol{\alpha}$ satisfies feasibility requirements* (8), *then* $\chi_{p \to j} = 1 \ \forall j \in N$.

Proof. The proof is obtained by induction on j in increasing topological order. For $j = p$, the only route from p to itself is the empty route at p, and the product over the empty set is one by definition. Suppose the statement is true for all nodes of lower topological order, including all predecessors of j; that is, $\forall a \in A_p$: $a_h = j$, $\chi_{p \to a_t} = 1$.

$$\begin{aligned} \chi_{p \to j} &= \sum_{s \in R_{pj}[A_p]} \prod_{a \subseteq s} \alpha_a = \sum_{a \in A_p: a_h = j} \alpha_a \cdot \left(\sum_{s' \in R_{pa_t}[A_p]} \prod_{a' \subseteq s'} \alpha_{a'} \right) \\ &= \sum_{a \in A_p: a_h = j} \alpha_a \cdot \chi_{p \to a_t} = \sum_{a \in A_p: a_h = j} \alpha_a = 1, \end{aligned} \tag{10}$$

where the last equality is due to feasibility requirement (8a). □

The route flow vector $\mathbf{h}(\boldsymbol{\alpha})$ satisfies conservation of flow, since by Lemma 4,

$$\sum_{r \in R_{pq}[A_p]} h_{rpq}(\boldsymbol{\alpha}) = \sum_{r \in R_{pq}[A_p]} d_q \cdot \prod_{a \subseteq r} \alpha_a = d_q \cdot \chi_{p \to q} = d_q. \tag{11}$$

Using Lemma 4, we can also evaluate the origin-based node, segment, and link flows:

$$\begin{aligned} g_j(\boldsymbol{\alpha}) &= \sum_{q \in N} \sum_{r \in R_{pq}[A_p]: j \in r} h_{rpq}(\boldsymbol{\alpha}) \\ &= \sum_{q \in N} \chi_{p \to j} \cdot \chi_{j \to q} \cdot d_q = \sum_{q \in N} \chi_{j \to q} \cdot d_q. \end{aligned} \tag{12}$$

$$g_s(\boldsymbol{\alpha}) = \sum_{q\in N} \sum_{r\in R_{pq}[A_p]:s\subseteq r} h_{rpq}(\boldsymbol{\alpha}) = \sum_{q\in N} \chi_{p\to s_t} \cdot \prod_{a\subseteq s} \alpha_a \cdot \chi_{s_h\to q} \cdot d_q = \prod_{a\subseteq s} \alpha_a \cdot g_{s_h}(\boldsymbol{\alpha}), \tag{13}$$

$$f_a(\boldsymbol{\alpha}) = g_a(\boldsymbol{\alpha}) = \alpha_a \cdot g_{a_h}(\boldsymbol{\alpha}), \tag{14}$$

demonstrating that Condition (6) is satisfied as proposed.

For every node $j \neq p$ we choose one approach $b_j \in A_p$: $(b_j)_h = j$ as the basic approach, and denote all other approaches (if there are any) as the nonbasic approaches $\mathrm{NB}_j = \{a \in A_p\colon a_h = j;\ a \neq b_j\}$; $\mathrm{NB} = \bigcup_{j\in N} \mathrm{NB}_j$. Using Condition (8a), $\boldsymbol{\alpha}$ can be viewed as a function of $\boldsymbol{\alpha}^{\mathrm{NB}}$:

$$\alpha_{b_j}(\boldsymbol{\alpha}^{\mathrm{NB}}) = 1 - \sum_{a\in \mathrm{NB}_j} \alpha_a^{\mathrm{NB}} \quad \forall j \in N\setminus\{p\}, \tag{15a}$$

$$\alpha_a(\boldsymbol{\alpha}^{\mathrm{NB}}) = \alpha_a^{\mathrm{NB}} \quad \forall a \in \mathrm{NB}. \tag{15b}$$

To derive optimality conditions, we shall examine the partial derivatives of $\mathbf{h}(\boldsymbol{\alpha})$ and $\mathbf{f}(\boldsymbol{\alpha})$ with respect to α_a for every approach link a; and the partial derivatives of $\mathbf{f}(\boldsymbol{\alpha}(\boldsymbol{\alpha}^{NB}))$ and $T(\boldsymbol{\alpha}(\boldsymbol{\alpha}^{NB}))$ with respect to $\boldsymbol{\alpha}_a^{NB}$ for every nonbasic approach a. For brevity, denote the termination node by $j = a_h$, its basic approach by $b = b_j = b_{(a_h)}$, and replace $\mathbf{g}(\boldsymbol{\alpha})$ by $\mathbf{g}$.

$$\frac{\partial h_{rpq}(\boldsymbol{\alpha})}{\partial \alpha_a} = \delta_{ra} \cdot d_q \cdot \prod_{a'\subseteq r:\, a'\neq a} \alpha_{a'}, \tag{16}$$

$$\frac{\partial f_{a'}(\boldsymbol{\alpha})}{\partial \alpha_a} = \sum_{q\in N} \sum_{r\in R_{pq}[A_p]:\, a'\subseteq r} \frac{\partial h_{rpq}(\boldsymbol{\alpha})}{\partial \alpha_a} = \begin{cases} g_j & a' = a, \\ 0 & a'_h = a_h;\ a' \neq a, \\ \alpha_{a'} \cdot \chi_{a'_h\to a_t} \cdot g_{a_h} & o(a'_h) < o(a_h), \\ \chi_{a_h\to a'_t} \cdot \alpha_{a'} \cdot g_{a'_h} & o(a'_h) > o(a_h), \end{cases} \tag{17}$$

$$\frac{\partial f_{a'}(\boldsymbol{\alpha}(\boldsymbol{\alpha}^{\mathrm{NB}}))}{\partial \alpha_a^{\mathrm{NB}}} = \begin{cases} g_j & a' = a, \\ -g_j & a' = b, \\ 0 & a'_h = a_h;\ a' \neq a;\ a' \neq b, \\ \alpha_{a'} \cdot g_j \cdot (\chi_{a'_h\to a_t} - \chi_{a'_h\to b_t}) & o(a'_h) < o(a_h), \\ 0 & o(a'_h) > o(a_h), \end{cases} \tag{18}$$

$$\frac{\partial T(\boldsymbol{\alpha}(\boldsymbol{\alpha}^{\mathrm{NB}}))}{\partial \alpha_a^{\mathrm{NB}}} = \sum_{a'\in A_p} \frac{\partial T}{\partial f_{a'}} \cdot \frac{\partial f_{a'}(\boldsymbol{\alpha}(\boldsymbol{\alpha}^{\mathrm{NB}}))}{\partial \alpha_a^{\mathrm{NB}}} = t_a \cdot g_j - t_b \cdot g_j + \sum_{a'\in A_p:\, o(a'_h)<o(j)} t_{a'} \cdot \alpha_{a'} \cdot g_j \cdot \left(\chi_{a'_h\to a_t} - \chi_{a'_h\to b_t}\right). \tag{19}$$

For more detailed derivation see Bar-Gera (1999, Equations (2.23)–(2.34)). Equation (19) can be simplified using average cost to a node and average approach cost, which are now defined. Consider the average cost weighted by flow over all route segments from the origin p to node i, and use the origin-based segment flow given by (13) to obtain

$$\frac{\sum_{r\in R_{pi}[A_p]} g_r \cdot c_r}{\sum_{r\in R_{pi}[A_p]} g_r} = \frac{\sum_{r\in R_{pi}[A_p]} \prod_{a\subseteq r} \alpha_a \cdot g_i \cdot c_r}{g_i} = \sum_{r\in R_{pi}[A_p]} c_r \cdot \prod_{a\subseteq r} \alpha_a. \tag{20}$$

Notice that the left-hand term is well defined only if the denominator, i.e., the node flow, is not zero, while the right-hand term is well defined even if the node flow is zero. Following this derivation, we define σ_i as the average cost to node i by

$$\sigma_i = \sum_{r\in R_{pi}[A_p]} c_r \cdot \prod_{a\subseteq r} \alpha_a. \tag{21}$$

If $a_t = i$ we can add link a to every route segment from p to i and obtain the set of route segments from p to $a_h = j$ that arrives at node j through approach a. Similar to Definition (21), the average cost for the set of route segments that arrive at node j through approach a, referred to as the *average approach cost* μ_a, is defined as follows:

$$\mu_a = \sum_{r\in R_{pj}[A_p]:\, a\subseteq r} c_r \cdot \prod_{a'\subseteq r:\, a'\neq a} \alpha_{a'} = t_a + \sigma_{a_t}. \tag{22}$$

Notice that

$$\sigma_j = \sum_{a\in A_p:\, a_h=j} \alpha_a \cdot \sum_{r\in R_{pj}[A_p]:\, a\subseteq r} c_r \cdot \prod_{a'\subseteq r:\, a'\neq a} \alpha_{a'} = \sum_{a\in A_p:\, a_h=j} \alpha_a \cdot \mu_a. \tag{23}$$

As shown in Bar-Gera (1999, Equation (2.41)),

$$\begin{aligned}\mu_a &= t_a + \sum_{r\in R_{pa_t}[A_p]} \prod_{a''\subseteq r} \alpha_{a''} \cdot \sum_{a'\subseteq r} t_{a'} \\ &= t_a + \sum_{a'\in A_p} t_{a'} \cdot \chi_{p\to a'_t} \cdot \alpha_{a'} \cdot \chi_{a'_h\to a_t} \\ &= t_a + \sum_{a'\in A_p;\, o(a'_h)<o(a_h)} t_{a'} \cdot \alpha_{a'} \cdot \chi_{a'_h\to a_t}. \end{aligned} \tag{24}$$

