NISS

Origin-based algorithms for Transportation Network Modeling

Hillel Bar-Gera

Technical Report Number 103
November, 1999

National Institute of Statistical Sciences
19 T. W. Alexander Drive
PO Box 14006
Research Triangle Park, NC 27709-4006
www.niss.org

ORIGIN-BASED ALGORITHMS FOR TRANSPORTATION NETWORK MODELING

Hillel Bar-Gera

Technical Report #103
October, 1999

National Institute of Statistical Sciences
P.O. Box 14006
Research Triangle Park, NC 27709

For additional copies please contact:

Hillel Bar-Gera or David Boyce at
Civil and Materials Engineering Department (MC 246)
University of Illinois at Chicago
842 W. Taylor St. Chicago, IL 60607

hbarge1@uic.edu, dboyce@uic.edu

PREFACE

Convergent algorithms for solving the deterministic user-optimal traffic assignment (route choice) problem with fixed demand have been studied since the 1960s. Although several innovative approaches for solving this problem have been advanced during the past 30 years, none seems to offer compelling practical advantages for solving large-scale problems over the method of Frank and Wolfe (1956). This may be regarded as surprising since the convergence of the Frank-Wolfe method is known to be extremely slow. See Patriksson (1994, Ch. 4) for a historical review.

Shortly after we first met in September, 1997, Hillel Bar-Gera and I discussed this issue in the context of the recent interest sparked by the route-based DSD algorithm of Larsson and Patriksson (1992). When Hillel suggested to me that an origin-based approach might be more effective, I cautioned him to be careful, since many capable researchers had taken up this problem.

This report is the result of Hillel's investigation of this problem. The contents of the report are identical, except in format, to his Ph.D. thesis, which he successfully defended in August, 1999. Based on his computational findings, as well as his thorough mathematical analysis of the problem, I have concluded that his findings represent a major breakthrough of substantial practical importance to professional practitioners.

The Bar-Gera Algorithm enables one to compute the solution to large-scale assignment problems with very high accuracy and much more detail than link-based methods, such as the Frank-Wolfe method, and with less computational effort than is conventionally used to obtain the approximate solutions used in practice. Moreover, the method requires a relatively modest amount of memory by today's standard.

One still hears the view expressed that the Frank-Wolfe method is adequate since the available data don't warrant better solutions. I believe this viewpoint is ill-founded. Practitioners expect that their assignment results should allow them to distinguish between the performance of alternative transportation plans at the level of individual link flows. Therefore, it is implausible that they should be satisfied with a solution method whose link flows do vary substantially from iteration to iteration. Moreover, some software solves the assignment problem using integer arithmetic for link flows. Such software cannot yield solutions to the accuracy required for plan evaluation and air quality performance measures.

Prior to enrolling in the Ph.D. program in civil engineering at UIC, Hillel Bar-Gera studied mathematics, computer science and physics at Hebrew University, Jerusalem. He also studied transportation engineering at North Carolina State University. During his two years as a Research Assistant at UIC, he made numerous important contributions to our research aimed at implementing, estimating and validating a combined urban travel choice model for the Chicago region. Since October, 1999, he is serving as a Post-Doctoral Research Associate on this project.

I would like to express my appreciate to the National Institute of Statistical Sciences, for the financial support of this research, and for the encouragement and collaboration provided by Dr. Jerome Sacks, Director. The support of the National Science Foundation through NISS is gratefully acknowledged.

David Boyce
Principal Investigator
Regional Travel Model Validation Project

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

TAP	Traffic Assignment Problem
RSOTAP	Restricted Single Origin Traffic Assignment Problem
MEUE	Maximum Entropy User Equilibrium
FW	Frank-Wolfe Algorithm

LIST OF NOTATION

flows:

d_{pq}	O-D flow (demand)
$\hat{d}$	total flow
h_{rpq}	route flow
f_{ap}	flow on link a from origin p
$\mathbf{f}_p$	origin-based link flow vector for origin p
$\mathbf{f}$	origin-based link flow array
$f_{a\bullet}$	total flow on link a
$\mathbf{f}_\bullet$	total link flow vector
g_{spq}	origin-destination route segment flow
g_{sp}	origin-based route segment flow
g_{jp}	origin-based node flow
α_{ap}	origin-based approach proportion
$\boldsymbol{\alpha}_p$	approach proportion vector for origin p
$\boldsymbol{\alpha}$	approach proportion array
$\chi_{i\rightarrow j}$	proportion of flow to j through i
H	the set of feasible route flow assignments
H^*	the set of user equilibrium route flow solutions
F	the set of feasible origin-based assignments
$F_\bullet$	the set of feasible total link flow assignments

costs:

$t_a, \mathbf{t}$	link cost, link cost vector
$t'_a, \mathbf{t}'$	link cost derivatives
c_s	route segment cost
C_{pq}	minimum O-D cost
μ_a	average cost for approach a
μ_b	average cost for basic approach b
σ_j	average cost to node j
ν_a	approximated derivative of μ_a cost with respect to f_a
ρ_j	approximated derivative of σ_j with respect to g_j
u_i	maximum cost to node i
w_i	minimum cost to node i
T	objective function

network:

$G = (N, A)$	directed graph
N	nodes
N_o	origins
$N_d(p)$	destinations for origin p
A	links
$a \equiv [a_t, a_h]$	link
a_h	head node of link a
a_t	tail node of link a
$r = [v_1, \ldots, v_n]$	route, or route segment
R	all simple routes
R_{ij}	simple route segments from i to j
$r + s$	route segments combination (concatenation)
δ_{ra}	link-route incidence matrix
p	origin
q	destination
i, j, v	nodes

subnetworks:

A_p	restricting subnetwork
$o(j)$	topological order
A_p^c	contributing subnetwork
A_p^u	used subnetwork
$R_{ij}[A_p]$	route segments in a restricting subnetwork A_p
lcn_j	last common node to j
b_j	basic approach to node j
B	basic tree
NB_j	non-basic approaches to node j
NB	all non-basic approaches

algorithm:

$z_{a \to b}$	desirable shift
Θ	algorithmic map
$\mathcal{A}$	restriction update map

SUMMARY

The transportation planning process relies on travel forecasts that result from various transportation models. Some of the well-known models are formulated as non-linear convex optimization problems. Solving these problems is quite challenging due to their non-linear nature and their combinatorial structure. Large scale networks of practical interest increase the need for computationally efficient algorithms. The goal of this research is to improve upon existing algorithms for various models.

At the heart of most transportation models stands the traffic assignment problem, which is to predict the route choice of travelers given the origin and destination of each traveler, 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 keeps track of all used routes and the flow on each of those routes. This approach allows to achieve higher accuracy, but at the expense of large memory requirements that are often regarded as impractical for large scale networks.

This research presents a different approach in which the solution is represented by origin-based link flows aggregated over all destinations. This approach provides highly accurate detailed solutions, its memory requirements are reasonable even for large-scale networks, while computation times are substantially lower than for link-based algorithms.

A key point in the proposed algorithm is that only a-cyclic solutions are considered through all iterations. Indeed in the equilibrium (optimal) solution, the links used by travelers from a specific origin form an a-cyclic sub-network. The absence of cycles allows most of the necessary computations to be performed in a time that is a linear function of network size. An efficient (and rather simple) procedure allows to dis-aggregate a-cyclic origin-based link flows into route flows. In that sense the detail

provided by a-cyclic origin-based link flows is practically equivalent to the detail provided by route flows. On the other hand, an origin-based solution does not require as much memory as an equivalent route-based solution. For the implementation of the origin-based algorithm a special data structure was developed that reduces memory requirements even further. For some large scale network the proposed data structure requires 50 times less memory than the equivalent route-based representation. Computational efficiency in time and memory makes this algorithm highly suitable for large scale networks.

In general, the cost minimizing assumption allows only total link flows to be determined. For many applications total link flows are sufficient; however, for certain analyses practitioners are interested in fully detailed solutions. To determine route flows an additional assumption is needed. Route flow entropy maximization has been proposed as a criterion for the most likely route flow pattern. In this research an alternative behavioral assumption, referred to as the bypass proportionality assumption, is proposed, which leads to rather similar results. The two assumptions are studied both in a general context and also in an origin-based context.

1. BACKGROUND ON TRANSPORTATION MODELING

1.1 Introduction

Transportation plays a key role in modern life. People use transportation to get from one activity to the other. Modern commerce also relies heavily on transportation of materials and products. Improving transportation systems is therefore essential to the quality of our lives. Planning improvements in such systems requires careful consideration of the various alternatives. The different alternatives are evaluated using models that attempt to capture the nature of transportation systems, and thus allow one to predict the effect of future changes on system performance. Measures of performance include efficiency in time and/or money, safety, social and environmental impacts and more.

Operational studies typically focus on the performance of a single element in the transportation system, like a roadway segment, an intersection, a rail line, and so forth. The performance of some elements is determined only by the available infrastructure; for example, train schedules are supposedly independent of the number of travelers. The travel time along a roadway on the other hand is clearly highly dependent on the level of congestion. The performance of a roadway segment or an intersection is therefore typically described as a function of traffic flows. This is only an approximate description, as it does not consider other influencing factors like arrival patterns and differences in driver behavior; nevertheless, such a description does seem to capture the main effects of traffic congestion.

To predict the actual performance of the transportation system, an estimate of the traffic flow pattern is needed. The overall flow pattern results from the decisions of many users. Users may avoid using a certain roadway segment by choosing an alternative route, an alternative mode of travel like public transportation, walking or riding a bike, or even by not making the trip altogether. These travel decisions depend on decisions like where to live, where to work, where to shop, and other questions that are often considered in a wider context of regional economics, land-use planning, and related disciplines. People make many other choices that affect location and travel decsision; for example, the choice to own a vehicle, choices of entertainment and recreational activities, and even the choice to have a family and how many children to have.

From a research point of view it is almost impossible to study all of these decisions simultaneously; hence social sciences have evolved into various disciplines that concern different aspects of people decisions. In reality, however, people do make many of these decisions simultaneously. Almost every pair of decisions mentioned earlier are interrelated. Perhaps one of the most complicated examples is the question of where to live. In answering that question the work/study places of all household members are considered, as well as many other considerations like shopping and entertainment opportunities, the availability of public transportation, parking availability, and the effort required for traveling from the location of one's home to various activities.

This research deals mainly with the modeling of travelers' route choice on the highway network, commonly known as the traffic assignment problem. For many years this problem was studied as a stand alone problem. A model integrating travel demand and route choice was proposed earlier by Beckmann et al. (1956); the first efficient method for its solution was presented by Evans (1976). Current efforts are directed at further integrating this model with regional economic and location models. At the other extreme, route choice effects should also be taken into account when local operational decisions are made, for example in the case of traffic signal timing.

The methods and results presented here are highly applicable to any of these problems. This study is therefore useful not only in the traditional context of transportation planning, but also in a wide range of applications varying from traffic signal timing to studies in regional economics.

1.2 Definitions

A transportation model considers a specific *study area* which 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, and A the set of directed links. Nodes represent intersections, interchanges, zone centroids, toll booths, and any other point of interest in the network. Links connect the nodes in the network, and as such they mainly represent roadway segments. In addition they may represent virtual connectors between zone centroids and the actual network. In some applications they may also represent turning movements within intersections.

A (simple) route segment is a sequence of (distinct) nodes $[v_1, \ldots, v_k]$ such that $[v_l, v_{l+1}] \in A \quad \forall 1 \leq l \leq k-1$. 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. Some refer to such graphs

as simple.) 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 $a \in A$.

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 $(r + s) = [i = v_1, \ldots v_{n-1}, v_n = j = u_1, u_2 \ldots, u_m = k]$. In general, a combination of simple route segments may not be simple; if it is simple, then $(r + s) \in R_{ik}$. 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 1 if link a is part of route r and zero otherwise. Comment: 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}. For the most part we assume that the sets $N_d(p)$ include only those destinations with positive flow, that is $d_{pq} > 0 \quad \forall p \in N_o; \forall q \in N_d(p)$. The total flow $\hat{d}$, is defined as the sum of flows over all O-D pairs, that is:

$$\hat{d} = \sum_{p \in N_o} \sum_{q \in N_d(p)} d_{pq} \tag{1.1}$$

The flow along route $r \in R_{pq}$ from origin p to destination q is denoted by h_{rpq}. Aggregating route flows through a link over all destinations results in origin-based link flows

$$f_{ap}(\mathbf{h}) = \sum_{q \in N_d(p)} \sum_{r \in R_{pq}; a \subseteq r} h_{rpq} \tag{1.2}$$

Further aggregating those over all origins result in total link flows

$$f_{a\bullet}(\mathbf{h}) = \sum_{p \in N_o} f_{ap} = \sum_{r \in R; a \subseteq r} h_{rpq} \tag{1.3}$$

A non-negative vector of route flows $\mathbf{h}$ represents a feasible assignment if it satisfies the origin-destination (O-D) flow. The set of feasible route flow assignments is therefore

$$H = \left\{ \mathbf{h} \in \mathbb{R}^{|R|} : \quad \mathbf{h} \geq 0; \ \sum_{r \in R_{pq}} h_{rpq} = d_{pq} \ \forall p \in N_o; \forall q \in N_d(p) \right\} \tag{1.4}$$

An origin-based link flow array $\mathbf{f}$ is feasible iff it is the aggregation of some feasible route flow vector $\mathbf{h}$. The set of feasible origin-based link flow arrays is therefore

$$F = \{\mathbf{f}(\mathbf{h}) : \mathbf{h} \in H\} \tag{1.5}$$

Similarly, a vector of total link flows $\mathbf{f}_\bullet$ is feasible iff it is the aggregation of some feasible route flow vector $\mathbf{h}$. The set of feasible total link flow vectors is therefore

$$F_\bullet = \{\mathbf{f}_\bullet(\mathbf{h}) : \mathbf{h} \in H\} \tag{1.6}$$

In general determining whether a vector of total link flows is feasible or not is a nontrivial task. For an origin-based link flow array, however, feasibility can be determined directly using the following definitions. Let $\mathbf{E} = (E_{ai})_{a \in A; i \in N}$ be the link-node incidence matrix, defined by

$$E_{ai} = \begin{cases} 1 & i = a_h \\ -1 & i = a_t \\ 0 & \text{otherwise} \end{cases} \tag{1.7}$$

Let $\mathbf{e} = (e_{pi})_{p \in N_o; i \in N}$ be the expanded O-D flow matrix, defined by

$$e_{pi} = \begin{cases} d_{pq} & i \in N_d(p) \setminus \{p\} \\ -\sum_{q \in N_d(p) \setminus \{p\}} d_{pq} & i = p \\ 0 & i \notin N_d(p); i \neq p \end{cases} \tag{1.8}$$

Consider the following set of origin-based link flow arrays

$$\hat{F} = \left\{ \mathbf{f} \in \mathbb{R}^{|A| \times |N_o|} : \quad \mathbf{f} \geq 0; \ \mathbf{f}^t \cdot \mathbf{E} = \mathbf{e} \right\} \tag{1.9}$$

The condition $\mathbf{f}^t \cdot \mathbf{E} = \mathbf{e}$ ensures that the origin-based link flow array satisfies the O-D flow while maintaining conservation of flow at every node. This condition is clearly met by every feasible origin-based link flow array, hence $F \subseteq \hat{F}$. The set

$\hat{F}$ may contain origin-based link flow arrays for which every decomposition to route flows involves cyclic routes; hence in general $\hat{F} \not\subseteq F$. One may consider $\hat{F}$ rather than F as the set of feasible origin-based link flow arrays. Our main interest here is in a-cyclic origin-based link flow arrays, meaning that the links used by each origin form an a-cyclic subnetwork. An a-cyclic origin-based link flow array $\mathbf{f}$ is in $\hat{F}$ only if $\mathbf{f} \in F$; hence in this context the conditions in (1.9) allow us to determine feasibility directly, and the distinction between F and $\hat{F}$ is of minor significance.

Comment: in this work the O-D flow pattern and the resulting route flow pattern are assumed to be static in time for the entire period covered by the model; usually a congested travel period. This assumption yields tractable consistent models, even though transportation systems are clearly dynamic in nature. Generalization of this assumption to dynamic models is an area of active, ongoing research.

The main decision attribute captured by transportation models is the cost of travel. The term cost is used here in the most general way, and can be interpreted as travel time, monetary cost, some combination of those, or any other measure of disutility of traveling along a specific route. The cost of travel along route segment s, c_s is assumed additive, i.e. $c_{s_1+s_2} = c_{s_1} + c_{s_2}$. As a result the cost of an empty route segment $[i]$ must be zero since $c_{[i]} = c_{[i]+[i]} = c_{[i]} + c_{[i]} = 0$. Intersection delays may be attached to the approaching links, or to special links representing lane movements if such links are included in the network representation. The vector of link costs is denoted by $\mathbf{t} = (t_a)_{a \in A}$, hence $c_s = \sum_{a \subseteq s} t_a$. Due to congestion effects, the cost of traveling along a link is likely to depend on the flow of that link, and perhaps on the flows of other links as well. In general, the cost of link a may be a function of the entire link flow vector, $t_a(\mathbf{f}_\bullet)$. For the sake of simplicity we assume here that link costs are separable, that is $t_a(\mathbf{f}_\bullet) = t_a(f_{a\bullet})$. For the most part link costs are further assumed to be non-negative, monotonically non-decreasing, and continuously differentiable functions of total link flows. In some cases stronger assumptions are made, that link costs are strictly positive and/or strictly increasing. For brevity the cost derivative of link a is denoted by $t'_a = \mathrm{d}t_a/\mathrm{d}f_{a\bullet}$.

1.3 The Traffic Assignment Problem (TAP)

Given the O-D flows the traffic assignment problem is to allocate those flows to specific routes according to a given behavioral hypothesis. A common hypothesis in transportation research is that users seek to minimize the cost associated with their chosen routes. These flow-dependent costs are assumed to be known perfectly in advance. Under these assumptions, known as Wardrop's user equilibrium principle, for every pair, origin p and destination q, the equilibrium flow on route r can be

positive only if the equilibrium cost of route r is not greater than the equilibrium cost of any alternative route r' from p to q (Wardrop, 1952). The user-equilibrium Traffic Assignment Problem (TAP) is to find such an assignment, either in terms of route flows, or in terms of the resulting total link flows.

Suppose that

$$T(\mathbf{f}_\bullet) = \int_0^{\mathbf{f}_\bullet} \mathbf{t}\left(\mathbf{f}_\bullet\right) \cdot \mathrm{d}\mathbf{f}_\bullet \tag{1.10}$$

is well defined, i.e. path-independent. Consider the set

$$H^* = \operatorname{argmin}\left\{T\left(\mathbf{f}_\bullet(\mathbf{h})\right) : \ \mathbf{h} \in H\right\} \tag{1.11}$$

which is the set of non-negative route flow vectors that minimize the Lagrangian

$$L(\mathbf{h}) = T(\mathbf{f}_\bullet(\mathbf{h})) + \sum_{p \in N_o} \sum_{q \in N_d(p)} \phi_{pq} \cdot \left(d_{pq} - \sum_{r \in R_{pq}} h_{rpq} \right) \tag{1.12}$$

Notice that

$$\begin{aligned} \frac{\partial L}{\partial h_{rpq}}(\mathbf{h}) &= \nabla_{\mathbf{f}_\bullet} T(\mathbf{f}_\bullet(\mathbf{h})) \cdot \frac{\partial \mathbf{f}_\bullet}{\partial h_{rpq}}(\mathbf{h}) - \phi_{pq} \\ &= \sum_{a \in A} \frac{\partial T}{\partial f_{a\bullet}}(\mathbf{f}_\bullet(\mathbf{h})) \cdot \delta_{ra} - \phi_{pq} \\ &= \sum_{a \subseteq r} t_a(\mathbf{f}_\bullet(\mathbf{h})) - \phi_{pq} \\ &= c_r(\mathbf{f}_\bullet(\mathbf{h})) - \phi_{pq} \end{aligned} \tag{1.13}$$

Therefore, by the first order necessary optimality conditions (using linear independence constraint qualifications) for every $\mathbf{h} \in H^*$ and for all $p \in N_o; q \in N_d(p); r \in R_{pq}$

$$\frac{\partial L}{\partial h_{rpq}}(\mathbf{h}) = c_r(\mathbf{f}_\bullet(\mathbf{h})) - \phi_{pq} \geq 0 \tag{1.14a}$$

$$h_{rpq} \cdot \frac{\partial L}{\partial h_{rpq}}(\mathbf{h}) = h_{rpq} \cdot (c_r(\mathbf{f}_\bullet(\mathbf{h})) - \phi_{pq}) = 0 \tag{1.14b}$$

or equivalently

$$c_r(\mathbf{f}_\bullet(\mathbf{h})) \geq \phi_{pq} \tag{1.15a}$$
$$h_{rpq} > 0 \Rightarrow c_r(\mathbf{f}_\bullet(\mathbf{h})) = \phi_{pq} \tag{1.15b}$$

meaning that $\mathbf{h}$ satisfies the user equilibrium conditions.

We assumed that link costs are separable, hence

$$T(\mathbf{f}_\bullet) = \int_0^{\mathbf{f}_\bullet} \mathbf{t}\,(\mathbf{f}_\bullet) \cdot \mathrm{d}\mathbf{f}_\bullet = \sum_{a \in A} \int_0^{f_{a\bullet}} t_a(x)\mathrm{d}x \tag{1.16}$$

which is well defined. Furthermore, link costs were assumed monotonically non-decreasing, hence $T(\mathbf{f}_\bullet)$ is a convex function, meaning that the first order conditions are also sufficient for optimality. In conclusion, under the separability and monotonicity assumption H^* is the set of user equilibrium route flows. Aggregating user equilibrium route flows yields the set of user equilibrium origin-based link flows F^*, and the set of user equilibrium total link flows $F^*_\bullet$.

If link costs are strictly increasing then T is strictly convex as a function of $\mathbf{f}_\bullet$; hence the total link flow equilibrium solution is unique. It should be noted that the objective function T depends on total link flows only, regardless of the specific route flow pattern. In general, different route flow patterns may lead to the same total link flow pattern. Therefore, even if there is a unique optimal solution for TAP in terms of total link flows, this solution may have many route flow representations; hence, typically, the route flow equilibrium solution is not unique. Similar arguments show that in most cases the user-equilibrium origin-based link flow solution is also not unique.

One of the well known properties of user-equilibrium solutions is that they do not contain cyclic flows. It is relatively easy to demonstrate that routes that contain cycles need not be used by a user-equilibrium solution. If link costs are strictly positive then the cost of every cycle is positive, hence a route that contains a cycle can not be of minimum cost, and therefore at equilibrium such a route can not be used. If there is a cycle with zero cost, such a cycle can be a part of a route that is used at equilibrium, but there is always another user-equilibrium solution where that route is not used.

Origin-based solutions are considered to be a-cyclic if for every origin p every directed cycle $s = [v_1, \ldots, v_n = v_1]$ contains at least one unused link $a \subseteq s; f_{ap} = 0$. By the following lemma (1) if link costs are strictly positive and if $\mathbf{f}$ is not a-cyclic then there exists $\mathbf{f}' \in F$ such that $T(\mathbf{f}_\bullet(\mathbf{f}')) < T(\mathbf{f}_\bullet(\mathbf{f}))$, hence $\mathbf{f}$ is not an optimal solution of TAP. In general it is always possible to choose $\mathbf{f} \in F^*$ such that $\forall \mathbf{f}' \in F^*; \mathbf{f}' \leq \mathbf{f} \Rightarrow \mathbf{f}' = \mathbf{f}$. Such a flow pattern $\mathbf{f}$ must be a-cyclic, otherwise by lemma 1 there exists $\mathbf{f}' \in F; \mathbf{f}' \neq \mathbf{f}$

such that $\mathbf{f}' \leq \mathbf{f}$ hence $T(\mathbf{f}_\bullet(\mathbf{f}')) \leq T(\mathbf{f}_\bullet(\mathbf{f}))$ and therefore $\mathbf{f}' \in F^*$. For additional discussion see Hagstrom and Tseng (1998).

Lemma 1. *If* $\mathbf{f} \in F$ *is not a-cyclic then there exists* $\mathbf{f}' \in F$ *such that* $\mathbf{f}' \leq \mathbf{f}$ *and* $f'_{ap} < f_{ap}$ *for some* a, p.

Proof:
If $\mathbf{f}$ is not a-cyclic then there exists origin p_0 and a cycle $s = [v_1, \ldots, v_n = v_1]$ such that $\forall a \subseteq s; f_{ap_0} > \epsilon$ for some $\epsilon > 0$. Consider $\mathbf{f}'$ where $f'_{ap_0} = f_{ap_0} - \epsilon$ for all $a \subseteq s$, and $f'_{ap} = f_{ap}$ otherwise. Clearly $\mathbf{f}'$ is non-negative and it maintains conservation of flow, hence it is a feasible origin-based solution, that is, $\mathbf{f}' \in F$. Furthermore, $\mathbf{f}' \leq \mathbf{f}$, and for every $a \subseteq s; f'_{ap_0} < f_{ap_0}$. □

1.4 Review of algorithms for TAP

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

All of the existing solution methods for TAP are iterative; i.e. they start by considering some initial assignment, calculate the costs using the flows of the considered assignment, then modify the assignment and update the costs. One way to categorize solution methods is by the level of aggregation in which they store the solution between iterations. The most aggregated approach is the link-based approach of storing total link flows, aggregated over all origin-destination pairs. The main advantage of this approach is its relatively modest memory requirements. The most disaggregated approach is the route-based approach of storing all used routes and the flow on each. Route-based methods have been shown to achieve better solutions; the main disadvantage is their large memory requirements.

The most common solution method for TAP is based on the general nonlinear optimization method of Frank and Wolfe (1956). In each iteration of the Frank and Wolfe (FW) method, 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. Also see section 4.1 for some 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 multi-dimensional search over the convex hull of all previous subproblem solutions. That is if $\bar{\mathbf{f}}_\bullet^l$ are subproblem solutions from previous iterations, the main problem solution at iteration $k+1$ is obtained by solving the following multidimensional nonlinear problem:

$$\mathbf{f}_\bullet^{k+1} \in \operatorname{argmin}\left\{T\left(\mathbf{f}_\bullet\right) :\ \mathbf{f}_\bullet \in \operatorname{conv}\left\{\bar{\mathbf{f}}_\bullet^1, \ldots, \bar{\mathbf{f}}_\bullet^k\right\}\right\}$$

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, where $\bar{\mathbf{f}}_\bullet^l$ are still subproblem solutions, except that the subproblems are nonlinear, rather than linear, approximations of the main problem.

All of the above methods are link-based methods; that is, only total link flows aggregated over all O-D pairs must be stored between iterations. More 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 in a sequentially, 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 implemented by Gibert (1968). Bothner and Lutter (1982) implemented a similar route-based method that is used in practice in Germany.

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. Bruynooghe, Gibert and Sakarovitch (1969) made an attempt to develop such a method; however, Gibert (1968) subsequently concluded that the presence of cycles makes origin-based methods quite complicated. As we shall see in chapter 2, cycles and the ways to avoid them have a key role in the method proposed in this thesis.

In the late 70's and early 80's Gallager and Bertsekas developed algorithms for routing in packet-switched communication networks, a problem that is mathematically equivalent to the traffic assignment problem. (Gallager, 1977; Bertsekas, 1979; Bertsekas et al. 1979; Bertsekas et al. 1984; Bertsekas, 1998, pp. 390-391) The main concepts of the algorithm presented here, which was developed independently by the author, are similar to the ones used by Gallager and Bertsekas. These concepts, which we find to be rather involved, are described and explained in detail in chapter 2. The discussion of the work of Gallager and Bertsekas is therefore postponed to section 2.10.

1.5 Route flow solutions

In many cases the information provided by a link-based solution is sufficient. Many global performance measures like the total amount of vehicle miles traveled (VMT), the total amount of time spent on traveling, and the average (space mean) speed on the network, which is the ratio between the two, do not require more than total link flows. The knowledge of the total flow on each and every link also allows local analyses of bottlenecks and related issues.

Nevertheless, link-based solutions have some limitations, as they do not provide a complete picture of the travel pattern. Given total link flows only, obtaining one possible route flow pattern is not a trivial task, let alone determining which route flow pattern best represents reality.

From a theoretical-methodological point of view the absence of an underlying route flow pattern causes several difficulties. The first problem is that there is no direct way to verify feasibility of total link flow solutions, while the feasibility of origin-based

or route-based solutions can be directly verified as demonstrated in section 1.2. In addition, total link flows are not sufficient for the implementation of certain concepts used by some of the advanced algorithms mentioned in section 1.4. Moreover, when traffic assignment is one component of a larger problem, it has to be solved many times for slightly different O-D flows and/or slightly different network conditions. Equilibrium link costs from a previous solution may provide a reasonable initial estimate for link costs under the new conditions. However, in general, equilibrium total link flows from a previous solution can not provide a feasible initial solution under the new conditions, especially if for some O-D pairs the new O-D flows are lower than the previous O-D flows, or if some links from the previous network are omitted in the new network. More detailed solutions, either origin-based or route-based, can be easily adjusted for such changes, keeping the rest of the solution intact. Notice that in all of these theoretical-methodological problems it does not matter which specific origin-based representation or route-based representation is used.

Practitioners may also be interested in the detailed route flow pattern. Rossi et al. (1989) raise the question of consistent impact fee assessment. In this situation, a transportation agency is responsible for determining the anticipated impact of several developments, either residential or commercial, on the traffic pattern in the area. The agency then needs to plan improvement projects to accommodate the increased traffic. Finally the agency needs to determine how much of the cost of each project should be levied on each of the developers. To perform the last task in a consistent fashion the agency needs to know how many travelers from a specific development will enjoy each of the projects, in other words how much of the traffic increase on every link is due to each of the developments. Link-based solutions for the before and after scenarios can only provide the total increase on each link. To obtain the changes in link usage a more detailed solution is required.

Another example for a practical application of the detailed route flow solution is the estimation of vehicle emissions. Simplistic emission models suggest a constant amount of emission per distance traveled. Such simplistic models do not take into account the effect of "cold starts", that is the substantially higher emission rate of engines during the first few minutes of driving, while the engine is still cold. More advanced emission models require the knowledge of the entire trip profile, including acceleration periods and so forth. Static macroscopic models may not provide a fully detailed trip profile; however, the knowledge of route flows is still a major improvement upon total link flows.

The problem of "window" models is another example where total link flows do not provide sufficient information. In this application there is an available transportation model for a large region, and the goal is to construct a model for a smaller area, referred to as a "window", within the larger region (Hearn, 1984). One of the main difficulties in this problem is the representation of external trips, that is trips that

either start or end outside of the window, but use links within the window, thereby affecting the congestion in the area of interest. External flows are represented by entry and exit nodes. Total link flows allow one to determine the total flow from each entry node and to each exit node; however, to determine the distribution of the flows from a specific entry node among the internal destination as well as the exit nodes, the complete route flow pattern in the original regional model is required. See Hearn (1984) for additional discussion.

As mentioned in section 1.3 the user equilibrium condition does not allow one in general to determine the route flow pattern uniquely. In the applications described above practitioners need to know which is the route flow pattern that represents reality in the best way. Rossi et al. (1989) proposed maximum entropy as a criterion for the most likely route flow solution, subject to user equilibrium. Rydergren recently presented an effective dual method to obtain the entropy maximizing solution once equilibrium link flows are known (Larsson et al., 1998). The contribution of our research in this area is presented in chapter 3.

The proposed solution method for the traffic assignment problem is presented in chapter 2. Chapter 3 discusses the bypass proportionality assumption and its relation to the entropy maximizing route flow pattern. Chapter 4 presents computational results of the origin-based solution method for the basic traffic assignment problem on real size practical networks. The conclusions from this research are presented in chapter 5.

2. ORIGIN-BASED ALGORITHM FOR THE TRAFFIC ASSIGNMENT PROBLEM

2.1 Overview

The origin-based solution method for the traffic assignment problem described herein is an iterative method, which considers flows from each origin separately in a sequential fashion. Flows from different origins interact on the network through congestion effects; however, this interaction has a minor roll in the description of the algorithm and the proof of its convergence. To simplify the discussion and the notation we focus on the single origin traffic assignment problem. Nevertheless, the proposed method is perfectly capable of dealing with the regular multiple origin problem as is briefly described towards the end of this section, and further discussed in section 2.9. From now on, unless specified otherwise, we assume that there is only one origin, denoted by p.

One of the main concepts in this algorithm is the separation of the assignment problem into two questions:

1. which links may be used?
2. how much flow should be assigned to each of these links?

A temporary answer to the first question is described by a subnetwork $A_p \subseteq A$, which includes all the links that might be used by flows originating at p. This network is referred to as the *restricting subnetwork*. Once a restricting subnetwork has been determined, we address the second question of determining the flows within this subnetwork, while assuming that the flow on any link outside A_p is zero.

The algorithm therefore consists of two main steps

1. update the restricting subnetwork A_p
2. shift flows within A_p

A key point in this method is that restricting subnetworks are always a-cyclic, i.e. they do not contain any directed cycles. As shown in section 1.3 there is always an a-cyclic equilibrium solution, and if link costs are strictly positive then any equilibrium solution is a-cyclic; however, most solution methods do not maintain the a-cyclic property throughout the iterative process. Maintaining a-cyclic restricting subnetworks ensures that the solution in this method is a-cyclic at every iteration.

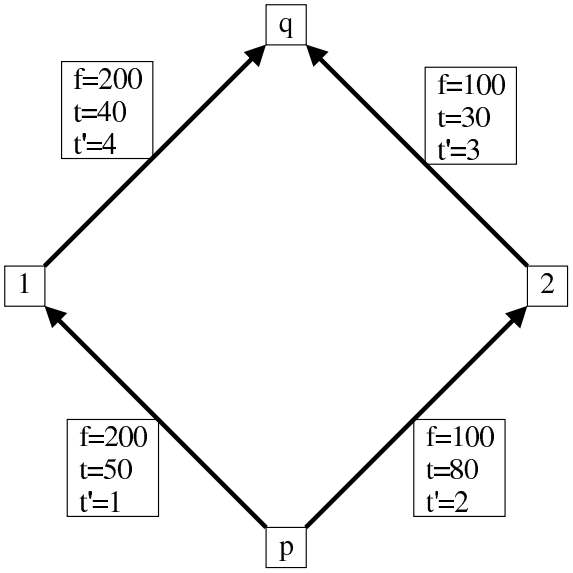
The restriction to a-cyclic solutions has several essential advantages. It permits a simple route flow interpretation. It enables a definition of maximum cost. It also allows for a definition of topological order, i.e. a one to one function $o : N \to \{1, 2, 3, ..., |N|\}$ such that $[i, j] \in A_p \Rightarrow o(i) < o(j)$. It is evident that $o(p) = 1$. The main reason for restricting to a-cyclic solutions is computational efficiency. (The topological order is also used quite intensively in proving convergence, but the problems that it overcomes may not occur if solutions are not restricted to be a-cyclic.) Many computations on a-cyclic networks can be done in a *single pass* over the nodes, either in ascending or descending topological order. In such computations each link a may be considered only when either its head a_h or its tail a_t are considered. As a result, the time required by such a computation is a linear function of the number of links in the network. Most computations in the proposed method belong to this category, as will be pointed out along the way. This is probably one of the main reasons for the excellent convergence performance of this method which will be demonstrated in section 4.1.

In addition to being a-cyclic, we require that the restricting subnetwork be spanning, i.e. that it contains at least one route from the origin to every other node. This requirement is somewhat technical, but still helpful in the definition of maximum cost, average cost, etc.

One possible initial restricting subnetwork is a tree of minimum cost routes under free flow travel conditions. When using such a tree as the restricting subnetwork, there is only one possible assignment, commonly known as the "all or nothing assignment".

To update the restricting subnetwork, first unused links are removed, then the maximum cost u_i among all 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. It will be shown that the new restricting subnetwork is spanning, a-cyclic, and contains the current solution. (Comment: the terms *unused* links and *maximum* cost were used here in an intuitive imprecise fashion. Precise definitions are given in section 2.8.) The way flows are shifted within the given restricting subnetwork is more complicated to explain, and is demonstrated here schematically using examples.

Figure 1 shows an extremely simple network with one origin, one destination and two routes. The current flows and costs indicated suggest that flows should be shifted



f-flow; t-cost; t'-cost derivative

Figure 1. Flow shift between simple alternatives

from the right route to the left route; the more difficult question is to determine how much flow to shift. First-order optimization methods suggest using the negative gradient as a search direction, and a line search to determine how much to move along that search direction. In section 2.4 we show that the first-order constrained derivative (reduced gradient) corresponds in this case to the difference between route costs, and the resulting search direction agrees with intuition. On the other hand, the cost difference by itself does not provide any reasonable basis to determine how much flow to shift. In fact there is a clear mismatch between the units of cost, say minutes, and the units of flow, vehicles per hour (vph). This units mismatch between the first-order derivative and the variable space is often ignored in the optimization literature. To overcome this mismatch the gradient should be divided (or multiplied) by some conversion factor. Using the gradient as it is actually implies that the gradient is divided by a conversion factor of 1 minute/vph. If we change cost units either to seconds or to hours, the same logic may lead us to use conversion factors of 1 second/vph or 1 hour/vph respectively. These are clearly substantially different conversion factors.

The importance of more appropriate conversion factors depends on the circumstances. In a fully sequential algorithm each flow shift is considered separately; hence a better conversion factor may provide a better starting point for the line search, but not much more than that. To better utilize the advantages associated with efficient computations on a-cyclic networks, one may wish to consider all flow shifts for a

single origin simultaneously. In that case a multi-dimensional search direction is needed. If the same conversion factor is used in all dimensions, then the direction remains the same, and the benefit is again mainly a better starting point for the line search. However, it is often more appropriate to use different conversion factors for different components of the gradient. Using such factors may yield a fairly different, and hopefully better search direction.

In the following discussion we provide some intuition for the conversion factors used in the proposed algorithm. Consider Figure 1 again. Since both link costs and cost derivatives are known, we can estimate how much flow δ should be shifted from the right route to the left route in order to equalize the costs along both routes. By first order approximations route costs are linear functions of δ

$$\begin{aligned} c_r &\approx (80+30)_{[minutes]} - \delta_{[vph]} \cdot (2+3)_{[minutes/vph]} \\ c_l &\approx (50+40)_{[minutes]} + \delta_{[vph]} \cdot (1+4)_{[minutes/vph]} \end{aligned} \tag{2.1}$$

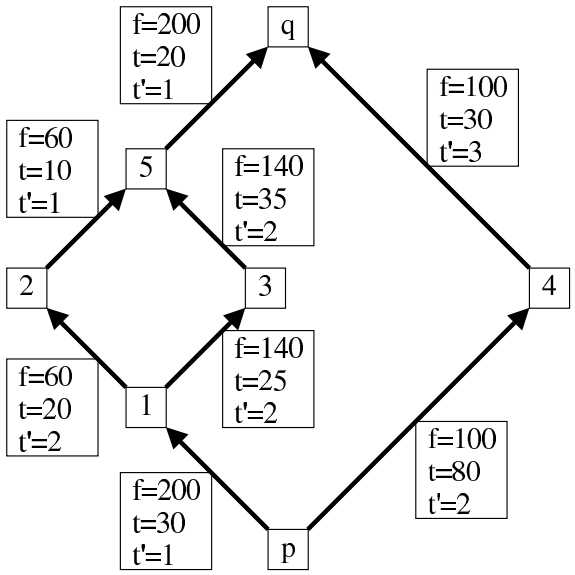
and they become equal for

$$\delta = \frac{(80+30)-(50+40)_{[minutes]}}{(2+3)+(1+4)_{[minutes/vph]}} = 2_{[vph]}$$

This derivation suggests that in this example the appropriate conversion factor is the sum of route cost derivatives. Notice that units in this case do match.