Therefore, Equation (19) can be rewritten as

$$\frac{\partial T\left(\boldsymbol{\alpha}\left(\boldsymbol{\alpha}^{\mathrm{NB}}\right)\right)}{\partial \alpha_a^{\mathrm{NB}}} = g_j \cdot (\mu_a - \mu_b). \tag{25}$$

Note that the social pressure associated with shifting δ from α_a to α_b or $(\delta\cdot g_j)$ vph from approach a to approach b is $\delta\cdot g_j\cdot(\mu_a-\mu_b) = \delta\cdot\left(\partial T\left(\boldsymbol{\alpha}\left(\boldsymbol{\alpha}^{NB}\right)\right)/\partial\alpha_a^{NB}\right)$, as might be expected. If we assume that the basic approach is an approach of minimum average cost, then by (25) the necessary conditions for optimality are

$$\mu_a \ge \mu_{b_j} \quad \forall j\in N\setminus\{p\};\, \forall a\in \mathrm{NB}_j, \tag{26a}$$

$$\alpha_a\cdot g_j\cdot(\mu_a-\mu_{b_j}) = 0 \quad \forall j\in N\setminus\{p\};\, \forall a\in \mathrm{NB}_j. \tag{26b}$$

These necessary conditions are not sufficient for optimality because in general T is not convex as a function of $\boldsymbol{\alpha}$. The following lemma shows that by omitting the node flow from (26b) sufficient conditions for optimality are obtained. In the proof of this lemma, and thereon, we refer to link a with $\alpha_a > 0$ as a *contributing link*, in contrast to a *used link* which must have strictly positive flow $f_a > 0$. Accordingly, we consider as contributing route segments those that contain contributing links only. Notice that by (6) every used link is necessarily a contributing one, and every used route is also a contributing route.

Lemma 5. *If*

$$\mu_a \ge \mu_{b_j} \quad \forall j\in N\setminus\{p\};\; \forall a\in NB_j, \tag{27a}$$

$$\alpha_a\cdot(\mu_a-\mu_{b_j}) = 0 \quad \forall j\in N\setminus\{p\};\; \forall a\in NB_j, \tag{27b}$$

then $\boldsymbol{\alpha}$ *is an equilibrium (optimal) solution for RSOTAP.*

Proof. Suppose there is a contributing route segment $r = (r' + [ik]) \in R_{pk}$ that has an alternative route segment $s = (s' + [jk]) \in R_{pk}$ of lower cost. Without loss of generality, assume that k is the node of lowest topological order for which such routes exist. Clearly $i \ne j$; otherwise, r', s' would be such route segments for i, and $o(i) < o(k)$. Since $o(i) < o(k)$, the cost of all contributing route segments from p to i are equal, and in particular $c_r = \mu_{[ik]}$. Because $o(j) < o(k)$, the cost of every contributing route segment from the origin p to j and the average of such costs is not greater than the cost of any alternative route segment; hence $c_s \ge \mu_{[jk]}$. Therefore $\mu_{[ik]} = c_r > c_s \ge \mu_{[jk]} \ge \mu_{b_k}$ and $\alpha_{[ik]} > 0$, in contradiction to Conditions (27). □

Notice that Conditions (26) are different from Conditions (27) only at nodes with zero node flow. At such nodes, modifying approach proportions does not affect link flows and link costs. As a result, satisfying (27) is not more demanding than satisfying (26). From here on, therefore, we only consider Sufficiency Conditions (27).

5. The Algorithm

This section provides a formal mathematical definition of the algorithm. The basic component of the algorithm is a Newton-type shift of flow from nonbasic to basic approaches. An aggregation of such flow shifts yields a search direction which, together with the boundary search procedure, provides an algorithmic map to solve RSOTAP. Unrestricted equilibrium for a single origin is achieved by iterative updates of the restricting subnetwork. Finally, the multiple-origin case is discussed.

The Newton-type shift of flow is based on the ratio between the first-order derivative (25) and an approximation of the objective function second-order derivative, which is the target of the following derivation. The diagonal second-order derivatives of the flow on any link a' with respect to any other approach proportion α_a and nonbasic approach proportion α_a^{NB} is always zero. $\partial^2 f_{a'}/\partial{\alpha_a}^2 = \partial^2 f_{a'}/\partial{\alpha_a^{\mathrm{NB}}}^2 = 0$. The second-order derivative of the objective function with respect to link flow is $\partial^2 T/\partial {f_a}^2 = \partial t_a/\partial f_a = t'_a$, and with respect to nonbasic approach proportions

$$\frac{\partial^2 T}{\partial {\alpha_a^{\mathrm{NB}}}^2} = \sum_{a'\in A_p}\left[\frac{\partial^2 T}{\partial {f_{a'}}^2}\cdot\left(\frac{\partial f_{a'}}{\partial \alpha_a^{\mathrm{NB}}}\right)^2 + \frac{\partial T}{\partial f_{a'}}\cdot\left(\frac{\partial^2 f_{a'}}{\partial {\alpha_a^{\mathrm{NB}}}^2}\right)\right]$$

$$= t'_a \cdot g_j^2 + t'_b \cdot g_j^2 + \sum_{\substack{a' \in A_p \\ o(a'_h) < o(j)}} t'_{a'} \cdot {\alpha_{a'}}^2 \cdot g_j^2 \left(\chi_{a'_h \to a_t} - \chi_{a'_h \to b_t}\right)^2. \quad (28)$$

In some cases the topological structure of A_p dictates that $\chi_{a'_h \to a_t} = \chi_{a'_h \to b_t}$, and hence the relevant term in the last summation vanishes. Suppose that node i is common to all routes that arrive at node j; that is, $i \in s \ \forall s \in R_{pj}[A_p]$, then $\chi_{p\to j} = \chi_{p\to i} \cdot \chi_{i \to j}$, hence $\chi_{i\to j} = 1$. Furthermore, for every node i' such that $o(i') \le o(i)$, we find that $\chi_{i' \to j} = \chi_{i'\to i} \cdot \chi_{i \to j} = \chi_{i' \to i}$. Note that the origin p and the node itself j are always common nodes for j. We define the *last common node* of node j in A_p, lcn_j, as the common node of highest topological order, excluding j itself. Some examples for common nodes and last common nodes in an acyclic restricting subnetwork are given in Figure 4. One can verify that lcn_j is also a common node for the tail of any approach to j. In particular, $l = \mathrm{lcn}_{(a_h)}$ is also a common node to a_t and to b_t; therefore, if $o(a'_h) \le o(\mathrm{lcn}_{(a_h)})$, then $\chi_{a'_h \to a_t} = \chi_{a'_h \to b_t} = \chi_{a'_h \to l}$. As a result

$$\frac{\partial^2 T}{\partial {\alpha_a^{\mathrm{NB}}}^2} = t'_a \cdot g_j^2 + t'_b \cdot g_j^2 + \sum_{\substack{a' \in A_p \\ o(\mathrm{lcn}_j) < o(a'_h) < o(j)}} t'_{a'} \cdot {\alpha_{a'}}^2 \cdot g_j^2 \cdot \left(\chi_{a'_h \to a_t} - \chi_{a'_h \to b_t}\right)^2. \quad (29)$$

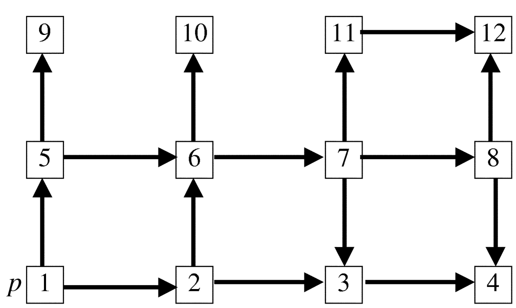


Figure 4 Common Nodes and Last Common Nodes in an Acyclic Network

Note.

node (j)	common nodes of j	lcn_j
9	1,5,9	5
12	1,6,7,12	7
4	1,4	1

Unfortunately, we know of no efficient way to compute this expression. As shown in Bar-Gera (1999, Equations (2.52)–(2.54)), the following recursive definitions

$$\rho_p = 0, \quad (30a)$$

$$\rho_j = \sum_{a \in A_p:\, a_h = j} \alpha_a^2 \cdot \nu_a \quad (\forall j \ne p), \quad (30b)$$

$$\nu_a = t'_a + \rho_{a_t}, \quad (30c)$$

provide a reasonable approximation of the second-order derivative by

$$\frac{\partial^2 T}{\partial {\alpha_a^{\mathrm{NB}}}^2} \approx g_j^2 \cdot (\nu_a + \nu_b - 2 \cdot \rho_{\mathrm{lcn}_j}). \quad (31)$$

These recursive definitions have an important advantage as they can be computed in a single ascending pass over the nodes.