Figure 2 shows a slightly more complicated network, still with one origin and one destination, but this time the left alternative consists of two routes: $[p,1,2,5,q]$ and $[p,1,3,5,q]$. The average cost incurred by travelers shifted to the left alternative depends on the way that these travelers are distributed among the two routes. We assume that the new travelers are distributed by the same proportions as the current travelers, i.e. 30% use route $[p,1,2,5,q]$ incurring a cost of $80_{[minutes]}$ and 70% use route $[p,1,3,5,q]$ incurring a cost of $110_{[minutes]}$. Therefore the average cost of the left alternative is $0.3 \cdot 80 + 0.7 \cdot 110 = 101_{[minutes]}$. As discussed later, there are efficient ways to compute these average costs, which are basic elements of the proposed algorithm.

Assuming that the shifted travelers are distributed by the same proportions as the current travelers, we can also find the linear first order approximation of the new route costs and average costs. Suppose that δ vph (vehicles per hour) are shifted from the right alternative to the left alternative. This shift results in a decrease of δ vph on links $[p,4],[4,q]$, an increase of δ vph on links $[p,1],[5,q]$, an increase of 0.3δ



f-flow; t-cost; t'-cost derivative

Figure 2. Flow shift between composite alternatives

vph on links $[1,2],[2,5]$ and an increase of 0.7δ vph on links $[1,3],[3,5]$. By linear approximation the link costs are (units are omitted for simplicity)

$$\begin{aligned} t_{[p,4]} &\approx 80 - 2\cdot\delta & t_{[1,2]} &\approx 20 + 2\cdot 0.3\delta \\ t_{[4,q]} &\approx 30 - 3\cdot\delta & t_{[2,5]} &\approx 10 + 1\cdot 0.3\delta \\ t_{[p,1]} &\approx 30 + 1\cdot\delta & t_{[1,3]} &\approx 25 + 2\cdot 0.7\delta \\ t_{[5,q]} &\approx 20 + 1\cdot\delta & t_{[3,5]} &\approx 35 + 2\cdot 0.7\delta \end{aligned}$$

the route costs are

$$\begin{aligned} c_{[p,4,q]} &\approx (80+30) - (2+3)\cdot\delta \\ c_{[p,1,2,5,q]} &\approx (30+20+10+20) + (1+0.3\cdot 2+0.3\cdot 1+1)\cdot\delta \\ c_{[p,1,3,5,q]} &\approx (30+25+35+20) + (1+0.7\cdot 2+0.7\cdot 2+1)\cdot\delta \end{aligned}$$

and the average costs are

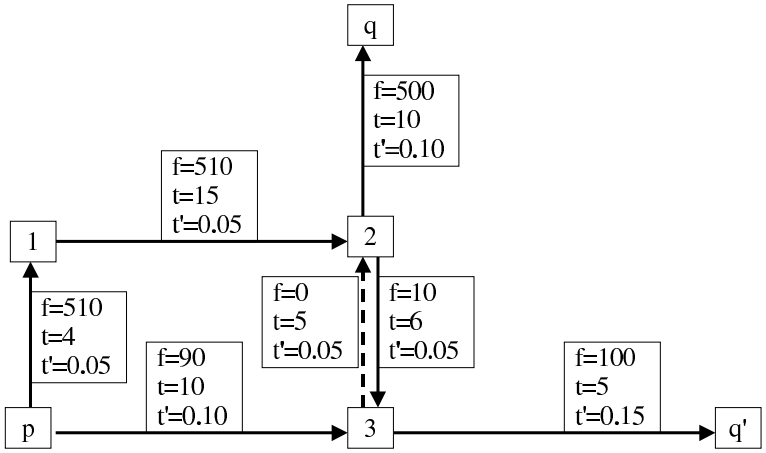
$$\begin{aligned} c_r &= c_{[p,4,q]} \approx (80+30) - (2+3)\cdot\delta \\ c_l &= 0.3\cdot c_{[p,1,2,5,q]} + 0.7\cdot c_{[p,1,3,5,q]} \\ &\approx 0.3\cdot(30+20+10+20) + 0.7\cdot(30+25+35+20) \\ &\quad + \left[(0.3+0.7)\cdot 1 + 0.3^2\cdot 2 + 0.3^2\cdot 1 + 0.7^2\cdot 2 + 0.7^2\cdot 2 + (0.3+0.7)\cdot 1\right]\cdot\delta \end{aligned}$$

To equalize the average costs of both alternatives the flow shift should be equal to the difference between average costs divided by some, rather complicated, conversion factor. Computing conversion factors in this way may become even more complicated as the network grows, and as interactions between the various routes become more complex; for example, adding link $[3,4]$ to the above figure makes this computation much more complicated. To avoid this complication the proposed algorithm uses an approximation, which is described in detail in section 2.5, that can be computed rather efficiently. The ratio between average cost difference and the approximated conversion factor is considered as the desirable flow shift.

The accurate conversion factors described above correspond, not suprisingly, to objective function second-order constrained derivatives, i.e. to the diagonal elements of the Hessian matrix. The desirable flow shift can therefore be viewed as a second-order Newton-type correction, i.e. as a product of an approximation of the inverse Hessian and the gradient. Like any other second-order method the units of the resulting search direction always match the variable units.

Second-order methods are known to provide better convergence rate in terms of accuracy improvement per iteration at the neighborhood of the optimum (equilibrium) point. On the other hand, the overhead associated with inverse Hessian computation often prohibits pure second-order methods, and in many cases even approximated second-order methods are known to be inferior to first-order methods in the initial stages of the optimization. In the proposed algorithm the time required to compute conversion factors, i.e. the approximation of the inverse Hessian, is similar to the time required to compute average costs, i.e. first-order derivatives. This amounts to probably around one quarter of the overall computation time (this last estimate is fairly crude and does not rely on any quantitative assessment). The computational results in chapter 4 suggest that this is a reasonable tradeoff between computation headover and search direction effectiveness.

In the discussion of desirable flow shifts so far we ignored feasibility constraints. These desirable flow shifts may very well lead to violation of the non-negativity constraints, and hence they must be truncated. The aggregation of truncated shifts can be used as a search direction, which may then be scaled by a step size to obtain the optimal point along that search direction. This regular convex line search procedure tends to lead to residual flows that reduce the effectiveness of the restrictions update procedure, and may in fact prevent convergence. Section 2.7 describes an alternative boundary search procedure in which the desirable flow shifts are first scaled by the step size, and only then truncated if necessary and aggregated to obtain a new solution. This procedure is shown to be effective in eliminating residual flows. We prove that the incorporation of boundary search in the proposed method guarantees global convergence.



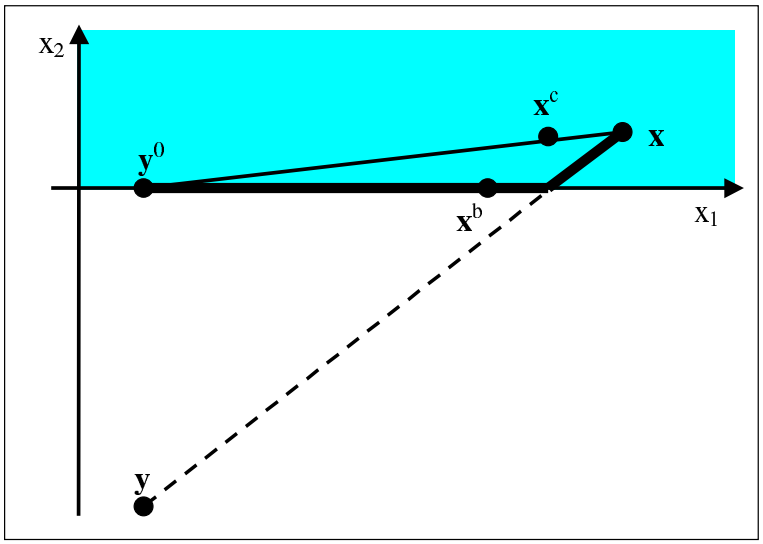
f-flow; t-cost; t'-cost derivative

Figure 3. Residual flow

To better understand the difference between the convex and the boundary search procedures, consider the situation in Figure 3. In this figure solid lines represent the current restricting subnetwork. It is evident that the minimum cost route to destination q includes the dashed link from 3 to 2. The restrictions update procedure does not add this link to the network because the maximum cost to node 3 is 25, which is more than 19, the maximum cost to node 2. In fact, adding the dashed link [3,2] creates a cycle with the solid link [2,3]. This, then, is an essential issue and not just a technical problem with this specific restrictions update procedure.

In order to add link [3,2] without creating a cycle, one must first eliminate the flow on the link [2,3]. In this situation shifting the flow of 10 vph from route $[p, 1, 2, 3, q']$ to route $[p, 3, q']$ seems a fairly reasonable thing to do. The desirable shift computed by the Newton-type technique described above suggests a shift of $(25-10)/(0.15+0.10) = 60$ vph. This computation of the desirable shift does not take feasibility into account; as a result the desirable shift in this case is higher than the maximum feasible shift of 10 vph. To ensure feasibility the convex line search procedure suggests considering only a shift of 10 vph, and aggregating it together with other shifts into a feasible search direction. Then a step size, i.e. a scaling factor, that minimizes the objective function along that search direction is chosen. If the chosen step size is 0.25, the result is to shift 2.5 vph and leave 7.5 vph on link [2,3]. No matter how many iterations are performed, unless a step size of 1.0 is chosen in one of these iterations, some residual flow would always remain on link [2,3].

In the boundary search, if a step size of 0.25 is considered, the desirable shift of 60 vph is first scaled by 0.25 resulting in a shift of 15 vph, which is only then truncated



x-current solution; **y**-estimated minimum; $\mathbf{y}^0$- projection of **y**
$\mathbf{x}^c$-convex search new solution; $\mathbf{x}^b$-boundary search new solution

Figure 4. Boundary search

if necessary, and finally aggregated with other shifts. As a result the flow on link [2,3] is eliminated with any step size between 0.167 and 1.0. In addition, even a smaller step size of say 0.1 leads to a shift of 6 vph, which is much more substantial than a shift of 1 vph in the case of the convex search.

Figure 4 provides another attempt to describe the basic ideas behind the boundary search in the context of a general two-dimensional convex minimization problem, where non-negativity of the variables are the only constraints. In this figure the current solution is denoted by $\mathbf{x}$, the estimated minimum point is denoted by $\mathbf{y}$. Its projection onto the feasible (non-negative) region is denoted by $\mathbf{y}^o$. The thin line describes the convex line search, and $\mathbf{x}^c$ denotes the chosen solution along that line. The dotted line describes the unconstrained line search, and the thick line which is the projection of that line on the feasible set describes the boundary search. $\mathbf{x}^b$ denotes the chosen point for the boundary search. As implied from its name, the boundary search is much more likely to find boundary solutions, although it does consider inner points as well. The application of these ideas to our problem is not straightforward. Projection is done in the decision variables space, the non-basic approach proportions, which have not been defined yet. This projection affects link flows in a non-linear complex fashion. Convex combinations are meaningful mainly in the link flow space. A precise description of the boundary search is given in section 2.7.

In general the broken line describing the boundary search travels along facets of the feasible polyhedron. As such, the number of segments in this broken line can be as high as the number of dimensions in the problem. Determining the vertices of this broken line may therefore be a rather demanding computational task. Fortunately, it is not necessary to perform such a task in the proposed method. The method evaluates step sizes of 2^{-k} for $k = 0, 1, 2, 3, \ldots$ one by one, and stops once a certain condition is satisfied which guarantees descent of the objective function. Therefore it is sufficient to produce the point along the search line that corresponds to each step size as needed. The computation required to obtain one such point, i.e. to scale the unconstrained shifts, then truncate them if necessary, and finally aggregate them to obtain a new solution is similar to the computation required for a single assignment procedure. If all origins are considered simultaneously, and the same step is applied to all of them, the computational effort to evaluate one possible step size value is on the order of the number of origins times the number of links. In comparison, the convex search requires a similar effort to obtain the feasible search direction, i.e. the point for step size 1.0; however, evaluation of additional step size values is done by averaging the two vectors of total link flows, which requires a substantially lower computational effort. In the proposed method flows from each origin are considered separately in a sequential fashion, and a different step size is chosen and applied to each of them separately. In that context the time required to evaluate additional step size values is a linear function of the number of links for both the convex search and the boundary search.

Building on the two main steps - update restricting subnetwork, and shift flows, an algorithm for the single origin problem may be described as follows:

```
Find initial solution
Until convergence
    update restrictions
    shift flows
```

Updating the restrictions involves a substantial amount of data structure reorganization; therefore, this step typically requires much more computational effort than shifting flows. Therefore, it may be useful to perform several flow shift iterations after every update of the restrictions.

To apply this method to the multiple origin problem, one could perform these steps for each origin in a sequential fashion. There are several different ways to organize these steps. The results reported here were obtained using a procedure of the following form:

Find initial solution
Until convergence
    Perform a full iteration:
        for every origin p
            update restricting subnetwork A_p
            shift flows within A_p
    Perform x quick iterations:
        for every origin p
            shift flows within the current restricting subnetwork A_p

The remainder of this chapter is organized as follows. Sections 2.3-2.7 describe the details of the flow shift step, in the context of the restricted single origin traffic assignment problem, which is formally introduced in section 2.2. Approach proportions are proposed as solution variables for this problem, and their properties are examined in section 2.3. Optimality conditions are derived in section 2.4. Section 2.5 discusses second order derivatives, the difficulties in computing them accurately, and motivates the approximation to be used thereafter. Algorithmic maps for flow shifts and their aggregation are presented in section 2.6. The boundary search is described in section 2.7 followed by a proof of convergence of the resulting iterative method for the restricted single origin problem. Section 2.8 deals with the unrestricted single origin problem, the procedure for updating the restrictions is defined precisely, and a proof of convergence for the resulting method is given. Finally, in section 2.9 the complete multiple origin traffic assignment problem is considered, and a method for its solution is presented.

2.2 Restricted single origin traffic assignment problem

Throughout most of this chapter we assume that there is only one origin p. To simplify the notation we therefore omit the p index from almost all variables. In this context f_a denotes the origin-based link flow, and d_j denotes the O-D flow (demand) from origin p to node j, which is defined for every node $j \in N$, even though it might be zero for most of them.

It should be noted that 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 the next few sections we further assume that the solution is restricted by some given subnetwork, $A_p \subseteq A$, which is an a-cyclic spanning subnetwork rooted at p, i.e. it does not contain any directed cycles and it contains at least one route from p to every node $j \in N$. In particular it can not contain any link terminating at p, and the only route from p to itself is the empty route $[p]$. The set of route segments from node i to node j that are included in the subnetwork A_p is denoted by $R_{ij}[A_p] = \{r \in R_{ij} : a \subseteq r \Rightarrow a \in A_p\}$.

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

$$\begin{aligned} \text{[RSOTAP]} \quad \min \quad & T = \sum_{a \in A} \int_0^{f_a} t_a(x)\mathrm{d}x \\ s.t. \quad & \sum_{r \in R_{pj}[A_p]} h_{rpj} = d_j \quad \forall j \in N \end{aligned} \tag{2.2}$$

2.3 Approach proportions

For reasons that are discussed later, we prefer to use *approach proportions* as solution variables. The relationships between origin-based link flows, route flows, and approach proportions are studied in chapter 3. The derivation in that chapter also provides some motivation for the following definitions. Except for this motivation, the following construction is completely self contained.

Suppose $\boldsymbol{\alpha} = (\alpha_a)_{a \in A}$ is a vector that satisfies:

$$\sum_{a \in A_p; a_h = j} \alpha_a = 1 \qquad \forall j \in N; j \neq p \tag{2.3a}$$

$$\alpha_a = 0 \qquad \forall a \notin A_p \tag{2.3b}$$

$$\boldsymbol{\alpha} \geq 0 \tag{2.3c}$$

Since A_p is spanning, there is at least one link terminating at every node other than the origin, and therefore the above conditions can be satisfied.

Consider the following formal expression

$$\chi_{i\to j} = \sum_{s\in R_{ij}[A_p]} \left(\prod_{a\subseteq s} \alpha_a \right) \tag{2.4}$$

An intuitive interpretation for this value is provided by (2.16), once some properties of this expression are explored.

Lemma 2. *If* $\boldsymbol{\alpha}$ *satisfies feasibility requirements* (2.3) *then*

$$\chi_{p\to j} = 1 \quad \forall j \in N \tag{2.5}$$

Proof:
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 1 by definition. Suppose the statement is true for all nodes of lower topological order; in particular, it holds for all predecessors of j, that is $\forall a \in A_p; a_h = j \quad \chi_{p\to a_t} = 1$ and therefore:

$$\begin{aligned} \chi_{p\to j} &= \sum_{s\in R_{pj}[A_p]} \prod_{a\subseteq s} \alpha_a = \sum_{a\in A_p; a_h=j} \alpha_a \cdot \left(\sum_{s'\in R_{pa_t}[A_p]} \prod_{a'\subseteq s'} \alpha_{a'} \right) \\ &= \sum_{a\in A_p; a_h=j} \alpha_a \cdot \chi_{p\to a_t} = \sum_{a\in A_p; a_h=j} \alpha_a = 1 \end{aligned} \tag{2.6}$$

where the last equality is feasibility requirement (2.3a). □

Notice that

$$\chi_{p\to i} \cdot \chi_{i\to j} = \sum_{s\in R_{pj}[A_p]; i\in s} \prod_{a\subseteq s} \alpha_a \le \sum_{s\in R_{pj}[A_p]} \prod_{a\subseteq s} \alpha_a = \chi_{p\to j} \tag{2.7}$$

therefore, by lemma (2)

$$0 \le \chi_{i\to j} \le 1 \quad \forall i, j \in N \tag{2.8}$$

Given a feasible vector $\boldsymbol{\alpha}$, consider the following route flows:

$$h_{rpq}(\boldsymbol{\alpha}) = d_q \cdot \prod_{a\subseteq r} \alpha_a \quad \forall r \in R_{pq} \tag{2.9}$$

If route r contains a link $a \notin A_p$ then by (2.3b) $\alpha_a = 0$ hence $h_{rpq}(\boldsymbol{\alpha}) = 0$. Summing route flows over R_{pq} or over $R_{pq}[A_p]$ is therefore the same. These route flows represent a feasible assignment since by lemma (2)

$$\sum_{r \in R_{pq}[A_p]} h_{rpq}(\boldsymbol{\alpha}) = \chi_{p \to q} \cdot d_q = d_q \tag{2.10}$$

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

$$g_s(\boldsymbol{\alpha}) = \sum_{q \in N} \sum_{r \in R_{pq}[A_p]: s \subseteq r} h_{rpq}(\boldsymbol{\alpha}) \tag{2.11}$$

Notice that even if $s \in R_{pq}$, it is possible that $g_s(\boldsymbol{\alpha}) \neq h_{spq}(\boldsymbol{\alpha})$ since there may be flows that use the route segment s and then continue to some other destination. These flows are included in the origin-based segment flow g_s, but not in the route flow h_{spq}. A special case of origin-based segment flow for s=[j] is the *origin-based node flow* $g_{[j]} = g_j$, which is the aggregation of all flows that originate at p and arrive at j, either on their way to another destination, or stop at j, if j is their destination.

$$g_j(\boldsymbol{\alpha}) = \sum_{q \in N} \sum_{r \in R_{pq}[A_p]: j \in r} h_{rpq}(\boldsymbol{\alpha}) = \sum_{q \in N} \chi_{p \to j} \cdot \chi_{j \to q} \cdot d_q = \sum_{q \in N} \chi_{j \to q} \cdot d_q \tag{2.12}$$

For a general segment

$$g_s(\boldsymbol{\alpha}) = \sum_{q \in N} \sum_{r \in R_{pq}[A_p]; s \subseteq r} h_{rpq}(\boldsymbol{\alpha}) = \sum_{q \in N} \chi_{p \to s_t} \cdot \prod_{a \subseteq s} \alpha_a \cdot \chi_{s_h \to q} \cdot d_q = \prod_{a \subseteq s} \alpha_a \cdot g_{s_h}(\boldsymbol{\alpha}) \tag{2.13}$$

where s_t, s_h are the first (tail) and last (head) nodes of route segment s respectively. The *origin-based link flow* may be viewed as a special case of origin-based segment flow for $s = a$

$$f_a(\boldsymbol{\alpha}) = g_a(\boldsymbol{\alpha}) = \alpha_a \cdot g_{a_h}(\boldsymbol{\alpha}) \tag{2.14}$$

For every node except the origin,

$$\sum_{a \in A; a_h = j} f_a = \sum_{a \in A; a_h = j} \alpha_a \cdot g_j = g_j \quad \forall j \in N; j \neq p \tag{2.15}$$

We refer to the collection of route segments from p to j that include link a where $a_h = j$ as the *a approach* to j. α_a is therefore interpreted as the *approach proportion*. Using (2.14) one may obtain approach proportions from origin-based link flows, but only if the origin-based node flow is strictly positive. The advantages of having

approach proportions even when origin-based node flow is zero may not be apparent at this point, and are discussed later.

A similar interpretation for $\chi_{i\to j}$ is obtained by considering the total origin-based flow on all route segments that pass through node i on their way to node j; that is

$$\begin{aligned} g_{[i,*,j]}(\boldsymbol{\alpha}) &= \sum_{s\in R_{ij}[A_p]} g_s(\boldsymbol{\alpha}) = \sum_{s\in R_{ij}[A_p]} \sum_{q\in N} \sum_{r\in R_{pq}[A_p];s\subseteq r} h_{rpq}(\boldsymbol{\alpha}) \\ &= \sum_{q\in N} \chi_{p\to i}\cdot\chi_{i\to j}\cdot\chi_{j\to q}\cdot d_q = \chi_{i\to j}\cdot g_j \end{aligned} \tag{2.16}$$

$\chi_{i\to j}$ is therefore the proportion of flow that arrives at node j through node i.

Our next goal is to find efficient ways of computing origin-based link flows and origin-based node flows from the vector of approach proportions $\boldsymbol{\alpha}$. For that purpose we use the following descending recursive equation for χ

$$\chi_{i\to j} = \begin{cases} 1 & i = j \\ 0 & o(i) > o(j) \\ \sum\limits_{a\in A_p;a_t=i} \alpha_a\cdot\chi_{a_h\to j} & o(i) < o(j) \end{cases} \tag{2.17}$$

Notice that if $o(i) > o(j)$ then $R_{ij}[A_p] = \emptyset$ and hence $\chi_{i\to j} = 0$. There is also an ascending recursive equation for χ,

$$\chi_{i\to j} = \begin{cases} 1 & i = j \\ 0 & o(i) > o(j) \\ \sum\limits_{a\in A_p;a_h=j} \chi_{i\to a_t}\cdot\alpha_a & o(i) < o(j) \end{cases} \tag{2.18}$$

which is used later on. Using (2.17) we find that

$$\begin{aligned} g_j &= \sum_{q\in N} \chi_{j\to q}\cdot d_q = d_j + \sum_{q\in N;o(j)<o(q)} \left(\sum_{a\in A_p;a_t=j} \alpha_a\cdot\chi_{a_h\to q} \right)\cdot d_q \\ &= d_j + \sum_{a\in A_p;a_t=j} \alpha_a\cdot\left(\sum_{q\in N} \chi_{a_h\to q}\cdot d_q \right) = d_j + \sum_{a\in A_p;a_t=j} \alpha_a\cdot g_{a_h} \\ &= d_j + \sum_{a\in A_p;a_t=j} f_a \end{aligned} \tag{2.19}$$

Equation (2.19) together with (2.15) shows that

$$\sum_{a\in A;a_h=j} f_a = g_j = d_j + \sum_{a\in A_p;a_t=j} f_a \quad \forall j \in N; j \neq p \tag{2.20}$$

which is the common flow conservation requirement for origin-based link flows. Equations (2.19) and (2.14) allow one to compute origin-based link flows and origin-based node flows from $\boldsymbol{\alpha}$ in a single descending pass over the nodes.

2.4 Optimality conditions

$\mathbf{f}(\boldsymbol{\alpha}) : [0,1]^{|A|} \to \mathbb{R}^{|A|}$ is a function from the restricted feasible set of approach proportions onto the restricted feasible set of origin-based link flows. Minimizing $T(\mathbf{f})$ over the latter is equivalent to minimizing $T(\mathbf{f}(\boldsymbol{\alpha}))$ over the prior. Under the monotonicity and separability conditions assumed above, it is known that $T(\mathbf{f})$ is convex as a function of $\mathbf{f}$. As is shown below, in general $T(\mathbf{f}(\boldsymbol{\alpha}))$ is not convex as a function of $\boldsymbol{\alpha}$.

One common technique for dealing with such constrained optimization problems is to form a Lagrangian. Unfortunately, in this case a direct Lagrangian derivation was not found to be helpful, and the Lagrange multipliers do not seem to have any meaningful interpretation. An alternative method is to choose a set of basic variables whose values are obtained from the non-basic variables using the constraints. In our case, once A_p has been determined, the constraints are easily decomposed into a Cartesian product by head node. For each head node j 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 non-basic approaches $NB_j = \{a \in A_p; a_h = j; a \neq b_j\}$; $NB = \bigcup_{j\in N} NB_j$. Feasibility constraint (2.3a) is then replaced by

$$\alpha_{b_j} = 1 - \sum_{a\in NB_j} \alpha_a \qquad \forall j \in N \setminus \{p\} \tag{2.21}$$

The set of all basic approaches $B = \{b_j : j \in N; j \neq p\} \subseteq A_p$ is a-cyclic, spanning, and contains one predecessor for every node (other than the origin). B is therefore a spanning tree, referred to as the *basic tree*.

From definitions one can easily verify that $\mathbf{h}, \mathbf{f}, \mathbf{t}, T$ are all continuously differentiable functions of $\boldsymbol{\alpha}$. When taking the partial derivatives of a general function of the approach proportions $x(\boldsymbol{\alpha})$ we should distinguish between two possible interpretations:

1. Ignore feasibility constraint (2.21), and assume that approach proportions are completely independent. This unconstrained derivative is denoted by the usual derivative notation $\frac{\partial x}{\partial \alpha_a}$.

2. Consider constraint (2.21). The constrained derivative is denoted by $\frac{\partial x}{\partial^c \alpha_a}$. Notice that

$$\frac{\partial x}{\partial^c \alpha_a} = \frac{\partial x}{\partial \alpha_a} - \frac{\partial x}{\partial \alpha_{b_{(a_h)}}} \tag{2.22}$$

The goal of this section is to derive the constrained first order optimality conditions. We start with the unconstrained derivative of route flow

$$\frac{\partial h_{rpq}}{\partial \alpha_a} = \delta_{ra} \cdot d_q \cdot \prod_{a' \subseteq r; a' \neq a} \alpha_{a'} \tag{2.23}$$

The unconstrained derivative of the origin-based link flow is

$$\frac{\partial f_{a'}}{\partial \alpha_a} = \sum_{q \in N} \sum_{r \in R_{pq}[A_p]; a' \subseteq r} \frac{\partial h_{rpq}}{\partial \alpha_a} \tag{2.24}$$

which we evaluate for four cases:

1. $a = a'$

$$\begin{aligned} \frac{\partial f_a}{\partial \alpha_a} &= \sum_{q \in N} \sum_{r \in R_{pq}[A_p]; a \subseteq r} d_q \cdot \prod_{a'' \subseteq r; a'' \neq a} \alpha_{a''} \\ &= \sum_{q \in N} \chi_{p \to a_t} \cdot d_{pq} \cdot \chi_{a_h \to q} = g_{a_h} \end{aligned} \tag{2.25}$$

2. $o(a_h) = o(a'_h); a' \neq a$

$$\{r \in R_{pq}[A_p]; a \subseteq r; a' \subseteq r\} = \emptyset \quad \Longrightarrow \quad \frac{\partial f_{a'}}{\partial \alpha_a} = 0 \tag{2.26}$$

3. $o(a'_h) < o(a_h)$

$$\begin{aligned}\frac{\partial f_{a'}}{\partial \alpha_a} &= \sum_{q \in N} \sum_{r \in R_{pq}[A_p]; a \subseteq r; a' \subseteq r} d_q \cdot \prod_{a'' \subseteq r; a'' \neq a} \alpha_{a''} \\ &= \sum_{q \in N} d_q \cdot \chi_{p \to a'_t} \cdot \alpha_{a'} \cdot \chi_{a'_h \to a_t} \cdot \chi_{a_h \to q} = \alpha_{a'} \cdot \chi_{a'_h \to a_t} \cdot g_{a_h} \end{aligned} \quad (2.27)$$

4. $o(a_h) < o(a'_h)$

$$\begin{aligned}\frac{\partial f_{a'}}{\partial \alpha_a} &= \sum_{q \in N} \sum_{r \in R_{pq}[A_p]; a \subseteq r; a' \subseteq r} d_q \cdot \prod_{a'' \subseteq r; a'' \neq a} \alpha_{a''} \\ &= \sum_{q \in N} d_q \cdot \chi_{p \to a_t} \cdot \chi_{a_h \to a'_t} \cdot \alpha_{a'} \cdot \chi_{a'_h \to q} = \alpha_{a'} \cdot \chi_{a_h \to a'_t} \cdot g_{a'_h} \end{aligned} \quad (2.28)$$

To find the constrained derivatives with respect to the approach proportion of some non-basic approach a, denote the head node by $j = a_h$, its basic approach $b = b_j = b_{(a_h)}$, and consider the same four cases:

1. $a' = a$ or $a' = b$

$$\frac{\partial f_a}{\partial^c \alpha_a} = g_j \quad (2.29)$$

$$\frac{\partial f_b}{\partial^c \alpha_a} = -g_j \quad (2.30)$$

2. $o(a_h) = o(a'_h); a' \neq a; a' \neq b$

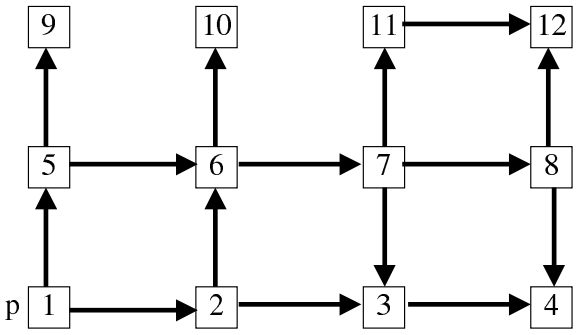
$$\frac{\partial f_{a'}}{\partial^c \alpha_a} = 0 \quad (2.31)$$

3. $o(a'_h) < o(a_h)$

$$\frac{\partial f_{a'}}{\partial^c \alpha_a} = \alpha_{a'} \cdot g_j \cdot \left(\chi_{a'_h \to a_t} - \chi_{a'_h \to b_t}\right) \quad (2.32)$$

4. $o(a_h) < o(a'_h)$

$$\frac{\partial f_{a'}}{\partial^c \alpha_a} = \alpha_{a'} \cdot g_{a'_h} \cdot \left(\chi_{a_h \to a'_t} - \chi_{b_h \to a'_t}\right) = 0 \quad (2.33)$$



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

Figure 5. Common nodes and last common nodes in an a-cyclic network

where the last equality is due to the fact that $b_h = a_h$. Intuitively, approach proportions affect the distribution of flows towards the origin; indeed, the zero partial derivative in case 4 shows that shifting proportions between approaches to a certain node does not impact flows on succeeding links. This fact is not easily captured in a direct Lagrangian derivation, hence complicating such derivation.

Case 3, of course, is the interesting one. In some cases it is possible to demonstrate that $\chi_{a'_h \to a_t} = \chi_{a'_h \to b_t}$ and hence the constrained partial derivative is zero even in this case. Suppose that node i is common to all routes that arrive at node j; that is $i \in s \quad \forall s \in R_{pj}[A_p]$, 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 $o(i') \leq o(i)$ 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.

Definition: the *last common node* of node j in A_p, lcn_j is the common node of highest topological order, excluding j itself.

Some examples for common nodes and last common nodes in an a-cyclic restricting subnetwork are given in Figure 5.

If A_p contains only one link terminating at j, that is j is a single termination node, then the last common node of j is the predecessor node. Otherwise, if j is a multiple termination node, i.e. there are at least two links terminating at j, then lcn_j must

come prior to all of those links, and can not be the tail of any one of them. In both cases lcn_j is a common node to all predecessor nodes of j, that is to all nodes i such that $[i,j] \in A_p$.

Let us now reconsider the constrained partial derivative in case 3 if in addition we assume that $o(a'_h) \leq o(lcn_{(a_h)})$. In this case $l = lcn_{(a_h)}$ is also a common node to a_t and to b_t; therefore $\chi_{a'_h \to a_t} = \chi_{a'_h \to b_t} = \chi_{a'_h \to l}$ hence $\frac{\partial f_{a'}}{\partial^c \alpha_a} = 0$.

In conclusion, the effect of an approach proportion is limited to the portion of the network connecting the last common node to the approach termination node.

We are now ready to derive first order optimality conditions for an approach proportion of a specific non-basic approach a. Denote the termination node by $j = a_h$, and its basic approach $b = b_j = b_{(a_h)}$.

$$\begin{aligned}\frac{\partial T}{\partial^c \alpha_a} &= \sum_{a' \in A_p} \frac{\partial T}{\partial f_{a'}} \cdot \frac{\partial f_{a'}}{\partial^c \alpha_a} \\ &= t_a \cdot g_j - t_b \cdot g_j + \sum_{a' \in A_p; o(a'_h) < o(j)} t_{a'} \cdot \alpha_{a'} \cdot g_j \cdot \left(\chi_{a'_h \to a_t} - \chi_{a'_h \to b_t}\right) \end{aligned} \tag{2.34}$$

At a first glance the expression for the first order derivative (2.34) may not reveal any intuitive interpretation. In the following paragraphs we examine the average cost to a node, and the approach average cost. This derivation eventually leads to a rather simple and intuitive expression for the first order derivative given in (2.42).

Consider the average cost weighted by flow over all route segments from the origin p to node j, which we evaluate using (2.13).

$$\frac{\sum_{r \in R_{pj}[A_p]} g_r \cdot c_r}{\sum_{r \in R_{pj}[A_p]} g_r} = \frac{\sum_{r \in R_{pj}[A_p]} \prod_{a \subseteq r} \alpha_a \cdot g_j \cdot c_r}{g_j} = \sum_{r \in R_{pj}[A_p]} c_r \cdot \prod_{a \subseteq r} \alpha_a \tag{2.35}$$

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. This is one case where using approach proportions simplifies the discussion. Following this derivation we define σ_j as the average cost to node j by

$$\sigma_j = \sum_{r \in R_{pj}[A_p]} c_r \cdot \prod_{a \subseteq r} \alpha_a \tag{2.36}$$

The $a = [i, j]$ approach was defined as the set of all route segments from the origin p to node j that use link a. A similar derivation for flow-weighted average approach cost is given by

$$\frac{\sum_{r \in R_{pj}[A_p]; a \subseteq r} g_r \cdot c_r}{\sum_{r \in R_{pj}[A_p]; a \subseteq r} g_r} = \frac{\sum_{r \in R_{pj}[A_p]; a \subseteq r} \prod_{a' \subseteq r} \alpha_{a'} \cdot g_j \cdot c_r}{f_a} = \sum_{\substack{r \in R_{pj}[A_p] \\ a \subseteq r}} c_r \cdot \prod_{\substack{a' \subseteq r \\ a' \neq a}} \alpha_{a'} \tag{2.37}$$

leads to the definition of *average approach cost* μ_a as

$$\mu_a = \sum_{r \in R_{pj}[A_p]; a \subseteq r} c_r \cdot \prod_{a' \subseteq r; a' \neq a} \alpha_{a'} \tag{2.38}$$

One can verify the following recursive equations

$$\mu_a = t_a + \sigma_{a_t} \tag{2.39}$$

$$\sigma_j = \sum_{a \in A_p; a_h = j} \alpha_a \cdot \mu_a \tag{2.40}$$

which allow one to compute these average cost values for all nodes and all approaches in a single ascending pass over the network. Finally

$$\begin{aligned}
\mu_a &= t_a + \sigma_{a_t} \\
&= t_a + \sum_{r \in R_{pa_t}[A_p]} \prod_{a'' \subseteq r} \alpha_{a''} \cdot \sum_{a' \subseteq r} t_{a'} \\
&= t_a + \sum_{a' \in A_p} t_{a'} \cdot \sum_{r \in R_{pa_t}[A_p]: a' \subseteq r} \prod_{a'' \subseteq r} \alpha_{a''} \\
&= t_a + \sum_{a' \in A_p} t_{a'} \cdot \chi_{p \to a'_t} \cdot \alpha_{a'} \cdot \chi_{a'_h \to a_t} \\
&= t_a + \sum_{a' \in A_p; o(a'_h) < o(a_h)} t_{a'} \cdot \alpha_{a'} \cdot \chi_{a'_h \to a_t}
\end{aligned} \tag{2.41}$$

Using (2.41) we can rewrite (2.34) as

$$\frac{\partial T}{\partial^c \alpha_a} = g_j \cdot (\mu_a - \mu_b) \tag{2.42}$$

This expression can be interpreted as follows: shifting $\delta\%$ from approach proportion α_a to approach proportion α_b is equivalent to shifting $(\delta \cdot g_j)$ vph from approach a to approach b. The cost incurred by these travelers while in approach a was on the

average μ_a; once shifted to approach b the cost incurred by them is μ_b. The total cost difference is

$$\delta \cdot g_j \cdot (\mu_a - \mu_b) = \delta \cdot \frac{\partial T}{\partial^c \alpha_a} \tag{2.43}$$

as might be expected intuitively.

From (2.42) the first order necessary conditions for optimality are that at least one of the following conditions holds

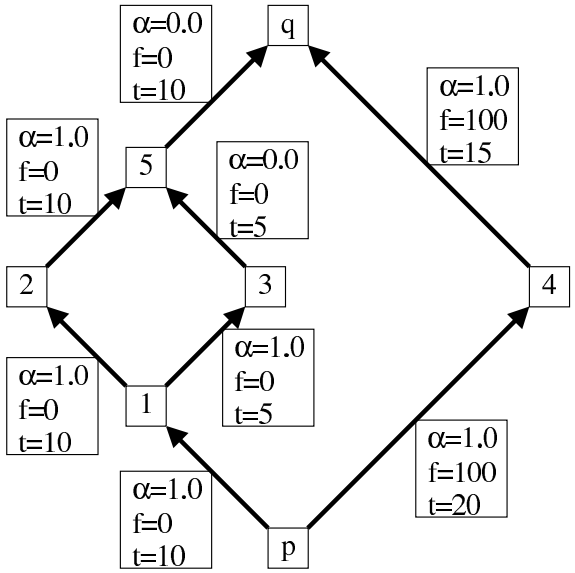
$$g_j = 0 \tag{2.44a}$$
$$\mu_a = \mu_b \tag{2.44b}$$
$$\alpha_a = 0; \quad \mu_a > \mu_b \tag{2.44c}$$
$$\alpha_b = 0; \quad \mu_b > \mu_a \tag{2.44d}$$

So far we did not make any assumption about the way basic approaches are chosen. From now on we assume that the basic approach is an approach of minimum average cost; therefore, the necessary conditions for optimality are

$$\mu_a \geq \mu_{b_j} \qquad \forall j \in N \setminus \{p\}; \forall a \in NB_j \tag{2.45a}$$
$$\alpha_a \cdot g_j \cdot \left(\mu_a - \mu_{b_j}\right) = 0 \qquad \forall j \in N \setminus \{p\}; \forall a \in NB_j \tag{2.45b}$$

Figure 6 shows an example of a network that satisfies conditions (2.45), but still does not satisfy the user equilibrium conditions. In this example $c_{[p,4,q]} = 35$; $c_{[p,1,2,5,q]} = 40$; $c_{[p,1,3,5,q]} = 30$; therefore the cost of the used route $[p,4,q]$ is greater than the cost of one of the unused routes $[p,1,3,5,q]$. $\mu_{[4,q]} = 35 < 40 = \mu_{[5,q]}$; therefore, $[4,q]$ is the basic approach, and conditions (2.45) hold at node q. As for node 5, $\mu_{[2,5]} = 30 > 20 = \mu_{[3,5]}$ hence $[3,5]$ is the basic approach. Condition (2.45b) holds in this case, but only because $g_5 = 0$.

First order conditions are not sufficient for optimality only if the function T is not convex. The non-convexity of T as a function of $\boldsymbol{\alpha}$ in this case can be verified directly. For simplicity assume that link costs are fixed. The solution can be described by a vector of the non-basic approach proportions, $[2,5];[5,q]$. The current solution is represented by $(1,0)$. Consider the direction $d = (-1,1)$. The directional derivative along that direction is positive; however, if we continue in the same direction all the way to $(0,1)$ we obtain a solution of lower objective function, which contradicts convexity.



necessary conditions hold; sufficient conditions are violated
α-approach proportion; f-flow; t-cost

Figure 6. Average approach cost optimality conditions

The following lemma shows that by omitting the node flow from (2.45b) sufficient conditions for optimality are obtained.

Lemma 3. *Conditions*

$$\mu_a \geq \mu_{b_j} \qquad \forall j \in N \setminus \{p\}; \forall a \in NB_j \tag{2.46a}$$

$$\alpha_a \cdot \left(\mu_a - \mu_{b_j}\right) = 0 \qquad \forall j \in N \setminus \{p\}; \forall a \in NB_j \tag{2.46b}$$

are sufficient for restricted user equilibrium.

Proof:
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 (2.14) every used link is necessarily a contributing one, therefore if there is a used route that violates the user equilibrium conditions, it can also be viewed as a contributing route. 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. w.l.o.g. assume that k is the node of lowest topological order for which such routes exist. Clearly $i \neq j$; 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 $c_r = \mu_{[ik]}$. Since

$o(j) < o(k)$ the cost of every contributing route segment from the p to j and the average of such costs is not greater than the cost of any alternative route segment, hence $c_s \geq \mu_{[jk]}$. Therefore $\mu_{[ik]} = c_r > c_s \geq \mu_{[jk]} \geq \mu_{b_k}$ and $\alpha_{[ik]} > 0$, in contradiction to conditions (2.46). □

As was shown in Figure 6, in some cases it is essential to require that the average approach cost condition holds even if the node flow is zero. In other cases such a requirement is not necessary. For example, if in the same figure the cost of link $[p, 4]$ was 10 instead of 20, then the cost of the used route would be $c_{[p,4,q]} = 25$, meaning this is a route of minimum cost. Therefore, this solution would be at equilibrium even though the sufficient average approach cost conditions (2.46) still does not hold at node 5. However, conditions (2.46) can be easily satisfied in this case ($t_{[p,4]} = 10$) by setting $\alpha_{[2,5]} = 0.0; \alpha_{[3,5]} = 1.0$. Since every user equilibrium $\boldsymbol{\alpha}$ satisfies (2.45), a similar correction at nodes of zero flow yields a solution that satisfies (2.46), while keeping the same origin-based link flows and the same route flows. From here on, therefore, we only consider sufficiency conditions (2.46).

2.5 Second order derivatives and their approximation

To minimize T one should shift flow from non-basic to basic approaches. A Newton-type shift of flow is based on the ratio of the first order derivative and the second order derivative. The first goal of this section is to derive the second order derivative. Unfortunately this derivation leads to a rather complicated expression, which we do not how know to compute efficiently. Therefore, in the following sections an approximation is used. Motivation for that approximation is provided in this section. The formal recursive definition for the approximated second order derivative is given in the next section.

The diagonal second order derivatives of the flow on any link a' with respect to any other approach proportion a is always zero.

$$\frac{\partial^2 f_{a'}}{\partial {\alpha_a}^2} = \frac{\partial^2 f_{a'}}{\partial^c {\alpha_a}^2} = 0 \tag{2.47}$$

The off-diagonal second order derivatives are not zero, as link flows are multiplicative and not linear functions of approach proportions. The second order derivative of the objective function with respect to link flow is

$$\frac{\partial^2 T}{\partial {f_a}^2} = \frac{\partial t_a}{\partial f_a} = t'_a \tag{2.48}$$

and with respect to approach proportions

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

$$= t'_a \cdot g_j^2 + t'_b \cdot g_j^2 + \sum_{\substack{a' \in A_p \\ o(a'_h) < o(j)}} t'_{a'} \cdot {\alpha_{a'}}^2 \cdot g_j^2 \cdot \left(\chi_{a'_h \to a_t} - \chi_{a'_h \to b_t} \right)^2 \tag{2.49}$$

We have shown that changing the approach proportion of link a terminating at j does not affect link flows prior to lcn_j. Remember that lcn_j is also a common node to a_t and to b_t hence

$$\chi_{i \to a_t} = \chi_{i \to b_t} = \chi_{i \to lcn_j} \qquad \forall i \in N; o(i) \leq o(lcn_j) \tag{2.50}$$

$$\frac{\partial^2 T}{\partial^c {\alpha_a}^2} = t'_a \cdot g_j^2 + t'_b \cdot g_j^2 + \sum_{\substack{a' \in A_p \\ o(lcn_j) < o(a'_h) < o(j)}} t'_{a'} \cdot {\alpha_{a'}}^2 \cdot g_j^2 \cdot \left(\chi_{a'_h \to a_t} - \chi_{a'_h \to b_t} \right)^2 \tag{2.51}$$

Developing recursive equations similar to (2.39-2.40) for the second order derivative is more challenging, due to the interaction between approaches, that is, the fact that route segments from two approaches are likely to share links in a complicated manner. If the interaction between approaches is ignored, the following approximation can be made

$$\begin{aligned}
\nu_a &= t'_a + \sum_{a' \in A_p; o(a'_h) < o(j)} t'_{a'} \cdot {\alpha_{a'}}^2 \cdot \chi^2_{a'_h \to a_t} \\
&= t'_a + \sum_{a' \in A_p; a'_h = a_t} t'_{a'} \cdot \alpha^2_{a'} + \sum_{a' \in A_p; o(a'_h) < o(a_t)} t'_{a'} \cdot {\alpha_{a'}}^2 \cdot \left(\sum_{a'' \in A_p; a''_h = a_t} \alpha_{a''} \cdot \chi_{a'_h \to a''_t} \right)^2 \\
&\approx t'_a + \sum_{a' \in A_p; a'_h = a_t} t'_{a'} \cdot \alpha^2_{a'} + \sum_{a' \in A_p; o(a'_h) < o(a_t)} t'_{a'} \cdot {\alpha_{a'}}^2 \cdot \sum_{a'' \in A_p; a''_h = a_t} \left(\alpha_{a''} \cdot \chi_{a'_h \to a''_t} \right)^2 \\
&= t'_a + \sum_{a'' \in A_p; a''_h = a_t} \alpha^2_{a''} \left(t''_{a''} + \sum_{a' \in A_p; o(a'_h) < o(a''_t)} t'_{a'} \cdot {\alpha_{a'}}^2 \cdot \chi^2_{a'_h \to a''_t} \right) \\
&= t'_a + \sum_{a'' \in A_p; a''_h = a_t} \alpha^2_{a''} \cdot \nu_{a''}
\end{aligned} \tag{2.52}$$

Using (2.50) again, we find a similar approximation for

$$\begin{aligned}
&\sum_{a'\in A_p;o(a'_h)\le o(lcn_j)} t'_{a'}\cdot {\alpha_{a'}}^2\cdot \chi^2_{a'_h\to a_t} = \sum_{a'\in A_p;o(a'_h)\le o(lcn_j)} t'_{a'}\cdot {\alpha_{a'}}^2\cdot \chi^2_{a'_h\to lcn_j}\\
&= \sum_{a'\in A_p;a'_h=lcn_j} t'_{a'}\cdot \alpha^2_{a'} + \sum_{a'\in A_p;o(a'_h)<o(lcn_j)} t'_{a'}\cdot {\alpha_{a'}}^2\cdot\left(\sum_{a''\in A_p;a''_h=lcn_j}\alpha_{a''}\cdot\chi_{a'_h\to a''_t}\right)^2\\
&\approx \sum_{a'\in A_p;a'_h=lcn_j} t'_{a'}\cdot \alpha^2_{a'} + \sum_{a'\in A_p;o(a'_h)<o(lcn_j)} t'_{a'}\cdot {\alpha_{a'}}^2\cdot\sum_{a''\in A_p;a''_h=lcn_j}\left(\alpha_{a''}\cdot\chi_{a'_h\to a''_t}\right)^2\\
&= \sum_{a''\in A_p;a''_h=lcn_j}\alpha^2_{a''}\left(t''_{a''}+\sum_{a'\in A_p;o(a'_h)<o(a''_t)} t'_{a'}\cdot {\alpha_{a'}}^2\cdot\chi^2_{a'_h\to a''_t}\right) = \sum_{a''\in A_p;a''_h=lcn_j}\alpha^2_{a''}\cdot\nu_{a''}
\end{aligned} \tag{2.53}$$

Denoting

$$\rho_j = \sum_{a\in A_p;a_h=j}\alpha^2_a\cdot\nu_a \tag{2.54}$$

and using the assumption that there is no interaction between approaches, (2.51) can be rewritten as

$$\begin{aligned}
\frac{\partial^2 T}{\partial^c {\alpha_a}^2} &= t'_a\cdot g^2_j + t'_b\cdot g^2_j + \sum_{\substack{a'\in A_p\\ o(lcn_j)<o(a'_h)<o(j)}} t'_{a'}\cdot {\alpha_{a'}}^2\cdot g^2_j\cdot\left(\chi_{a'_h\to a_t}-\chi_{a'_h\to b_t}\right)^2\\
&\approx t'_a\cdot g^2_j + \sum_{a'\in A_p;o(lcn_j)<o(a'_h)<o(j)} t'_{a'}\cdot {\alpha_{a'}}^2\cdot g^2_j\cdot\chi^2_{a'_h\to a_t}\\
&\quad + t'_b\cdot g^2_j + \sum_{a'\in A_p;o(lcn_j)<o(a'_h)<o(j)} t'_{a'}\cdot {\alpha_{a'}}^2\cdot g^2_j\cdot\chi^2_{a'_h\to b_t}\\
&\approx g^2_j\cdot\left(\nu_a+\nu_b-2\cdot\rho_{lcn_j}\right)
\end{aligned} \tag{2.55}$$

Equations (2.52) and (2.54) allow us to compute ν_a for all links and ρ_j for all nodes in a single ascending pass.

2.6 Flow shifts and their aggregation

For this section average cost functions are redefined recursively by

$$\sigma_p(\boldsymbol{\alpha}, \mathbf{t}) = 0 \tag{2.56a}$$

$$\sigma_j(\boldsymbol{\alpha}, \mathbf{t}) = \sum_{a \in A_p; a_h = j} \alpha_a \cdot \mu_a(\boldsymbol{\alpha}, \mathbf{t}) \qquad (\forall j \neq p) \tag{2.56b}$$

$$\mu_a(\boldsymbol{\alpha}, \mathbf{t}) = t_a + \sigma_{a_t}(\boldsymbol{\alpha}, \mathbf{t}) = t_a + \sum_{a' \in A_p; a'_h = a_t} \alpha_{a'} \cdot \mu_{a'}(\boldsymbol{\alpha}, \mathbf{t}) \tag{2.56c}$$

and similarly the derivative approximating functions are redefined recursively by

$$\rho_p(\boldsymbol{\alpha}, \mathbf{t}') = 0 \tag{2.57a}$$

$$\rho_j(\boldsymbol{\alpha}, \mathbf{t}') = \sum_{a \in A_p; a_h = j} \alpha_a^2 \cdot \nu_a(\boldsymbol{\alpha}, \mathbf{t}') \qquad (\forall j \neq p) \tag{2.57b}$$

$$\nu_a(\boldsymbol{\alpha}, \mathbf{t}') = t'_a + \rho_{a_t}(\boldsymbol{\alpha}, \mathbf{t}') = t'_a + \sum_{a' \in A_p; a'_h = a_t} \alpha_{a'}^2 \cdot \nu_{a'}(\boldsymbol{\alpha}, \mathbf{t}') \tag{2.57c}$$

These definitions are motivated by the discussion in the previous section. Notice that in these functions $\mathbf{t}, \mathbf{t}'$ and $\boldsymbol{\alpha}$ are separate variables, since we wish to use these functions in cases where $\mathbf{t} \neq \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha}))$ and $\mathbf{t}' \neq \mathbf{t}'(\mathbf{f}(\boldsymbol{\alpha}))$.

The first step in building the search direction is to determine the amount of flow to be shifted between two alternative approaches a and b; $a_h = b_h = j$. Assume w.l.o.g. that $\mu_b \leq \mu_a$, and choose b as the current basic approach for j. Our goal is to choose a value θ to be subtracted from α_a and added to α_b. By (2.42) and (2.55) a Newton type shift would be

$$\frac{\frac{\partial T}{\partial^c \alpha_a}}{\frac{\partial^2 T}{\partial^c {\alpha_a}^2}} \approx \frac{g_j \cdot (\mu_a - \mu_b)}{g_j^2 \cdot \left(\nu_a + \nu_b - 2 \cdot \rho_{lcn_j}\right)} = \frac{1}{g_j} \cdot \frac{\mu_a - \mu_b}{\nu_a + \nu_b - 2 \cdot \rho_{lcn_j}} \tag{2.58}$$

Here, g, μ, ν, ρ are all continuous functions of $\mathbf{t}$, $\mathbf{t}'$, and $\boldsymbol{\alpha}$; however, both the node flow and the cost derivatives may have a value of zero. In order to obtain a well-defined shift for all possible values of $\mathbf{t}$, $\mathbf{t}'$, and $\boldsymbol{\alpha}$, some modification is required. We wish to make the modification in such a way that the resulting algorithmic map is also closed, a property that helps in the proof of convergence. To overcome the problem of zero derivative estimate, we choose a small positive constant $\epsilon_\nu > 0$, and define

$$z_{a \to b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}') = \frac{\mu_a(\boldsymbol{\alpha}, \mathbf{t}) - \mu_b(\boldsymbol{\alpha}, \mathbf{t})}{\max(\epsilon_\nu, \nu_a(\alpha, \mathbf{t}') + \nu_b(\boldsymbol{\alpha}, \mathbf{t}') - 2 \cdot \rho_{lcn_j}(\boldsymbol{\alpha}, \mathbf{t}'))} \tag{2.59}$$

$z_{a\to b}$ is a continuous function that can be interpreted as the desirable amount of flow (in units of flow, say vph) that should be shifted from a to b in order to equalize costs, ignoring feasibility constraints.

Handling zero node flow in a way that yields a closed and converging algorithmic map requires a different treatment. We define the change in approach proportions by the point to set map $\Theta_{a\to b} : [0,1]^{|A|} \times \Re_+^{|A|} \times \Re_+^{|A|} \to 2^{[0,1]}$ as follows

$$\Theta_1^{a\to b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}') = \begin{cases} \left\{\min\left(\alpha_a, \frac{z_{a\to b}(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}')}{g_j(\boldsymbol{\alpha})}\right)\right\} & g_j > 0 \\ \{\alpha_a\} & g_j = 0; \mu_a > \mu_b \\ [0, \alpha_a] & g_j = 0; \mu_a = \mu_b \end{cases} \tag{2.60}$$

Since $\Theta_1^{a\to b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}') \subseteq [0, \alpha_a]$, feasibility is guaranteed.

Lemma 4. *$\Theta_1^{a\to b}$ is a closed map; that is if $\boldsymbol{\alpha}^k \to \boldsymbol{\alpha}$, $\mathbf{t}^k \to \mathbf{t}$, $\mathbf{t}'^k \to \mathbf{t}'$, $\theta^k \in \Theta_1^{a\to b}(\boldsymbol{\alpha}^k, \mathbf{t}^k, \mathbf{t}'^k)$, and $\theta^k \to \theta$, then $\theta \in \Theta_1^{a\to b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}')$.*

Proof:
There are three cases:

1. $g_j(\boldsymbol{\alpha}) > 0$
   The function $\Theta_1^{a\to b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}')$ is a continuous point to point function in the neighborhood of $\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}'$.

2. $g_j(\boldsymbol{\alpha}) = 0$, $\mu_a(\boldsymbol{\alpha}, \mathbf{t}) - \mu_b(\boldsymbol{\alpha}, \mathbf{t}) > \epsilon > 0$.
   Choose M such that

   $$M > \nu_a(\boldsymbol{\alpha}, \mathbf{t}') + \nu_b(\boldsymbol{\alpha}, \mathbf{t}') - 2 \cdot \rho_{lcn_j}(\boldsymbol{\alpha}, \mathbf{t}') \quad ; \quad M > \epsilon_\nu$$

   There is k_0 such that for every $k > k_0$

   $$\mu_a(\boldsymbol{\alpha}^k, \mathbf{t}^k) - \mu_b(\boldsymbol{\alpha}^k, \mathbf{t}^k) > \epsilon$$

   $$\nu_a(\boldsymbol{\alpha}^k, \mathbf{t}'^k) + \nu_b(\boldsymbol{\alpha}^k, \mathbf{t}'^k) - 2 \cdot \rho_{lcn_j}(\boldsymbol{\alpha}^k, \mathbf{t}'^k) < M$$

   $$g_j(\boldsymbol{\alpha}^k) < \frac{\epsilon}{M}$$

   Therefore, $\frac{z_{a\to b}(\boldsymbol{\alpha}^k, \mathbf{t}^k, \mathbf{t}')}{g_j(\boldsymbol{\alpha}^k)} > 1 > \boldsymbol{\alpha}^k$, hence

   $$\Theta_1^{a\to b}(\boldsymbol{\alpha}^k, \mathbf{t}^k, \mathbf{t}'^k) = \{\alpha_a^k\}$$
   $$\theta^k = \alpha_a^k \to \alpha_a$$
   $$\theta = \alpha_a$$

and $\Theta_1^{a\to b}(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}') = \{\alpha_a\}$ so $\theta \in \Theta_1^{a\to b}(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}')$.

3. $g_j(\boldsymbol{\alpha}) = 0$, $\mu_a(\boldsymbol{\alpha},\mathbf{t}) = \mu_b(\boldsymbol{\alpha},\mathbf{t})$.
   $\theta^k \in \Theta_1^{a\to b}(\boldsymbol{\alpha}^k,\mathbf{t}^k,\mathbf{t}'^k) \subseteq [0,\alpha^k]$, therefore $\theta \in [0,\alpha_a] = \Theta_1^{a\to b}(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}')$.

□

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

$$\Delta\mathbf{f}(\boldsymbol{\alpha},\Delta\boldsymbol{\alpha}) = \mathbf{f}(\boldsymbol{\alpha}+\Delta\boldsymbol{\alpha}) - \mathbf{f}(\boldsymbol{\alpha}) \tag{2.61}$$

Suppose $\Delta\alpha_a \in -\Theta_1^{a\to b}(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}')$; $\Delta\alpha_b = -\Delta\alpha_a$; $\Delta\alpha_{a'} = 0 \quad \forall a' \neq a, a' \neq b$. By (2.47) $f_{a'}$ is a linear function of α_a (zero second order derivative), therefore

$$\begin{aligned}
\Delta f_{a'}(\boldsymbol{\alpha},\Delta\boldsymbol{\alpha}) &= f_{a'}(\boldsymbol{\alpha}+\Delta\boldsymbol{\alpha}) - f_{a'}(\boldsymbol{\alpha}) = \Delta\alpha_a \cdot \frac{\partial f_{a'}}{\partial^c \alpha_a} \\
&= \begin{cases} \Delta\alpha_a \cdot g_j(\boldsymbol{\alpha}) & a' = a \\ -\Delta\alpha_a \cdot g_j(\boldsymbol{\alpha}) & a' = b \\ \Delta\alpha_a \cdot \alpha_{a'} \cdot g_j(\boldsymbol{\alpha}) \cdot \left(\chi_{a'_h \to a_t} - \chi_{a'_h \to b_t}\right) & o(a'_h) < o(a'_t) \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{2.62}$$

The directional derivative of T along $\Delta\mathbf{f}$ is

$$\begin{aligned}
\Delta\mathbf{f}(\boldsymbol{\alpha},\Delta\boldsymbol{\alpha})\cdot\mathbf{t} &= \Delta\alpha_a \cdot t_a \cdot g_j(\boldsymbol{\alpha}) - \Delta\alpha_a \cdot t_b \cdot g_j(\boldsymbol{\alpha}) \\
&\quad + \sum_{a' \in A_p; o(a'_h) < o(j)} \Delta\alpha_a \cdot t_{a'} \cdot \alpha_{a'} \cdot g_j(\boldsymbol{\alpha}) \cdot \left(\chi_{a'_h \to a_t} - \chi_{a'_h \to b_t}\right) \\
&= \Delta\alpha_a \cdot g_j(\boldsymbol{\alpha}) \cdot (\mu_a(\boldsymbol{\alpha},\mathbf{t}) - \mu_b(\boldsymbol{\alpha},\mathbf{t})) \leq 0
\end{aligned} \tag{2.63}$$

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 method, shifts are computed and applied for each non-basic approach sequentially. Link costs are updated after each shift. Such a method does not take full advantage of the a-cyclic structure of the solution. We mentioned that the 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 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.

Aggregating shifts from approaches to the same node is a relatively simple matter, as they are independent in the following sense: if a, a' are two alternative non-basic approaches to node $j = a_h = a'_h$ then $g_j, \mu_a, \mu_b, \nu_a, \nu_b, \rho_{lcn_j}$ are all independent of $\alpha_{a'}$; hence $z_{a\to b}$ and $\Theta_1^{a\to b}$ are also independent of $\alpha_{a'}$. As a result, applying such shifts simultaneously or in any sequential order produces the same results, assuming that only the weights for averaging are updated and not link costs. If the basic approach b is given, the aggregated shift is defined by the following map

$$\Theta_1^{j;b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}') = \left\{ \Delta\boldsymbol{\alpha} : \begin{array}{ll} \Delta\alpha_a \in -\Theta_1^{a\to b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}') & \forall a \in NB_j \\ \Delta\alpha_b = -\sum_{a\in NB_j} \Delta\alpha_a & \\ \Delta\alpha_{a'} = 0 & a'_h \neq j \end{array} \right\} \tag{2.64}$$

This aggregated shift satisfies feasibility requirements. Since one route can not contain two approaches to the same node, the change in link flows for the aggregated shift is the sum of the changes of the pairwise shifts. Therefore, if $\Delta\boldsymbol{\alpha} \in \Theta_1^{j;b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}')$ then $\Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha})$ is either zero or a direction of descent of T.

In many cases there is only one minimum cost approach, and therefore a unique way to choose the basic approach. In general, however, several alternative approaches may have equal cost; in this case the choice of the basic approach is arbitrary. The algorithmic map for node-shifts takes that arbitrariness into account and is defined as follows

$$\Theta_1^{j}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}' : A_p) = \bigcup_{\substack{b \in A_p : b_h = j; \\ \mu_b \leq \mu_a \quad \forall a \in A_p; a_h = j}} \Theta_1^{j;b}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}') \tag{2.65}$$

Notice that the choice of the basic approach depends not only on the current values of $\mathbf{t}$, $\mathbf{t}'$ and $\boldsymbol{\alpha}$, but also on the given restricting subnetwork A_p.

Lemma 5. *$\Theta_1^j(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}' : A_p)$ is a closed map.*

Proof:
Suppose $\boldsymbol{\alpha}^k \to \boldsymbol{\alpha}$, $\mathbf{t}^k \to \mathbf{t}$, $\mathbf{t}'^k \to \mathbf{t}'$, $\Delta\boldsymbol{\alpha}^k \in \Theta_1^j(\boldsymbol{\alpha}^k, \mathbf{t}^k, \mathbf{t}'^k : A_p)$, and $\Delta\boldsymbol{\alpha}^k \to \Delta\boldsymbol{\alpha}$. Let b^k be the basic approach used to find $\Delta\boldsymbol{\alpha}^k$. There are only a finite number of possible basic approaches; therefore, there is a subsequence K that uses the same basic approach b^0, that is $b^k = b^0 \quad \forall k \in K$. If there is an alternative approach a of lower cost, $\mu_a(\boldsymbol{\alpha}, \mathbf{t}) < \mu_{b^0}(\boldsymbol{\alpha}, \mathbf{t})$ then for all $k > k_0$ also $\mu_a(\boldsymbol{\alpha}^k, \mathbf{t}^k) < \mu_{b^0}(\boldsymbol{\alpha}^k, \mathbf{t}^k)$

contradicting the choice of b^k as a basic approach. Therefore, b^0 is a legitimate basic approach for $\boldsymbol{\alpha}$, $\mathbf{t}$, $\mathbf{t}'$.

Since $\Theta_1^{a\to b^0}(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}')$ is a closed map, $\Delta\alpha_a \in \Theta_1^{a\to b^0}(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}') \quad \forall a \in NB_j$ hence

$$\Delta\boldsymbol{\alpha} \in \Theta_1^{j;b^0}(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}') \subseteq \Theta_1^{j}(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}' : A_p)$$

□

As mentioned before, the aggregation to origin-based shift can be done in three ways.

Simultaneous: node shifts are computed independently

$$\Theta_1(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}' : A_p) = \left\{ \Delta\boldsymbol{\alpha} = \sum_{j\in N; j\neq p} \Delta\boldsymbol{\alpha}^j : \Delta\boldsymbol{\alpha}^j \in \Theta_1^j(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}' : A_p) \right\} \tag{2.66}$$

Ascending: each node shift takes into account the shifts at nodes of lower topological order

$$\Theta_1^{\uparrow}(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}' : A_p) = \left\{ \Delta\boldsymbol{\alpha} = \sum_{j\in N; j\neq p} \Delta\boldsymbol{\alpha}^j : \begin{array}{c} \Delta\boldsymbol{\alpha}^j \in \Theta_1^j(\boldsymbol{\alpha}^j,\mathbf{t},\mathbf{t}' : A_p) \\ \boldsymbol{\alpha}^j = \boldsymbol{\alpha} + \sum\limits_{j'\in N; o(j')<o(j)} \Delta\boldsymbol{\alpha}^{j'} \end{array} \right\} \tag{2.67}$$

Descending: each node shift takes into account the shifts at nodes of higher topological order

$$\Theta_1^{\downarrow}(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}' : A_p) = \left\{ \Delta\boldsymbol{\alpha} = \sum_{j\in N; j\neq p} \Delta\boldsymbol{\alpha}^j : \begin{array}{c} \Delta\boldsymbol{\alpha}^j \in \Theta_1^j(\boldsymbol{\alpha}^j,\mathbf{t},\mathbf{t}' : A_p) \\ \boldsymbol{\alpha}^j = \boldsymbol{\alpha} + \sum\limits_{j'\in N; o(j')>o(j)} \Delta\boldsymbol{\alpha}^{j'} \end{array} \right\} \tag{2.68}$$

Notice in all three cases link costs and cost derivatives are not updated. All three maps are feasible and closed. Evaluating the directional derivative in the simultaneous aggregation is somewhat cumbersome, due to the multiplicative interaction among the approach proportions. As for the sequential aggregations, we find that the resulting vector of link flow changes is a sum of node-shift changes, using different values of $\boldsymbol{\alpha}$, but the same values of $\mathbf{t}$ and $\mathbf{t}'$. As a result the aggregated change is either zero or a direction of descent of T. For example if $\Delta\boldsymbol{\alpha} \in \Theta_1^{\downarrow}(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}' : A_p)$ then $\Delta\mathbf{f}(\boldsymbol{\alpha},\Delta\boldsymbol{\alpha}) = \sum\limits_{j\in N} \Delta\mathbf{f}(\boldsymbol{\alpha}^j,\Delta\boldsymbol{\alpha}^j)$; therefore $\Delta\mathbf{f}(\boldsymbol{\alpha},\Delta\boldsymbol{\alpha})\cdot\mathbf{t} \leq 0$, and equality holds if and only if $\Delta\mathbf{f}(\boldsymbol{\alpha},\Delta\boldsymbol{\alpha}) = 0$.

Comment: in the multiple-origin problem one might want to aggregate the flow shifts over all nodes to come up with a global search direction. Such aggregation causes no problems, since feasibility closedness, and non-positive directional derivative are guaranteed.

2.7 Boundary search

In the previous section various possible search directions at different levels of aggregation were described. In all cases we showed that applying the search direction as is (with step 1.0) leads to a new feasible solution. We also showed that locally the search direction has a non-positive directional derivative. This property although highly desirable does not guarantee that applying the search direction as is leads to a lower value of the objective function. To overcome this problem, it is common to choose a convex combination of the current solution and the new solution that minimizes the objective function. Since the feasible set is convex, the combination is guaranteed to be feasible.

The main problem with the common convex line search is that it tends to lead to inner point solutions. In the traffic assignment problem this implies that it tends to leave residual flows on sub-optimal routes. Residual flows may have a negligible effect on the objective function value; thus in many methods they are not a reason for concern. Residual flows cause significant problems when restrictions are to be updated, as was discussed in section 2.1, and may in fact prevent global convergence. Even proving restricted convergence was found to be quite difficult if convex search is used. Experimental results also indicated that the convex search is problematic.

The proposed alternative is boundary search, which was described schematically in the overview. As mentioned there, the key point in the boundary search is to apply the step size prior to any consideration of feasibility constraints. In our case this is done while determining the shift between a pair of approaches, that is by replacing equation (2.60) with

$$\Theta_{\lambda}^{a\to b}(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}') = \begin{cases} \left\{\min\left(\alpha_a, \lambda\cdot\frac{z_{a\to b}(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}')}{g_j(\boldsymbol{\alpha})}\right)\right\} & g_j > 0 \\ \{\alpha_a\} & g_j = 0; \mu_a > \mu_b \\ [0,\alpha_a] & g_j = 0; \mu_a = \mu_b \end{cases} \tag{2.69}$$

where λ is the step size. Aggregation to $\Theta_{\lambda}^{j}(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}':A_p)$ and $\Theta_{\lambda}^{\downarrow}(\boldsymbol{\alpha},\mathbf{t},\mathbf{t}':A_p)$ is done in the same way as before.

Finding the step size that minimizes T in this case is more complicated than in the convex search for two reasons. First, for every value of λ there are several possible values of $\Delta\boldsymbol{\alpha}$ and hence several possible values of $\Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha})$. Second, even if there was a way to make the choices consistent, thereby obtaining a function $\Delta\mathbf{f}(\lambda)$, the direction of $\Delta\mathbf{f}$ would vary with λ, thus complicating the evaluation of the directional derivative substantially. The boundary search considers a broken line that changes direction every time the boundary is reached by one of the components. Figure 4 shows one such change of direction for a two dimensional problem. In problems of realistic size, the number of dimensions is more likely to be in the range of hundreds or thousands; thus changes of direction can come quite frequently.

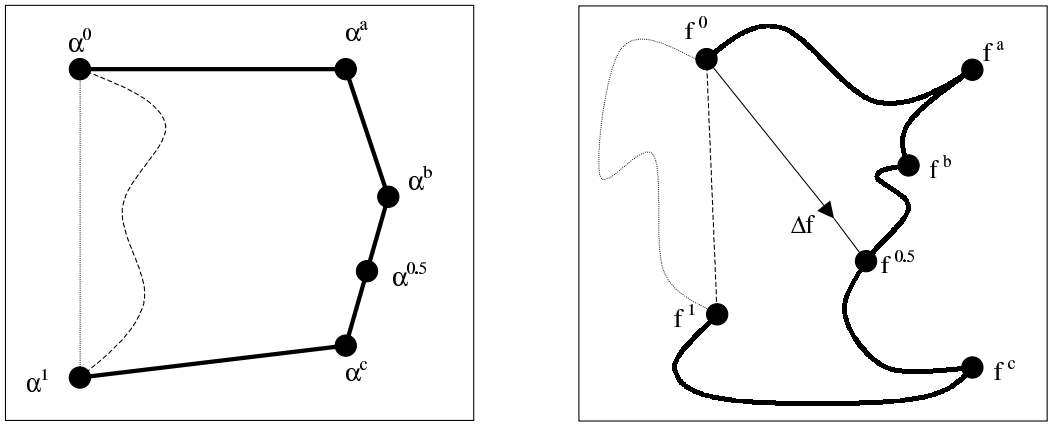
On the other hand, even with convex line search, finding the optimal step size is not necessarily cost effective, and typically some approximation is used. One possible strategy is to examine $\lambda = 2^{-k}$ and to choose the largest λ (smallest $k \geq 0$) such that

$$\Delta\mathbf{f} \cdot \mathbf{t}(\mathbf{f} + \lambda \cdot \Delta\mathbf{f}) \leq \mathbf{0} \tag{2.70}$$

Assuming that the search direction is a direction of descent, i.e. $\Delta\mathbf{f} \cdot \mathbf{t}(\mathbf{f}) < \mathbf{0}$, the continuity of cost guarantees that when λ is small enough, condition (2.70) is satisfied. We can also find $0 < \lambda' \leq \lambda$ for which condition (2.70) holds with strong inequality. Since $T(\mathbf{f})$ is a convex function of $\mathbf{f}$, condition (2.70) ensures that

$$\begin{aligned}
\Delta\mathbf{f} \cdot \mathbf{t}(\mathbf{f} + \mathbf{x} \cdot \Delta\mathbf{f}) &\leq 0 \quad \forall 0 \leq x \leq \lambda && (2.71)\\
\Delta\mathbf{f} \cdot \mathbf{t}(\mathbf{f} + \mathbf{x} \cdot \Delta\mathbf{f}) &< 0 \quad \forall 0 \leq x \leq \lambda' && (2.72)\\
T(\mathbf{f} + \lambda \cdot \Delta\mathbf{f}) - T(\mathbf{f}) &= \int_0^{\lambda} \Delta\mathbf{f} \cdot \mathbf{t}(\mathbf{f} + \mathbf{x} \cdot \Delta\mathbf{f}) \mathrm{d}\mathbf{x} \\
&\leq \int_0^{\lambda'} \Delta\mathbf{f} \cdot \mathbf{t}(\mathbf{f} + \mathbf{x} \cdot \Delta\mathbf{f}) \mathrm{d}\mathbf{x} < \mathbf{0} && (2.73)\\
T(\mathbf{f} + \Delta\mathbf{f}) &< T(\mathbf{f}) && (2.74)
\end{aligned}$$

Figure 7 illustrates the different possible search procedures in both the approach proportions and link flows spaces. $\boldsymbol{\alpha}^0$ denotes the current approach proportions solution, and $\boldsymbol{\alpha}^1$ the new solution for step size 1.0. $\mathbf{f}^0$ and $\mathbf{f}^1$ denote the corresponding link flows solutions. The thin line represents convex combinations of $\boldsymbol{\alpha}^0$ and $\boldsymbol{\alpha}^1$, this line segment corresponds to the thin curve in the link flows space. Notice that the same substantial amount of computational effort is required to produce each additional point along that line. The dashed line represents convex combinations of $\mathbf{f}^0$ and $\mathbf{f}^1$, this line corresponds to the dashed curve in the approach proportions space; searching along that line has the advantage that once the end points are known, new



a) Approach proportions space b) Link flows space

Figure 7. Boundary search and its alternatives

points along the line segment are obtained by simple averaging, thus reducing the computational effort especially if all origins are considered simultaneously.

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 interset to us, since T is not convex as a function of $\boldsymbol{\alpha}$. $\Delta\mathbf{f}$ is the vector connecting $\mathbf{f}^0$ and $\mathbf{f}^{0.5}$. We know that the directional derivative along that vector at $\mathbf{f}^0$ is negative. Also, by convexity, we know that the directional derivative along the line segment connecting $\mathbf{f}^0$ and $\mathbf{f}^{0.5}$ is monotonically increasing. Therefore, to ensure descent of the objective function, all we need to do is to verify that the directional derivative along $\Delta\mathbf{f}$ at $\mathbf{f}^{0.5}$ is non-positive. If it is, we are done; otherwise, we examine smaller step size values.

Demonstrating that condition (2.70) holds for some positive λ in the case of the boundary search requires a more careful argument, since $\Delta\mathbf{f}$ changes with λ. Furthermore, even if λ converges to zero, $\Delta\boldsymbol{\alpha}$ does not necessarily converge to zero; only $\Delta\mathbf{f}$ does. Using this result and the continuity of approach average cost, one can show that if $\mu_a(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f})) > \mu_b(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f}))$ then for a sufficiently small perturbation

$\mu_a(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f} + \Delta\mathbf{f})) > \mu_b(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f} + \Delta\mathbf{f}))$. Therefore, for a small enough λ if $\Delta\boldsymbol{\alpha} \in \Theta^{\downarrow}_{\lambda}(\boldsymbol{\alpha}, \mathbf{t}, \mathbf{t}')$, $\Delta\mathbf{f} = \Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha})$, then

$$\Delta\alpha_a \cdot g_j(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}) \cdot (\mu_a(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f} + \Delta\mathbf{f})) - \mu_b(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f} + \Delta\mathbf{f}))) \leq 0 \tag{2.75}$$

$$\Delta\mathbf{f} \cdot \mathbf{t}(\mathbf{f} + \Delta\mathbf{f}) \leq 0 \tag{2.76}$$

If $\Delta\mathbf{f} \neq 0$ then $\Delta\mathbf{f} \cdot \mathbf{t} < 0$ and the same arguments yield a strong inequality. The following lemma shows a stronger result, that there is actually a strictly positive lower bound on the step size that ensures this descent condition.