We are now ready to state the formal definition of the algorithm. For that purpose we redefine σ, μ, ρ, ν as functions of $\mathbf{t}$, $\mathbf{t}'$, and $\boldsymbol{\alpha}$ as separate variables. These definitions are motivated by the previous discussion; however, they allow us to handle cases where $\mathbf{t} \ne \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha}))$ and $\mathbf{t}' \ne \mathbf{t}'(\mathbf{f}(\boldsymbol{\alpha}))$.

$$\sigma_p(\boldsymbol{\alpha}, \mathbf{t}) = 0, \quad (32a)$$

$$\sigma_j(\boldsymbol{\alpha}, \mathbf{t}) = \sum_{a \in A_p:\, a_h = j} \alpha_a \cdot \mu_a(\boldsymbol{\alpha}, \mathbf{t}), \quad (32b)$$

$$\mu_a(\boldsymbol{\alpha}, \mathbf{t}) = t_a + \sigma_{a_t}(\boldsymbol{\alpha}, \mathbf{t}), \quad (32c)$$

$$\rho_p(\boldsymbol{\alpha}, \mathbf{t}') = 0, \quad (33a)$$

$$\rho_j(\boldsymbol{\alpha}, \mathbf{t}') = \sum_{a \in A_p:\, a_h = j} \alpha_a^2 \cdot \nu_a(\boldsymbol{\alpha}, \mathbf{t}'), \quad (33b)$$

$$\nu_a(\boldsymbol{\alpha}, \mathbf{t}') = t'_a + \rho_{a_t}(\boldsymbol{\alpha}, \mathbf{t}'). \quad (33c)$$

Consider a nonbasic approach a to node $j = a_h$, and an alternative basic approach $b = b_j$. Recall that by assumption, $\mu_b \le \mu_a$. By (25) and (31), a Newton-type shift from α_a to α_b is given by

$$\frac{\partial T / \partial \alpha_a^{\mathrm{NB}}}{\partial^2 T / \partial {\alpha_a^{\mathrm{NB}}}^2} \approx \frac{g_j \cdot (\mu_a - \mu_b)}{g_j^2 \cdot (\nu_a + \nu_b - 2 \cdot \rho_{\mathrm{lcn}_j})} = \frac{1}{g_j} \cdot \frac{\mu_a - \mu_b}{\nu_a + \nu_b - 2 \cdot \rho_{\mathrm{lcn}_j}}. \quad (34)$$

We choose a small positive constant $\epsilon_\nu > 0$, and define

$$z_{a\to b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}') = \frac{\mu_a(\boldsymbol{\alpha}, \mathbf{t}) - \mu_b(\boldsymbol{\alpha}, \mathbf{t})}{\max(\epsilon_\nu, \nu_a(\boldsymbol{\alpha}, \mathbf{t}') + \nu_b(\boldsymbol{\alpha}, \mathbf{t}') - 2\cdot\rho_{\mathrm{lcn}_j}(\boldsymbol{\alpha}, \mathbf{t}'))}. \tag{35}$$

$z_{a\to b}$ is a continuous function that can be interpreted as the desirable amount of flow (in vph) that should be shifted from a to b in order to equalize costs, ignoring feasibility constraints. The boundary search requires us to apply the step size $0 < \lambda \leq 1$ prior to any consideration of feasibility constraints. The shift of proportions between alternative approaches for step size λ is therefore defined by the following point to set map $\Theta_\lambda^{a\to b}$: $[0,1]^{|A|} \times \mathbb{R}_+^{|A|} \times \mathbb{R}_+^{|A|} \to 2^{[0,1]}$, where 2^X stands for the set of subsets of X.

$$\Theta_\lambda^{a\to b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}') = \begin{cases} \left\{\min\left(\alpha_a, \lambda\cdot\dfrac{z_{a\to b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}')}{g_j(\boldsymbol{\alpha})}\right)\right\}, & g_j(\boldsymbol{\alpha}) > 0, \\ \{\alpha_a\}, & g_j(\boldsymbol{\alpha}) = 0,\ \mu_a(\boldsymbol{\alpha}, \mathbf{t}) > \mu_b(\boldsymbol{\alpha}, \mathbf{t}), \\ [0, \alpha_a], & g_j(\boldsymbol{\alpha}) = 0,\ \mu_a(\boldsymbol{\alpha}, \mathbf{t}) = \mu_b(\boldsymbol{\alpha}, \mathbf{t}). \end{cases} \tag{36}$$

Because $\Theta_\lambda^{a\to b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}') \subseteq [0, \alpha_a]$ holds, feasibility is guaranteed.

Lemma 6. *$\Theta_\lambda^{a\to b}$ is a closed map for all $\lambda > 0$; that is, if $\boldsymbol{\alpha}^k \to \boldsymbol{\alpha}$, $\mathbf{t}^k \to \mathbf{t}$, $\mathbf{t}'^k \to \mathbf{t}'$, $\theta^k \in \Theta_\lambda^{a\to b}(\boldsymbol{\alpha}^k, \mathbf{t}^k, \mathbf{t}'^k)$, and $\theta^k \to \theta$, then $\theta \in \Theta_\lambda^{a\to b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}')$.*

Proof. If $g_j(\boldsymbol{\alpha}) > 0$, then the function $\Theta_\lambda^{a\to b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}')$ is a continuous point-to-point function in the neighborhood of $\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}'$. If $g_j(\boldsymbol{\alpha}) = 0$ and $\mu_a(\boldsymbol{\alpha}, \mathbf{t}) = \mu_b(\boldsymbol{\alpha}, \mathbf{t})$, then $\theta^k \in \Theta_\lambda^{a\to b}(\boldsymbol{\alpha}^k, \mathbf{t}^k, \mathbf{t}'^k) \subseteq [0, \alpha_a^k]$ and therefore $\theta \in [0, \alpha_a] = \Theta_\lambda^{a\to b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}')$. Suppose that $g_j(\boldsymbol{\alpha}) = 0$ and $\mu_a(\boldsymbol{\alpha}, \mathbf{t}) - \mu_b(\boldsymbol{\alpha}, \mathbf{t}) > \epsilon > 0$. Since ρ and ν are bounded, and since $\lambda > 0$, there is k_0 such that for every $k > k_0$ the desirable shift is infeasible, $\lambda \cdot z_{a\to b}(\boldsymbol{\alpha}^k, \mathbf{t}^k, \mathbf{t}') > g_j(\boldsymbol{\alpha}^k)$, hence $\Theta_\lambda^{a\to b}(\boldsymbol{\alpha}^k, \mathbf{t}^k, \mathbf{t}'^k) = \{\alpha_a^k\}$, hence $\theta^k = \alpha_a^k$, and $\theta = \alpha_a \in \{\alpha_a\} = \Theta_\lambda^{a\to b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}')$. □

The next task is to determine the effect of the change in approach proportions on link flows. For any vector of changes in approach proportion $\Delta\boldsymbol{\alpha}$ define

$$\Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha}) = \mathbf{f}(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}) - \mathbf{f}(\boldsymbol{\alpha}). \tag{37}$$

Suppose $-\Delta\alpha_a \in \Theta_\lambda^{a\to b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}')$; $\Delta\alpha_b = -\Delta\alpha_a$; $\Delta\alpha_{a'} = 0$ $\forall a' \neq a$, $a' \neq b$. Since $f_{a'}$ is a linear function of α_a^{NB},

$$\Delta f_{a'}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha}) = f_{a'}(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}) - f_{a'}(\boldsymbol{\alpha}) = \Delta\alpha_a \cdot \frac{\partial f_{a'}}{\partial\alpha_a^{\mathrm{NB}}}. \tag{38}$$

By (18) and (24) the directional derivative of T along $\Delta\mathbf{f}$ is

$$\begin{aligned}\Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha}) \cdot \nabla_{\mathbf{f}} T &= \Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha}) \cdot \mathbf{t} \\ &= \Delta\alpha_a \cdot g_j(\boldsymbol{\alpha}) \cdot (\mu_a(\boldsymbol{\alpha}, \mathbf{t}) - \mu_b(\boldsymbol{\alpha}, \mathbf{t})) \leq 0. \end{aligned} \tag{39}$$

Equality holds only if: $\Delta\alpha_a = 0$; $g_j(\boldsymbol{\alpha}) = 0$; or $\mu_a(\boldsymbol{\alpha}, \mathbf{t}) = \mu_b(\boldsymbol{\alpha}, \mathbf{t})$, and in all three cases $\Delta\mathbf{f} = 0$. $\Delta\mathbf{f}$ is therefore either zero or a feasible direction of descent of T.

In a fully sequential algorithm, shifts are computed and applied for each nonbasic approach sequentially. Link costs are updated after each shift. Such an algorithm does not take full advantage of the acyclic structure of the solution. By (32) and (33), average approach cost μ_a and estimated derivatives ν_a and ρ_j can be computed efficiently for the entire network in a single ascending pass over the nodes. Once new approach proportions are determined, then new origin-based link flows can be computed by (5) and (6) in a single descending pass over the nodes. If all shifts are to be determined simultaneously, the complete search direction for the origin could be computed efficiently in one ascending pass and one descending pass over the nodes. It is not demonstrated here, but the search direction can be computed as efficiently if shifts are determined sequentially in either ascending or descending topological order, as long as link costs remain fixed and are not updated.

For a given basic approach b, the aggregated shift is defined by the following map:

$$\Theta_\lambda^{j:b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}') = \left\{\Delta\boldsymbol{\alpha}\colon -\Delta\alpha_a \in \Theta_\lambda^{a\to b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}'),\ \forall a \in \mathrm{NB}_j;\ \Delta\alpha_b = -\sum_{a\in\mathrm{NB}_j} \Delta\alpha_a;\ \Delta\alpha_{a'} = 0,\ a_h' \neq j\right\}. \tag{40}$$

This aggregated shift satisfies feasibility requirements. The change in link flows for the aggregated shift is the sum of the changes of the pairwise shifts. Therefore, if $\Delta\boldsymbol{\alpha} \in \Theta_\lambda^{j:b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}')$, then $\Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha})$ is either zero or a direction of descent of T. The algorithmic map for node shifts requires that the basic approach is of minimum cost, but allows an arbitrary choice if there is more than one minimum cost approach.

$$\Theta_\lambda^j(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}' : A_p) = \bigcup_{\substack{b \in A_p : b_h = j \\ \mu_b \le \mu_a \ \forall a \in A_p ; a_h = j}} \Theta_\lambda^{j:b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}'). \quad (41)$$

LEMMA 7. *$\Theta_\lambda^j(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}' : A_p)$ is a closed map.*

PROOF. There is a subsequence K that uses the same basic approach b^0, which is therefore a legitimate choice in the limit. □