Lemma 6. *There exists $\lambda_0 > 0$ such that for all $0 < \lambda \leq \lambda_0$ and for all $\boldsymbol{\alpha}$ and A_p, every choice of $\Delta\boldsymbol{\alpha} \in \Theta^{\downarrow}_{\lambda}(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha})), \mathbf{t}'(\mathbf{f}(\boldsymbol{\alpha})) : A_p)$ satisfies*

$$\Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha}) \cdot \mathbf{t}\left(\mathbf{f} + \Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha})\right) \leq \mathbf{0} \tag{2.77}$$

Proof:
The main idea of the proof is as follows. Under current conditions the cost of every non-basic approach is not lower than the cost of the alternative basic approach. For some approaches this situation remains even after the change; therefore, their contribution to the directional derivative is negative (which is desirable). For those non-basic approaches that have lower cost after the change, the original cost difference cannot be too large, and hence the flow shift can not be too large. Then the contribution of such changes to the directional derivative although positive can not be too large in magnitude. Finally, if there is any substantial change in the flows, it must be due to some relatively large cost difference for some non-basic approach, which remains relatively large even after the change, thus contributing a negative component to the directional derivative. We show that for a sufficiently small step size, the latter negative component dominates the prior positive component.

In the descending aggregation map the shift from every non-basic approach is determined by an average approach cost based on the original approach proportions $\boldsymbol{\alpha}$, and a node flow based on the new approach proportions $\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}$. To simplify the notation in this proof we use the following abbreviations:

$$g_j = g_j(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}) \tag{2.78}$$

$$\mu_a = \mu_a(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha}))) \tag{2.79}$$

$$\hat{\mu}_a = \mu_a(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}))) \tag{2.80}$$

$$z_{a\to b} = z_{a\to b}(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha})), \mathbf{t}'(\mathbf{f}(\boldsymbol{\alpha}))) \tag{2.81}$$

$$\Delta\mathbf{f} = \Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha}) \tag{2.82}$$

Let x be the largest flow shift, in units of flow, from non-basic to basic approach, that is

$$x = \max_{j \in N \setminus \{p\}; a \in NB_j} -\Delta\alpha_a \cdot g_j(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}) \tag{2.83}$$

Remark: 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.

$\forall a \in NB$

$$\Delta f_a = \Delta\alpha_a \cdot g_{a_h} + \sum_{\substack{j \in N \\ o(j) > o(a_h)}} \sum_{a' \in NB_j} \Delta\alpha_{a'} \cdot g_j \cdot \alpha_a \cdot \left(\chi_{a_h \to a'_t} - \chi_{a_h \to (b_j)_t}\right) \tag{2.84}$$

$\forall b \in B$

$$\Delta f_b = -\sum_{a \in NB_{b_h}} \Delta\alpha_a \cdot g_{a_h} + \sum_{\substack{j \in N \\ o(j) > o(a_h)}} \sum_{a' \in NB_j} \Delta\alpha_{a'} \cdot g_j \cdot \alpha_b \cdot \left(\chi_{b_h \to a'_t} - \chi_{b_h \to (b_j)_t}\right) \tag{2.85}$$

By (2.8) $0 \leq \chi_{i \to j} \leq 1$; therefore, $\forall a \in A_p$

$$|\Delta f_a| \leq |A| \cdot x \tag{2.86}$$

Let $t'_{max} = \max\left\{t'_a(\mathbf{f}(\boldsymbol{\alpha})) : a \in A; \boldsymbol{\alpha} \in [0,1]^{|A_p|}\right\}$ be the maximum cost derivative; then

$$|t_a(\mathbf{f}(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha})) - t_a(\mathbf{f}(\boldsymbol{\alpha}))| \leq |\Delta f_a \cdot t'_{max}| \leq |A| \cdot x \cdot t'_{max} \tag{2.87}$$

$$|\hat{\mu}_a - \mu_a| \leq \left|\sum_{a \in A_p} \Delta f_a \cdot t'_{max}\right| \leq |A|^2 \cdot x \cdot t'_{max} \tag{2.88}$$

For all $j \in N \setminus \{p\}; a \in NB_j; b = b_j; \quad \mu_a \geq \mu_b$ hence

$$\hat{\mu}_a - \hat{\mu}_b \geq -2 \cdot |A|^2 \cdot x \cdot t'_{max} \tag{2.89}$$

For those $j \in N \setminus \{p\}; a \in NB_j; b = b_j$ such that $\hat{\mu}_a \geq \hat{\mu}_b$

$$g_j \cdot \Delta\alpha_a \cdot (\hat{\mu}_a - \hat{\mu}_b) \leq 0 \tag{2.90}$$

For those $j \in N \setminus \{p\}; a \in NB_j; b = b_j$ such that $\hat{\mu}_a < \hat{\mu}_b$

$$\begin{aligned}
\mu_a - \mu_b &\leq 2 \cdot |A|^2 \cdot x \cdot t'_{max} && (2.91)\\
z_{a\to b} &\leq 2 \cdot |A|^2 \cdot x \cdot t'_{max}/\epsilon_\nu && (2.92)\\
-\Delta\alpha_a \cdot g_j &\leq \lambda \cdot z_{a\to b} && (2.93)\\
g_j \cdot \Delta\alpha_a \left(\hat{\mu}_a - \hat{\mu}_b\right) &\leq 4\frac{\lambda}{\epsilon_\nu} \cdot |A|^4 \cdot x^2 \cdot {t'_{max}}^2 && (2.94)\\
\sum_{j\in N} \sum_{\substack{a\in NB_j \\ \hat{\mu}_a<\hat{\mu}_b}} g_j \cdot \Delta\alpha_a \cdot \left(\hat{\mu}_a - \hat{\mu}_b\right) &\leq 4\frac{\lambda}{\epsilon_\nu} \cdot |A|^5 \cdot x^2 \cdot {t'_{max}}^2 && (2.95)
\end{aligned}$$

There exist $j_0 \in N; a_0 \in NB_{j_0}; b_0 = b_{j_0}$ such that $x = -\Delta\alpha_{a_0} \cdot g_{j_0}$; hence

$$\begin{aligned}
x = -\Delta\alpha_{a_0} \cdot g_{j_0} &\leq \lambda \cdot z_{a_0\to b_0} \leq \lambda \cdot \left(\mu_{a_0} - \mu_{b_0}\right)/\epsilon_\nu && (2.96)\\
\mu_{a_0} - \mu_{b_0} &\geq x \cdot \frac{\epsilon_\nu}{\lambda} && (2.97)\\
\hat{\mu}_{a_0} - \hat{\mu}_{b_0} &\geq x \cdot \frac{\epsilon_\nu}{\lambda} - 2 \cdot x \cdot |A|^2 \cdot t'_{max} && (2.98)\\
g_{j_0} \cdot \Delta\alpha_{a_0} \cdot \left(\hat{\mu}_{a_0} - \hat{\mu}_{b_0}\right) &\leq -x^2 \cdot \left(\frac{\epsilon_\nu}{\lambda} - 2 \cdot |A|^2 \cdot t'_{max}\right) && (2.99)
\end{aligned}$$

where the transformation from (2.97) to (2.98) is by (2.88).

$$\begin{aligned}
\Delta\mathbf{f} \cdot \mathbf{t}(\mathbf{f} + \Delta\mathbf{f}) &= \sum_{j\in N} \sum_{a\in NB_j} g_j \cdot \Delta\alpha_a \cdot \left(\hat{\mu}_a - \hat{\mu}_b\right)\\
&\leq 4\frac{\lambda}{\epsilon_\nu} \cdot |A|^5 \cdot x^2 \cdot {t'_{max}}^2 - x^2 \cdot \left(\frac{\epsilon_\nu}{\lambda} - 2 \cdot |A|^2 \cdot t'_{max}\right)\\
&= x^2 \cdot \left(4\frac{\lambda}{\epsilon_\nu} \cdot |A|^5 \cdot {t'_{max}}^2 + 2 \cdot |A|^2 \cdot t'_{max} - \frac{\epsilon_\nu}{\lambda}\right) && (2.100)
\end{aligned}$$

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. □

Using λ_0 from Lemma 6, we define the algorithmic map for the boundary search as follows

$$\Theta^{\downarrow}(\boldsymbol{\alpha} : A_p) = \left\{ \Delta\boldsymbol{\alpha} : \begin{array}{l} \exists\lambda \in [\lambda_0, 1] : \Delta\boldsymbol{\alpha} \in \Theta^{\downarrow}_{\lambda}(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha})), \mathbf{t}'(\mathbf{f}(\boldsymbol{\alpha})) : A_p) \\ \Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha}) \cdot \mathbf{t}\left(\mathbf{f} + \Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha})\right) \leq \mathbf{0} \end{array} \right\} \quad (2.101)$$

According to this definition, one can choose a step size of λ_0 at every iteration. Apparently the proof of convergence is valid even with such an inefficient choice of

step size. This map also allows for the approximated search procedure described above; that is, to consider $\lambda = 2^{-k}$ and choose the largest step size that leads to a value of $\Delta\boldsymbol{\alpha}$ that satisfies the above descent condition. In cases where the map for a certain step size allows for several options, it is sufficient to examine one of them arbitrarily.

By Lemma 6, the map $\Theta^{\downarrow}$ is non-empty. The map $\Theta^{\downarrow}$ is closed, since if $\Delta\boldsymbol{\alpha}^k \in \Theta^{\downarrow}_{\lambda^k}(\boldsymbol{\alpha}^k : A_p)$,$\boldsymbol{\alpha}^k \to \boldsymbol{\alpha}$,$\Delta\boldsymbol{\alpha}^k \to \Delta\boldsymbol{\alpha}$, then for some subsequence $\lambda^k \to \lambda \in [\lambda_0, 1]$ and $\Delta\boldsymbol{\alpha} \in \Theta^{\downarrow}_{\lambda}(\boldsymbol{\alpha} : A_p)$.

Lemma 7. *If* $\Delta\boldsymbol{\alpha} \in \Theta^{\downarrow}_{\lambda}(\boldsymbol{\alpha} : A_p)$ *then* $T(\mathbf{f}(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha})) \leq T(\mathbf{f}(\boldsymbol{\alpha}))$ *and equality holds iff* $\Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha}) = 0$.

Proof:
The same arguments used to show (2.74) are valid here, since if $\Delta\mathbf{f}(\boldsymbol{\alpha}, \Delta\boldsymbol{\alpha}) \neq 0$ then $\Delta\mathbf{f} \cdot \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha})) < 0$, and $\Delta\mathbf{f} \cdot \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha})) \leq 0$. □

The next theorem proves convergence of the algorithm described above for RSOTP.

Theorem 1. *The algorithm defined by* $\Theta^{\downarrow}(\boldsymbol{\alpha} : A_p)$ *converges to restricted equilibrium.*

Proof:
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 exist a subsequence K such that $\boldsymbol{\alpha}^{k+l} \to \boldsymbol{\alpha}^{*l}$ $\forall 0 \leq l \leq |N|$. Hence

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

$$\begin{array}{ccccccccc}
\cdots & \boldsymbol{\alpha}^{k_0+1} & \to & \boldsymbol{\alpha}^{k_0+2} & \cdots \to & \boldsymbol{\alpha}^{k_0+l} & \cdots \to & \boldsymbol{\alpha}^{k_0+|N|} & \cdots \\
\cdots & \boldsymbol{\alpha}^{k_1+1} & \to & \boldsymbol{\alpha}^{k_1+2} & \cdots \to & \boldsymbol{\alpha}^{k_1+l} & \cdots \to & \boldsymbol{\alpha}^{k_1+|N|} & \cdots \\
 & \vdots & & \vdots & & \vdots & & \vdots & \\
\cdots & \boldsymbol{\alpha}^{k+1} & \to & \boldsymbol{\alpha}^{k+2} & \cdots \to & \boldsymbol{\alpha}^{k+l} & \cdots \to & \boldsymbol{\alpha}^{k+|N|} & \cdots \\
 & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\
 & \boldsymbol{\alpha}^{*1} & \to & \boldsymbol{\alpha}^{*2} & \cdots \to & \boldsymbol{\alpha}^{*l} & \cdots \to & \boldsymbol{\alpha}^{*|N|} &
\end{array} \tag{2.102}$$

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 non-increasing sequence, hence the limiting point of every subsequence is equal, and thus

$$T(\mathbf{f}(\boldsymbol{\alpha}^{*l})) = T^* \qquad \forall 0 \leq l \leq |N| \tag{2.103}$$

Therefore, $\Delta\mathbf{f}(\boldsymbol{\alpha}^{*l}, \Delta\boldsymbol{\alpha}^{*l}) = 0$;

$$\mathbf{f}(\boldsymbol{\alpha}^{*l}) = \mathbf{f}^* \qquad \forall 0 \leq l \leq |N| \tag{2.104}$$

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 restricted approach equilibrium conditions (2.46) for 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 non-zero components only for links terminating at nodes of topological order higher then l. Therefore equilibrium conditions will remain for all nodes of topological order no greater then 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 $\Delta\mathbf{f}(\boldsymbol{\alpha}^{*l}, \Delta\boldsymbol{\alpha}^{*l}) \neq 0$, contradiction. If $g_j = 0$ then $\Delta\alpha_a^{*l} = -\alpha_a^{*l}$ and therefore $\alpha_a^{*l+1} = 0$ and the approach equilibrium conditions are met. □

2.8 Restrictions update

So far the restricting subnetwork was assumed given. Determining a restricting subnetwork that includes the globally optimal solution is not a trivial task, and an iterative scheme seems necessary. A reasonable initial guess may be the tree of minimum cost routes under free flow travel conditions; however, the only feasible assignment under such restrictions is the all-or-nothing assignment. This section presents a simple procedure to determine a new restricting subnetwork based on the current approach proportions solution. This procedure is used to define the algorithmic map for a full unrestricted iteration; a proof of global convergence for the resulting algorithm follows.

It was assumed that the restricting subnetwork is a-cyclic and spanning. The tree of minimum cost routes does satisfy these requirements. We will show that given any feasible solution the proposed procedure produces a spanning a-cyclic restricting subnetwork. Another useful property of this procedure is that the new restricted feasible set contains the current solution. Without this property the solution must be modified while updating the restrictions. Unless carefully performed, such modifications may increase the objective function value, and thus disturb the global convergence of the algorithm.

In the description of the procedure we distinguish between the following two terms. A link a is considered to be a *used link* if its flow is strictly positive, $f_a > 0$. It is considered to be a *contributing link* if $\boldsymbol{\alpha}_a > 0$. The *used subnetwork* is defined accordingly by $A_p^u(\mathbf{f}) = \{a \in A : f_a > 0\}$, and the *contributing subnetwork* is defined by $A_p^c(\boldsymbol{\alpha}) = \{a \in A : \alpha_a > 0\}$. Clearly every used link is also a contributing one, but not necessarily vice versa: even if all approaches to a certain node carry zero flow, approach proportions must still sum to one. One advantage of the contributing

subnetwork over the used subnetwork is that the former is spanning; i.e., it contains at least one route from the origin to every other node, while the latter is not. To some extent this is only a technical issue; however, it does simplify the discussion substantially.

In order for the new restricted feasible set to include the current solution, the new restricting subnetwork must include the contributing subnetwork. Feasibility requirements ensure that the contributing subnetwork $A_p^c(\boldsymbol{\alpha})$ is spanning and a-cyclic. Taking the contributing subnetwork $A_p^c(\boldsymbol{\alpha})$ as the basis for the new restricting subnetwork guarantees it is spanning; the remaining question is which links if any should be added in order to allow improved solutions. Links from the current tree of minimum cost routes are possible candidates, but determining whether any of them creates a cycle and which ones should be excluded can be a rather cumbersome and time consuming operation.

The proposed condition is based on the *maximum contributing cost* from the origin p to node j, defined as

$$u_j(\boldsymbol{\alpha}, \mathbf{t}) = \max_{r \in R_{pj}\left[A_p^c(\boldsymbol{\alpha})\right]} c_r(\mathbf{t}) = \max_{r \in R_{pj}\left[A_p^c(\boldsymbol{\alpha})\right]} \sum_{a \in r} t_a \tag{2.105}$$

In this definition the only role of $\boldsymbol{\alpha}$ is to determine which routes to consider, while the cost of each of these routes is based on some link costs $\mathbf{t}$ that do not necessarily depend on $\boldsymbol{\alpha}$. This flexibility in the definition is used mainly for proving convergence. When updating the restrictions, consistent approach proportions and link costs are used. Once the value of $u_j = u_j(\boldsymbol{\alpha}, \mathbf{t}(\mathbf{f}(\boldsymbol{\alpha})))$ is computed for all the nodes, every link $[i,j] \in A$ such that $u_i < u_j$ is added to the restricting subnetwork. The resulting restricting subnetwork is

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

This condition is relatively easy to verify since maximum contributing costs to all nodes can be computed in a single ascending pass over the nodes. Note that the contributing subnetwork is spanning; hence the maximum contributing cost is well defined for every node.

Comment: A similar definition of maximum *used* cost is valid only for used nodes. Hagstrom (1997) proposed a rigorous but quite involved method to extend the definition of maximum used cost to all nodes. Algorithm performance is expected to be similar whether the condition for adding links is based on either maximum contributing cost or extended maximum used cost. The condition based on maximum

contributing cost is preferred as it simplifies the discussion and the code implementation.

Lemma 8. *$\mathcal{A}_p(\boldsymbol{\alpha})$ is a-cyclic.*

Proof:
Suppose that $[v_0, v_1, \ldots, v_n = v_0]$ is a cycle such that $[v_k, v_{k+1}] \in \mathcal{A}_p(\boldsymbol{\alpha}) \quad \forall 0 \leq k \leq n-1$. 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 $u_{v_k} \leq u_{v_k} + t_{[v_k, v_{k+1}]} \leq u_{v_{k+1}}$. Since $A_p^c(\boldsymbol{\alpha})$ is a-cyclic 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 $u_{v_0} \leq u_{v_1} \leq \ldots \leq u_{v_{k_0}} < u_{v_{k_0+1}} \leq \ldots \leq u_{v_n} = u_{v_0}$, contradiction. □

Comment: Dial (1971) defines *efficient links* by a similar condition, using minimum cost rather than maximum contributing cost. Intuitively Dial's condition is more appealing. Indeed when applied once, the subnetwork of efficient links is a-cyclic. In our case restrictions are updated iteratively. Efficient links by current solution link costs are likely to create cycles with both the contributing and the used subnetworks.

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

$$\Theta^{\downarrow}(\boldsymbol{\alpha}) = \Theta^{\downarrow}(\boldsymbol{\alpha} : \mathcal{A}_p(\boldsymbol{\alpha})) \tag{2.106}$$

Theorem 2. *If $\boldsymbol{\alpha}^1$ is a feasible solution for 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 TAP.*

The first part of this proof is similar to the proof of Theorem 1. The main difference is that in the previous proof it was sufficient to show that the limiting sequence is 'well behaved', while here we must show that from some point onwards every sequence of consecutive iterations is 'well behaved'. This part of the proof becomes somewhat less cumbersome under the assumption that link costs are strictly positive. A proof under this assumption is given first, followed by a proof for the general case where zero link costs are allowed. The first part of the proof is common to both assumptions.

Proof:
The number of possible restricting subnetworks is finite, and therefore in any sequence of restricting subnetworks there is one subnetwork that appears infinitely many times. The feasible set is compact and therefore every sequence has a converging

subsequence. Applying these arguments $|N|$ times yields a series of $|N|$ consecutive subsequences of the form:

$$\begin{array}{ccccccccc}
\cdots & \boldsymbol{\alpha}^{k_0+1} & \rightarrow & \boldsymbol{\alpha}^{k_0+2} & \cdots \rightarrow & \boldsymbol{\alpha}^{k_0+l} & \cdots \rightarrow & \boldsymbol{\alpha}^{k_0+|N|} & \cdots \\
\cdots & \boldsymbol{\alpha}^{k_1+1} & \rightarrow & \boldsymbol{\alpha}^{k_1+2} & \cdots \rightarrow & \boldsymbol{\alpha}^{k_1+l} & \cdots \rightarrow & \boldsymbol{\alpha}^{k_1+|N|} & \cdots \\
 & \vdots & & \vdots & & \vdots & & \vdots & \\
\cdots & \boldsymbol{\alpha}^{k+1} & \rightarrow & \boldsymbol{\alpha}^{k+2} & \cdots \rightarrow & \boldsymbol{\alpha}^{k+l} & \cdots \rightarrow & \boldsymbol{\alpha}^{k+|N|} & \cdots \\
 & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\
 & \boldsymbol{\alpha}^{*1} & \rightarrow & \boldsymbol{\alpha}^{*2} & \cdots \rightarrow & \boldsymbol{\alpha}^{*l} & \cdots \rightarrow & \boldsymbol{\alpha}^{*|N|} &
\end{array} \tag{2.107}$$

such that each subsequence converges, and for every subsequence the restricting subnetwork produced by all solutions is the same. Formally, there exists a subsequence K such that $\forall k \in K$, $\forall l : 1 \leq l \leq |N|$

$$\mathcal{A}_p(\boldsymbol{\alpha}^{k+l}) = A_p^{*l} \tag{2.108}$$

$$\boldsymbol{\alpha}^{k+l} \rightarrow \boldsymbol{\alpha}^{*l} \tag{2.109}$$

As a result $\boldsymbol{\alpha}^{k+l+1} - \boldsymbol{\alpha}^{k+l} = \Delta\boldsymbol{\alpha}^{k+l} \rightarrow \Delta\boldsymbol{\alpha}^{*l} = \boldsymbol{\alpha}^{*l+1} - \boldsymbol{\alpha}^{*l} \quad \forall l : 1 \leq l < |N|$. 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 $\Delta\boldsymbol{\alpha}^{*l} \in \Theta^{\downarrow}\left(\boldsymbol{\alpha}^{*l} : A_p^{*l}\right)$.

In every iteration the objective function value either decreases or remains the same. $T\left(\boldsymbol{\alpha}^{k}\right)$ is therefore a bounded monotonically non-increasing series; hence it converges to some value T^*, in particular $T\left(\boldsymbol{\alpha}^{*l}\right) = T^*$. To prove that T^* is the global unrestricted minimum value of the objective function it is sufficient to show that $\boldsymbol{\alpha}^{*|N|}$ is an equilibrium solution for TAP. Since $T\left(\boldsymbol{\alpha}^{k}\right) \rightarrow T^*$ for every subsequence K', showing that T^* is a global minimum proves that every limit point of $\left\{\boldsymbol{\alpha}^{k}\right\}$ is an equilibrium solution for TAP.

By Lemma 7 $T\left(\boldsymbol{\alpha}^{*l}\right) = T\left(\boldsymbol{\alpha}^{*l+1}\right)$ only if $\Delta\mathbf{f}\left(\boldsymbol{\alpha}^{*l}, \Delta\boldsymbol{\alpha}^{*l}\right) = 0$ and therefore $\mathbf{f}\left(\boldsymbol{\alpha}^{*l}\right) = \mathbf{f}^*$ for every $l : 1 \leq l \leq |N|$. Eventually it is shown that these are the equilibrium link flows, but it is important that we do not make such an assumption during the proof; therefore, 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

$$w_j^* = \min_{r \in R_{pj}} c_r(\mathbf{t}^*) \tag{2.110}$$

consider the subnetwork

$$A_p^* = \left\{ a \in A : w_{a_t}^* + t_a^* = w_{a_h}^* \right\} \tag{2.111}$$

and denote

$$R_{ij}^* = R_{ij}[A_p^*] \tag{2.112}$$

$$R^* = \bigcup_{i,j \in N} R_{ij}^* \tag{2.113}$$

It follows that these are the sets of minimum cost routes under the limiting conditions, in particular $r \in R_{pj}^* \iff c_r(\mathbf{t}^*) = w_j^*$. Since A is finite there exists a strictly positive value $\epsilon > 0$ such that

$$\forall a \in A \setminus A_p^* : \quad w_{a_t}^* + t_a^* > w_{a_h}^* + \epsilon \tag{2.114}$$

Since $\mathbf{f}(\boldsymbol{\alpha}^{k+l}) \to \mathbf{f}^*$ there exists k_0 such that $\forall k \in K; k \geq k_0; \forall l : 1 \leq l \leq |N|$;

$$\begin{aligned} &\forall a \in A : && \left| f_a^{k+l} - f_a^* \right| < \tfrac{\lambda_0 \cdot \epsilon}{4 \cdot |A| \cdot t'_{max}} \\ &\forall r \in R : && \left| c_r\left(\mathbf{f}\left(\boldsymbol{\alpha}^{k+l}\right)\right) - c_r\left(\mathbf{f}^*\right) \right| < \tfrac{\epsilon}{4} \end{aligned}$$

where $t'_{max} = \max_{\boldsymbol{\alpha}} \max_{a \in A} \{ t'_a(\mathbf{f}(\boldsymbol{\alpha})) \}$.

Assumption A: $\mathbf{t} > 0$ (link costs are *strictly* positive)

We want to show that every contributing route in $\boldsymbol{\alpha}^{k+|N|}$ for all $k \in K; k > k_0$ is a "good" route, in the sense that it is included in A_p^*. This does not necessarily mean that it is a route of minimum cost for the current solution, since $\mathbf{t}(\mathbf{f}(\boldsymbol{\alpha}^{k+|N|}))$ and $\mathbf{t}^*$ can be slightly different. However, if this is true for all $k \in K; k > k_0$, it is also true in the limit. Hence $\boldsymbol{\alpha}^{*|N|}$ is an equilibrium solution for TAP.

For that purpose we are interested to know at every iteration $\boldsymbol{\alpha}^{k+l}$ which are the "good nodes", i.e. those nodes that all contributing routes to them are "good" routes. Formally we define

$$\breve{N}^{k+l} = \left\{ j \in N : R_{pj}[A_p^c(\boldsymbol{\alpha}^{k+l})] \subseteq R_{pj}^* \right\} \tag{2.115}$$

which is considered the set of temporary "good nodes" since it is possible that nodes enter and leave this set from one iteration to the next within a specific sequence (row).

Comment: suppose $k_1, k_2 \in K; k_0 < k_1; k_1 + |N| < k_2$. Showing that all contributing routes are "good" at iteration $k_1 + |N|$ does not immediately guarantee anything

about iteration $k_2 + 1$, since we do not know anything about the conditions in the iterations between, except for the fact that the objective function cannot increase. The only iterations for which we monitor flows and costs are those included in one of the sequences $\{k + l : 1 \leq l \leq |N|\}$ for some $k \in K; k \geq k_0$.

Under assumption A, that link costs are strictly positive, A_p^* is a-cyclic and has a topological order o^*. Using this topological order we can define the set

$$\breve{\breve{N}}^{k+l} = \left\{ j \in \breve{N}^{k+l} : \forall i \in N; o^*(i) < o^*(j) \Rightarrow i \in \breve{N}^{k+l} \right\} \tag{2.116}$$

which is considered the set of permanent "good nodes", since it will be shown that $\breve{\breve{N}}^{k+l} \subseteq \breve{\breve{N}}^{k+l+1}$.

Comment: the following parts of the proof are rather intensive in notation. To help tracking we adopt the following convention, variables with smile above $(\breve{i}, \breve{r})$ are associated with "good" routes, while variables with frown above $(\hat{i}, \hat{r})$ are associated with "bad" routes.

We show by induction on l that $\forall k \in K; k \geq k_0$; $\left|\breve{\breve{N}}^{k+l}\right| \geq l$, in particular $\breve{\breve{N}}^{k+|N|} = N$. Hence

$$\begin{aligned} R_{pj}[A_p^c(\boldsymbol{\alpha}^{k+|N|})] &\subseteq R_{pj}^* \qquad \forall j \in N; \forall k \in K : k > k_0 \\ R_{pj}[A_p^c(\boldsymbol{\alpha}^{*|N|})] &\subseteq R_{pj}^* \qquad \forall j \in N \end{aligned}$$

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, $\left|\breve{\breve{N}}^{k+l}\right| \geq l$, show for $l+1$. Suppose $\left|\breve{\breve{N}}^{k+l+1}\right| \leq l < N$, let

$$j = \operatorname{argmin} \left\{ o^*(j') : j' \in N \setminus \breve{\breve{N}}^{k+l+1} \right\} \tag{2.117}$$

By the choice of j, for every $i \in N : o^*(i) < o^*(j) \Rightarrow i \in \breve{\breve{N}}^{k+l+1}$. So $o^*(j) - 1 = \left|\breve{\breve{N}}^{k+l+1}\right| \leq l$ 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 $j \notin \breve{N}^{k+l+1}$. This means that there exists a "bad" contributing route to j

$$\hat{r} = \hat{s} + [\hat{i}, j] \in R_{pj}[A_p^c(\boldsymbol{\alpha}^{k+l+1})] \setminus R_{pj}^* \tag{2.118}$$

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 $\hat{i} \in N \setminus \breve{\breve{N}}^{k+l+1}$; but $[\hat{i}, j] \in A_p^*$ also implies that $o^*(i) < o^*(j)$, in contradiction to the choice of j; hence $[\hat{i}, j] \notin A_p^*$.

Case 1: $j \in \breve{\breve{N}}^{k+l}$

$$[\hat{i}, j] \notin A_p^* \Rightarrow [\hat{i}, j] \notin A_p^c(\boldsymbol{\alpha}^{k+l}) \Rightarrow \alpha_{[\hat{i},j]}^{k+l} = 0 \tag{2.119}$$

Let $\breve{a} = [\breve{i}, j]$ be any contributing approach, i.e. $[\breve{i}, j] \in A_p^c(\boldsymbol{\alpha}^{k+l})$. Since $j \in \breve{\breve{N}}^{k+l}$, every contributing route is a "good route"; hence, for every contributing approach $\mu_{[\breve{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) = w_j^*$ and $\mu_{[\breve{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) < w_j^* + \frac{\epsilon}{4}$.

On the other hand, $[\hat{i}, j] \notin A_p^*$; therefore, $w_{\hat{i}}^* + t_{[\hat{i},j]}^* > w_j^* + \epsilon$ and hence

$$\begin{aligned} \mu_{[\hat{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) &> \mu_{[\hat{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) - \frac{\epsilon}{4} \geq w_{\hat{i}}^* + t_{[\hat{i},j]}^* - \frac{\epsilon}{4} > w_j^* + \frac{3\epsilon}{4} \\ \mu_{[\hat{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) &> \mu_{[\breve{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) \end{aligned} \tag{2.120}$$

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 (2.118), showing that indeed $\breve{\breve{N}}^{k+l} \subseteq \breve{\breve{N}}^{k+l+1}$ as proposed earlier.

Case 2: $j \notin \breve{\breve{N}}^{k+l}$
$[\hat{i}, j] \notin A_p^*$ hence $w_{\hat{i}}^* + t_{[\hat{i},j]}^* > w_j^* + \epsilon$; therefore

$$\mu_{[\hat{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) > \mu_{[\hat{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) - \frac{\epsilon}{4} \geq w_{\hat{i}}^* + t_{[\hat{i},j]}^* - \frac{\epsilon}{4} > w_j^* + \frac{3\epsilon}{4} \tag{2.121}$$

In contrast we show that the average cost of the basic approach is not greater than $w_j^* + \frac{\epsilon}{4}$. A_p^* is spanning; hence, there is $[\breve{i}, j] \in A_p^*$, $o^*(\breve{i}) < o^*(j) \leq l + 1$. By

the induction assumption $\left|\breve{\breve{N}}^{k+l}\right| \geq l$, so $o^*(\breve{i}) \leq l$ implying that $\breve{i} \in \breve{\breve{N}}^{k+l}$; hence $\mu_{[\breve{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) = w^*_{\breve{i}} + t^*_{[\breve{i},j]} = w^*_j$.

If $u_{\breve{i}}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) < u_j(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l}))$, then the link $[\breve{i}, j]$ must be in the restricting subnetwork A^{*l}_p. Hence the average cost of the basic approach can not be higher than the average cost of the $[\breve{i}, j]$ approach; that is

$$\mu_{b_j}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) \leq \mu_{[\breve{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) < \mu_{[\breve{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) + \frac{\epsilon}{4} = w^*_j + \frac{\epsilon}{4} \tag{2.122}$$

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

$$\begin{aligned} \mu_{b_j}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) &\leq u_j(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) \leq u_{\breve{i}}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) \\ &< u_{\breve{i}}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) + \frac{\epsilon}{4} = w^*_{\breve{i}} + \frac{\epsilon}{4} \leq w^*_j + \frac{\epsilon}{4} \end{aligned} \tag{2.123}$$

Let $\hat{a} = [\hat{i}, j]$; then

$$\mu_{\hat{a}}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) - \mu_{b_j}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) > \frac{\epsilon}{2} \tag{2.124}$$

$$z_{\hat{a}\to b_j}(\boldsymbol{\alpha}, \mathbf{t}(\boldsymbol{\alpha}^{k+l}), \mathbf{t}'(\boldsymbol{\alpha}^{k+l})) > \frac{\epsilon}{2 \cdot |A| \cdot t'_{max}} > 0 \tag{2.125}$$

where $t'_{max} = \max_{\boldsymbol{\alpha}} \max_{a \in A} \{t'_a(\mathbf{f}(\boldsymbol{\alpha}))\}$ The desirable shift is first scaled by the step size, and then truncated only if non-negativity is about to be violated. $\alpha_{\hat{a}}$ can remain a contributing approach only if the scaled desirable shift is applied as is and nothing is truncated. Since the cost difference and the desirable shift are strictly positive, this can only happen if the amount of flow in the approach is greater than the scaled desirable shift. In such a case

$$-\Delta\alpha^{k+l}_{\hat{a}} = \lambda^{k+l} \cdot \frac{z_{\hat{a}\to b_j}(\boldsymbol{\alpha}, \mathbf{t}(\boldsymbol{\alpha}^{k+l}), \mathbf{t}'(\boldsymbol{\alpha}^{k+l}))}{g_j(\boldsymbol{\alpha}^{k+l+1})} \geq \frac{\lambda_0 \cdot \epsilon}{2 \cdot |A| \cdot t'_{max} \cdot g_j(\boldsymbol{\alpha}^{k+l+1})} \tag{2.126}$$

so the actual shift can not be too small. This shift is aggregated with other shifts; however, shifts due to other nodes change the flow on $\hat{a}$ and b_j in the same direction, while the shift due to node j decrease the flow on $\hat{a}$ and increase the flow on b_j. In other words, if $g_j(\boldsymbol{\alpha}^{k+l+1}) \leq g_j(\boldsymbol{\alpha}^{k+l})$ then

$$\begin{aligned} \Delta f_{\hat{a}}(\boldsymbol{\alpha}^{k+l}, \Delta\boldsymbol{\alpha}^{k+l}) &= \Delta\alpha^{k+l}_{\hat{a}} \cdot g_j(\boldsymbol{\alpha}^{k+l+1}) + \alpha^{k+l}_{\hat{a}} \cdot \left(g_j(\boldsymbol{\alpha}^{k+l+1}) - g_j(\boldsymbol{\alpha}^{k+l})\right) \\ &\leq -\frac{\lambda_0 \cdot \epsilon}{2 \cdot |A| \cdot t'_{max}} \end{aligned} \tag{2.127}$$

while if $g_j(\boldsymbol{\alpha}^{k+l+1}) > g_j(\boldsymbol{\alpha}^{k+l})$ then

$$\Delta f_{b_j}(\boldsymbol{\alpha}^{k+l}, \Delta\boldsymbol{\alpha}^{k+l}) = \sum_{a \in NB_j} \Delta\alpha_a^{k+l} \cdot g_j(\boldsymbol{\alpha}^{k+l+1}) + \alpha_{b_j}^{k+l} \cdot \left(g_j(\boldsymbol{\alpha}^{k+l+1}) - g_j(\boldsymbol{\alpha}^{k+l})\right)$$
$$\geq \Delta\alpha_{\hat{a}}^{k+l} \cdot g_j(\boldsymbol{\alpha}^{k+l+1}) \geq \frac{\lambda_0 \cdot \epsilon}{2 \cdot |A| \cdot t'_{max}} \tag{2.128}$$

Therefore, there is at least one link in which the actual aggregated change of flow is at least $\frac{\lambda_0 \cdot \epsilon}{2 \cdot |A| \cdot t'_{max}}$ in magnitude. On the other hand k_0 was chosen so that for all $k \in K; k \geq k_0; \forall 1 \leq l \leq |N|; \forall a \in A$

$$|\Delta f_a(\boldsymbol{\alpha}^{k+l}, \Delta\boldsymbol{\alpha}^{k+l})| \leq |f_a(\boldsymbol{\alpha}^{k+l}) - f_a^*| + |f_a(\boldsymbol{\alpha}^{k+l+1}) - f_a^*| < \frac{\lambda_0 \cdot \epsilon}{2 \cdot |A| \cdot t'_{max}} \tag{2.129}$$

and this is a contradiction. End of proof (assumption A: $\mathbf{t} > 0$).

Assumption B: $\mathbf{t} \geq 0$ (allow zero link costs)

As under assumption A, we consider the set of temporary "good nodes"

$$\breve{N}^{k+l} = \left\{ j \in N : R_{pj}[A_p^c(\boldsymbol{\alpha}^{k+l})] \subseteq R_{pj}^* \right\} \tag{2.130}$$

In this case, since link costs may be zero, A_p^* may contain cycles; hence, we can not assume that there is a topological order for this subnetwork. This is the main difficulty in proving the theorem under assumption B.

We show that a set of permanent "good nodes" can be defined using the minimum limiting cost values $\mathbf{w}^*$ as follows

$$\breve{\breve{N}}^{k+l} = \left\{ j \in \breve{N}^{k+l} : \forall i \in N; w_i^* < w_j^* \Rightarrow i \in \breve{N}^{k+l} \right\} \tag{2.131}$$

As before, we show by induction on l that $\forall k \in K; k \geq k_0; \left|\breve{\breve{N}}^{k+l}\right| \geq l$; in particular, $\breve{\breve{N}}^{|N|} = N$ and hence

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

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

For $l=1$, $p \in \breve{\breve{N}}^{k+1}$. Suppose $\left|\breve{\breve{N}}^{k+l}\right| \geq l$. First we show that $\breve{\breve{N}}^{k+l} \subseteq \breve{\breve{N}}^{k+l+1}$, then we show that $\breve{\breve{N}}^{k+l} \subsetneq N \Rightarrow \breve{\breve{N}}^{k+l} \subsetneq \breve{\breve{N}}^{k+l+1}$.