Node shifts are aggregated to origin-based shift in descending order

$$\begin{aligned}&\Theta_\lambda^\downarrow(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}' : A_p)\\ &\quad = \Bigg\{ \Delta\boldsymbol{\alpha} = \sum_{j \in N : j \ne p} \Delta\boldsymbol{\alpha}^j : \Delta\boldsymbol{\alpha}^j \in \Theta_\lambda^j\left(\boldsymbol{\alpha}^j, \mathbf{t}, \mathbf{t}' : A_p\right),\\ &\qquad\qquad \boldsymbol{\alpha}^j = \boldsymbol{\alpha} + \sum_{j' \in N : o(j') > o(j)} \Delta\boldsymbol{\alpha}^{j'} \Bigg\}. \quad (42)\end{aligned}$$

If $\Delta\boldsymbol{\alpha} \in \Theta_\lambda^\downarrow(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}' : A_p)$ holds, then $\Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha}) = \sum_{j \in N} \Delta\mathbf{f}(\boldsymbol{\alpha}^j, \Delta\boldsymbol{\alpha}^j)$; therefore $\Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha}) \cdot \mathbf{t} \le 0$, and equality holds if and only if $\Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha}) = 0$. $T(\mathbf{f})$ is a convex function of $\mathbf{f}$; hence $\Delta\mathbf{f} \cdot \mathbf{t}(\mathbf{f} + \beta \cdot \Delta\mathbf{f})$ is a monotonically increasing function of β. Therefore, objective function descent for nonzero shift can be guaranteed by nonnegative social pressure, that is, by nonpositive directional derivative $\Delta\mathbf{f} \cdot \mathbf{t}(\mathbf{f} + \Delta\mathbf{f}) \le \mathbf{0}$. The following lemma shows that this condition can always be satisfied and provide a positive lower bound on the required step size.

LEMMA 8. *There exists $\lambda_0 > 0$ such that for all $0 < \lambda \le \lambda_0$ and for all $\boldsymbol{\alpha}$ and A_p, every choice of $\Delta\boldsymbol{\alpha} \in \Theta_\lambda^\downarrow(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha})), \mathbf{t}'(\mathbf{f}(\boldsymbol{\alpha})) : A_p)$ satisfies*

$$\Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha}) \cdot \mathbf{t}(\mathbf{f} + \Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha})) \le \mathbf{0}. \quad (43)$$

PROOF. See Appendix. □

By Lemma 8, examining $\lambda = 2^{-k}$; $k = 0, 1, 2, \ldots$ until the social pressure condition (43) is satisfied, is a finite process. To prove convergence, we can allow more flexibility in the choice of the step size, and define the algorithmic map for RSOTAP as follows:

$$\begin{aligned}&\Theta^\downarrow(\boldsymbol{\alpha} : A_p)\\ &\quad = \big\{ \Delta\boldsymbol{\alpha} : \exists \lambda \in [\lambda_0, 1] : \Delta\boldsymbol{\alpha} \in \Theta_\lambda^\downarrow(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha})), \mathbf{t}'(\mathbf{f}(\boldsymbol{\alpha})) : A_p),\\ &\qquad\qquad \Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha}) \cdot \mathbf{t}(\mathbf{f} + \Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha})) \le \mathbf{0} \big\}. \quad (44)\end{aligned}$$

By Lemma 8, the map $\Theta^\downarrow$ is nonempty. Descent is guaranteed in the sense that for all $\Delta\boldsymbol{\alpha} \in \Theta_\lambda^\downarrow(\boldsymbol{\alpha} : A_p)$, $T(\mathbf{f}(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha})) \le T(\mathbf{f}(\boldsymbol{\alpha}))$. The map $\Theta^\downarrow$ is closed, since if $\Delta\boldsymbol{\alpha}^k \in \Theta_{\lambda^k}^\downarrow(\boldsymbol{\alpha}^k : A_p)$, $\boldsymbol{\alpha}^k \to \boldsymbol{\alpha}$, $\Delta\boldsymbol{\alpha}^k \to \Delta\boldsymbol{\alpha}$, then for some subsequence $\lambda^k \to \lambda \in [\lambda_0, 1]$ and $\Delta\boldsymbol{\alpha} \in \Theta_\lambda^\downarrow(\boldsymbol{\alpha} : A_p)$. A crucial point in the above argument is that at the limit, λ is strictly positive and not zero. This is guaranteed by Lemma 8.

THEOREM 1. *The algorithm defined by $\Theta^\downarrow(\boldsymbol{\alpha} : A_p)$ converges to an optimal solution for RSOTAP.*

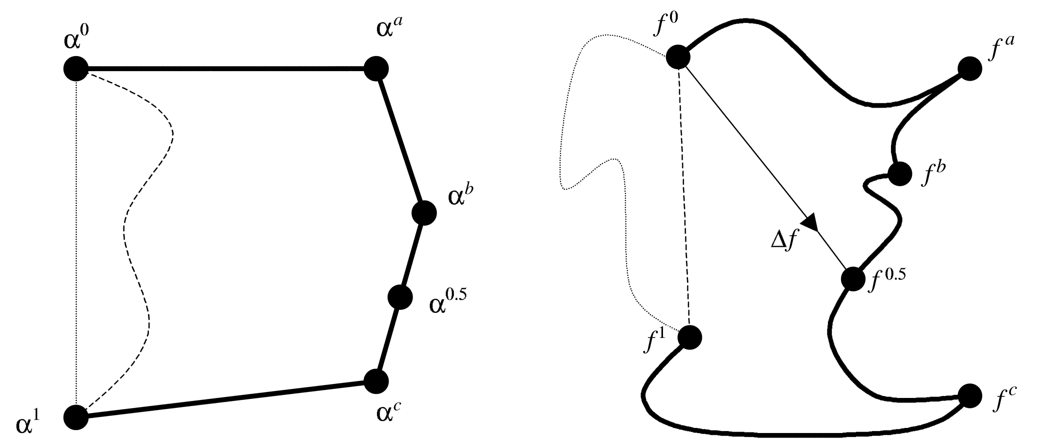
PROOF. See Appendix. □

The search for step size in the algorithm given by (44) is referred to as a *boundary search*, as opposed to the common convex line search. The difference between the two is illustrated in Figure 5, in both approach proportions and link flows spaces. Current solution is represented by $\boldsymbol{\alpha}^0$,$\mathbf{f}^0$. The new solution for step size 1.0 is represented by $\boldsymbol{\alpha}^1$,$\mathbf{f}^1$. Convex combinations of $\boldsymbol{\alpha}^0$ and $\boldsymbol{\alpha}^1$ are depicted by solid thin lines. Convex combinations of $\mathbf{f}^0$ and $\mathbf{f}^1$ are depicted by dashed thin lines.

The boundary search is described in the approach proportions space by the thick piecewise line connecting $\boldsymbol{\alpha}^0$, $\boldsymbol{\alpha}^a$, $\boldsymbol{\alpha}^b$, $\boldsymbol{\alpha}^c$, and $\boldsymbol{\alpha}^1$. The corresponding thick curve in the link flows space is a general curve that connects $\mathbf{f}^0$, $\mathbf{f}^a$, $\mathbf{f}^b$, $\mathbf{f}^c$, and $\mathbf{f}^1$. For any step-size value, we choose a point along the piecewise line in the approach proportions space and then compute the corresponding link flows. For example, $\boldsymbol{\alpha}^{0.5}$ and $\mathbf{f}^{0.5}$ represent the approach proportions and link flows for step size 0.5, respectively. The line segment connecting $\boldsymbol{\alpha}^0$ and $\boldsymbol{\alpha}^{0.5}$ is of no particular interest to us, because T is not convex as a function of $\boldsymbol{\alpha}$. $\Delta\mathbf{f}$ is the vector connecting $\mathbf{f}^0$ and $\mathbf{f}^{0.5}$. The directional derivative along $\Delta\mathbf{f}$ at $\mathbf{f}^0$ is negative, and by convexity it is monotonically increasing along the line segment connecting $\mathbf{f}^0$ and $\mathbf{f}^{0.5}$. If the directional derivative along $\Delta\mathbf{f}$ at $\mathbf{f}^{0.5}$ is less than or equal to zero, descent of the objective function is guaranteed; otherwise, a smaller step size is examined.

The last component of the algorithm is the update of restricting subnetworks, which is based on *maximum contributing cost*

$$u_j(\boldsymbol{\alpha}, \mathbf{t}) = \max_{r \in R_{pj}\left[A_p^c(\boldsymbol{\alpha})\right]} c_r(\mathbf{t}) = \max_{r \in R_{pj}\left[A_p^c(\boldsymbol{\alpha})\right]} \sum_{a \in r} t_a, \quad (45)$$



(a) Approach Proportions Space

(b) Link Flows Space

Figure 5 **Boundary Search and Its Alternatives**

where $A_p^c(\boldsymbol{\alpha}) = \{a \in A : \alpha_a > 0\}$ is the *contributing subnetwork*. Note the distinction between the term *used links* indicating strictly positive origin-based link flow $f_a > 0$ and the term *contributing links*, indicating strictly positive origin-based approach proportion $\alpha_a > 0$. The new restricting subnetwork is

$$\mathcal{A}_p(\boldsymbol{\alpha}) = A_p^c(\boldsymbol{\alpha}) \cup \big\{[i,j] \in A\colon u_i(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha}))) < u_j(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha})))\big\}. \quad (46)$$