Show $\breve{\breve{N}}^{k+l} \subseteq \breve{\breve{N}}^{k+l+1}$. Suppose the opposite and let

$$\mathcal{J}_0 = \operatorname{argmin}\left\{w_j^* : j \in \breve{\breve{N}}^{k+l} \setminus \breve{\breve{N}}^{k+l+1}\right\} \tag{2.132}$$

$$j = \operatorname{argmin}\left\{o^{*l}(j') : j' \in \mathcal{J}_0\right\} \tag{2.133}$$

where o^{*l} is a topological order for the restricting subnetwork A_p^{*l}. Since $j \in \mathcal{J}_0 \subseteq \breve{\breve{N}}^{k+l}$, for every $i \in N : w_i^* < w_j^* \Rightarrow i \in \breve{\breve{N}}^{k+l}$. On the other hand since $\mathcal{J}_0$ is defined as the argmin, $w_i^* < w_j^*$ implies that $i \notin \breve{\breve{N}}^{k+l} \setminus \breve{\breve{N}}^{k+l+1}$. In conclusion, $i \in N : w_i^* < w_j^* \Rightarrow i \in \breve{\breve{N}}^{k+l+1}$. Therefore, 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 $j \notin \breve{N}^{k+l+1}$. Since $j \notin \breve{N}^{k+l+1}$, there exists a "bad" contributing route to j

$$\hat{r} = \hat{s} + [\hat{i}, j] \in R_{pj}[A_p^c(\boldsymbol{\alpha}^{k+l+1})] \setminus R_{pj}^* \tag{2.134}$$

Consider two cases, either $\alpha_{[\hat{i},j]}^{k+l} > 0$ or $\alpha_{[\hat{i},j]}^{k+l} = 0$.

Case 1: $\alpha_{[\hat{i},j]}^{k+l} > 0 \iff [\hat{i}, j] \in A_p^c(\boldsymbol{\alpha}^{k+l})$

Since $j \in \breve{\breve{N}}^{k+l}$, for every $s \in R_{p\hat{i}}[A_p^c(\boldsymbol{\alpha}^{k+l})]$, $s+[\hat{i}, j] \in R_{pj}[A_p^c(\boldsymbol{\alpha}^{k+l})] \subseteq R_{pj}^*$; therefore, $s \in R_{p\hat{i}}^*$ and $[\hat{i}, j] \in A_p^*$. So $R_{p\hat{i}}[A_p^c(\boldsymbol{\alpha}^{k+l})] \subseteq R_{p\hat{i}}^*$ hence $\hat{i} \in \breve{N}^{k+l}$. $[\hat{i}, j] \in A_p^*$ implies $w_{\hat{i}}^* \leq w_j^*$. Since $w_{\hat{i}}^* \leq w_j^*$, for every $i \in N$ such that $w_i^* < w_{\hat{i}}^*$ then $w_i^* < w_j^*$, and since in addition $j \in \breve{\breve{N}}^{k+l}$ we find that $i \in \breve{N}^{k+l}$. Combining that with the fact that $\hat{i} \in \breve{N}^{k+l}$ we get that $\hat{i} \in \breve{\breve{N}}^{k+l}$; i.e. $\hat{i}$ is a permanent "good node".

Now $\hat{r} = \hat{s} + [\hat{i}, j] \notin R_{pj}^*$, but $[\hat{i}, j] \in A_p^*$; therefore $\hat{s} \notin R_{p\hat{i}}^*$. Since $\hat{s} \in R_{p\hat{i}}[A_p^c(\boldsymbol{\alpha}^{k+l+1})]$ we obtain that $\hat{i} \notin \breve{N}^{k+l+1}$ and in particular $\hat{i} \notin \breve{\breve{N}}^{k+l+1}$. We found that $\hat{i} \in \breve{\breve{N}}^{k+l} \setminus \breve{\breve{N}}^{k+l+1}$, and also that $w_{\hat{i}}^* \leq w_j^* = \min\left\{w_j^* : j \in \breve{\breve{N}}^{k+l} \setminus \breve{\breve{N}}^{k+l+1}\right\}$; therefore, $\hat{i} \in \mathcal{J}_0$.

Finally, since $j = \operatorname{argmin}\left\{o^{*l}(j') : j' \in \mathcal{J}_0\right\}$, $\hat{i} \in \mathcal{J}_0$ implies that $o^{*l}(j) < o^{*l}(\hat{i})$, but o^{*l} is a topological order of A_p^{*l} and $[\hat{i}, j] \in A_p^c(\boldsymbol{\alpha}^{k+l}) \subseteq A_p^{*l}$, so by the definition of topological order $o^{*l}(\hat{i}) < o^{*l}(j)$, which is a contradiction.

The arguments for Case 1 can be summarized as follows:

$$j \in \breve{\breve{N}}^{k+l}; [\hat{i}, j] \in A_p^c(\boldsymbol{\alpha}^{k+l}) \Rightarrow [\hat{i}, j] \in A_p^*; \hat{i} \in \breve{N}^{k+l} \tag{2.135}$$

$$[\hat{i}, j] \in A_p^* \Rightarrow w_{\hat{i}}^* \leq w_j^* \tag{2.136}$$

$$\hat{i} \in \breve{N}^{k+l}; w_{\hat{i}}^* \leq w_j^*; j \in \breve{\breve{N}}^{k+l} \Rightarrow \hat{i} \in \breve{\breve{N}}^{k+l} \tag{2.137}$$

$$\hat{r} = \hat{s} + [\hat{i}, j] \notin R_{pj}^*; [\hat{i}, j] \in A_p^* \Rightarrow \hat{s} \notin R_{p\hat{i}}^* \tag{2.138}$$

$$\hat{s} \notin R_{p\hat{i}}^*; \hat{s} \in R_{p\hat{i}}[A_p^c(\boldsymbol{\alpha}^{k+l+1})] \Rightarrow \hat{i} \notin \breve{N}^{k+l+1}$$

$$\Rightarrow \hat{i} \notin \breve{\breve{N}}^{k+l+1} \tag{2.139}$$

$$\hat{i} \in \breve{\breve{N}}^{k+l} \setminus \breve{\breve{N}}^{k+l+1}; w_{\hat{i}}^* \leq w_j^* \Rightarrow \hat{i} \in \mathcal{J}_0 \tag{2.140}$$

$$\hat{i} \in \mathcal{J}_0; j = \operatorname{argmin}\left\{o^{*l}(j') : j' \in \mathcal{J}_0\right\} \Rightarrow o^{*l}(\hat{i}) > o^{*l}(j) \tag{2.141}$$

$$[\hat{i}, j] \in A_p^c(\boldsymbol{\alpha}^{k+l}) \subseteq A_p^{*l} \Rightarrow o^{*l}(\hat{i}) < o^{*l}(j) \tag{2.142}$$

and (2.141) contradicts (2.142).

Case 2: $\alpha_{[\hat{i},j]}^{k+l} = 0 \iff [\hat{i}, j] \notin A_p^c(\boldsymbol{\alpha}^{k+l})$

$[\hat{i}, j]$ was not a contributing approach at iteration $k+l$, but it became a contributing approach at iteration $k+l+1$. Therefore, link $[\hat{i}, j]$ is a new link which was added to the restricting subnetwork at this iteration, meaning that it satisfied the condition $u_i(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) < u_j(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l}))$.

For $[\hat{i}, j]$ to be a contributing approach at iteration $k+l+1$ it must be in the restricting subnetwork A_p^{*l}, but this is not sufficient. Since $\alpha_{[\hat{i},j]}^{k+l} = 0,\ \alpha_{[\hat{i},j]}^{k+l+1} > 0 \Rightarrow \Delta\alpha_{[\hat{i},j]}^{k+l} > 0$; this can only happen if $[\hat{i}, j]$ is a basic approach at iteration $k+l$.

Let $\breve{a} = [\breve{i}, j]$ be any contributing approach, i.e. $[\breve{i}, j] \in A_p^c(\boldsymbol{\alpha}^{k+l})$. Since $j \in \breve{\breve{N}}^{k+l}$, every contributing route is a "good route"; hence, for every contributing approach $\mu_{[\breve{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) = w_j^*$. As a basic approach, the average cost of the $[\hat{i}, j]$ approach at iteration $k+l$ can not be greater than the cost of any other approach. So

$$\begin{aligned} w_{\hat{i}}^* + t_{[\hat{i},j]}^* &\leq \mu_{[\hat{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) \leq \mu_{[\hat{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) + \frac{\epsilon}{4} \\ &\leq \mu_{[\breve{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) + \frac{\epsilon}{4} \leq \mu_{[\breve{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) + \frac{\epsilon}{2} = w_j^* + \frac{\epsilon}{2} \end{aligned} \tag{2.143}$$

Therefore $[\hat{i}, j] \in A_p^*$. Now $\hat{r} = \hat{s} + [\hat{i}, j] \notin R_{pj}^*$, but $[\hat{i}, j] \in A_p^*$; therefore $\hat{s} \notin R_{p\hat{i}}^*$, Since $\hat{s} \in R_{p\hat{i}}[A_p^c(\boldsymbol{\alpha}^{k+l+1})]$ we obtain that $\hat{i} \notin \breve{N}^{k+l+1}$ and in particular $\hat{i} \notin \breve{\breve{N}}^{k+l+1}$.

We saw that the link $[\hat{i}, j]$ must be in the restricting subnetwork A_p^{*l}; therefore, the topological order of this restricting subnetwork must satisfy $o^{*l}(\hat{i}) < o^{*l}(j)$. The choice of j as argmin $\{o^{*l}(j') : j' \in \mathcal{J}_0\}$ implies that $\hat{i} \notin \mathcal{J}_0$.

We found that $[\hat{i}, j] \in A_p^*$; therefore, $w_{\hat{i}}^* \leq w_j^* = \min\left\{w_{j'}^* : j' \in \breve{N}^{k+l} \setminus \breve{N}^{k+l+1}\right\}$, and also that $\hat{i} \notin \breve{N}^{k+l+1}$, so the only way that $\hat{i} \notin \mathcal{J}_0$ is if $\hat{i} \notin \breve{N}^{k+l}$.

Since $j \in \breve{N}^{k+l}$ every $i \in N$ such that $w_i^* < w_{\hat{i}}^*$ must be a permanent "good node". In particular, since $\hat{i}$ is not a permanent "good node" $w_{\hat{i}}^* \geq w_j^*$; but we saw that $w_{\hat{i}}^* \leq w_j^*$; therefore, $w_{\hat{i}}^* = w_j^*$.

Furthermore, the only possible reason for $\hat{i}$ not to be a permanent "good node" is if it is not even a temporary "good node"; that is $\hat{i} \notin \breve{N}^{k+l}$. This means that there exists a "bad" contributing route to $\hat{i}$

$$\hat{s} \in R_{p\hat{i}}[A_p^c(\boldsymbol{\alpha}^{k+l})] \setminus R_{pj}^* \tag{2.144}$$

$$u_{\hat{i}}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) \geq c_s(\mathbf{t}^*) > w_{\hat{i}}^* + \epsilon \tag{2.145}$$

Combining that result with the fact that $w_{\hat{i}}^* = w_j^*$ implies that

$$u_{\hat{i}}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) \quad > \quad u_{\hat{i}}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) - \frac{\epsilon}{4} > w_{\hat{i}}^* + \frac{3\epsilon}{4} = w_j^* + \frac{3\epsilon}{4} \tag{2.146}$$

On the other hand $j \in \breve{N}^{k+l}$ so $R_{pj}[A_p^c(\boldsymbol{\alpha}^{k+l}] \subseteq R_{pj}^*$; therefore, $u_j(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) = w_j^*$, so

$$u_j(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) < u_j(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) + \frac{\epsilon}{4} = w_j^* + \frac{\epsilon}{4} < u_{\hat{i}}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) \tag{2.147}$$

which contradicts the fact that $[i, j]$ is a new link that was added to the restricting subnetwork by the condition $u_i(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) < u_j(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l}))$.

The arguments for Case 2 can be summarized as follows:

$$\alpha^{k+l}_{[\hat{i},j]} = 0; \alpha^{k+l+1}_{[\hat{i},j]} > 0 \Rightarrow u_i(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) < u_j(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) \tag{2.148}$$

$$\alpha^{k+l}_{[\hat{i},j]} = 0; \alpha^{k+l+1}_{[\hat{i},j]} > 0 \Rightarrow \Delta\alpha^{k+l}_{[\hat{i},j]} > 0 \tag{2.149}$$

$$\alpha^{k+l+1}_{[\hat{i},j]} > 0 \Rightarrow o^{*l}(\hat{i}) < o^{*l}(j) \tag{2.150}$$

$$j \in \breve{\breve{N}}^{k+l} \Rightarrow \mu_{[\breve{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) = w^*_j \tag{2.151}$$

$$\Delta\alpha^{k+l}_{[\hat{i},j]} > 0 \Rightarrow \mu_{[\hat{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) \leq \mu_{[\breve{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) \tag{2.152}$$

$$\Rightarrow [\hat{i}, j] \in A^*_p \qquad \text{by (2.143)} \tag{2.153}$$

$$[\hat{i}, j] \in A^*_p \Rightarrow w^*_{\hat{i}} \leq w^*_j; \hat{s} \notin R^*_{p\hat{i}} \tag{2.154}$$

$$\hat{s} \notin R^*_{p\hat{i}} \Rightarrow \hat{i} \notin \breve{N}^{k+l+1} \Rightarrow \hat{i} \notin \breve{\breve{N}}^{k+l+1} \tag{2.155}$$

$$o^{*l}(\hat{i}) < o^{*l}(j) \Rightarrow \hat{i} \notin \mathcal{J}_0 \tag{2.156}$$

$$\Downarrow$$

$$w^*_{\hat{i}} \leq w^*_j; \hat{i} \notin \breve{\breve{N}}^{k+l+1} \Rightarrow \hat{i} \notin \breve{\breve{N}}^{k+l} \tag{2.157}$$

$$\Downarrow$$

$$w^*_{\hat{i}} \leq w^*_j; j \in \breve{\breve{N}}^{k+l} \Rightarrow \hat{i} \notin \breve{N}^{k+l}; w^*_{\hat{i}} = w^*_j \tag{2.158}$$

$$\hat{i} \notin \breve{N}^{k+l} \Rightarrow u_{\hat{i}}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) > w^*_{\hat{i}} + \epsilon > u_j(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) \tag{2.159}$$

and (2.148) contradicts (2.159). End of proof ($N^{k+l} \subseteq N^{k+l+1}$).

It remains to show that if $\breve{\breve{N}}^{k+l} \subsetneq N$, then $\breve{\breve{N}}^{k+l} \subsetneq \breve{\breve{N}}^{k+l+1}$.

The main difficulty in this part of the proof is to choose a node that will become a permanent "good node" in this iteration. Given the lack of a topological order for the 'star' subnetwork A^*_p, this is not a trivial task. The topological order of the current restricting subnetwork is not as helpful as one might expect, since the current restricting subnetwork does not necessarily allow for an equilibrium solution; in other words some links may be in the 'wrong' direction.

The minimum limiting cost values $\mathbf{w}^*$ provide the first condition for choosing such a node. In addition we require that there is at least one "good" route to this node in the restricting subnetwork. Such nodes can be viewed as "mixed" nodes, since they have some "bad" contributing routes but also some "good" routes available in the restricting subnetwork. We choose that "mixed" node with minimum topological order according to the current restricting subnetwork. It turns out that every approach to this node is either completely "good", i.e. consists of "good" routes only, or com-

pletely "bad", i.e. consists of "bad" routes only. As a result in a single iteration all "bad" approaches are evacuated, and the node becomes a permanent "good" node.

We start the procedure of choosing a new "good" node by considering a set of preliminary candidates which is defined in a similar way to (2.132)

$$\mathcal{J}_0 \quad = \quad \operatorname{argmin}\left\{w^*_j : j \in N \setminus \breve{\breve{N}}^{k+l}\right\} \tag{2.160}$$

This set is not empty since $\breve{\breve{N}}^{k+l} \subsetneq N$. Define the set of "mixed" candidates by

$$\mathcal{J}_1 = \left\{j \in \mathcal{J}_0 : R_{pj}[A^{*l}_p] \cap R^*_{pj} \neq \emptyset\right\} \tag{2.161}$$

We need to show that this is not an empty set. Consider some $j_0 \in \mathcal{J}_0$, and a "good" route $\breve{r} \in R^*_{pj_0}$. Let i_1 be the last node in $\breve{r}$ that is included in $\breve{\breve{N}}^{k+l}$, and let j_1 be the following node in $\breve{r}$. $j_1 \notin \breve{\breve{N}}^{k+l}$ and $w^*_{j_1} \leq w^*_{j_0} = \min\left\{w^*_j : j \in N \setminus \breve{\breve{N}}^{k+l}\right\}$; therefore, $j_1 \in \mathcal{J}_0$. Furthermore, the only possible reason for j_1 not to be a permanent "good node" is if it is not even a temporary "good node", that is $j_1 \notin \breve{N}^{k+l}$. Consider any contributing route to i_1 at iteration $k+l$

$$s \in R_{pi_1}[A^c_p(\boldsymbol{\alpha}^{k+l})] \subseteq R_{pi_1}[A^{*l}_p] \cap R^*_{pi_1} \tag{2.162}$$

$\breve{r} \in R^*_{pj_0}$; $[i_1, j_1] \subseteq \breve{r}$ hence $[i_1, j_1] \in A^*_p$, so $s + [i_1, j_1] \in R^*_{pj_1}$.

$i_1 \in \breve{\breve{N}}^{k+l}$; therefore, $u_{i_1}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) = w^*_{i_1}$ and hence $u_{i_1}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) < w^*_{i_1} + \frac{\epsilon}{4}$. $j_1 \notin \breve{N}^{k+l}$; therefore, $u_{j_1}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) > w^*_{j_1} + \epsilon$ and hence $u_{j_1}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) > w^*_{j_1} + \frac{3\epsilon}{4}$. In addition, $[i_1, j_1] \in A^*_p$ therefore $w^*_{i_1} \leq w^*_{j_1}$. In conclusion

$$u_{i_1}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) < w^*_{i_1} + \frac{\epsilon}{4} < w^*_{j_1} + \frac{3\epsilon}{4} < u_{j_1}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) \tag{2.163}$$

Hence $[i_1, j_1]$ is in the restricting subnetwork A^{*l}_p, and therefore $s + [i_1, j_1] \in R_{pj_1}[A^{*l}_p]$. So j_1 is a "mixed" node as we wanted, in other words $j_1 \in \mathcal{J}_1$ hence $\mathcal{J}_1 \neq \emptyset$. Since $\mathcal{J}_1$ is not an empty set, we can choose

$$j \quad = \quad \operatorname{argmin}\left\{o^{*l}(j') : j' \in \mathcal{J}_1\right\} \tag{2.164}$$

where o^{*l} is a topological order of the restricting subnetwork A^{*l}_p. We know that $j \in \mathcal{J}_1 \subseteq \mathcal{J}_0$ hence $j \notin \breve{\breve{N}}^{k+l}$, i.e. j was not a "good" node in iteration $k+l$. We want to show that $j \in \breve{\breve{N}}^{k+l+1}$, i.e. that j is a permanent "good" node from iteration

$k+l+1$ untill the end of the sequence. Consider a contributing route to j at iteration $k+l+1$,

$$r = s + [i,j] \in R_{pj}[A_p^c(\boldsymbol{\alpha}^{k+l+1})] \tag{2.165}$$

If $i \in \breve{\breve{N}}^{k+l} \subseteq \breve{\breve{N}}^{k+l+1}$ and $[i,j] \in A_p^*$, then $s \in R_{pi}[A_p^c(\boldsymbol{\alpha}^{k+l+1})] \subseteq R_{pi}^*$; hence, $r = s + [i,j] \in R_{pj}^*$. Therefore, a contributing route

$$\hat{r} = \hat{s} + [\hat{i},j] \in R_{pj}[A_p^c(\boldsymbol{\alpha}^{k+l+1})] \tag{2.166}$$

may be "bad" only if either $\hat{i} \notin \breve{\breve{N}}^{k+l}$ or $[\hat{i},j] \notin A_p^*$. We show that in both cases not only the route is "bad", but the entire approach is completely "bad"; i.e. every contributing route in the $[\hat{i},j]$ approach is "bad". As a result we show that the $[\hat{i},j]$ approach is evacuated in this iteration, i.e. $\alpha_{[\hat{i},j]}^{k+l+1} = 0$ hence $[\hat{i},j] \notin A_p^c(\boldsymbol{\alpha}^{k+l+1})$ in contradiction to the choice (2.166).

We next show that $\mu_{[\hat{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) > w_j^* + \epsilon$.
If $[\hat{i},j] \notin A_p^*$ then $\mu_{[\hat{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) \geq w_{\hat{i}}^* + t_{[\hat{i},j]}^* > w_j^* + \epsilon$.
If $[\hat{i},j] \in A_p^*$ and $\hat{i} \notin \breve{\breve{N}}^{k+l}$, then $w_{\hat{i}}^* \leq w_j^* = \min\left\{w_{j'}^* : j' \in N \setminus \breve{\breve{N}}^{k+l}\right\}$. Hence $\hat{i} \in \mathcal{J}_0$, so $\hat{i}$ is a preliminary candidate. $[\hat{i},j] \in A_p^c(\boldsymbol{\alpha}^{k+l+1}) \subseteq A_p^{*l}$; therefore, $o^{*l}(\hat{i}) < o^{*l}(j)$, so from the choice of j in (2.164) we learn that $\hat{i} \notin \mathcal{J}_1$; i.e. $\hat{i}$ is not a "mixed" candidate. Since $\hat{i}$ is a preliminary candidate, but not a "mixed" candidate, the $[\hat{i},j]$ approach consists of "bad" routes only; formally

$$\hat{i} \in \mathcal{J}_0; \hat{i} \notin \mathcal{J}_1 \Rightarrow R_{p\hat{i}}[A_p^{*l}] \cap R_{p\hat{i}}^* = \emptyset \tag{2.167}$$

In particular, every contributing route $s \in R_{p\hat{i}}[A_p^c(\boldsymbol{\alpha}^{k+l})] \subseteq R_{p\hat{i}}[A_p^{*l}]$ is a "bad" route; i.e. $s \notin R_{p\hat{i}}^*$ and hence $c_s(\mathbf{t}^*) > w_{\hat{i}}^* + \epsilon$; a similar result can be obtained for any average of contributing routes in the approach. Combining this with the assumption that $[\hat{i},j] \in A_p^*$, and hence $w_{\hat{i}}^* + t_{[\hat{i},j]} = w_j^*$ leads to the desired result that $\mu_{[\hat{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) > w_{\hat{i}}^* + \epsilon + t_{[\hat{i},j]} = w_j^* + \epsilon$.

Our conclusion so far is that if $\hat{r} = \hat{s} + [\hat{i},j]$ is a "bad" contributing route, then $\mu_{[\hat{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) > w_j^* + \epsilon$ and therefore $\mu_{[\hat{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) > w_j^* + \frac{3\epsilon}{4}$.

The next step is to show that $\mu_{b_j} \leq w_j^* + \frac{\epsilon}{4}$. Since $j \in \mathcal{J}_1$ is a "mixed" node, there exists a route $\breve{r} = \breve{s} + [\breve{i},j] \in R_{pj}[A_p^{*l}] \cap R_{pj}^*$. $[\breve{i},j]$ is in the restricting subnetwork A_p^{*l}; therefore, $o^{*l}(\breve{i}) < o^{*l}(j)$, so from the choice of j in (2.164) we learn that $\breve{i} \notin \mathcal{J}_1$; i.e. $\breve{i}$

is not a "mixed" candidate. $\breve{s} \in R_{p\breve{i}}[A_p^{*l}] \cap R^*_{p\breve{i}}$ and hence $\breve{i} \notin \mathcal{J}_1$ only if $\breve{i} \notin \mathcal{J}_0$; i.e. $\breve{i}$ is not even a preliminary candidate.

In addition $[\breve{i}, j] \in A_p^*$ implies $w_{\breve{i}}^* \leq w_j^* = \min\left\{w_{j'}^* : j' \in N \setminus \breve{\breve{N}}^{k+l}\right\}$; therefore, $\breve{i} \notin \mathcal{J}_0$ only if $\breve{i} \in \breve{\breve{N}}^{k+l}$, and as a result

$$\mu_{[\breve{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) = w_{\breve{i}}^* + t^*_{[\breve{i},j]} = w_j^* \tag{2.168}$$

$$\mu_{b_j}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) \leq \mu_{[\breve{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) < \mu_{[\breve{i},j]}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}^*) + \frac{\epsilon}{4} = w_j^* + \frac{\epsilon}{4} \tag{2.169}$$

as proposed. The rest of the proof continues in a similar fashion to the proof under assumption A. Let $\hat{a} = [\hat{i}, j]$; then

$$\mu_{\hat{a}}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) - \mu_{b_j}(\boldsymbol{\alpha}^{k+l}, \mathbf{t}(\boldsymbol{\alpha}^{k+l})) > \frac{\epsilon}{2} \tag{2.170}$$

$$z_{\hat{a}\to b_j}(\boldsymbol{\alpha}, \mathbf{t}(\boldsymbol{\alpha}^{k+l}), \mathbf{t}'(\boldsymbol{\alpha}^{k+l})) > \frac{\epsilon}{2 \cdot |A| \cdot t'_{max}} > 0 \tag{2.171}$$

where $t'_{max} = \max_{\boldsymbol{\alpha}} \max_{a \in A}\{t'_a(\mathbf{f}(\boldsymbol{\alpha}))\}$

The desirable shift is first scaled by the step size, and then truncated only if non-negativity is about to be violated. $\alpha_{\hat{a}}$ can remain a contributing approach only if the scaled desirable shift is applied as is and not truncated. Since the cost difference and the desirable shift are strictly positive, this can only happen if the amount of flow in the approach is greater than the scaled desirable shift. In such a case

$$-\Delta\alpha_{\hat{a}}^{k+l} = \lambda^{k+l} \cdot \frac{z_{\hat{a}\to b_j}(\boldsymbol{\alpha}, \mathbf{t}(\boldsymbol{\alpha}^{k+l}), \mathbf{t}'(\boldsymbol{\alpha}^{k+l}))}{g_j(\boldsymbol{\alpha}^{k+l+1})} \geq \frac{\lambda_0 \cdot \epsilon}{2 \cdot |A| \cdot t'_{max} \cdot g_j(\boldsymbol{\alpha}^{k+l+1})} \tag{2.172}$$

so the actual shift can not be too small.

This shift is aggregated with other shifts; however, shifts due to other nodes change the flow on $\hat{a}$ and b_j in the same direction, while the shift due to node j decreases the flow on $\hat{a}$ and increases the flow on b_j. In other words, if $g_j(\boldsymbol{\alpha}^{k+l+1}) \leq g_j(\boldsymbol{\alpha}^{k+l})$ then

$$\begin{aligned} \Delta f_{\hat{a}}(\boldsymbol{\alpha}^{k+l}, \Delta\boldsymbol{\alpha}^{k+l}) &= \Delta\alpha_{\hat{a}}^{k+l} \cdot g_j(\boldsymbol{\alpha}^{k+l+1}) + \alpha_{\hat{a}}^{k+l} \cdot \left(g_j(\boldsymbol{\alpha}^{k+l+1}) - g_j(\boldsymbol{\alpha}^{k+l})\right) \\ &\leq -\frac{\lambda_0 \cdot \epsilon}{2 \cdot |A| \cdot t'_{max}} \end{aligned} \tag{2.173}$$

while if $g_j(\boldsymbol{\alpha}^{k+l+1}) > g_j(\boldsymbol{\alpha}^{k+l})$ then

$$\Delta f_{b_j}(\boldsymbol{\alpha}^{k+l}, \Delta\boldsymbol{\alpha}^{k+l}) = \sum_{a \in NB_j} \Delta\alpha_a^{k+l} \cdot g_j(\boldsymbol{\alpha}^{k+l+1}) + \alpha_{b_j}^{k+l} \cdot \left(g_j(\boldsymbol{\alpha}^{k+l+1}) - g_j(\boldsymbol{\alpha}^{k+l})\right)$$
$$\geq \Delta\alpha_{\hat{a}}^{k+l} \cdot g_j(\boldsymbol{\alpha}^{k+l+1}) \geq \frac{\lambda_0 \cdot \epsilon}{2 \cdot |A| \cdot t'_{max}} \tag{2.174}$$

Therefore, there is at least one link in which the actual aggregated change of flow is at least $\frac{\lambda_0 \cdot \epsilon}{2 \cdot |A| \cdot t'_{max}}$ in magnitude.

On the other hand, k_0 was chosen so that $\forall k \in K, k \geq k_0$, $\forall 1 \leq l \leq |N|, \forall a \in A$,

$$|\Delta f_a(\boldsymbol{\alpha}^{k+l}, \Delta\boldsymbol{\alpha}^{k+l})| \leq |f_a(\boldsymbol{\alpha}^{k+l}) - f_a^*| + |f_a(\boldsymbol{\alpha}^{k+l+1}) - f_a^*| < \frac{\lambda_0 \cdot \epsilon}{2 \cdot |A| \cdot t'_{max}} \tag{2.175}$$

and this is a contradiction. End of proof (assumption B: $\mathbf{t} \geq 0$). □

2.9 Multiple origins

The origin-based solution method presented above can be easily extended to the case of multiple origins. An initial solution is found by assigning all the flows to the routes of minimum cost under free flow travel conditions, which is known as the all-or-nothing solution. Given a feasible solution the procedure described above is applied to each origin separately in a cyclic fashion.

A formal discussion of the multiple origins problem requires the addition of origin indices that were omitted so far. In particular the origin-based link flow vector becomes a two dimensional array, $\mathbf{f} = (f_{ap})_{a \in A; p \in N_o}$. Similarly the origin-based approach proportions array is $\boldsymbol{\alpha} = (\alpha_{ap})_{a \in A; p \in N_o}$. 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 method 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 $\Delta\boldsymbol{\alpha}_p \in \Theta^{\downarrow}(\boldsymbol{\alpha}_p)$; $\Delta\boldsymbol{\alpha}_{p'} = 0$ $(\forall p' \neq p)$ 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 method, consider a sequence $\boldsymbol{\alpha}^{k,i}$; $0 \leq i \leq n$, where $\boldsymbol{\alpha}^{1,0}$ is some feasible a-cyclic 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 exist a subsequence K such that $\forall k \in K$; $\forall l : 1 \leq l \leq |N|$;

$$\mathcal{A}_{p_i}(\boldsymbol{\alpha}_{p_i}^{k+l,i-1}) = A_{p_i}^{*l} \quad \forall 1 \leq i \leq n \tag{2.176}$$

$$\boldsymbol{\alpha}^{k+l,i} \rightarrow \boldsymbol{\alpha}^{*l,i} \quad \forall 0 \leq i \leq n \tag{2.177}$$

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

$$\boldsymbol{\alpha}^{k+l,i} - \boldsymbol{\alpha}^{k+l,i-1} = \Delta\boldsymbol{\alpha}^{k+l,i} \rightarrow \Delta\boldsymbol{\alpha}^{*l,i} = \boldsymbol{\alpha}^{*l,i} - \boldsymbol{\alpha}^{*l,i-1} \tag{2.178}$$

$$\Delta\boldsymbol{\alpha}^{*l,i} \in \Theta^{\downarrow p_i}\left(\boldsymbol{\alpha}^{*l,i-1} : A_{p_i}^{*l}\right) \tag{2.179}$$

The objective function value is a bounded monotonically non-increasing series, and hence converges to T^*; in particular, $T\left(\boldsymbol{\alpha}^{*l,i}\right) = T^*$. By Lemma 7, $T\left(\boldsymbol{\alpha}^{*l,i}\right) = T\left(\boldsymbol{\alpha}^{*l,i-1}\right)$ 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 $1 \leq l \leq |N|$; $0 \leq i \leq n$. 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.

2.10 Algorithm of Gallager and Bertsekas

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, the mathematical formulation of the routing problem is equivalent to TAP.

Gallager (1977) proposed a destination-based point of view, which is equivalent by symmetry to our origin-based point of view. For every destination q and every node i he defines routing variables that determine what portion of the node flow from i to q continues on each of the links that go out from node i. These routing variables are equivalent to our approach proportions. He requires that the solution described by these routing variables is spanning and loopfree, i.e. a-cyclic. He derives necessary and sufficient conditions for optimality similar to section 2.4. In every iteration Gallager's algorithm "reduces the fraction of traffic sent on non-optimal links and increase the fraction on the best link;" in our terminology this is described as a shift of flow from non-basic to basic approach. The magnitude of the shift in Gallager's method is determined by the difference between average approach costs divided by

node flow. This shift is scaled by a predetermined fixed step size, and then truncated if necessary to maintain feasibility. There is no discussion of the order in which shifts are computed and aggregated; we may therefore assume that they occur simultaneously.

The most important difference between Gallager's algorithm and our algorithm is the way new links are introduced into the solution. Gallager's condition is based on the definition of *improper link*. In our terminology an improper link is a contributing link (positive approach proportion) such that the average cost to its tail is higher than the average cost to its head, i.e. $\sigma_{a_t} > \sigma_{a_h}; \alpha_a > 0$. In fact since the step size is predetermined, the shift $\Delta\alpha_a$ is known, and hence link a may be considered as improper only if it remains contributing after the current iteration, that is if $\alpha_a \geq \Delta\alpha_a$. Cycles are avoided by prohibiting the introduction of any new link if there is a contributing route to the link tail that contains an improper link. This condition is quite different from the maximum cost condition described in section 2.8. The advantages and disadvantages of the two conditions have yet to be studied.

Bertsekas (1979) improved Gallager's algorithm in two aspects. First he used approximations of the second order derivative of the objective function. He argued that the recursive definitions (2.57) provide a lower bound, while replacing them with

$$\bar{\rho}_p(\boldsymbol{\alpha}, \mathbf{t}') = 0 \tag{2.180a}$$

$$\bar{\rho}_j(\boldsymbol{\alpha}, \mathbf{t}') = \sum_{a \in A_p; a_h = j} \alpha_a^2 \cdot t'_a + \left(\sum_{a \in A_p; a_h = j} \alpha_a \cdot \sqrt{\bar{\rho}_{a_t}(\boldsymbol{\alpha}, \mathbf{t}')} \right)^2 \qquad \forall j \neq p \tag{2.180b}$$

$$\bar{\nu}_a(\boldsymbol{\alpha}, \mathbf{t}') = t'_a + \bar{\rho}_{a_t}(\boldsymbol{\alpha}, \mathbf{t}') \tag{2.180c}$$

provides an upper bound in the sense that $\rho_j \leq \frac{\partial^2 T}{\partial d_j^2} \leq \bar{\rho}_j$.

Bertsekas et al. (1984) proposed an improved upper bound that takes the interaction between alternative routes into account. For nodes with two approaches only, one basic and one non-basic, Bertsekas proposed a shift similar to (2.59), except that the denominator term due to the approximated second order derivative $\nu_a + \nu_b - 2 \cdot \rho_{lcn_j}$ is replaced by $\bar{\nu}_a + \bar{\nu}_b$. If there are more than two approaches to a single node, he suggested either to consider each pair separately the way we did, or to determine new proportions for all approaches to a single node simultaneously by solving a piecewise linear equation. Comment: the notion of last common node, which we used to improve the approximation of the second order derivative, does not appear in any of the reviewed papers (Gallager, 1977; Bertsekas, 1979; Bertsekas et al. 1979; Bertsekas et al. 1984).

The second improvement introduced by Bertsekas is the line search. He proposed a piece-wise line search similar to our boundary search as described in section 2.7. He discussed the importance of such search techniques for the elimination of residual flows, especially on improper links. He also proposed considering step sizes of $\beta^k; k = 0, 1, 2, \ldots$ where $\beta \in [0.1, 0.5]$, again similar to our search strategy. The only difference between Bertsekas's search procedure and our search procedure is the stopping condition. Bertsekas uses an Armijo type condition

$$T(\boldsymbol{\alpha}^k) - T(\boldsymbol{\alpha}^{k+1}) \geq \sigma \nabla T(\boldsymbol{\alpha}^k) \cdot \Delta \mathbf{f} \tag{2.181}$$

where $\sigma \in [0.1, 0.01]$, while we consider a bisection type condition

$$\nabla T(\boldsymbol{\alpha}^{k+1}) \cdot \Delta \mathbf{f} \leq 0 \tag{2.182}$$

To summarize, the main algorithmic concepts used in our method are similar to the ones used by Gallager and Bertsekas, in particular the origin-based (destination-based) point of view, the absence of cycles, and the use of approach proportions (routing variables). The most important difference is the way restricting subnetworks are updated and new links are introduced. Other differences include the approximation of second order derivatives, the consideration of last common nodes, and the line search stopping condition. The specific implementations may also differ in the data structure used as well as in other details like the introduction of quick iterations in which restricting subnetworks are not modified.

3. ROUTE FLOW ENTROPY MAXIMIZATION AND BYPASS PROPORTIONALITY

The previous chapter presented a method for finding an equilibrium origin-based link flow solution for the standard (separable) traffic assignment problem. As mentioned in section 1.3, this solution is not unique, since by Wardrop's user equilibrium condition only total link flows are determined uniquely, while origin-based and route-based solutions are not. The importance of obtaining a realistic route flow pattern is discussed in section 1.5. In this chapter additional assumptions regarding the route flow pattern are considered in an attempt to determine which is the most likely one.

Rossi et al. (1989) and several other researchers suggested that the entropy maximizing route flow solution is the most likely one. We propose an alternative condition, referred to as the bypass proportionality condition, which is more intuitive and straightforward. This condition is described in detail in section 3.1, first from an intuitive point of view, and then in a formal mathematical fashion.

As it turns out, the two conditions lead to similar results. In section 3.2 we show that bypass proportionality is a necessary but not sufficient condition for route flow entropy maximization under any feasible constraint on total link flows. In particular, the Maximum Entropy User Equilibrium (MEUE) solution must satisfy the bypass proportionality condition.

Section 3.3 shows that the bypass proportionality assumption provides a constructive route flow interpretation for any feasible a-cyclic origin-based link flow array. This interpretation also maximizes route flow entropy. Therefore, in the context of route flow interpretations for origin-based solutions the two assumptions are equivalent. The specific route flow interpretation described provides motivation for the definitions in section 2.3. Section 3.4 extends the constructive solution of section 3.3 to the general case when only total link flows are known.