Lemma 9. *$\mathcal{A}_p(\boldsymbol{\alpha})$ is spanning and acyclic.*

Proof. $\mathcal{A}_p(\boldsymbol{\alpha})$ is clearly spanning as it contains $A_p^c(\boldsymbol{\alpha})$. Suppose that $[v_0, v_1, \ldots, v_n = v_0]$ is a cycle such that $[v_k, v_{k+1}] \in \mathcal{A}_p(\boldsymbol{\alpha})$ $\forall 0 \le k \le n-1$. For brevity let $u_j = u_j(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha})))$. Notice that if $[v_k, v_{k+1}] \in A_p^c(\boldsymbol{\alpha})$, then by definition, $u_{v_k} \le u_{v_k} + t_{[v_k, v_{k+1}]} \le u_{v_{k+1}}$. Since $A_p^c(\boldsymbol{\alpha})$ is acyclic there must be at least one new link $[v_{k_0}, v_{k_0+1}]$ for which $u_{v_{k_0}} < u_{v_{k_0+1}}$. Therefore, $u_{v_0} \le \cdots \le u_{v_{k_0}} < u_{v_{k_0+1}} \le \cdots \le u_{v_n} = u_{v_0}$, a contradiction. □

The algorithmic map for a full unrestricted iteration is defined as follows:

$$\Theta^{\downarrow}(\boldsymbol{\alpha}) = \Theta^{\downarrow}(\boldsymbol{\alpha}\colon \mathcal{A}_p(\boldsymbol{\alpha})). \quad (47)$$

Theorem 2. *If $\boldsymbol{\alpha}^1$ is a feasible solution for the single-origin TAP, $\Delta\boldsymbol{\alpha}^k \in \Theta^{\downarrow}(\boldsymbol{\alpha}^k)$ and $\boldsymbol{\alpha}^{k+1} = \boldsymbol{\alpha}^k + \Delta\boldsymbol{\alpha}^k$, then every limit point of $\{\boldsymbol{\alpha}^k\}$ is a global equilibrium solution of the single-origin TAP.*

Proof. See Appendix. □

We now return to the multiple-origin case, where origin-based link flows are represented by the two-dimensional array, $\mathbf{f} = (f_{ap})_{a\in A;\ p\in N_o}$, and the origin-based approach proportions are represented by the array $\boldsymbol{\alpha} = (\alpha_{ap})_{a\in A;\ p\in N_o}$. The definition of the algorithmic map $\Theta^{\downarrow}$ uses many components that are origin dependent, including the topological order, average approach cost and its derivative, maximum contributing cost, and more. The extension of the algorithm to the multiple-origin case relies on the algorithmic map $\Theta^{\downarrow p}(\boldsymbol{\alpha})$, which is basically defined in the same way as $\Theta^{\downarrow}$; that is, $\Delta\boldsymbol{\alpha} \in \Theta^{\downarrow p}(\boldsymbol{\alpha})$ if and only if $\Delta\boldsymbol{\alpha}_p \in \Theta^{\downarrow}(\boldsymbol{\alpha}_p)$; $\Delta\boldsymbol{\alpha}_{p'} = 0$ $(\forall p' \ne p)$ where in $\Theta^{\downarrow}$ link costs and their derivatives are based on total flows, aggregated over all origins, using the given origin-based approach proportion array. In the complete algorithm, the changes obtained by $\Theta^{\downarrow p}$ are applied to the origins in cyclic order, given by $N_o = \{p_1, p_2, \ldots, p_n\}$.

To prove convergence of this algorithm, consider a sequence $\boldsymbol{\alpha}^{k,i}$; $0 \le i \le n$, where $\boldsymbol{\alpha}^{1,0}$ is some feasible acyclic solution, $\boldsymbol{\alpha}^{k,i} = \boldsymbol{\alpha}^{k,i-1} + \Delta\boldsymbol{\alpha}^{k,i}$; $\Delta\boldsymbol{\alpha}^{k,i} \in \Theta^{\downarrow p_i}(\boldsymbol{\alpha}^{k,i-1})$; $\boldsymbol{\alpha}^{k+1,0} = \boldsymbol{\alpha}^{k,n}$. As in the single-origin case, there exists a subsequence K such that $\forall k \in K$; $\forall l: 1 \le l \le |N|$;

$$\mathcal{A}_{p_i}(\boldsymbol{\alpha}_{p_i}^{k+l,i-1}) = A_{p_i}^{*l} \quad \forall i\colon 1 \le i \le n, \quad (48)$$

$$\boldsymbol{\alpha}^{k+l,i} \to \boldsymbol{\alpha}^{*l,i} \quad \forall i\colon 0 \le i \le n. \quad (49)$$

As a result, $\forall l$: $1 \le l < |N|$; $\forall i$: $1 \le i \le n$,

$$\boldsymbol{\alpha}^{k+l,i} - \boldsymbol{\alpha}^{k+l,i-1} = \Delta\boldsymbol{\alpha}^{k+l,i} \to \Delta\boldsymbol{\alpha}^{*l,i}$$

$$= \boldsymbol{\alpha}^{*l,i} - \boldsymbol{\alpha}^{*l,i-1}, \tag{50}$$

$$\Delta\boldsymbol{\alpha}^{*l,i} \in \Theta^{\downarrow p_i}\left(\boldsymbol{\alpha}^{*l,i-1} : A^{*l}_{p_i}\right). \tag{51}$$

The objective function value is a bounded, monotonically nonincreasing series, and hence converges to T^*; in particular, $T\left(\boldsymbol{\alpha}^{*l,i}\right) = T^*$. $T\left(\boldsymbol{\alpha}^{*l,i}\right) = T\left(\boldsymbol{\alpha}^{*l,i-1}\right)$ only if $\Delta\mathbf{f}\left(\boldsymbol{\alpha}^{*l,i-1}, \Delta\boldsymbol{\alpha}^{*l,i}\right) = 0$, and therefore $\mathbf{f}\left(\boldsymbol{\alpha}^{*l,i}\right) = \mathbf{f}^*$ for $1 \le l \le |N|$; $0 \le i \le n$. From here on the same proof as for Theorem 2 can be applied to each origin separately to show that $\mathbf{f}^*$ is indeed a user-equilibrium solution.

6. Experimental Results

The fixed-demand static traffic assignment problem for several networks was solved by the origin-based algorithm described above as well as by the state-of-the-practice Frank-Wolfe algorithm. The latter algorithm used the L-deque minimum cost routes algorithm of Pape (1974), considered by Pallottino and Scutella (1998) to be one of the best choices for transportation networks at the current state-of-the-art. All experiments were conducted with double precision arithmetic on a SUN Ultra 10, 333 MHz, 576 MB RAM, using the Solaris v2.6 operating system. All coding was done in C. Basic characteristics of the test networks are presented in Table 1.

Table 1 Basic Characteristics of Test Networks

Network	Zones (origins)	Nodes	Links	O-D pairs
Sioux Falls	24	24	76	528
Barcelona	110	1,020	2,522	7,922
Winnipeg	147	1,052	2,836	4,345
Chicago Sketch	387	933	2,950	93,513
Chicago Regional	1,790	12,982	39,018	2,297,945
Philadelphia	1,525	13,389	40,003	1,151,166

Figure 6 shows the convergence of the two algorithms for the Chicago Regional Network as measured by the relative gap as a function of CPU time, both on logarithmic scale. This figure demonstrates the clear advantage of the origin-based algorithm over the Frank-Wolfe algorithm. An extrapolation of the linear trend in the Frank-Wolfe case suggests that to obtain a relative gap of 1.0E-14 using the Frank-Wolfe algorithm would take about 80,000 years! Similar trends were observed in other networks (Bar-Gera 1999) and are summarized in Table 2. Larsson and Patriksson (1992) report solving Sioux-Falls to relative gap of 6.0E-3 in 7.04 seconds, a slightly different version of Barcelona to relative gap of 1.7E-3 in 485

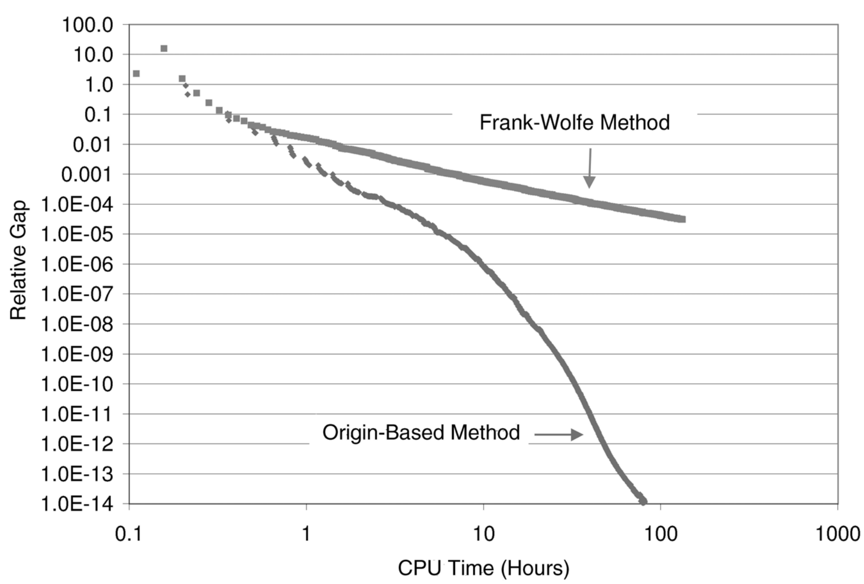


Figure 6 Relative Gap vs. CPU Time for the Chicago Regional Network

Table 2 Computation Time for Various Networks and Various Accuracy Levels

		Relative Gap					
Network	Algorithm	0.1	0.01	0.001	0.0001	0.00001	1.0E-14
Sioux Falls	FW	0.05s	0.18s	1.18s	12.50s	124.61s	
	OBA	0.05s	0.09s	0.17s	0.20s	0.39s	5s
Barcelona	FW	3.1s	5.9s	12.0s	24.9s	140.5s	
	OBA	6.3s	11.1s	16.0s	18.0s	21.9s	85s
Winnipeg	FW	4.1s	11.7s	33.4s	134.0s	>600s	
	OBA	11.1s	20.7s	28.8s	36.7s	43.2s	132s
Chicago Sketch	FW	0.4m	1.5m	7.1m	55.0m	>250m	
	OBA	0.4m	0.6m	1.0m	1.5m	2.8m	21m
Chicago Regional	FW	0.4h	1.5h	6.9h	43.8h	>120h	
	OBA	0.5h	0.8h	1.4h	3.0h	5.9h	82h
Philadelphia	FW	0.6h	1.2h	3.8h	>8h	>8h	
	OBA	0.9h	1.2h	1.9h	2.8h	4.0h	

Note. s = seconds; m = minutes; h = hours.

seconds, and a slightly different version of Winnipeg to relative gap of 0.99E-2 in 271 seconds. The differences in computation times are, at least in part, due to the differences between machines.