So far we assumed that link costs are non-negative, but may be zero. It should be noted that with zero cost links some equilibrium routes may contain cycles, and some routes may in fact be infinite. As demonstrated by Akamatsu (1996), the entropy maximizing solution uses all of these routes, including infinite routes. To avoid this unappealing situation, we assume in this chapter that all user equilibrium routes are simple, i.e. they contain no cycles, either because all link costs are strictly positive, or because of the special structure of the network.

3.1 Bypass Proportionality

Consider a segment of a main road route with an alternative bypass. Wardrop's user equilibrium assumption implies that the proportion of users choosing the bypass is such that the cost of each alternative is the same. Interpreting that proportion as the probability that a certain user chooses the bypass, one may ask whether that probability depends on the trip origin or trip destination. The basic user equilibrium traffic assignment model assumes that all users are identical, in the sense that they all decide in the same way to minimize route cost, which is the same for all users regardless of their origin and destination. More complex models suggest that route generalized cost may vary by trip purpose, user group and other attributes, but typically not by origin and destination. Hence, it seems reasonable to assume that the probability of choosing the bypass is independent of the origin and the destination.

The same arguments suggest that the probability of choosing the bypass is also independent of decisions made prior to the point of diversion, and after the merge point. The *bypass proportionality assumption* is that the proportion of users choosing a bypass is the same for all origins, destinations, initial routes (the route segment from the origin to the bypass diverge), and final routes (the route segment from the bypass merge to the destination).

For example, consider the network of Figure 8. In this network the main route passes through nodes 1, 2, 3, 4, 5, and 6. Suppose that 800 vph use main route segment $3 \rightarrow 4 \rightarrow 5$, while 200 vph divert to bypass $3 \rightarrow 8 \rightarrow 5$. The bypass proportionality assumption suggests that in this case every user remains on the main route with probability 0.8 and chooses the bypass with probability 0.2. In particular if the demand from origin B to destination D is 200 vph, then 80% of that flow, i.e. 160 vph, chooses main route B$\rightarrow 2 \rightarrow 3 \rightarrow \mathbf{4} \rightarrow 5 \rightarrow 6 \rightarrow$D, while 20%, i.e. 40 vph, divert to the bypass and use route B$\rightarrow 2 \rightarrow 3 \rightarrow \mathbf{8} \rightarrow 5 \rightarrow 6 \rightarrow$D.

Similarly, suppose that the demand from origin A to destination C is 350 vph; of this flow 150 vph begin on initial route A$\rightarrow 7 \rightarrow 2 \rightarrow 3$ from origin A to diverge node 3. Suppose that out of this flow, 100 vph end their trip on the direct link from merge node 5 to destination C, while the other 50 vph choose final route $5 \rightarrow 6 \rightarrow$C. The bypass proportionality assumption is that the same proportions (80/20) apply to each of these groups; in particular 80% of the flow in the last group, i.e. 40 vph follow main route A$\rightarrow 7 \rightarrow 2 \rightarrow 3 \rightarrow \mathbf{4} \rightarrow 5 \rightarrow 6 \rightarrow$C, and the remaining 10 vph choose the bypass and follow route A$\rightarrow 7 \rightarrow 2 \rightarrow 3 \rightarrow \mathbf{8} \rightarrow 5 \rightarrow 6 \rightarrow$C.

The term *bypass proportionality* stems intuitively from the situation presented above. In general it may not be possible to distinguish between the "main route" and the

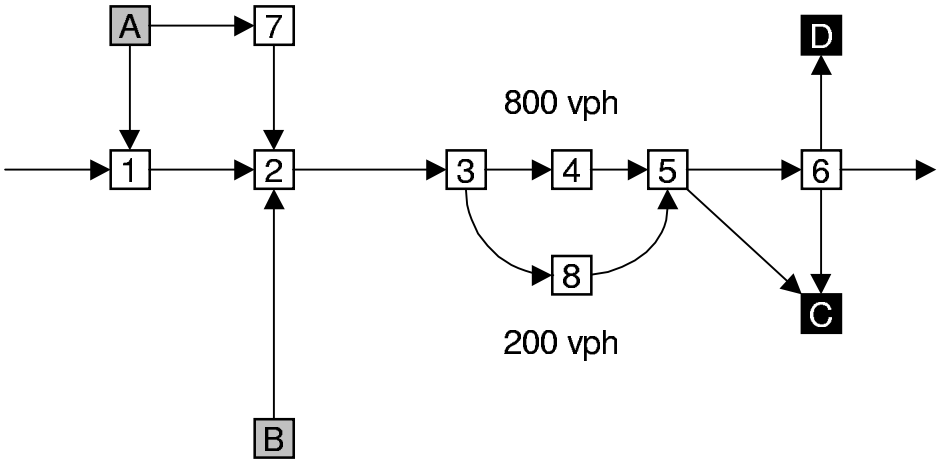


Figure 8. Bypass proportionality assumption

"bypass;" therefore a formal discussion must consider any pair of alternative route segments $s, s' \in R_{n_d n_m}$ that begin at some diverge node $n_d \in N$ and end at some merge node $n_m \in N$. To formulate the bypass proportionality condition mathematically consider two groups of users: one group begins at origin p_1, uses an initial route segment r_1^i to the diverge node n_d, continues through either alternative segment s or s' to the merge node n_d, and ends through a final route segment r_1^f to their destination q_1. The other group begins at origin p_2, uses another initial route segment r_2^i to the diverge node n_d, chooses between the same alternative segments s and s', and ends through a final route segment r_2^f to their destination q_2. Consider the following route combinations: $r_1^i + s + r_1^f, r_1^i + s' + r_1^f, r_2^i + s + r_2^f, r_2^i + s' + r_2^f$. The bypass proportionality assumption suggests that the proportion of users that chooses each alternative segment is the same in both groups; hence, the flow ratios are equal

$$\frac{h_{(r_1^i+s+r_1^f)p_1q_1}}{h_{(r_1^i+s'+r_1^f)p_1q_1}} = \frac{h_{(r_2^i+s+r_2^f)p_2q_2}}{h_{(r_2^i+s'+r_2^f)p_2q_2}} \tag{3.1}$$

Cross multiplying implies that

$$h_{(r_1^i+s+r_1^f)p_1q_1} \cdot h_{(r_2^i+s'+r_2^f)p_2q_2} = h_{(r_1^i+s'+r_1^f)p_1q_1} \cdot h_{(r_2^i+s+r_2^f)p_2q_2} \tag{3.2}$$

Condition (3.1) is probably more intuitive; however, it is only applicable if the denominators are strictly positive, in which case (3.1) and (3.2) are equivalent. In the

following definition, condition (3.2) is used, since it can be applied to all possible combinations of zero flows as well.

Definition: A route flow vector $\mathbf{h} = (h_{rpq})_{p \in N_o; q \in N_d(p); r \in R_{pq}}$ satisfies the (strong) *bypass proportionality* condition iff it satisfies condition (3.2) for every diverge node $n_d \in N$, merge node $n_m \in N$, pair of alternative route segments $s, s' \in R_{n_d n_m}$, pair of origins $p_1, p_2 \in N_o$, pair of destinations $q_1 \in N_d(p_1)$, $q_2 \in N_d(p_2)$, initial routes $r_1^i \in R_{p_1 n_d}; r_2^i \in R_{p_2 n_d}$, and final routes $r_1^f \in R_{n_m q_1}; r_2^f \in R_{n_m q_2}$.

Notice that this definition requires that (3.2) holds even if some of the route combinations contain cycles. For the sake of simplicity, in the following flows are explicitly restricted to simple routes, i.e. to routes that do not contain cycles. In that context an alternative condition is considered, referred to as the weak bypass proportionality condition which requires that (3.2) holds only if all four route combinations are simple. That is $(r_1^i + s + r_1^f), (r_1^i + s' + r_1^f) \in R_{p_1 q_1}$, and $(r_2^i + s + r_2^f), (r_2^i + s' + r_2^f) \in R_{p_2 q_2}$.

The bypass proportionality *assumption* corresponds to a behavioral postulate that actual flows satisfy the bypass proportionality condition.

3.2 Route flow representation for total link flows

Suppose that $\mathbf{f}_\bullet \in F_\bullet$ is a feasible vector of total link flows. A route flow representation for $\mathbf{f}_\bullet$ is a feasible route flow pattern that is consistent with the given total link flows, that is a vector $\mathbf{h}$ such that

$$\sum_{p \in N_o} \sum_{q \in N_d(p)} \sum_{r \in R_{pq}; a \subseteq r} h_{rpq} = f_{a\bullet} \qquad \forall a \in A \tag{3.3a}$$

$$\sum_{r \in R_{pq}} h_{rpq} = d_{pq} \qquad \forall p \in N_o;\ \forall q \in N_d(p) \tag{3.3b}$$

$$h_{rpq} \geq 0 \qquad \forall p \in N_o;\ \forall q \in N_d(p);\ \forall r \in R_{pq} \tag{3.3c}$$

Notice that constraint (3.3a) on total link flows does not guarantee that the demand is satisfied; hence an explicit constraint on the demand (3.3b) is necessary. The feasibility of $\mathbf{f}_\bullet$ does ensure, by the definition of $F_\bullet$, that it has at least one route flow representation. In general such representation is not likely to be unique, and the expected question is which is the most likely or most reasonable route flow representation.

One possible criterion is route flow entropy maximization. When applied to the user equilibrium total link flows, this criterion leads to the MEUE solution mentioned earlier; however, the results of the following discussion are valid for any feasible total link flows vector. The route flow entropy maximizing representation of $\mathbf{f}_\bullet$ is the optimal route flow solution for:

$$\max \quad E(\mathbf{h}) = -\sum_{p\in N_o}\sum_{q\in N_d(p)}\sum_{r\in R_{pq}} h_{rpq}\left(\ln\left(\frac{h_{rpq}}{d_{pq}}\right)-1\right) \tag{3.4a}$$

subject to

$$\sum_{p\in N_o}\sum_{q\in N_d(p)}\sum_{\substack{r\in R_{pq}\\ a\subseteq r}} h_{rpq} = f_{a\bullet} \qquad \forall a\in A \tag{3.4b}$$

$$\sum_{r\in R_{pq}} h_{rpq} = d_{pq} \qquad \forall p\in N_o;\ \forall q\in N_d(p) \tag{3.4c}$$

$$h_{rpq} \geq 0 \qquad \forall p\in N_o;\ \forall q\in N_d(p);\ \forall r\in R_{pq} \tag{3.4d}$$

Comment: we assumed that the sets $N_d(p)$ are such that $d_{pq} > 0$ for all $p\in N_o; q\in N_d(p)$; hence the division by d_{pq} in the summation above is valid. Let $\hat{R}_{pq}$ denote the set of routes $r\in R_{pq}$ such that there is some feasible solution for (3.4) with $h_{rpq} > 0$. Problem (3.4) is therefore equivalent to:

$$\max \quad E(\mathbf{h}) = -\sum_{p\in N_o}\sum_{q\in N_d(p)}\sum_{r\in \hat{R}_{pq}} h_{rpq}\left(\ln\left(\frac{h_{rpq}}{d_{pq}}\right)-1\right) \tag{3.5a}$$

subject to

$$\sum_{p\in N_o}\sum_{q\in N_d(p)}\sum_{\substack{r\in \hat{R}_{pq}\\ a\subseteq r}} h_{rpq} = f_{a\bullet} \qquad \forall a\in A \tag{3.5b}$$

$$\sum_{r\in \hat{R}_{pq}} h_{rpq} = d_{pq} \qquad \forall p\in N_o;\ \forall q\in N_d(p) \tag{3.5c}$$

$$h_{rpq} \geq 0 \qquad \forall p\in N_o;\ \forall q\in N_d(p);\ \forall r\in \hat{R}_{pq} \tag{3.5d}$$

In the optimal solution every route in $\hat{R} = \bigcup_{p \in N_o} \bigcup_{q \in N_d(p)} \hat{R}_{pq}$ must have positive flow; therefore, the optimal solution of (3.5) is an inner point, at which the objective function is differentiable. The Lagrangian is:

$$L = -\sum_{p \in N_o} \sum_{q \in N_d(p)} \sum_{r \in \hat{R}_{pq}} h_{rpq} \left(\ln\left(\frac{h_{rpq}}{d_{pq}}\right) - 1 \right) - \sum_{a \in A} \beta_a \left(f_{a\bullet} - \sum_{p \in N_o} \sum_{q \in N_d(p)} \sum_{\substack{r \in \hat{R}_{pq} \\ a \subseteq r}} h_{rpq} \right)$$
$$- \sum_{p \in N_o} \sum_{q \in N_d(p)} \gamma_{pq} \left(d_{pq} - \sum_{r \in \hat{R}_{pq}} h_{rpq} \right) \tag{3.6}$$

and the inner point optimality conditions are that for every origin p, destination q and route $r \in \hat{R}_{pq}$

$$\frac{\partial L}{\partial h_{rpq}} = -\ln\left(\frac{h_{rpq}}{d_{pq}}\right) + \sum_{a \subseteq r} \beta_a + \gamma_{pq} = 0 \tag{3.7}$$

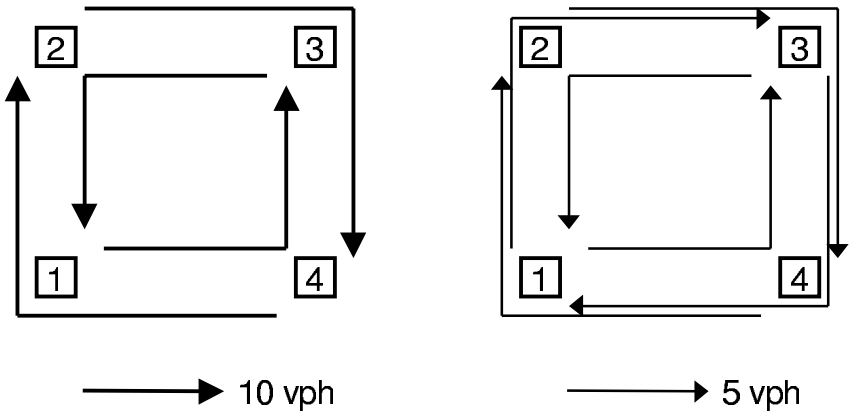
$$h_{rpq} = d_{pq} \cdot \exp\left(\gamma_{pq} + \sum_{a \subseteq r} \beta_a\right) \tag{3.8}$$

Using this derivation we can verify that the optimal solution for (3.4) satisfies the weak bypass proportionality condition. Suppose $n_d, n_m \in N$; $s, s' \in R_{n_d n_m}$; $p_1, p_2 \in N_o$; $q_1 \in N_d(p_1)$; $q_2 \in N_d(p_2)$; $r_1^i \in R_{p_1 n_d}$; $r_2^i \in R_{p_2 n_d}$; $r_1^f \in R_{n_m q_1}$; $r_2^f \in R_{n_m q_2}$. If all four route combinations can have positive flow, i.e. $(r_1^i + s + r_1^f), (r_1^i + s' + r_1^f) \in \hat{R}_{p_1 q_1}$ $(r_2^i + s + r_2^f), (r_2^i + s' + r_2^f) \in \hat{R}_{p_2 q_2}$, we can substitute (3.8) in (3.2), denote $w_r = \exp\left(\sum_{a \subseteq r} \beta_a\right)$ and obtain

$$\left[d_{p_1 q_1} \cdot \exp\left(\gamma_{p_1 q_1}\right) \cdot w_{r_1^i} \cdot w_s \cdot w_{r_1^f}\right] \cdot \left[d_{p_2 q_2} \cdot \exp\left(\gamma_{p_2 q_2}\right) \cdot w_{r_2^i} \cdot w_{s'} \cdot w_{r_2^f}\right] =$$
$$\left[d_{p_1 q_1} \cdot \exp\left(\gamma_{p_1 q_1}\right) \cdot w_{r_1^i} \cdot w_{s'} \cdot w_{r_1^f}\right] \cdot \left[d_{p_2 q_2} \cdot \exp\left(\gamma_{p_2 q_2}\right) \cdot w_{r_2^i} \cdot w_s \cdot w_{r_2^f}\right] \tag{3.9}$$

which is clearly true.

Suppose w.l.o.g. that $(r_1^i + s + r_1^f) \notin \hat{R}_{p_1 q_1}$ and hence $h_{r_1^i + s + r_1^f, p_1 q_1} = 0$. If the right hand side of (3.2) is not zero, then there is $\epsilon > 0$ such that $h_{(r_1^i + s' + r_1^f) p_1 q_1} > \epsilon > 0$ and $h_{(r_2^i + s + r_2^f) p_2 q_2} > \epsilon > 0$. If all four route combinations are simple, we can shift ϵ flow from $(r_1^i + s' + r_1^f)$ to $(r_1^i + s + r_1^f)$ and from $(r_2^i + s + r_2^f)$ to $(r_2^i + s' + r_2^f)$ and get



a) Bypass proportionality holds b) Maximum route flow entropy

Figure 9. Bypass proportionality vs. entropy maximization

another feasible solution where $h_{(r_1^i+s+r_1^f)p_1q_1} = \epsilon > 0$; hence $(r_1^i + s + r_1^f, p_1q_1) \in \hat{R}_{p_1q_1}$, contradiction.

We showed that the route flow entropy maximizing representation for any feasible total link flows constraint satisfies the weak bypass proportionality condition. In particular, the solution to MEUE, which is defined by maximizing route flow entropy under the user equilibrium total link flows, must satisfy the weak bypass proportionality condition.

Our conjecture is that if flows are not explicitly restricted to simple routes, the route flow entropy maximizing representation satisfies the strong bypass proportionality condition. In the case of user equilibrium, the restriction to simple routes is done implicitly; i.e. routes that contain cycles are not equilibrium routes, either because link costs are strictly positive, or because of the special structure of the network. Therefore the MEUE solution also satisfies the strong bypass proportionality condition.

The next question is whether satisfying bypass proportionality is sufficient for entropy maximization. To answer this question consider the networks in Figure 9, with diagonal O-D flows, $d_{1,3} = d_{3,1} = d_{2,4} = d_{4,2}$ =10 vph, and zero flow for all other O-D pairs. The total flow on each link in both figures is 10 vph; therefore, both are feasible solutions for the same route flow representation problem. In Figure 9a all flows from 1 to 3 and from 3 to 1 use the counter-clockwise links, while flows from 2 to 4 and from 4 to 2 use the clockwise links. In Figure 9b flows between each O-D pair are evenly distributed, half going clockwise, and half going counter-clockwise. One can

verify that the route flows representation in both networks satisfy the bypass proportionality condition; however, route flow entropy is maximized only by the route flow representation of Figure 9b. From this example we learn that bypass proportionality is only a necessary, but not a sufficient condition for entropy maximization.

3.3 Route flow interpretation for origin-based link flows

In the previous section, entropy maximization and bypass proportionality were considered in determining the most likely route flow representation for a given total link flow pattern. In this section similar criteria are considered when route flows are further restricted by a specific origin-based link flow array, $\mathbf{f} = (f_{ap})$. One reason to consider this question is when origin-based solution methods are used to find the entropy maximizing user equilibrium assignment; however, such methods have not yet been proposed. In chapter 2 an origin-based method to solve the basic user equilibrium traffic assignment problem was presented. During the iterative process this method produces feasible a-cyclic origin-based solutions that converge to a user equilibrium solution. Even though these solutions are not necessarily in agreement with the entropy maximizing route flow representation for the same total link flows, route flow interpretations for them are helpful in understanding the origin-based solutions, and useful for evaluation purposes. Therefore, in this section we use the two criteria, entropy maximization and bypass proportionality, to determine the most likely route flow interpretation for a general feasible a-cyclic origin-based link flow array.

Given a feasible a-cyclic origin-based link flow array, $\mathbf{f} = (f_{ap})$, a route flow *interpretation* is a non-negative vector of route flows $\mathbf{h}$ that satisfies $\sum_{q \in N_d(p)} \sum_{r \in R_{pq}; a \subseteq r} h_{rpq} = f_{ap}$ for every origin $p \in N_o$ and every link $a \in A$. Notice that in the case of origin-based solutions the feasibility of $\mathbf{f}$ is sufficient to ensure the feasibility of every route flow interpretation, without adding the O-D flow constraints explicitly. This was not the case in the previous section, where satisfying total link flow constraints was not sufficient and explicit O-D flow constraints were needed.

As before, we denote the used subnetwork for origin p by $A_p^u = \{a \in A : f_{ap} > 0\} \subseteq A$, and the set of route segments from node i to node j that are included in this subnetwork by $R_{ij}[A_p^u] = \left\{r \in R_{ij} : a \subseteq r \Rightarrow a \in A_p^u\right\}$. By assuming that $\mathbf{f}$ is a-cyclic we mean that each of the used subnetworks A_p^u is a-cyclic. Since used subnetworks are a-cyclic, every route segment included in them is simple, and every combination of such route segments is also simple. Therefore in this section, the earlier distinction between strong and weak bypass proportionality is irrelevant.

The entropy maximizing interpretation is the optimal solution for:

$$\max \quad E(\mathbf{h}) = -\sum_{p\in N_o}\sum_{q\in N_d(p)}\sum_{r\in R_{pq}} h_{rpq}\cdot\left(\ln\left(\frac{h_{rpq}}{d_{pq}}\right)-1\right) \tag{3.10a}$$

subject to

$$\sum_{q\in N_d(p)}\sum_{\substack{r\in R_{pq}\\ a\subseteq r}} h_{rpq} = f_{ap} \qquad \forall p\in N_o;\ \forall a\in A \tag{3.10b}$$

$$h_{rpq}\geq 0 \qquad \forall p\in N_o;\ \forall q\in N_d(p);\ \forall r\in R_{pq} \tag{3.10c}$$

In every interpretation of $\mathbf{f}$, a route r can carry a positive flow only if it is included in the used subnetwork, i.e. $h_{rpq}>0 \Rightarrow r\in R_{pq}[A^u_p]$. Therefore, problem (3.10) can be rewritten as:

$$\max \quad E(\mathbf{h}) = -\sum_{p\in N_o}\sum_{q\in N_d(p)}\sum_{r\in R_{pq}[A^u_p]} h_{rpq}\cdot\left(\ln\left(\frac{h_{rpq}}{d_{pq}}\right)-1\right) \tag{3.11a}$$

subject to

$$\sum_{q\in N_d(p)}\sum_{\substack{r\in R_{pq}[A^u_p]\\ a\subseteq r}} h_{rpq} = f_{ap} \qquad \forall p\in N_o;\ \forall a\in A \tag{3.11b}$$

$$h_{rpq}\geq 0 \qquad \forall p\in N_o;\ \forall q\in N_d(p);\ \forall r\in R_{pq}[A^u_p] \tag{3.11c}$$

The Lagrangian is:

$$\begin{aligned} L = &-\sum_{p\in N_o}\sum_{q\in N_d(p)}\sum_{r\in R_{pq}[A^u_p]} h_{rpq}\left(\ln\left(\frac{h_{rpq}}{d_{pq}}\right)-1\right)\\ &-\sum_{p\in N_o}\sum_{a\in A^u_p}\beta_{ap}\left(f_{ap}-\sum_{q\in N_d(p)}\sum_{\substack{r\in R_{pq}[A^u_p]\\ a\subseteq r}} h_{rpq}\right) \end{aligned} \tag{3.12}$$

and the inner point optimality conditions are that for every origin p, destination q and route $r \in R_{pq}[A^u_p]$

$$\frac{\partial L}{\partial h_{rpq}} = -\ln\left(\frac{h_{rpq}}{d_{pq}}\right) + \sum_{a \subseteq r} \beta_{ap} = 0 \tag{3.13}$$

$$h_{rpq} = d_{pq} \cdot \exp\left(\sum_{a \subseteq r} \beta_{ap}\right) \tag{3.14}$$

Notice that (3.14) is very similar to (3.8), except that the Lagrange multipliers related to the links are origin-dependent, and the O-D Lagrange multiplier is omitted. As in the previous section, this derivation allows us to examine the bypass proportionality condition for the optimal solution of (3.10). If all four route combinations are included in the used subnetworks, i.e. $(r_1^i + s + r_1^f)$, $(r_1^i + s' + r_1^f) \in R_{p_1q_1}[A^u_{p_1}]$; $(r_2^i + s + r_2^f)$, $(r_2^i + s' + r_2^f) \in R_{p_2q_2}[A^u_{p_2}]$, we can substitute (3.14) in (3.2) to obtain

$$\begin{aligned}\left[d_{p_1q_1} \cdot w_{r_1^i p_1} \cdot w_{sp_1} \cdot w_{r_1^f p_1}\right] \cdot \left[d_{p_2q_2} \cdot w_{r_2^i p_2} \cdot w_{s'p_2} \cdot w_{r_2^f p_2}\right] = \\ \left[d_{p_1q_1} \cdot w_{r_1^i p_1} \cdot w_{s'p_1} \cdot w_{r_1^f p_1}\right] \cdot \left[d_{p_2q_2} \cdot w_{r_2^i p_2} \cdot w_{sp_2} \cdot w_{r_2^f p_2}\right]\end{aligned} \tag{3.15}$$

where $w_{rp} = \exp\left(\sum_{a \subseteq r} \beta_{ap}\right)$. Equation (3.15) holds if

$$w_{sp_1} \cdot w_{s'p_2} = w_{s'p_1} \cdot w_{sp_2} \tag{3.16}$$

This equation is certainly true if $p_1 = p_2$; however, in general it may not hold.

It is also necessary to check the bypass proportionality condition if one of the four route combinations is not included in the used subnetwork. Suppose w.l.o.g. that $(r_1^i + s + r_1^f) \notin R_{p_1q_1}[A^u_{p_1}]$, and therefore either $r_1^i \notin R_{p_1n_d}[A^u_{p_1}]$, or $r_1^f \notin R_{n_mq_1}[A^u_{p_1}]$, or $s \notin R_{n_dn_m}[A^u_{p_1}]$. In any case, $h_{r_1^i + s + r_1^f, p_1q_1} = 0$. In the first two cases, $(r_1^i + s' + r_1^f)$ is also not in $R_{p_1q_1}[A^u_{p_1}]$, and condition (3.2) holds. However, in the last case, the fact that $s \notin R_{n_dn_m}[A^u_{p_1}]$ does not necessarily imply that $s \notin R_{n_dn_m}[A^u_{p_2}]$, unless p_1 and p_2 are the same origin.

In conclusion, solution (3.14) satisfies condition (3.2) for groups of users that start from the same origin. One should note that bypass proportions in a general feasible origin-based link flow array may be different from one origin to the other, in which case there is no route flow interpretation that satisfies the global bypass proportionality condition, as defined in section 3.1. Therefore a weaker condition should be considered, that equation (3.2) holds when the two groups of users have the same origin, i.e. $p_1 = p_2$. This is referred to as the *origin-based bypass proportionality*

condition. As shown above, the entropy maximizing route flow interpretation (3.14) does satisfy the origin-based bypass proportionality condition.

In section 3.2, when only total link flows were given, we found that there may be more than one route flow representation that satisfies the bypass proportionality condition. In the following we show that when the more detailed origin-based link flows are given, there is only one route flow interpretation that satisfies the origin-based bypass proportionality condition. As is shown, this interpretation also satisfies the inner point optimality conditions of the entropy maximization problem, thus demonstrating that the two conditions are equivalent.

The following derivation uses an additional definition for aggregating route flows. The *O-D segment flow* g_{spq} is the aggregation of all flows from origin p to destination q that share a specific route segment s. The route segment s does not have to start at the origin p or end at the destination q. It can be part of a route, or possibly part of several different routes from p to q. It is defined mathematically as the sum over all routes $r \in R_{pq}$ such that the route segment s is part of the route r, that is:

$$g_{spq} = \sum_{r \in R_{pq}; s \subseteq r} h_{rpq} \tag{3.17}$$

Recall that the *origin-based node flow* from origin p to node j, $g_{j,p}$ is the aggregation of all the flows that originate at p and arrive at j, either on their way to another destination, or to stop at j, if it is the destination. It was shown in (2.15) that $g_{jp} = \sum_{[i,j] \in A} f_{[i,j]p}$. When the node flow is strictly positive, the proportion of flow that arrives to j from a specific approach $[i,j] \in A$ is denoted by

$$\alpha_{[i,j]p} = \frac{f_{[i,j]p}}{g_{jp}} \tag{3.18}$$

For a given origin p and link $a = [i,j] \in A^u_p$ consider a route segment $s_1 + [i,j]$ for some $s_1 \in R_{pi}[A^u_p]$, an alternative route segment $s_2 \in R_{pj}[A^u_p]$, destinations $q, q' \in N_d(p)$ and final routes $r \in R_{jq}[A^u_p]$; $r' \in R_{jq'}[A^u_p]$. The origin-based bypass proportionality assumption (with empty initial routes $r_i = r'_i = [p]$) implies that:

$$h_{s_1+[i,j]+rpq} \cdot h_{s_2+r',pq'} = h_{s_2+rpq} \cdot h_{s_1+[i,j]+r',pq'} \tag{3.19}$$

Notice that:

$$\sum_{s_1 \in R_{pi}[A_p^u]} h_{s_1+[i,j]+rpq} = g_{[i,j]+rpq} \tag{3.20}$$

$$\sum_{s_2 \in R_{pj}[A_p^u]} \sum_{q' \in N_d(p)} \sum_{r' \in R_{jq'}[A_p^u]} h_{s_2+r',pq'} = g_{jp} \tag{3.21}$$

$$\sum_{s_2 \in R_{pj}[A_p^u]} h_{s_2+rpq} = g_{rpq} \tag{3.22}$$

$$\sum_{s_1 \in R_{pi}[A_p^u]} \sum_{q' \in N_d(p)} \sum_{r' \in R_{jq'}[A_p^u]} h_{(s_1+[i,j]+r')pq'} = f_{[i,j]p} \tag{3.23}$$

Summing (3.19) over all possible s_1, s_2, q', r' and using (3.20), (3.21), (3.22), and (3.23) we obtain

$$g_{([i,j]+r)pq} \cdot g_{jp} = g_{rpq} \cdot f_{[i,j]p} \tag{3.24}$$

Since $[i,j] \in A_p^u$, $f_{[i,j]p} > 0$, hence $g_{jp} > 0$; therefore, we can rewrite (3.24) as

$$g_{([i,j]+r)pq} = \alpha_{[i,j]p} \cdot g_{rpq} \tag{3.25}$$

which may be interpreted as an *approach proportionality* condition. Since A_p^u is a-cyclic, if $r \in R_{pq}[A_p^u]$ then there can not be a longer route $r' \in R_{pq}[A_p^u]$ that contains r; therefore the route flow and the O-D segment flow are equal, $h_{rpq} = g_{rpq}$. By a similar argument $g_{[q]pq} = d_{pq}$, and hence for any route $r \in R_{pq}[A_p^u]$

$$h_{rpq} = g_{rpq} = g_{[q]pq} \cdot \prod_{a \subseteq r} \alpha_{ap} = d_{pq} \cdot \prod_{a \subseteq r} \alpha_{ap} \tag{3.26}$$

The route flow interpretation given by (3.26) satisfies the inner point optimality conditions for the entropy maximization problem with Lagrange multipliers $\beta_{ap} = \ln(\alpha_{ap})$. In conclusion, when a feasible a-cyclic origin-based link flow array is given, route flow entropy maximization, the origin-based bypass proportionality condition and the approach proportionality condition are equivalent.

We can therefore view the route flows resulting from (3.26) as a function of the origin-based link flows $\mathbf{h}(\mathbf{f})$. Notice that approach proportions are immediately available for any feasible origin-based link flow array. (At nodes with zero origin-based flow, approach proportions can be chosen arbitrarily.) The main effort in obtaining the route flow interpretation by (3.26) is to enumerate all used routes. In some cases route

enumeration may be avoided; for example, as demonstrated by Akamatsu (1997), the expression for route flow entropy can be decomposed as follows,

$$\begin{aligned}
E(\mathbf{f}) = E(\mathbf{h}(\mathbf{f})) &= -\sum_{p\in N_o}\sum_{q\in N_d(p)}\sum_{r\in R_{pq}[A_p^u]} h_{rpq}\cdot\ln\left(\frac{h_{rpq}}{d_{pq}}\right)\\
&= -\sum_{p\in N_o}\sum_{q\in N_d(p)}\sum_{r\in R_{pq}[A_p^u]} h_{rpq}\cdot\ln\left(\prod_{a\subseteq r}\alpha_{ap}\right)\\
&= -\sum_{p\in N_o}\sum_{q\in N_d(p)}\sum_{r\in R_{pq}[A_p^u]}\sum_{a\subseteq r} h_{rpq}\cdot\ln\left(\alpha_{ap}\right)\\
&= -\sum_{p\in N_o}\sum_{a\in A}\ln\left(\alpha_{ap}\right)\cdot\left[\sum_{q\in N_d(p)}\sum_{r\in R_{pq}[A_p^u];a\subseteq r} h_{rpq}\right]\\
&= -\sum_{p\in N_o}\sum_{a\in A}\ln\left(\alpha_{ap}\right)\cdot f_{ap}\\
&= -\sum_{p\in N_o}\sum_{a\in A}\ln\left(\frac{f_{ap}}{g_{a_hp}}\right)\cdot f_{ap}\\
&= -\sum_{p\in N_o}\sum_{a\in A}\left(\ln\left(f_{ap}\right)-\ln\left(g_{a_hp}\right)\right)\cdot f_{ap}\\
&= -\sum_{p\in N_o}\sum_{a\in A}\ln\left(f_{ap}\right)\cdot f_{ap}+\sum_{p\in N_o}\sum_{j\in N}\ln\left(g_{jp}\right)\cdot\sum_{a\in A;a_h=j} f_{ap}\\
&= -\sum_{p\in N_o}\sum_{a\in A}\ln\left(f_{ap}\right)\cdot f_{ap}+\sum_{p\in N_o}\sum_{j\in N}\ln\left(g_{jp}\right)\cdot g_{jp} \qquad (3.27)
\end{aligned}$$

Akamatsu considers the first term as the link entropy

$$E_{link}(\mathbf{f}) = -\sum_{p\in N_o}\sum_{a\in A}\ln\left(f_{ap}\right)\cdot f_{ap} \qquad (3.28)$$

and the second term as the node entropy

$$E_{node}(\mathbf{f}) = -\sum_{p\in N_o}\sum_{j\in N}\ln\left(g_{jp}\right)\cdot g_{jp} \qquad (3.29)$$

Hence by this derivation, the overall entropy is the difference between the link entropy and the node entropy.

$$E(\mathbf{f}) = E_{link}(\mathbf{f}) - E_{node}(\mathbf{f}) \qquad (3.30)$$

The computation of route flow entropy by the last expression does not require route enumeration, and can be done substantially faster than computing entropy using route flows directly.

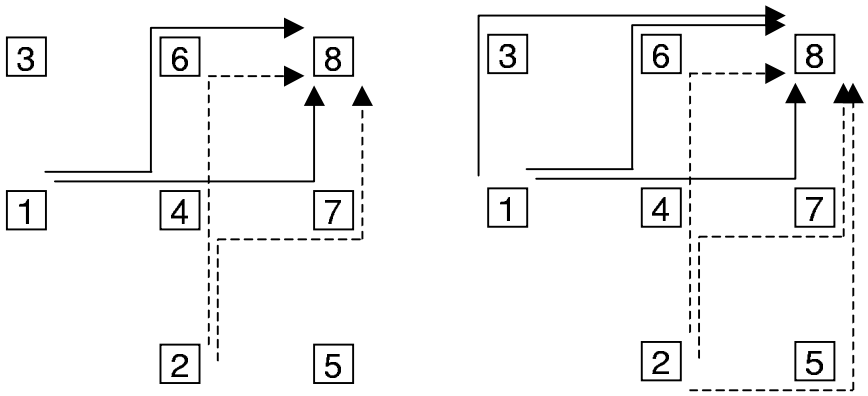
Practitioners are often interested in route-based solutions as they provide detail that is not available by link-based solutions. As discussed in section 1.5, such detail is important in several applications, impact fee assessment, certain emission estimation procedures, "window" models, and more. Using the approach proportionality condition (3.25) and the related route flow interpretation (3.26), the detail provided by an origin-based solution is practically equivalent to route-based solutions.

3.4 Extended approach proportionality

In section 3.3 the origin-based bypass proportionality assumption was used to derive the approach proportionality condition (3.25), which provided a constructive solution for the entropy maximizing route flow interpretation problem. In this section we examine the possibility to extend this result for solving (3.4) when only total link flows are given, using the strong bypass proportionality assumption. In particular we are looking for situations where the approach proportions are expected to be equal across origins, that is $\alpha_{ap_1} = \alpha_{ap_2}$ for some link $a = [i, j]$ and some origins p_1 and p_2. It is quite unlikely to expect the approach proportions to be equal for all origins; however, if the routes from two origins arrive at the termination node j from the same direction, such equality may hold. For example in Figure 10a the approach proportions of links [6,8] and [7,8] are the same for both origins 1 and 2. (All routes carry the same flow.) On the other hand in Figure 10b when additional routes are considered, these approach proportions differ by origin, even though the bypass proportionality assumption still holds. (Again all routes carry the same flow.) The general direction from the origins to the termination node is therefore not sufficient condition for equal approach proportions, and the specific structure of used routes must be considered.

To analyze the difference between the two cases, recall the following definitions. *Origin-based segment flow* $g_{sp\bullet}$ is the aggregation of O-D segment flows over all destinations, that is:

$$g_{sp\bullet} = \sum_{q \in N_d(p)} g_{spq} = \sum_{q \in N_d(p)} \sum_{r \in R_{pq}; s \subseteq r} h_{rpq} \tag{3.31}$$



a) Equal approach proportions and same last common node

b) Different approach proportions and different last common node

Figure 10. Extended approach proportionality

A *common node* from origin p to node j is a node that is common to all used route segments from p to j. The set of common nodes from p to j is

$$COM_{pj} = \bigcap_{s \in R_{pj}[A_p^u]} s \tag{3.32}$$

This definition is slightly different from the one used in chapter 2 as it is based on the used subnetwork, which is not necessarily spanning. As a result the above definition of common node is valid only for used nodes, that is nodes such that there is at least one used route segment from the origin to them.

In Figure 10a, node 4 is a common node from origin 1 to node 8; the same node is also a common node from origin 2 to node 8. In Figure 10b the only common nodes from origin 1 to node 8 is the origin - node 1, and the node 8 itself. Similarly, the only common nodes from origin 2 to node 8 is the origin - node 2, and the node 8 itself. The only node that is common to node 8 for both origins is the node 8 itself. The following lemma suggests that this is the essential criterion for approach proportions to be equal.

Lemma 9. *If* $\mathbf{h}$ *is a feasible route flow vector that satisfies the strong bypass proportionality condition, and if for some node* $j \in N$ *which is used by two origins* $p_1, p_2 \in N_o$*, there is a node* $n \neq j$ *which is a common node from* p_1 *to* j *and also from*

p_2 to j, then the proportion of every approach to j is the same for both origins. That is:

$$\{j\} \subsetneq COM_{p_1 j} \cap COM_{p_2 j} \Rightarrow \alpha_{[i,j]p_1} = \alpha_{[i,j]p_2} \quad \forall [i,j] \in A \tag{3.33}$$

Proof:
Let $n \in COM_{p_1 j} \cap COM_{p_2 j}; n \neq j$. Consider a specific approach $[i,j] \in A$. For any route segments $s \in R_{ni}$, and $s' \in R_{nj}$, destinations $q_1 \in N_d(p_1)$, $q_2 \in N_d(p_2)$ initial routes $r_1^i \in R_{p_1 n}$; $r_2^i \in R_{p_2 n}$, and final routes $r_1^f \in R_{jq_1}$; $r_2^f \in R_{jq_2}$, the strong bypass proportionality condition states that:

$$h_{(r_1^i+s+[i,j]+r_1^f)p_1q_1} \cdot h_{(r_2^i+s'+r_1^f)p_2q_2} = h_{(r_1^i+s'+r_1^f)p_1q_1} \cdot h_{(r_2^i+s+[i,j]+r_2^f)p_2q_2} \tag{3.34}$$

Sum over all possible s, s', q_1, q_2, r_1^i, r_1^f, r_2^i, r_2^f and note that:

$$\begin{aligned}
\sum_{q_1 \in N_d(p)} \sum_{r_1^i \in R_{p_1 n}} \sum_{s \in R_{ni}} \sum_{r_1^f \in R_{jq_1}} h_{(r_1^i+s+[i,j]+r_1^f)p_1q_1} &= f_{[i,j]p_1} \\
\sum_{q_2 \in N_d(p)} \sum_{r_2^i \in R_{p_2 n}} \sum_{s' \in R_{nj}} \sum_{r_2^f \in R_{jq_2}} h_{(r_2^i+s'+r_1^f)p_2q_2} &= g_{jp_2} \\
\sum_{q_1 \in N_d(p)} \sum_{r_1^i \in R_{p_1 n}} \sum_{s' \in R_{nj}} \sum_{r_1^f \in R_{jq_1}} h_{(r_1^i+s'+r_1^f)p_1q_1} &= g_{jp_1} \\
\sum_{q_2 \in N_d(p)} \sum_{r_2^i \in R_{p_2 n}} \sum_{s \in R_{ni}} \sum_{r_2^f \in R_{jq_2}} h_{(r_2^i+s+[i,j]+r_2^f)p_2q_2} &= f_{[i,j]p_2}
\end{aligned}$$

to obtain

$$f_{[i,j]p_1} \cdot g_{jp_2} = g_{jp_1} \cdot f_{[i,j]p_2} \tag{3.35}$$

Node j is used by both origins, hence $g_{jp_1} > 0$ and $g_{jp_2} > 0$, and therefore

$$\alpha_{[i,j]p_1} = \frac{f_{[i,j]p_1}}{g_{j,p_1}} = \frac{f_{[i,j]p_2}}{g_{j,p_2}} = \alpha_{[i,j]p_2} \tag{3.36}$$

□

The condition in (3.33) is fairly general, but it may not be so easy to verify. When all used routes from each origin are included in some a-cyclic subnetwork, an alternative condition can be derived, which is easier to verify. For every such a-cyclic subnetwork A_p^u, a topological order can be defined, i.e. a one-to-one function $o_p : N \to \{1, 2, 3 \ldots |N|\}$ such that $[i,j] \in A_p^u \Rightarrow o_p(i) < o_p(j)$. The *last common*

node, lcn_{pj} from origin p to node j is defined as the common node l with highest value of $o_p(l)$, except for j. If the last common node to j is the same for two origins, the condition in (3.33) is satisfied and approach proportions must be equal. The following lemma shows that it is sufficient to compare only the last common nodes.

Lemma 10. *If* $\mathbf{h}$ *is a feasible route flow vector that satisfies the strong bypass proportionality condition, where the flows from each origin* p *are restricted to some a-cyclic subnetwork* A^u_p*, and if for some node* $j \in N$ *and two origins* $p_1, p_2 \in N_o$*, there is a node* $n \neq j$ *which is a common node from* p_1 *to* j *and also from* p_2 *to* j*, then the last common node to* j *is the same for both origins. That is:*

$$\{j\} \subsetneq COM_{p_1 j} \cap COM_{p_2 j} \Rightarrow lcn_{p_1 j} = lcn_{p_2 j} \tag{3.37}$$

Proof:
Denote $l_1 = lcn_{p_1 j}$; $l_2 = lcn_{p_2 j}$. Suppose $n \in COM_{p_1 j} \cap COM_{p_2 j}; n \neq j$. By definition there is a destination $q_1 \in N_d(p_1)$ and route segments $r_1^i \in R_{p_1 n}\left[A^u_{p_1}\right]$, $s_1 \in R_{nj}\left[A^u_{p_1}\right]$, and $r_1^f \in R_{jq_1}\left[A^u_{p_1}\right]$ such that $h_{(r_1^i + s_1 + r_1^f)p_1 q_1} > 0$. Every used route segment from p_2 to j is of the form $(r_2^i + s_2)$, where $r_2^i \in R_{p_2 n}\left[A^u_{p_2}\right]$, and $s_2 \in R_{nj}\left[A^u_{p_2}\right]$. Again by definition there is a destination $q_2 \in N_d(p_2)$ and route segments $r_2^f \in R_{jq_2}\left[A^u_{p_2}\right]$ such that $h_{(r_2^i + s_2 + r_2^f)p_2 q_2} > 0$. The bypass proportionality assumption states that

$$h_{(r_1^i + s_1 + r_1^f)p_1 q_1} \cdot h_{(r_2^i + s_2 + r_2^f)p_2 q_2} = h_{(r_1^i + s_2 + r_1^f)p_1 q_1} \cdot h_{(r_2^i + s_1 + r_2^f)p_2 q_2} \tag{3.38}$$

which implies that $h_{(r_1^i + s_2 + r_1^f)p_1 q_1} > 0$; hence $(r_1^i + s_2)$ is a used route segment from p_1 to j, and therefore $l_1 \in (r_1^i + s_2)$. Using the topological order $o_{p_1}(l_1) \geq o_{p_1}(n)$; hence $l_1 \in s_2$, i.e. l_1 is common to all used route segments from p_2 to j, and if $l_1 \neq n$ then in each of these segments l_1 comes after n. Applying the same argument in the opposite direction where l_1 replaces n and l_2 replaces l_1 shows that l_2 is common to all used route segments from p_1 to j, and if $l_1 \neq l_2$ then in each of these segments l_2 comes after l_1. But this is a contradiction to the choice of l_1 as the last common node from p_1 to j. Therefore $l_1 = l_2$. □

The conclusion from this section is that if flows from each origin are restricted to a-cyclic subnetworks, then strong bypass proportionality implies that

$$lcn_{p_1 j} = lcn_{p_2 j} \Rightarrow \alpha_{[i,j]p_1} = \alpha_{[i,j]p_2} \quad \forall p_1, p_2 \in N_o; \forall [i,j] \in A \tag{3.39}$$

which we refer to as the extended approach proportionality condition.

The origin-based method described in chapter 2 uses a-cyclic restricting subnetworks, hence the definition of last common node is valid for the resulting solution. In fact, the current implementation finds the last common nodes in the approximation of the

objective function second order derivative (see section 2.5). This method therefore provides a good starting point for embedding the extended approach proportionality condition, so that whenever last common nodes are the same, approach proportions are equal. The resulting solution is closer to satisfying the bypass proportionality assumption and to maximizing route flow entropy. Additional research in that direction can hopefully lead to an origin-based solution method for finding the entropy maximizing user equilibrium route flows.

4. EXPERIMENTAL RESULTS

This chapter presents experimental results for three networks, Sioux-Falls (LeBlanc et al. 1975, where O-D flows are divided by 10 to reproduce results in previous literature), a sketch (aggregate) network for the Chicago region for the year 1990, and the fully detailed regional network of Chicago also for the year 1990. Basic characteristics of the three networks are presented in Table I.

The cost of travel in the Sioux-Falls network consists of travel time only, measured originally in hours and here in minutes. The cost of travel in the two networks of Chicago is a generalized cost, which is a linear combination of travel time, tolls (cents), and distance (miles), converted to minute equivalents. Conversion coefficients are given in Table II. Network and trip generation data for the Chicago networks was kindly provided by the Chicago Area Transportation Study (CATS). Cost conversion coefficients and the trip matrix were prepared by the UIC Transportation Laboratory as a part of another project.

The basic fixed demand static traffic assignment problem for each network was solved by the origin-based method described in chapter 2 as well as by the state-of-the-practice method of Frank and Wolfe. The Frank-Wolfe method 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.

Comparison of the convergence performance of the two methods is given in section 4.1. Analysis of equilibrium solutions and their characteristics is given in section 4.2,

Network	Sioux Falls	Chicago sketch	Chicago regional
Zones (origins)	24	387	1,790
Nodes	24	933	12,982
Links	76	2,950	39,018
O-D pairs	528	93,513	2,297,945

TABLE I: NETWORK CHARACTERISTICS

Network	Chicago sketch	Chicago regional
Tolls (minutes/cent)	0.02	0.1
Distance (minutes/mile)	0.04	0.25

TABLE II: COST CONVERSION COEFFICIENTS

followed by a discussion of the resulting memory requirements in section 4.3. A more detailed description of the progress of the two methods is given in section 4.4.

4.1 Convergence performance

Convergence performance is one of the main, if not the most important criteria in comparing solution methods. The first question in conducting such comparison is which convergence measure to use. Denoting the minimum O-D cost by:

$$C_{pq} = \min \{c_r : \ r \in R_{pq}\} \tag{4.1}$$

we can define the *route excess cost* ec_r as the difference between the route cost and the minimum O-D cost, that is:

$$ec_r = c_r - C_{pq} \quad \forall r \in R_{pq} \tag{4.2}$$

Clearly, at equilibrium the excess cost on all used routes must be zero. Excess cost on used routes is therefore a basic measure of the violation of Wardrop's user equilibrium principle.

The main global (aggregate) measure of convergence used here is the *average excess cost*, weighted by route flow over all used routes of all O-D pairs,

$$AEC = \left(1/\hat{d}\right) \cdot \sum\nolimits_{p \in N_o} \sum\nolimits_{q \in N_d(p)} \sum\nolimits_{r \in R_{pq}} ec_r \cdot h_{rpq} \tag{4.3}$$

where $\hat{d}$ was defined as the total O-D flow (demand). Average excess cost is equivalent to the difference between the objective function and the lower bound obtained from the solution to the linearized subproblem, divided by the total flow. This is a common measure that can be calculated using link-based, origin-based or route-based solutions. It is also common to consider the relative gap, which is the difference between the current solution objective function and the best lower bound obtained so far, divided by the absolute value of the best lower bound.

Another possible measure of convergence is the maximum excess cost over all used routes of all O-D pairs,

$$MEC = \max\{ec_r : \ r \in R, \ h_{rpq} > 0\} \tag{4.4}$$

Maximum excess cost is a sensitive and effective measure for solution accuracy; however, it requires a detailed solution, origin-based or route-based, and cannot be calculated from a link-based solution.

In different applications different convergence measures may be preferred, as well as different convergence criteria. A method that is more effective in achieving high accuracy levels, may be less effective in achieving low accuracy levels. It is therefore important to consider the entire convergence process as a function of computation time.

Figures 11, 15 and 19 show relative gap results for the three networks. Equivalent average excess cost results are shown in Figures 12, 16 and 20, together with maximum excess cost results. Every point in these figures represent one iteration, either full or quick in the case of the origin-based method. All of these figures show the clear advantage of the origin-based solution method over the Frank-Wolfe method, especially in achieving highly accurate solutions, as much as a relative gap of 1.0E-14 and an average excess cost of 1.0E-12 to 1.0E-15. Indeed, some of these results are possibly affected by the machine precision of approximately 2.6E-16. In fact additional iterations exhibit instability, probably due to truncation errors. The main purpose of solving the problem to such high accuracy is to examine the behavior of the method, which we found to be quite pleasing. It is also pleasing to see that the maximum excess cost in all three networks is in the range of 1.0E-9 to 1.0E-12. This means for example that in the solution found for the Chicago regional network the cost of any used route is not greater than the cost of any alternative route by more than 1.0E-9 minute equivalents, which is clearly far below the sensitivity of the most cautious traveler. The low values of maximum excess cost indicate that the origin-based method is rather efficient in eliminating residual flows, demonstrating that the boundary search described in section 2.7 works well.

Additional comparisons of the convergence performance of the two methods and the computational effort to achieve them are given in Tables III and IV. Again we can see the clear advantage of the origin-based method. In the same time required by the origin-based method to achieve machine accurate solutions with average excess costs of about 1.0E-13, the Frank-Wolfe method yields substantially less accurate solutions with average excess costs of about 1.0E-3. Equilibrium average route cost for each network are given as a reference for the average excess cost results reported. Using the Frank-Wolfe method for a substantially longer time provides only a limited improvement, leading to average excess cost in the range of 1.0E-3 to 1.0E-5.

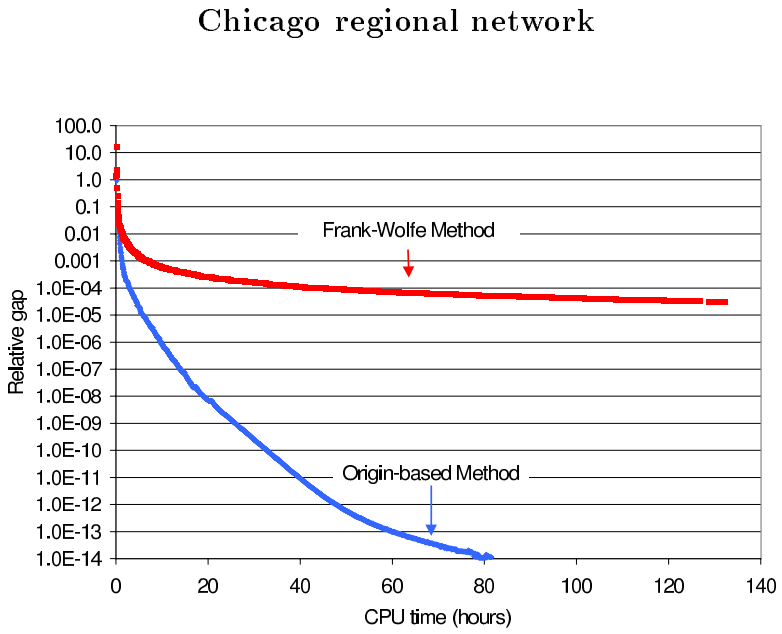


Figure 11. Relative gap vs. CPU time for the Chicago regional network

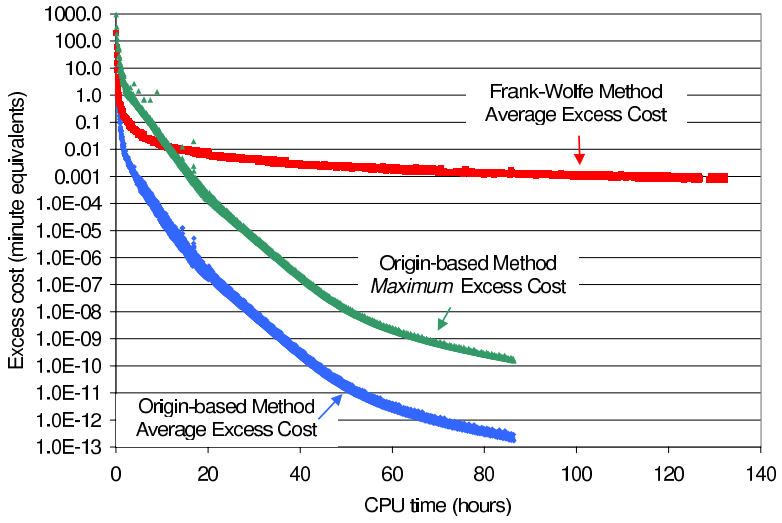


Figure 12. Excess cost vs. CPU time for the Chicago regional network

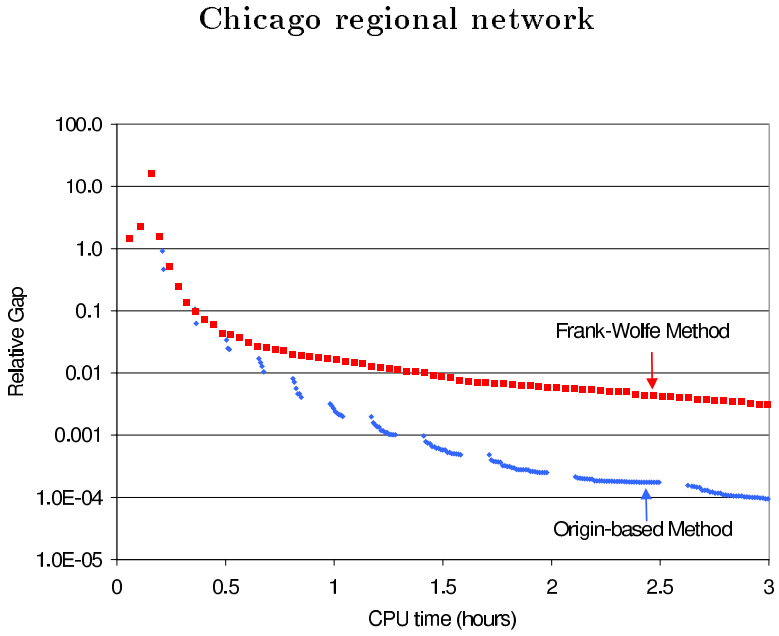


Figure 13. Detail of relative gap vs. CPU time for the Chicago regional network

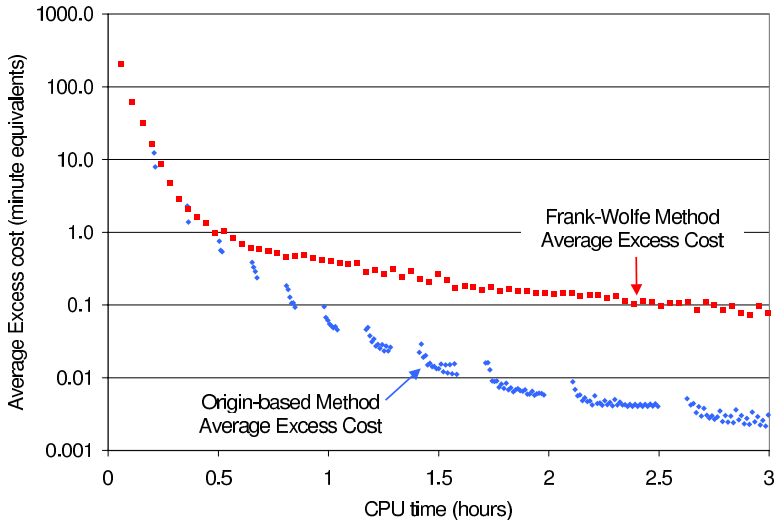


Figure 14. Detail of excess cost vs. CPU time for the Chicago regional network

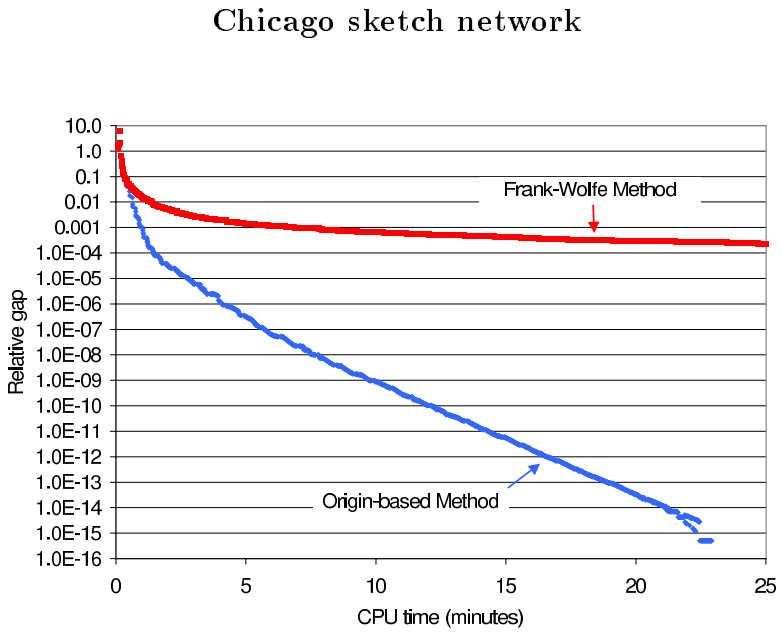


Figure 15. Relative gap vs. CPU time for the Chicago sketch network

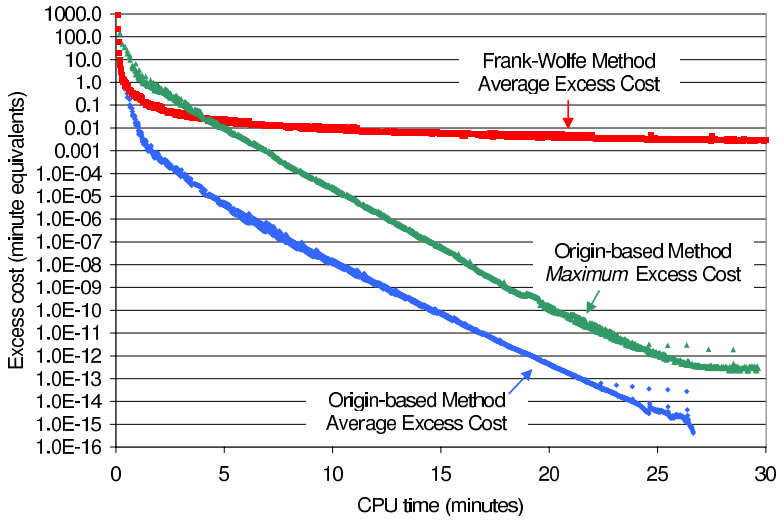


Figure 16. Excess cost vs. CPU time for the Chicago sketch network

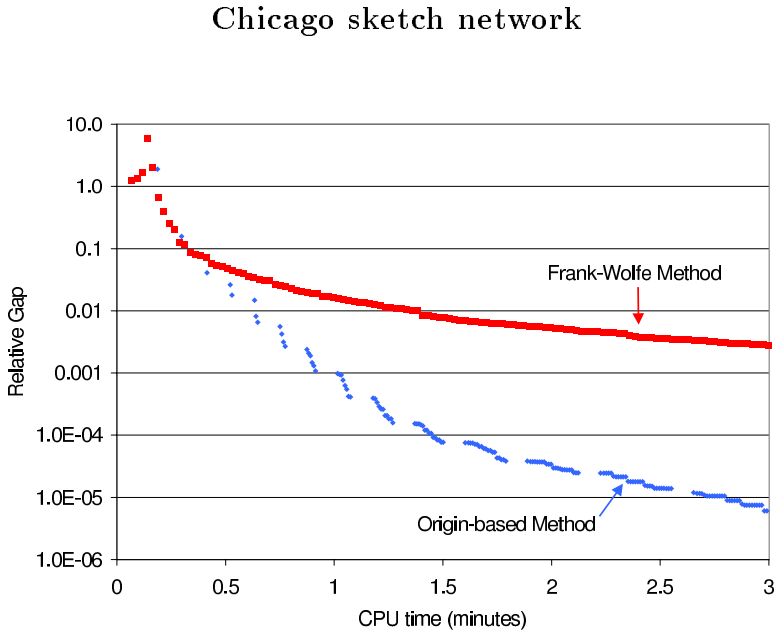


Figure 17. Detail of relative gap vs. CPU time for the Chicago sketch network

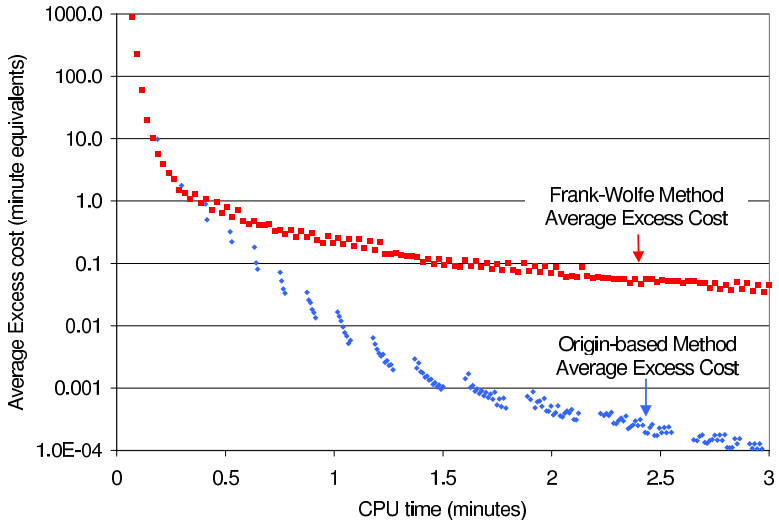


Figure 18. Detail of excess cost vs. CPU time for the Chicago sketch network

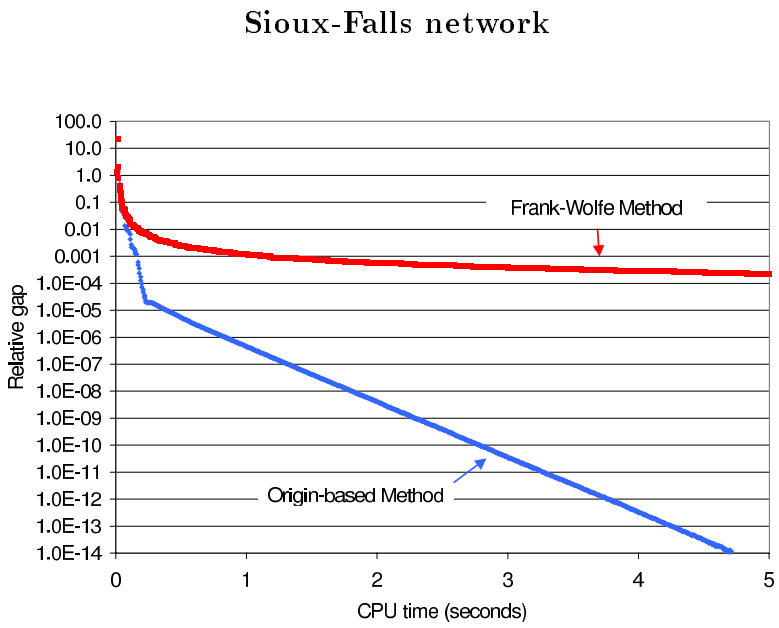


Figure 19. Relative gap vs. CPU time for the Sioux-Falls network

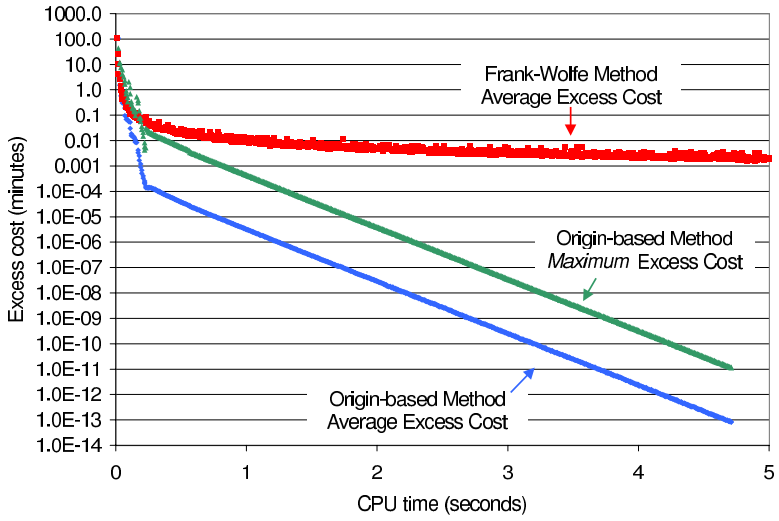


Figure 20. Excess cost vs. CPU time for the Sioux-Falls network

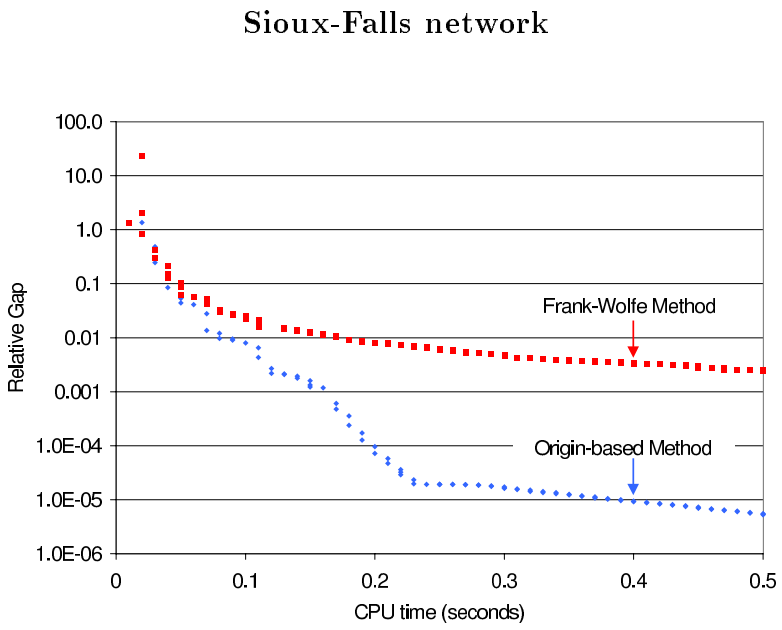


Figure 21. Detail of relative gap vs. CPU time for the Sioux-Falls network

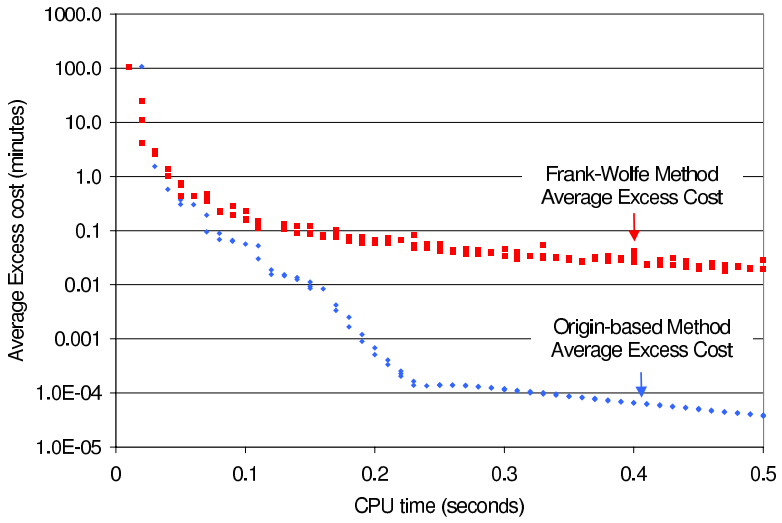


Figure 22. Detail of excess cost vs. CPU time for the Sioux-Falls network

Network	Sioux Falls	Chicago sketch	Chicago regional
CPU time	5 seconds	20 minutes	80 hours
FRANK-WOLFE			
Iterations	1095	806	1,961
Relative gap	2.23E-04	3.03E-04	5.20E-05
Average excess cost	2.07E-03	5.40E-03	1.18E-03
ORIGIN-BASED			
Iterations (full+quick)	985	1380	3277
Restriction updates	50	39	33
Relative gap	1.16E-14	3.29E-14	1.12E-14
Average excess cost	8.50E-14	4.08E-13	3.00E-13
Average route cost	12.45	20.60	28.59

TABLE III: CONVERGENCE COMPARISON FOR A GIVEN CPU TIME

Network	Sioux Falls	Chicago sketch	Chicago regional
CPU time	577 seconds	252 minutes	122 hours
Iterations	100,000	10,000	3,000
Relative gap	2.00E-06	2.20E-05	3.40E-05
Average excess cost	2.83E-05	3.44E-04	1.04E-03

TABLE IV: FRANK-WOLFE METHOD BEST RESULTS

From a practical point of view, spending 20 minutes of CPU time to solve a medium size network like the Chicago sketch network to machine accuracy is quite reasonable, and may often be worth while. When large scale networks like the Chicago regional network are considered, spending more than 80 hours of CPU time to solve one static fixed demand traffic assignment problem is perhaps possible, but probably not cost effective.

Practitioners so far have been satisfied with substantially less accurate solutions, using a relative gap criterion in the range of 0.1 to 0.001. Figures 13, 17 and 21 present relative gap results for this range of accuracies. Equivalent excess cost results are shown in Figures 14, 18 and 22. As can be seen in these figures, the origin-based solution method is not inferior to the Frank-Wolfe method at any accuracy level, and from a certain point onwards it becomes superior.

In the case of the Chicago regional network, for example, both methods have similar performance during the first 30 minutes of CPU time, during which the Frank-Wolfe method performs 10 iterations; the resulting solutions have a relative gap of about 0.05 and average excess cost of about one minute. It should be noted that if the average excess cost is as high as one minute, it is quite possible that a decent portion of the flow uses routes with excess costs of 5 or maybe even 10 minutes; in fact, the origin-based solution at this point has a maximum excess cost of about 80 minutes. Considering such solution as an "equilibrium" solution implies that travelers are not sensitive to cost differences of 5 or 10 minutes, which is probably not a realistic assumption.

From this point of view such a solution should not be considered as an accurate one. To get a more accurate solution with average excess cost of 0.1 minutes the origin-based method needs less than one hour of CPU time, while it takes more than two hours for the Frank-Wolfe method to achieve the same accuracy. The next accuracy level of 0.01 minutes average excess cost requires less than two hours of CPU time with the origin-based method, and more than 10 hours for the Frank-Wolfe method, which is five times longer. As accuracy requirements increase, the advantage of the origin-based method becomes more significant. The trends for the smaller networks are very similar, except that CPU times are of course much shorter.

Another perspective on the same results using a logarithmic time scale is given for relative gap in Figures 23, 25 and 27, and for excess cost in Figures 24, 26 and 28. These figures show the entire process while providing more detailed resolution for the first part of it. The advantage of the origin-based method over the Frank-Wolfe method is exemplified in these figures. The linear trends in the Frank-Wolfe case suggest a power function of the form

$$\text{relative gap} = a \cdot (\text{CPU time})^b \tag{4.5}$$

Regression values of a and b together with the r^2 measure of goodness of fit for the three networks are given in Table V. Extrapolating from these results one might estimate that to obtain a machine accurate solution with a relative gap of say 1.0E-14 using the Frank-Wolfe method would take about 80,000 years for the Chicago regional network, 10,000 years for the Chicago sketch network, and even for the miniature network of Sioux-Falls it would take about 3,000 years of CPU time.

Comment: in general the average excess cost and relative gap are very similar measures of convergence. The ratio between them for a given network is almost constant, which appears in the figures presented here as a simple translation along the ordinate. By preserving the cost units, excess cost also provides intuitive interpretations, like those discussed above. In addition, average excess cost captures the fluctuations inherent to these iterative methods, which are smoothed by the relative gap.