The efficiency in memory requirements can be demonstrated by the case of the Chicago Regional Network, where the origin-based solution did not exceed 112 MB at any time. In comparison, an equivalent route-based solution is estimated to require more than 6,000MB, or 50 times more than the origin-based solution. Even storing one set of trees of minimum cost routes requires as much as 93 MB.

7. Conclusions and Future Research

The origin-based algorithm presented here is highly efficient both in memory and in computation time, especially when highly accurate solutions are needed. The resulting solution provides detail that is equivalent to the detail of a route-based solution. These advantages make the origin-based algorithm attractive in practical applications, especially for large-scale networks.

The author plans to continue to test the algorithm on other networks, and to compare the results with other state-of-the-art algorithms. As indicated before, there are many possible extensions for this model, including nonsymmetric cost structures, combined models of travel demand and traffic assignment, stochastic user preferences, dynamic models, and more. The origin-based assignment may also provide a useful framework for determining the most likely route flows. These items provide a promising agenda for future research.

Acknowledgments

The author wishes to thank Professor David Boyce for his support, guidance, advice, and overall contribution to this research. The comments and suggestions of Professors Jane Hagstrom, Robert Abrams, and Donald Hearn are highly appreciated. Chicago network data were provided by the Chicago Area Transportation Study, Chicago, IL. Philadelphia data were provided by Thomas Walker, Delaware Valley Regional Planning Commission. Philadelphia experiments were conducted by Biljana Ralevic-Dekic. Winnipeg and Barcelona data were provided by Professor Michael Patriksson, Chalmers University, Sweden. Financial support was kindly provided by the National Science Foundation through the National Institute of Statistical Sciences, Research Triangle Park, NC.

Appendix

Proof of Lemma 8. To simplify the notation in this proof we use the following abbreviations: $g_j = g_j(\boldsymbol{\alpha}+\Delta\boldsymbol{\alpha})$; $\mu_a = \mu_a(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha})))$; $\hat{\mu}_a = \mu_a(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha}+\Delta\boldsymbol{\alpha})))$; $z_{a\to b} = z_{a\to b}(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha})), \mathbf{t}'(\mathbf{f}(\boldsymbol{\alpha})))$; $\Delta\mathbf{f} = \Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha})$. Let $x = \max\{-\Delta\alpha_a \cdot g_j(\boldsymbol{\alpha}+\Delta\boldsymbol{\alpha}) : j \in N\setminus\{p\}; a \in \mathrm{NB}_j\}$ and $t'_{\max} = \max\{t'_a(\mathbf{f}(\boldsymbol{\alpha})) : a \in A; \boldsymbol{\alpha} \in [0,1]^{|A_p|}\}$. Notice that if $x = 0$ for some positive λ, then $\Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha}) = 0$ for all λ and the lemma is proven. One can show that $\forall a \in A_p$; $|\Delta f_a| \le |A| \cdot x$; $|\hat{\mu}_a - \mu_a| \le |A|^2 \cdot x \cdot t'_{\max}$. Since $\mu_a \ge \mu_b$ for all $j \in N\setminus\{p\}$; $a \in \mathrm{NB}_j$; $b = b_j$; hence

$\hat{\mu}_a - \hat{\mu}_b \geq -2 \cdot |A|^2 \cdot x \cdot t'_{\max}$. For those $j \in N \setminus \{p\}$; $a \in \mathrm{NB}_j$; $b = b_j$ such that $\hat{\mu}_a < \hat{\mu}_b$,

$$\mu_a - \mu_b \leq 2 \cdot |A|^2 \cdot x \cdot t'_{\max}, \tag{52}$$

$$-\Delta\alpha_a \cdot g_j \leq \lambda \cdot z_{a\to b} \leq 2 \cdot \lambda \cdot |A|^2 \cdot x \cdot t'_{\max}/\epsilon_\nu. \tag{53}$$

Therefore,

$$\sum_{j\in N} \sum_{\substack{a\in \mathrm{NB}_j \\ \hat{\mu}_a < \hat{\mu}_b}} g_j \cdot \Delta\alpha_a \cdot (\hat{\mu}_a - \hat{\mu}_b) \leq 4\frac{\lambda}{\epsilon_\nu} \cdot |A|^5 \cdot x^2 \cdot t'^2_{\max}. \tag{54}$$

There exist $j_0 \in N$; $a_0 \in \mathrm{NB}_{j_0}$; $b_0 = b_{j_0}$ such that $x = -\Delta\alpha_{a_0} \cdot g_{j_0} \leq \lambda \cdot z_{a_0\to b_0} \leq \lambda \cdot \left(\mu_{a_0} - \mu_{b_0}\right)/\epsilon_\nu$; hence

$$\left(\hat{\mu}_{a_0} - \hat{\mu}_{b_0}\right) \geq \frac{x \cdot \epsilon_\nu}{\lambda} - 2 \cdot |A|^2 \cdot x \cdot t'_{\max}, \tag{55}$$

$$g_{j_0} \cdot \Delta\alpha_{a_0} \cdot \left(\hat{\mu}_{a_0} - \hat{\mu}_{b_0}\right) \leq -x^2 \cdot \left(\frac{\epsilon_\nu}{\lambda} - 2 \cdot |A|^2 \cdot t'_{\max}\right), \tag{56}$$

and as a result,

$$\Delta\mathbf{f} \cdot \mathbf{t}(\mathbf{f} + \Delta\mathbf{f}) \leq \mathbf{x}^2 \cdot \left(\mathbf{4}\frac{\lambda}{\epsilon_\nu} \cdot |\mathbf{A}|^5 \cdot \mathbf{t}'^2_{\max} + \mathbf{2} \cdot |\mathbf{A}|^2 \cdot \mathbf{t}'_{\max} - \frac{\epsilon_\nu}{\lambda}\right). \tag{57}$$

Notice that all the elements in parentheses except for λ are constants independent of λ, and that as λ converges to zero, this expression becomes negative. □

Proof of Theorem 1. Suppose $\boldsymbol{\alpha}^{k+1} = \boldsymbol{\alpha}^k + \Delta\boldsymbol{\alpha}^k$ and $\Delta\boldsymbol{\alpha}^k \in \Theta^{\downarrow}(\boldsymbol{\alpha}^k : A_p)$. There exists a subsequence K such that $\boldsymbol{\alpha}^{k+l} \to \boldsymbol{\alpha}^{*l}$ $\forall l$: $0 \leq l \leq |N|$. Hence,

$$\boldsymbol{\alpha}^{k+l+1} - \boldsymbol{\alpha}^{k+l} = \Delta\boldsymbol{\alpha}^{k+l} \to \Delta\boldsymbol{\alpha}^{*l} = \boldsymbol{\alpha}^{*l+1} - \boldsymbol{\alpha}^{*l} \quad \forall l : 1 \leq l < |N|,$$

$$\begin{array}{ccccccc}
\cdots\boldsymbol{\alpha}^{k_0+1} & \to & \boldsymbol{\alpha}^{k_0+2}\cdots & \to & \boldsymbol{\alpha}^{k_0+l}\cdots & \to & \boldsymbol{\alpha}^{k_0+|N|}\cdots \\
\cdots\boldsymbol{\alpha}^{k_1+1} & \to & \boldsymbol{\alpha}^{k_1+2}\cdots & \to & \boldsymbol{\alpha}^{k_1+l}\cdots & \to & \boldsymbol{\alpha}^{k_1+|N|}\cdots \\
\vdots & & \vdots & & \vdots & & \vdots \\
\cdots\boldsymbol{\alpha}^{k+1} & \to & \boldsymbol{\alpha}^{k+2}\cdots & \to & \boldsymbol{\alpha}^{k+l}\cdots & \to & \boldsymbol{\alpha}^{k+|N|}\cdots \\
\downarrow & & \downarrow & & \downarrow & & \downarrow \\
\boldsymbol{\alpha}^{*1} & \to & \boldsymbol{\alpha}^{*2}\cdots & \to & \boldsymbol{\alpha}^{*l}\cdots & \to & \boldsymbol{\alpha}^{*|N|}
\end{array} \tag{58}$$

Since the map is closed, $\Delta\boldsymbol{\alpha}^{*l} \in \Theta^{\downarrow}(\boldsymbol{\alpha}^{*l} : A_p)$. $T(\mathbf{f}(\boldsymbol{\alpha}^k))$ is a monotonically nonincreasing sequence, hence the limiting point of every subsequence is equal; that is, $T(\mathbf{f}(\boldsymbol{\alpha}^{*l})) = T^*$ $\forall 0 \leq l \leq |N|$. Therefore, $\Delta\mathbf{f}(\boldsymbol{\alpha}^{*l}, \Delta\boldsymbol{\alpha}^{*l}) = 0$; hence $\mathbf{f}(\boldsymbol{\alpha}^{*l}) = \mathbf{f}^*$ $\forall 0 \leq l \leq |N|$. Let $\mathbf{t}^* = \mathbf{t}(\mathbf{f}^*)$; $\mathbf{t}'^* = \mathbf{t}'(\mathbf{f}^*)$. To complete the proof we show by induction that $\boldsymbol{\alpha}^{*l}$ satisfies the approach proportions conditions for restricted equilibrium(27) at all nodes of topological order less than or equal to l. For $l = 1$, there are no approaches to the origin, hence the conditions hold in the empty sense. Suppose the theorem is true for l. $\Delta\boldsymbol{\alpha}^{*l}$ can have nonzero components only for links terminating at nodes of topological order higher than l. Therefore, equilibrium conditions will remain for all nodes of topological order no greater than l. Let j be the node of topological order $l+1$, $o(j) = l+1$. Suppose approach equilibrium conditions are not met for a, b; $a_h = b_h = j$; $\mu_a(\boldsymbol{\alpha}^{*l}, \mathbf{t}^*) > \mu_b(\boldsymbol{\alpha}^{*l}, \mathbf{t}^*)$; $\boldsymbol{\alpha}_a^{*l} > 0$. If $g_j > 0$, then $\Delta\mathbf{f}(\boldsymbol{\alpha}^{*l}, \Delta\boldsymbol{\alpha}^{*l}) \neq 0$, a contradiction. If $g_j = 0$, then $\Delta\alpha_a^{*l} = -\alpha_a^{*l}$ and therefore $\alpha_a^{*l+1} = 0$. □

Proof of Theorem 2. The following proof relies on the assumption that link costs are strictly positive. A proof for the case where zero cost links are allowed is given in Bar-Gera (1999).

There exists a subsequence K such that $\forall k \in K$; $\forall l$: $1 \leq l \leq |N|$, $\mathcal{A}_p(\boldsymbol{\alpha}^{k+l}) = A_p^{*l}$; $\boldsymbol{\alpha}^{k+l} \to \boldsymbol{\alpha}^{*l}$. As a result, $\boldsymbol{\alpha}^{k+l+1} - \boldsymbol{\alpha}^{k+l} = \Delta\boldsymbol{\alpha}^{k+l} \to \Delta\boldsymbol{\alpha}^{*l} = \boldsymbol{\alpha}^{*l+1} - \boldsymbol{\alpha}^{*l}$ $\forall l$: $1 \leq l < |N|$. The restriction update function $\mathcal{A}$ is not a closed map, and therefore A_p^{*l} and $\mathcal{A}_p(\boldsymbol{\alpha}^{*l})$ are not necessarily equal. The algorithmic map of the restricted iteration is closed, and therefore $\Delta\boldsymbol{\alpha}^{*l} \in \Theta^{\downarrow}(\boldsymbol{\alpha}^{*l} : A_p^{*l})$. The same arguments used in Theorem 1 to show that $\mathbf{f}(\boldsymbol{\alpha}^{*l}) = \mathbf{f}^*$ for every l: $1 \leq l \leq |N|$ are valid here as well. We refer to $\mathbf{f}^*$ as the limiting link flows. The limiting link costs and cost derivatives are denoted accordingly by $\mathbf{t}^* = \mathbf{t}(\mathbf{f}^*)$, $\mathbf{t}'^* = \mathbf{t}'(\mathbf{f}^*)$. Define the minimum limiting cost $w_j^* = \min_{r\in R_{pj}} c_r(\mathbf{t}^*)$. Consider the subnetwork $A_p^* = \{a \in A\colon w_{a_t}^* + t_a^* = w_{a_h}^*\}$ and denote $R_{ij}^* = R_{ij}[A_p^*]$; $R^* = \bigcup_{i,j\in N} R_{ij}^*$. Because A is finite, there exists a strictly positive value $\epsilon > 0$ such that $w_{a_t}^* + t_a^* > w_{a_h}^* + \epsilon$ $\forall a \in A \setminus A_p^*$. Since $\mathbf{f}(\boldsymbol{\alpha}^{k+l}) \to \mathbf{f}^*$, there exists k_0 such that $\forall k \in K$; $k \geq k_0$; $\forall l : 1 \leq l \leq |N|$;

$$\forall a \in A : \left|f_a^{k+l} - f_a^*\right| < \frac{\lambda_0 \cdot \epsilon}{4 \cdot |A| \cdot t'_{\max}}, \tag{59}$$

$$\forall r \in R : \left|c_r\left(\mathbf{f}\left(\boldsymbol{\alpha}^{k+l}\right)\right) - c_r\left(\mathbf{f}^*\right)\right| < \tfrac{\epsilon}{4}, \tag{60}$$

where $t'_{\max} = \max_{\boldsymbol{\alpha}} \max_{a\in A}\{t'_a(\mathbf{f}(\boldsymbol{\alpha}))\}$. Define the set of temporary "good nodes"

$$\check{N}^{k+l} = \left\{j \in N\colon R_{pj}[A_p^c(\boldsymbol{\alpha}^{k+l})] \subseteq R_{pj}^*\right\}. \tag{61}$$

Because link costs are assumed to be strictly positive, A_p^* is acyclic and has a topological order o^*. Using this topological order, define the set of permanent "good nodes"

$$\check{\check{N}}^{k+l} = \left\{j \in \check{N}^{k+l} : \forall i \in N;\, o^*(i) < o^*(j) \Rightarrow i \in \check{N}^{k+l}\right\}. \tag{62}$$

We show by induction on l that $\forall k \in K$; $k \geq k_0$; $|\check{\check{N}}^{k+l}| \geq l$; hence $\check{\check{N}}^{k+|N|} = N$ and

$$R_{pj}[A_p^c(\boldsymbol{\alpha}^{k+|N|})] \subseteq R_{pj}^* \quad \forall j \in N; \forall k \in K\colon k > k_0, \tag{63}$$

$$R_{pj}[A_p^c(\boldsymbol{\alpha}^{*|N|})] \subseteq R_{pj}^* \quad \forall j \in N, \tag{64}$$

implying that $\boldsymbol{\alpha}^{*|N|}$ is an equilibrium solution for TAP.

For $l = 1$, $R_{pp}[A_p^c(\boldsymbol{\alpha}^k)] = R_{pp}^* = [p]$. Assume for l, $|\check{\check{N}}^{k+l}| \geq l$; show for $l+1$. Suppose $|\check{\check{N}}^{k+l+1}| \leq l < N$, let

$$j = \arg\min\left\{o^*(j')\colon j' \in N \setminus \check{\check{N}}^{k+l+1}\right\}. \tag{65}$$

By the choice of j, for every $i \in N : o^*(i) < o^*(j) \Rightarrow i \in \check{\check{N}}^{k+l+1}$. So $o^*(j) - 1 \leq |\check{\check{N}}^{k+l+1}| \leq l$ or $o^*(j) \leq l+1$. We can also conclude that the only possible reason for j not to be a permanent "good node" is that it is not even a temporary "good node," that is, $j \notin \check{N}^{k+l+1}$. This means that there exists a "bad" contributing route to j,

$$\hat{r} = \hat{s} + [\hat{i}, j] \in R_{pj}[A_p^c(\boldsymbol{\alpha}^{k+l+1})] \setminus R_{pj}^*. \tag{66}$$

If $[\hat{i}, j] \in A_p^*$, then $\hat{s} \notin R_{p\hat{i}}^*$ and since $\hat{s} \in R_{p\hat{i}}[A_p^c(\boldsymbol{\alpha}^{k+l+1})]$, therefore $\hat{i} \in N \setminus \check{\check{N}}^{k+l+1}$. However $[\hat{i}, j] \in A_p^*$ also implies that $o^*(i) < o^*(j)$, in contradiction to the choice of j; hence $[\hat{i}, j] \notin A_p^*$. As a result, $w_{\hat{i}}^* + t_{[\hat{i},j]}^* > w_j^* + \epsilon$, and hence

$$\mu_{[\hat{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) > \mu_{[\hat{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) - \frac{\epsilon}{4}$$
$$\geq w_{\hat{i}}^* + t_{[\hat{i},j]}^* - \frac{\epsilon}{4} > w_j^* + \frac{3\epsilon}{4}. \tag{67}$$

Case 1. $j \in \check{\check{N}}^{k+l}$. $[\hat{i}, j] \notin A_p^* \Rightarrow [\hat{i}, j] \notin A_p^c(\boldsymbol{\alpha}^{k+l}) \Rightarrow \alpha_{[\hat{i},j]}^{k+l} = 0$. Let $\check{a} = [\check{i}, j]$ be any contributing approach; i.e., $[\check{i}, j] \in A_p^c(\boldsymbol{\alpha}^{k+l})$. Since $j \in \check{\check{N}}^{k+l}$, every contributing route is a "good route;" hence, for every contributing approach $\mu_{[\check{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) = w_j^*$ and $\mu_{[\check{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) < w_j^* + \epsilon/4 < \mu_{[\hat{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l}))$. Therefore, $[\hat{i}, j]$ is not a basic approach; hence $\Delta\alpha_{[\hat{i},j]}^{k+l} \leq 0$, and therefore $\alpha_{[\hat{i},j]}^{k+l+1} = 0 \iff [\hat{i}, j] \notin A_p^c(\boldsymbol{\alpha}^{k+l+1})$; but this contradicts the choice of the "bad" route in (66).

Case 2. $j \notin \check{\check{N}}^{k+l}$. A_p^* is spanning; hence, there is $[\check{i}, j] \in A_p^*$, $o^*(\check{i}) < o^*(j) \leq l+1$. By the induction assumption, $|\check{\check{N}}^{k+l}| \geq l$, so $o^*(\check{i}) \leq l$ implies that $\check{i} \in \check{\check{N}}^{k+l}$; hence $\mu_{[\check{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) = w_{\check{i}}^* + t_{[\check{i},j]}^* = w_j^*$. If $u_{\check{i}}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) < u_j(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l}))$, then the link $[\check{i}, j]$ must be in the restricting subnetwork A_p^{*l}. Hence, the average cost of the basic approach cannot be higher than the average cost of the $[\check{i}, j]$ approach; that is

$$\mu_{b_j}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) \leq \mu_{[\check{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l}))$$
$$< \mu_{[\check{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) + \frac{\epsilon}{4} = w_j^* + \frac{\epsilon}{4}. \tag{68}$$

Otherwise, $u_{\check{i}}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) \geq u_j(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l}))$; hence

$$\mu_{b_j}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) \leq u_j(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) \leq u_{\check{i}}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l}))$$
$$< u_{\check{i}}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) + \frac{\epsilon}{4} = w_{\check{i}}^* + \frac{\epsilon}{4} \leq w_j^* + \frac{\epsilon}{4}. \tag{69}$$

Either way, by (67),

$$\mu_{\hat{a}}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) - \mu_{b_j}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) > \frac{\epsilon}{2}, \tag{70}$$

$$z_{\hat{a}\to b_j}(\boldsymbol{\alpha}, \mathbf{t}(\boldsymbol{\alpha}^{k+l}), \mathbf{t}'(\boldsymbol{\alpha}^{k+l})) > \frac{\epsilon}{2 \cdot |A| \cdot t'_{\max}} > 0, \tag{71}$$

where $\hat{a} = [\hat{i}, j]$. However, since $\alpha_{\hat{a}}^{k+l+1} > 0$,

$$-\Delta\alpha_{\hat{a}}^{k+l} = \lambda^{k+l} \cdot \frac{z_{\hat{a}\to b_j}(\boldsymbol{\alpha}, \mathbf{t}(\boldsymbol{\alpha}^{k+l}), \mathbf{t}'(\boldsymbol{\alpha}^{k+l}))}{g_j(\boldsymbol{\alpha}^{k+l+1})}$$
$$\geq \frac{\lambda_0 \cdot \epsilon}{2 \cdot |A| \cdot t'_{\max} \cdot g_j(\boldsymbol{\alpha}^{k+l+1})}. \tag{72}$$

So, if $g_j(\boldsymbol{\alpha}^{k+l+1}) \leq g_j(\boldsymbol{\alpha}^{k+l})$, then

$$\Delta f_{\hat{a}}(\boldsymbol{\alpha}^{k+l}, \Delta\boldsymbol{\alpha}^{k+l}) = \Delta\alpha_{\hat{a}}^{k+l} \cdot g_j(\boldsymbol{\alpha}^{k+l+1}) + \alpha_{\hat{a}}^{k+l} \cdot (g_j(\boldsymbol{\alpha}^{k+l+1}) - g_j(\boldsymbol{\alpha}^{k+l}))$$
$$\leq -\frac{\lambda_0 \cdot \epsilon}{2 \cdot |A| \cdot t'_{\max}}, \tag{73}$$

contradicting the choice of k_0 in (59), while if $g_j(\boldsymbol{\alpha}^{k+l+1}) > g_j(\boldsymbol{\alpha}^{k+l})$, then

$$\Delta f_{b_j}(\boldsymbol{\alpha}^{k+l}, \Delta\boldsymbol{\alpha}^{k+l}) = \sum_{a \in \mathrm{NB}_j} \Delta\alpha_a^{k+l} \cdot g_j(\boldsymbol{\alpha}^{k+l+1})$$
$$+ \alpha_{b_j}^{k+l} \cdot (g_j(\boldsymbol{\alpha}^{k+l+1}) - g_j(\boldsymbol{\alpha}^{k+l}))$$
$$\geq \Delta\alpha_{\hat{a}}^{k+l} \cdot g_j(\boldsymbol{\alpha}^{k+l+1}) \geq \frac{\lambda_0 \cdot \epsilon}{2 \cdot |A| \cdot t'_{\max}}, \tag{74}$$

again contradicting the choice of k_0 in (59). □

References

Bar-Gera, H. 1999. Origin-based algorithms for transportation network modeling. Ph.D. thesis, Civil Engineering, University of Illinois at Chicago, Chicago, IL.

——, D. Boyce. 1999. Route flow entropy maximization in origin-based traffic assignment. A. Ceder, ed. *Transportation and Traffic Theory, Proceedings of the 14th International Symposium on Transportation and Traffic Theory, Jerusalem, Israel, 1999*. Elsevier Science, Oxford, U.K., 397–415.

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

Bertsekas, D. P. 1979. Algorithms for nonlinear multicommodity network flow problems. A. Bensoussan and J. L. Lions, eds. *Proceedings of the International Symposium on Systems Optimization and Analysis*. Springer-Verlag, Berlin, Germany, 210–224.

——. 1999. *Network Optimization—Continuous and Discrete Models*. Athena Scientific, Belmont, MA.

——, E. M. Gafni, R. G. Gallager. 1984. Second derivative algorithms for minimum delay distributed routing in networks. *IEEE Trans. Comm.* **COM-32** 911–919.

——, ——, K. S. Vastola. 1979. Validation of algorithms for optimal routing of flow in networks. *Proc. 1979 IEEE Conf. Decision and Control*, San Diego, CA, 220–227.

Bothner, P., W. Lutter. 1982. *Ein Direktes Verfahren Zur Verkehrsumlegung nach dem 1. Prinzip von Wardrop*, Forschungsbereich: Verkehrssysteme Arbeitsbericht 1, Universität Bremen, Bremen, Germany.

Bruynooghe, M., A. Gibert, M. Sakarovitch. 1969. Une méthode d'affectation du traffic. W. Leutzbach and P. Baron, eds. *Proceedings of the 4th International Symposium on the Theory of Road Traffic Flow, Karlsruhe, 1968*. Beiträge zur Theorie des Verkehrsflusses, Strassenbau und Strassenverkehrstechnik, Heft 86, Herausgegeben von Bundesminister für Verkehr, Abteilung Strassenbau, Bonn, Germany, 198–204.

Dafermos, S. C. 1968. Traffic assignment and resource allocation in transportation networks. Ph.D. thesis, Operations Research, Johns Hopkins University, Baltimore, MD.

——, F. T. Sparrow. 1969. The traffic assignment problem for a general network. *J. Res. National Bureau of Standards* **73B** 91–118.

Dial, R. B. 1971. A probabilistic multipath traffic assignment model which obviates path enumeration. *Transportation Res.* **5** 83–111.

Florian, M., D. Hearn. 1995. Network equilibrium models and algorithms. M. O. Ball et al., eds. *Network Routing, Handbooks in OR & MS*, vol. 8. Elsevier Science, Oxford, U.K., 485–549.

——, H. Spiess. 1983. Transport networks in practice. *Proc. Conf. Oper. Res. Soc. Italy*, Napoli, Italy, 29–52.

Frank, M., P. Wolfe. 1956. An algorithm for quadratic programming. *Naval Res. Logist. Quart.* **3** 95–110.

Fukushima, M. 1984. A modified Frank-Wolfe algorithm for solving the traffic assignment problem. *Transportation Res.* **18B** 169–177.

Gallager, R. G. 1977a. A minimum delay routing algorithm using distributed computation. *IEEE Trans. Comm.* **COM-25** 73–85.

——, 1977b. Loops in multicommodity flows. *Proceedings of the 10th IEEE Conf. Decision and Control*, New Orleans, TX, 819–825.

Gibert, A. 1968. A method for the traffic assignment problem. Report LBS-TNT-95. Transportation Network Theory Unit, London Business School, London, U.K.

Hagstrom, J. N., P. Tseng. 1998. Traffic equilibrium: Link flows, path flows and weakly/strongly acyclic solutions. Working paper, Department of Information and Decision Sciences, University of Illinois at Chicago, Chicago, IL.

Hearn, D. W., S. Lawphongpanich, J. A. Venture. 1987. Restricted simplicial decomposition: Computation and extensions. *Math. Programming Study* **31** 99–118.

Jayakrishnan, R., W. K. Tsai, J. N. Prashker, S. Rajadhyaksha. 1994. A faster path-based algorithm for traffic assignment. *Transportation Res. Record* **1443** 75–83.

Kupiszewska, D., D. Van Vliet. 1999. 101 uses for path-based assignment. *Transportation Planning Methods: Proceedings of Seminar C held at the PTRC Transport and Planning Summer Annual Meeting*. University of Sussex, U.K., no. 434.

Larsson, T., M. Patriksson. 1992. Simplicial decomposition with disaggregated representation for the traffic assignment problem. *Transportation Sci.* **26** 4–17.

——, ——, C. Rydergren. 1998. Application of simplicial decomposition with nonlinear column generation to nonlinear network flows. *Licentiate Thesis.* Clas Rydergren, Thesis No. 702, Linköping Institute of Technology, Linköping, Sweden.

LeBlanc, L. J., R. V. Helgason, D. E. Boyce. 1985. Improved efficiency of the Frank-Wolfe algorithm for convex network programs. *Transportation Sci.* **19** 445–462.

——, E. K. Morlok, W. P. Pierskalla. 1975. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Res.* **9** 309–318.

Lupi, M. 1986. Convergence of the Frank-Wolfe algorithm for solving the traffic assignment problem. *Civil Engr. Systems* **3** 7–15.

Pallottino, S., M. G. Scutella. 1998. Shortest path algorithms in transportation models: Classical and innovative aspects. P. Marcotte and S. Nguyen, eds. *Equilibrium and Advanced Transportation Modelling*. Kluwer Academic Publishers, Boston, MA, 245–281.

Pape, U. 1974. Implementation and efficiency of Moore-algorithms for the shortest route problem. *Math. Programming* **7** 212–222.

Patriksson, M. 1994. *The Traffic Assignment Problem—Models and Methods*. VSP, Utrecht, Netherlands.

Wardrop J.G. 1952. Some theoretical aspects of road traffic research. *Proc. Institution of Civil Engineers, Part II*, **1** 325–378.

Received: April 2000; revisions received: April 2001, August 2001; accepted: August 2001.