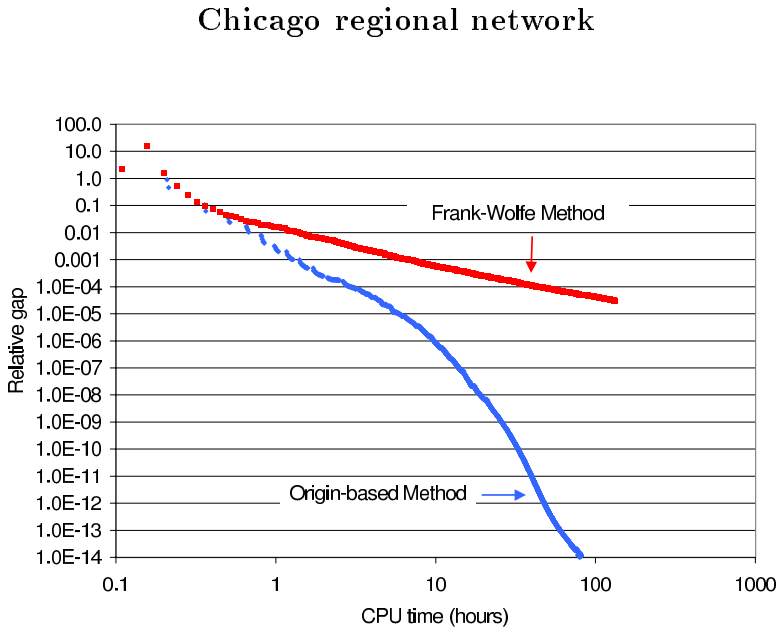


Figure 23. Relative gap vs. CPU time for the Chicago regional network (log)

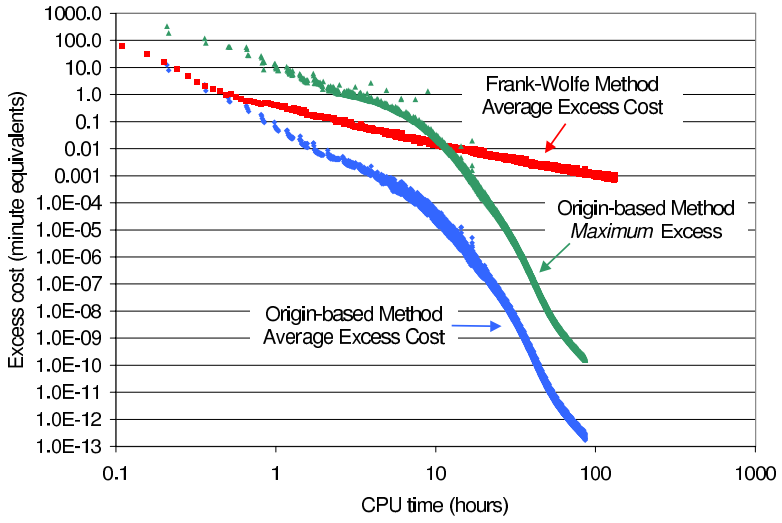


Figure 24. Excess cost vs. CPU time for the Chicago regional network (log)

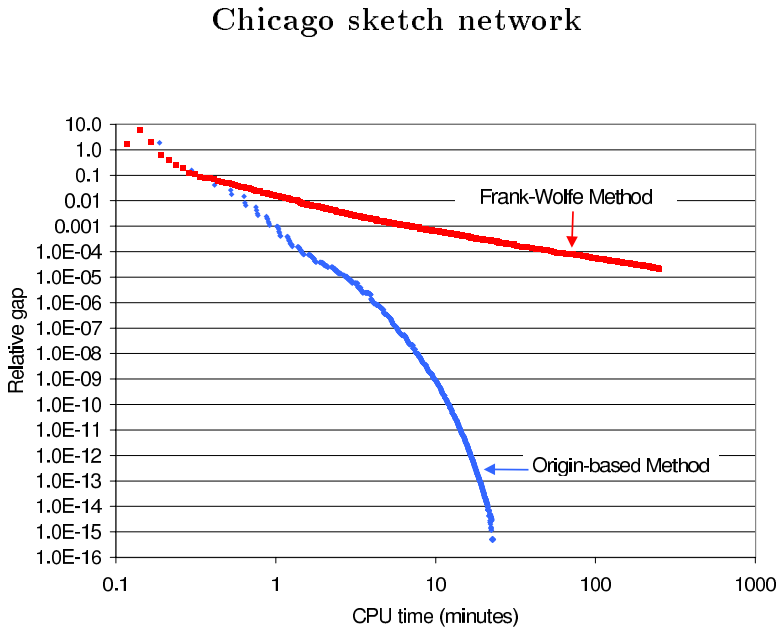


Figure 25. Relative gap vs. CPU time for the Chicago sketch network (log)

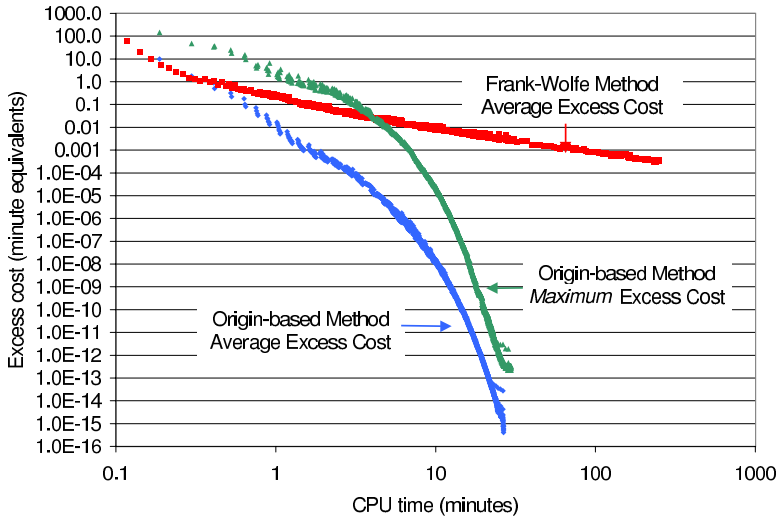


Figure 26. Excess cost vs. CPU time for the Chicago sketch network (log)

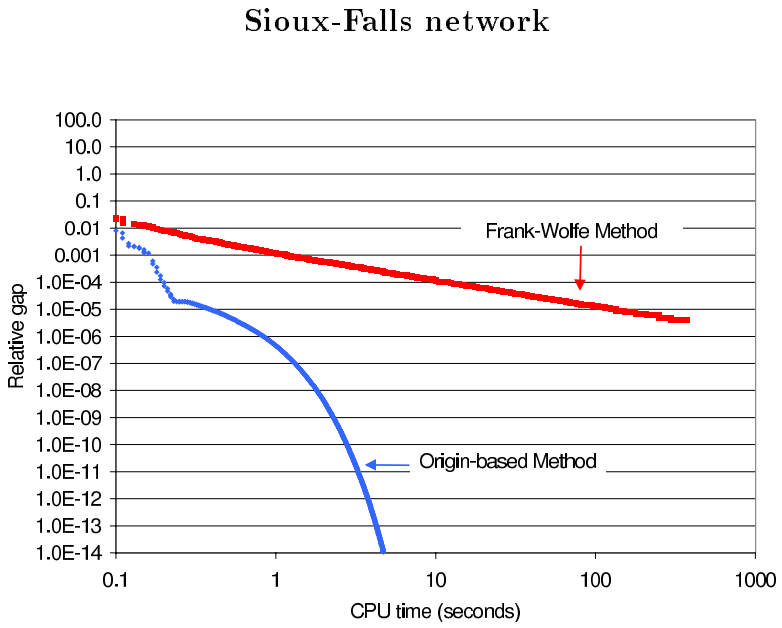


Figure 27. Relative gap vs. CPU time for the Sioux-Falls network (log)

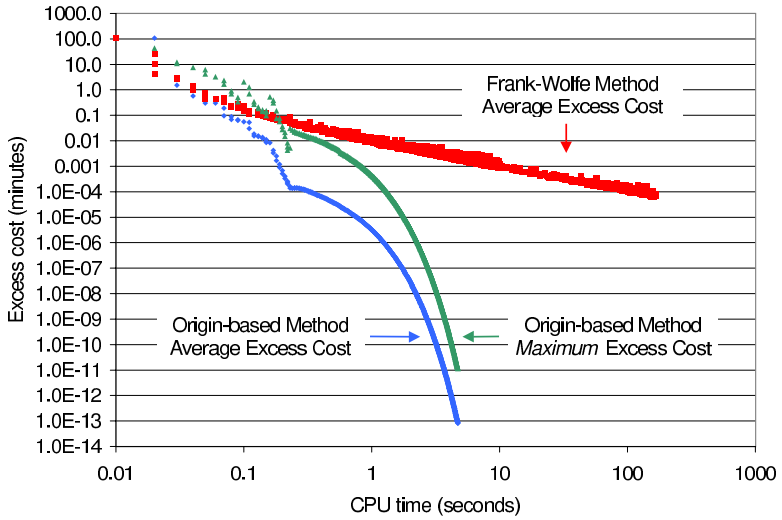


Figure 28. Excess cost vs. CPU time for the Sioux-Falls network (log)

Network	Sioux Falls	Chicago sketch	Chicago regional
a	0.0012	0.013	0.0163
b	-1.0071	-1.2468	-1.3822
r^2	0.9781	0.9747	0.989

TABLE V: FRANK-WOLFE CONVERGENCE REGRESSION

4.2 Characteristics of equilibrium solutions

In addition to fast convergence, origin-based solutions provide much more detail than link-based methods like Frank-Wolfe. To begin with, one can examine the flows from a specific origin, to gain better understanding of the equilibrium solution.

Figures 29-30 show two such examples from an equilibrium solution for the network of Sioux-Falls. The flows from origin 1 (Figure 29) form a relatively simple network, with only two additional (non-basic) links; that is, a tree may be obtained from this network by eliminating two links. Figure 30 shows the flows from origin 12 for that solution, which form a relatively complicated network with seven additional links. One may observe eight different routes from origin 12 to destination 16, which is the maximum number of routes for one O-D pair in the equilibrium solution of Sioux-Falls.

Several parameters describing the structure of the most accurate solutions obtained by the origin-based method are presented in Table VI. As shown in section 4.1 these solutions are extremely well converged and hence may be viewed as equilibrium solutions.

The key parameter describing the origin-based structure is the total number of used links, which is the sum over all origins of the number of links used by travelers from that origin. (Comment: in section 2.8 we distinguished between 'used' links and 'contributing' links; in this section the more intuitive term 'used' links is used, even though in some cases these links may only be contributing and not actually used.) The number of links in one set of spanning trees is given in comparison, this would be the number of used links if no alternative routes were used. The difference between the two is referred to as the number of additional links, which is also the number of non-basic links as defined in section 2.4. Notice that in all of the networks the number of additional links is substantially smaller than the total number of used links, 1-3% in the two Chicago networks, meaning that in general the origin-based subnetworks are fairly similar to trees. A similar conclusion can be drawn from the number of used links terminating at nodes. In a tree there can be only one link terminating at every

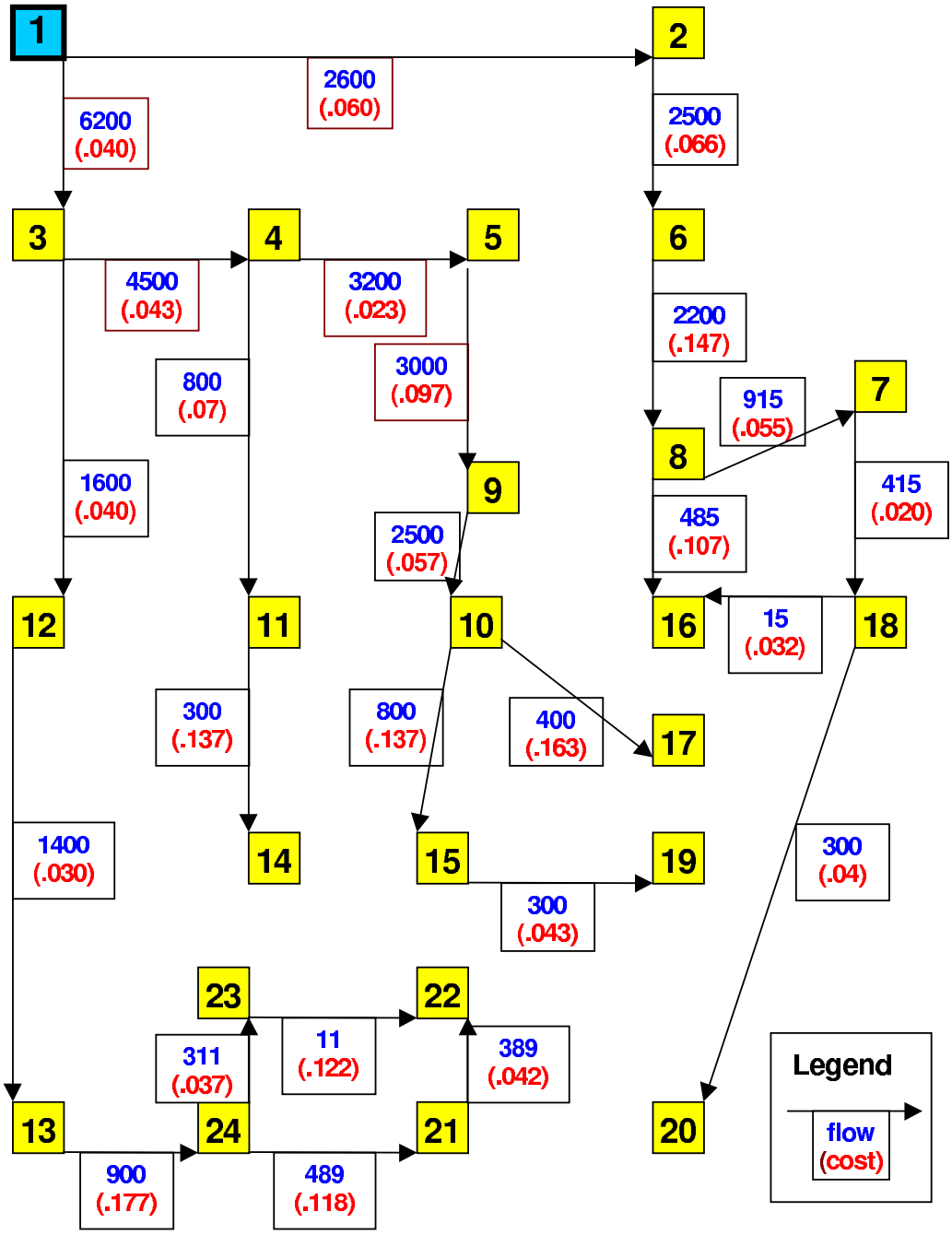


Figure 29. Sioux-Falls equilibrium solution - link flows from origin 1

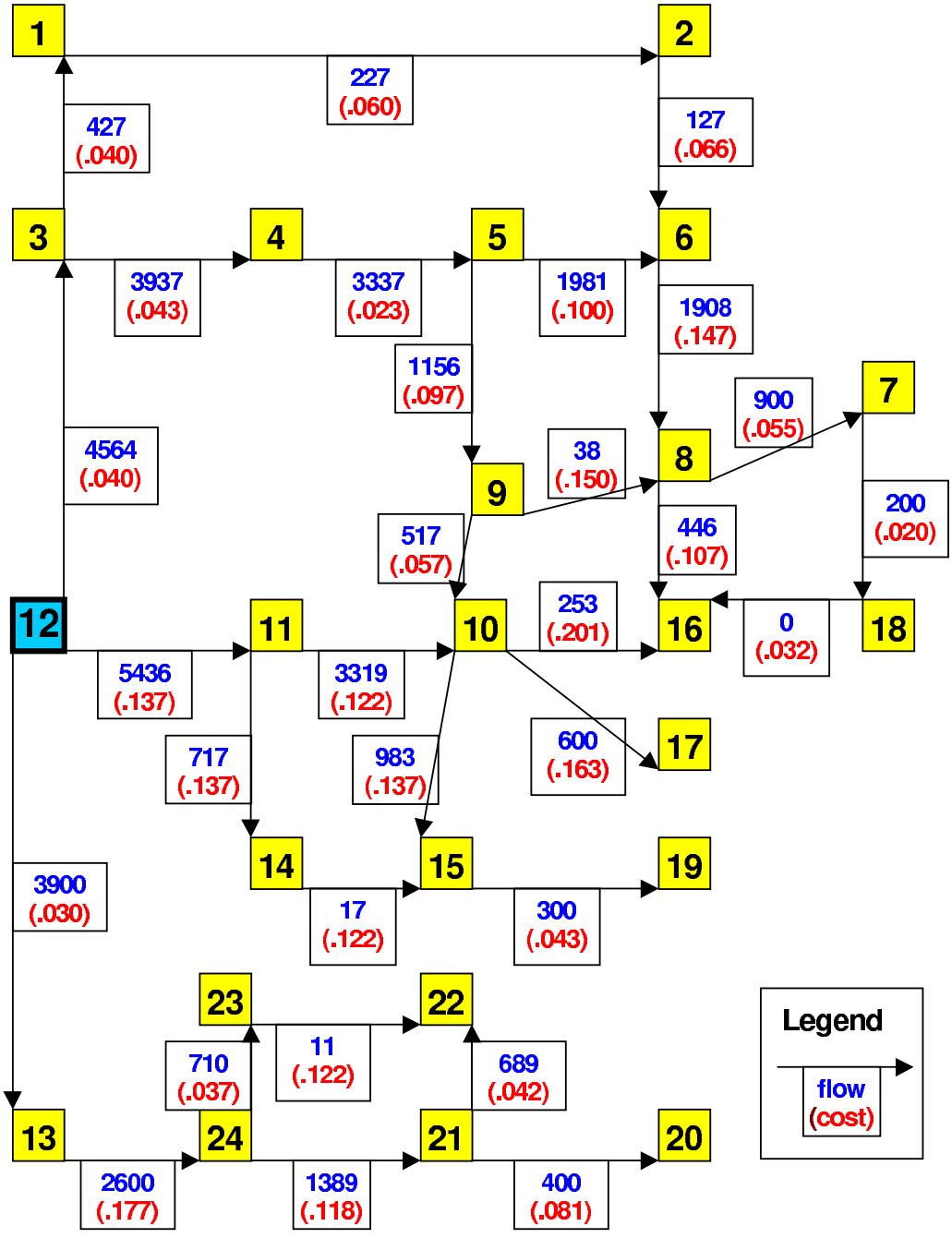


Figure 30. Sioux-Falls equilibrium solution - link flows from origin 12

Network	Sioux Falls	Chicago sketch	Chicago regional
Zones (origins)	24	387	1,790
Nodes	24	933	12,982
Links	76	2,950	39,018
O-D pairs	528	93,513	2,297,945
Links with flow > capacity	60 (80%)	1,165 (40%)	12,256 (31%)
Origin-based used links	632	370,787	23,623,279
# links in one set of trees	552	360,684	23,235,990
Additional links	80	10,103	387,289
Nodes with 1 approach	476	350,690	22,850,715
Nodes with 2 approaches	72	9,885	383,261
Nodes with 3+ approaches	<4	<109	<2,014
Routes	716	219,437	27,251,576
Average routes per O-D pair	1.36	2.35	11.86
Maximum routes per O-D pair	8	327	136,498
Average # links per route	3.5	14.8	55.8
Additional routes	188	125,924	24,953,631

TABLE VI: EQUILIBRIUM SOLUTION STRUCTURE

node. In the equilibrium solution we find that this is the case for the vast majority of the nodes. A small portion of the nodes, only 1-3% for the two Chicago networks, have two used links terminating, and a negligible number of nodes have three or more used links terminating.

The route-based structure is described by the total number of used routes, average number of routes per O-D pair, the maximum number of routes used by a single O-D pair, the average number of links in a route, and the number of additional routes, defined as the difference between the total number of used routes and the number of O-D pairs with positive flow. The number of additional routes is also the number of degrees of freedom, or the number of decision variables, in determining the equilibrium flows once the set of equilibrium routes is known. In a similar fashion the number of additional links can be viewed as the number of degrees of freedom in determining the origin-based link flows once the set of equilibrium routes and hence the equilibrium origin-based subnetworks are known. As can be seen from Table VI, the number of additional links is typically substantially smaller than the number of additional routes, suggesting a reduced optimization complexity, which probably contributes to the computational efficiency of the origin-based method.

From all the values presented in Table VI the most unexpected one is perhaps the maximum number of routes for a single O-D pair on the Chicago regional network. Having as many as 136,498 different alternative routes of equal cost between one O-D pair may seem at least at a first look as an exaggeration. On the other hand, the number of routes from a North-West corner to a South-East corner of a grid of 11 by 11 nodes when only North to South and West to East links are used is $\binom{20}{10} = 184,756$. The regional network of Chicago with almost 13,000 nodes and almost 40,000 links is a much bigger network, with many grid-like portions where many similar alternatives exist. Having 136,489 different equilibrium routes for a single O-D pair on such a network may therefore be somewhat surprising, but not completely inconceivable. This argument demonstrates that such a value is possible, but it may still be an outlier. To examine this issue, we consider the distribution of O-D pairs by the number of equilibrium routes.

Figure 31 presents the frequency of O-D pairs by their number of equilibrium routes for the range 1-100. As can be seen in this figure, the distribution in all three networks is rather fluctuating. It is interesting to point out that typically prime numbers of equilibrium routes are less frequent than product numbers. For example in the Chicago regional network, there are 6,880 O-D pairs with 48 equilibrium routes, and only 90 O-D pairs with 47 equilibrium routes; there are 2,346 O-D pairs with 96 equilibrium routes, and only 12 O-D pairs with 97 equilibrium routes. A product number of equilibrium routes can be the result of a route structure that consists of several sections, with some alternatives in each section, where the total number of equilibrium routes is the product of the number of alternatives in each section. For example if there are three sections, two alternatives in the first section, three alternatives in the second section, and two alternatives in the last section, then the total number of equilibrium routes will be $2 \cdot 3 \cdot 2 = 12$. If such structures are common, as suggested by Figure 31, a route-based representation is clearly inefficient, since every route is represented separately, and it does not take advantage of the special structure of the routes.

Given the fluctuations of the equilibrium route distribution, frequency representation is not likely to provide much insight, especially for large number of equilibrium routes. To reduce the effect of these fluctuations on the graphical representation we consider their cumulative distributions. In addition to fluctuations, the distribution of equilibrium routes is also extremely skewed, in the case of the Chicago regional network having a mode of 1, mean of 11.86 and maximum of 136,498. A regular ascending cumulative distribution representation for such data is dominated by the first few values, and the contribution of O-D pairs with large number of equilibrium routes is completely unobservable. To resolve this problem we consider in Figure 32 an inverted (descending) cumulative distribution for the three networks.

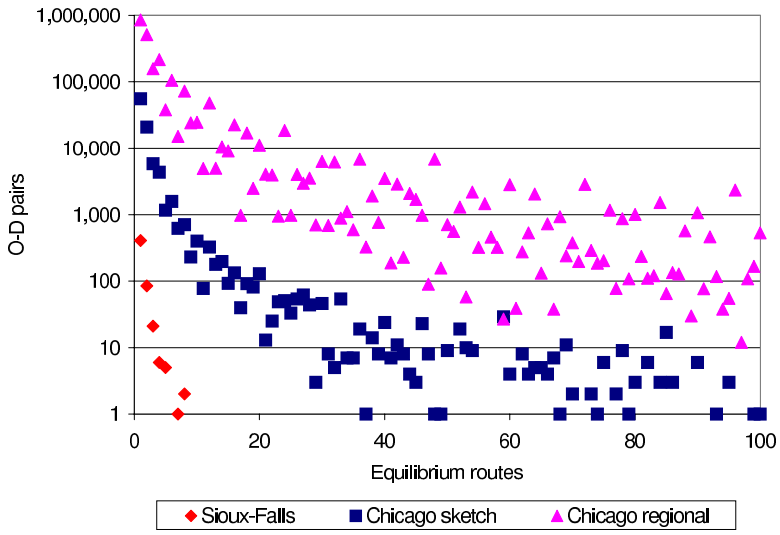


Figure 31. Frequency distribution of O-D pairs by equilibrium routes

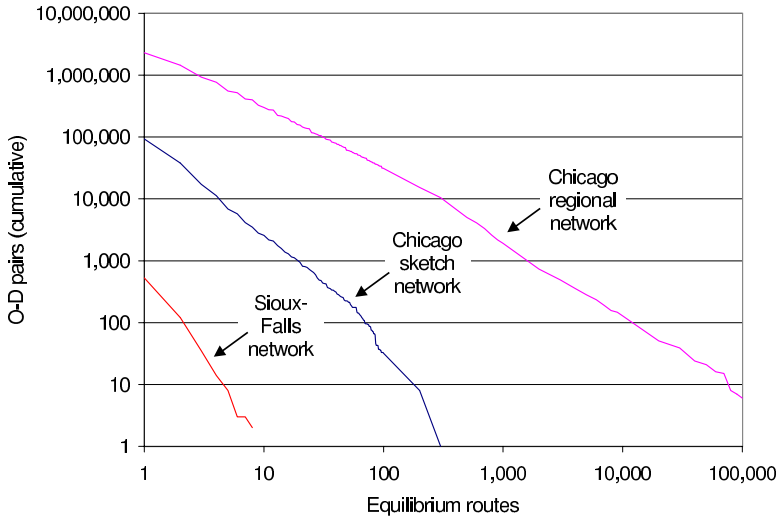


Figure 32. Inverted cumulative distribution of O-D pairs by equilibrium routes

For any given number of equilibrium routes, this figure presents the number of O-D pairs that have at least as many routes, or more. The results for the Chicago regional network should be interpreted as follows. All 2,297,945 O-D pairs have at least one route, 301,707 of them have 10 equilibrium routes or more, 31,147 of them have 100 equilibrium routes or more, 1,916 of them have 1,000 equilibrium routes or more, 128 of them have 10,000 equilibrium routes or more, and 6 of them have 100,000 equilibrium routes or more. From these values we learn that the O-D pair with 136,489 routes is not an outlier, as there are several other O-D pairs with almost as many equilibrium routes.

4.3 Memory requirements

In this section the memory required by link-based, origin-based, and route-based solution methods is discussed. Such a comparison depends on many assumptions regarding the implementation. We follow the assumptions used in our implementation; in particular 8 bytes double precision reals and 4 bytes integers are assumed throughout the analysis. In addition, this analysis considers only the main data structures. The remaining components have a negligible impact on the results for the large regional network of Chicago, which is our main interest. As for the network of Sioux-Falls, memory requirements are so small that even the length of an additional string may be significant, which is of course not taken into account. Table VII summarizes memory requirements of different components of the equilibrium solutions described in the previous section. Values quoted in the following discussion refer to the Chicago regional network.

One possible way to store an origin-based solution is by storing the entire origin-based link flow array, that is one floating point value for every link for each origin. This is a rather naive implementation, that leads to large memory requirements (558 MB). We developed a special data structure that reduces memory requirements substantially (to 112 MB). The memory required to store a link-based solution, that is to store the total flow on every link (0.3 MB), is still much smaller. On the other hand, it is always necessary to store the input data, and in particular the O-D trip table. The difference between the total memory required by a link-based method (at least 28 MB) and our origin-based method (140 MB) is still significant but not as dramatic. In addition, in many cases it is useful to store at least one set of minimum cost route trees even if it is not necessary for the solution method. In such a case, the additional 22 MB needed to store a fully detailed origin-based solution rather than storing only one set of trees is probably worth while.

Network	Sioux Falls	Chicago sketch	Chicago regional
Zones (origins)	24	387	1,790
Nodes	24	933	12,982
Links	76	2,950	39,018
O-D pairs	528	93,513	2,297,945
Trip Table	4.6 KB	1.20 MB	25.6 MB
Other input	3.0 KB	0.10 MB	1.6 MB
Link-based solution	0.6 KB	0.02 MB	0.3 MB
Set of trees	2.3 KB	1.40 MB	93.0 MB
Origin-based solution:			
naive implementation	14.6 KB	9.13 MB	558.7 MB
our implementation	5.9 KB	1.90 MB	111.5 MB
lower bound	2.3 KB	1.40 MB	93.0 MB
upper bound	48.0 KB	30.00 MB	1,861.6 MB
Route flows	5.7 KB	1.80 MB	218.0 MB
Route description:			
by link lists	10.0 KB	13.00 MB	6,086.9 MB
by trees	>8.0 KB	>27.70 MB	>89,661.4 MB

TABLE VII: MEMORY REQUIREMENTS

There are several different ways to store route-based solutions. In all cases the flow on every route is stored. This part by itself consumes a descent amount of memory, about as much as the entire origin-based solution in the case of the Sioux-Falls network and the Chicago sketch network, and almost twice as much as the entire origin-based solution in the case of the Chicago regional network. In addition it is also necessary to store a description of the routes. A common and simple way to do that is by storing a list of the links consisting each route. The amount of memory required for that purpose is extremely large, 6 GB, more than 50 times larger than the origin-based solution. Other data structures may be more efficient; for example it has been proposed to store several different routes from the same origin by one tree. We do not know what is the most efficient arrangement of routes in trees, and how much effort is involved in obtaining and maintaining the required data structure throughout the iterative process; in any case, the number of trees from a specific origin in such an arrangement must be at least as large as the maximum number of routes to a single destination from that origin. We observed that for some networks, especially less congested networks, such data structure may improve upon the link-list structure. As shown in Table VII, for the equilibrium solutions to the networks presented here, arranging the routes in trees is probably not useful, and may in fact lead to substantially larger memory requirement (89 GB).

For large machines currently available with 256MB–1GB of RAM, solving the Chicago regional network using a route-based method is probably not practical. In the future machines with more memory will become available. Even then, spending that much more memory on a route-based data structure, when origin-based solution provides equivalent detail (see section 3.3) seems quite hard to justify.

It should be noted that memory requirements for origin-based and route-based solutions depend not only on network size, but also on the characteristics of the specific equilibrium solution, and hence on the level of congestion. The lower bound on the memory requirement in the current implementation of the origin-based method, when there is only one equilibrium route for each O-D pair, is equivalent to storing one set of trees (93 MB). A route-based method that organizes route descriptions in trees may have a similar lower bound. The memory required by the more common link-list structure depends on the specific routes used. If we assume that the average number of links in a route is the same as in the equilibrium solution presented here (55.8), storing the description of one route for each O-D pair as a list of links for the Chicago regional network requires 128 MB.

The upper bound for the current implementation of the origin-based method is about 1.8 GB. Indeed the naive implementation for storing an origin-based solution requires only 558 MB regardless of the specific solution; however, our data structure provides other merits that are used in the implementation. For example it also stores the topological order for each of the restricting subnetworks. We find that the actual amount of memory used by our origin-based data structure does not vary too much with the level of congestion, and is typically fairly close to the lower bound.

As for route-based solutions, there does not seem to be any practical upper bound on their memory requirement, as the total number of simple routes increases in an exponential fashion with network size. Furthermore, it seems that the number of routes and hence the memory required by a route-based solution is much more sensitive to the congestion on the network. For example the equilibrium solution for the Chicago sketch network with half the demand yields a substantially less congested network with 335 (11%) links where flow exceeds capacity instead of 1,165 (40%). The resulting origin-based solution requires 1.6 MB of memory instead of 1.9 MB of memory. The number of equilibrium routes is only 116,670 which is about half the original value of 219,437 routes; as a result the memory requirement for the route-based solution is about one half of the values mentioned above.

4.4 Solution method progress

This section shows some details regarding the progress of the solution methods. First the magnitude of the step size in both the Frank-Wolfe and the origin-based methods is examined. Second the development of the restricting subnetworks structure in the origin-based case is studied.

Figures 33, 35 and 37, show the Frank-Wolfe step sizes for the three networks through all iterations. Figures 34, 36 and 38, show the Frank-Wolfe step sizes in the first 100 iterations. It is evident from these figures that except for the first 20-30 iterations, step sizes in the Frank-Wolfe method are rather small, less than 0.1, and they get smaller and smaller, as the method proceeds. This is quite expected, since as the solution approaches equilibrium the necessary shifts get smaller and smaller; however, the search direction in the Frank-Wolfe method represents shifting all the flow for each O-D pair to the route that currently has the lowest cost. It should be pointed out that the same step size is applied to all the shifts, those where the cost difference is large and a larger shift may be appropriate, those where the cost difference is small and a smaller shift may be better, and even those where the cost is equal or practically equal and there should not be any shift at all. This is probably one of the main reasons for the computational inefficiency of the Frank-Wolfe method as demonstrated in section 4.1.

In the origin-based method, at every iteration, a separate step size is chosen and applied to each of the origins. To describe the distribution of step sizes in every iteration, we divide the origins into seven categories. The first category includes the origins that appears to be at equilibrium, that is the method suggests that the flows from these origins should remain as they are, without any shifts at all. This is the case if all used routes from the specific origin to each of the destinations have the same cost as the route of minimum cost to the same destination in the current restricting subnetwork, at least up to the machine precision. In particular if the restricting subnetwork is a tree, where there is only one route to each destination, the origin will be considered as an equilibrium origin. It is more likely that the restricted subnetwork does contain some additional routes, from which flow has been eliminated in previous iterations, and the cost of these unused routes is still higher than the cost of the used route.

The second category includes those origins for which a step size of 1.0 was chosen, the third, fourth, and fifth categories include those origins for which a step size of 0.5, 0.25, 0.125 was chosen respectively. The sixth category include all other origins where some shift was applied, that is where the chosen step size is between 0.1 and 1E-10. When a step size of 1E-10 is reached, the search is stopped, the step size is

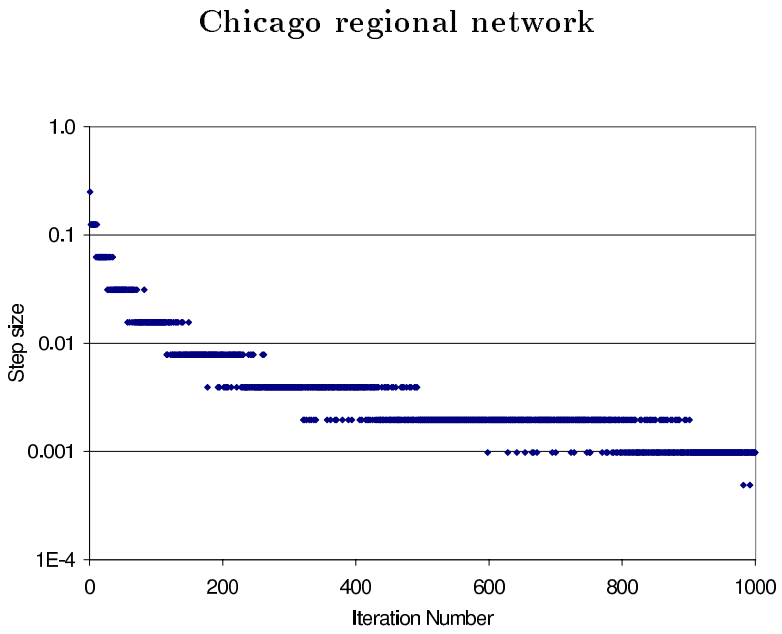


Figure 33. Frank-Wolfe method step size for the Chicago regional network

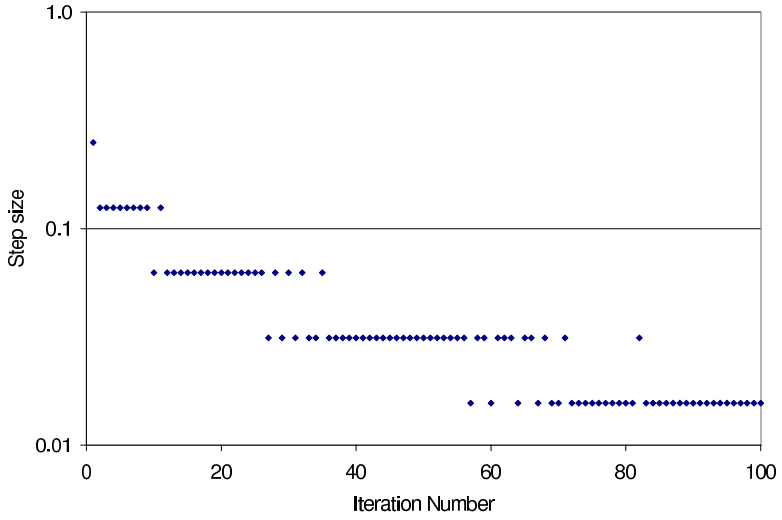


Figure 34. Detail of FW method step size for the Chicago regional network

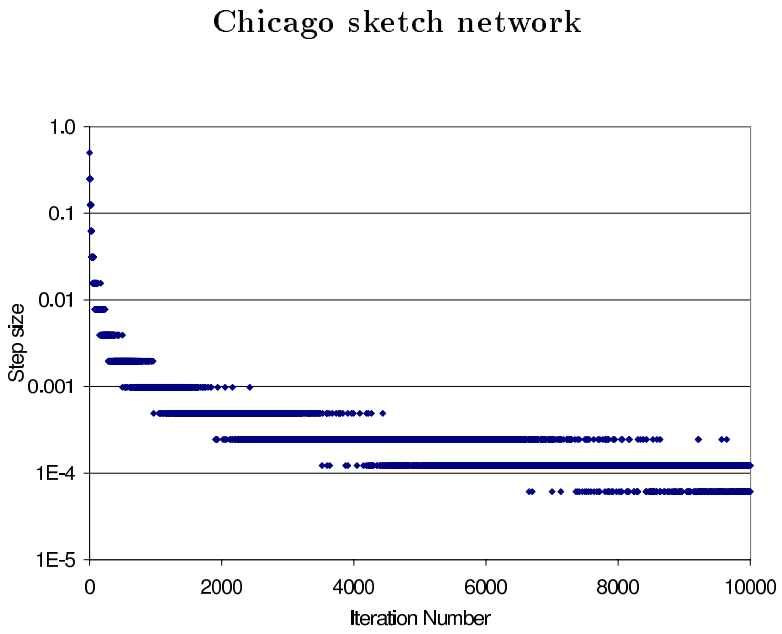


Figure 35. Frank-Wolfe method step size for the Chicago sketch network

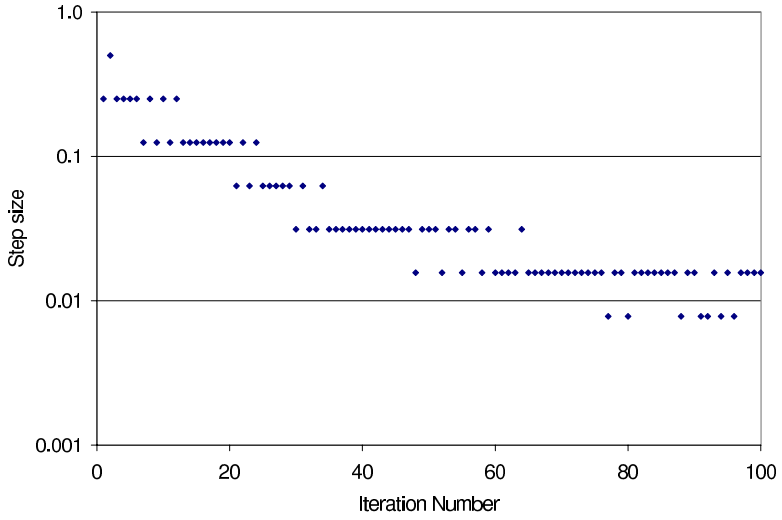


Figure 36. Detail of FW method step size for the Chicago sketch network

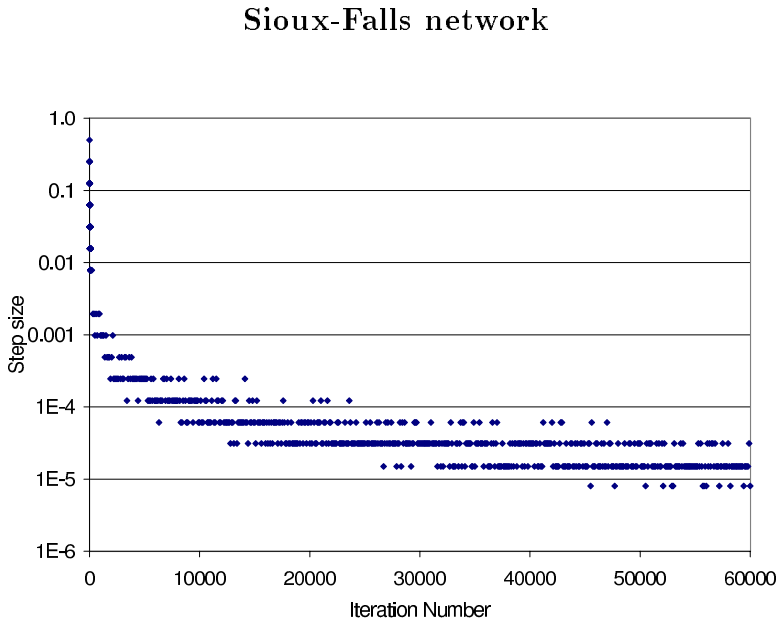


Figure 37. Frank-Wolfe method step size for the Sioux-Falls network

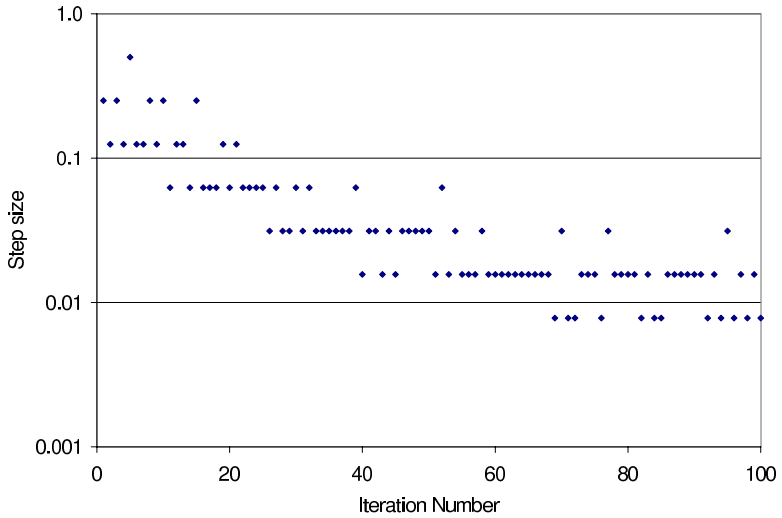


Figure 38. Detail of FW method step size for the Sioux-Falls network

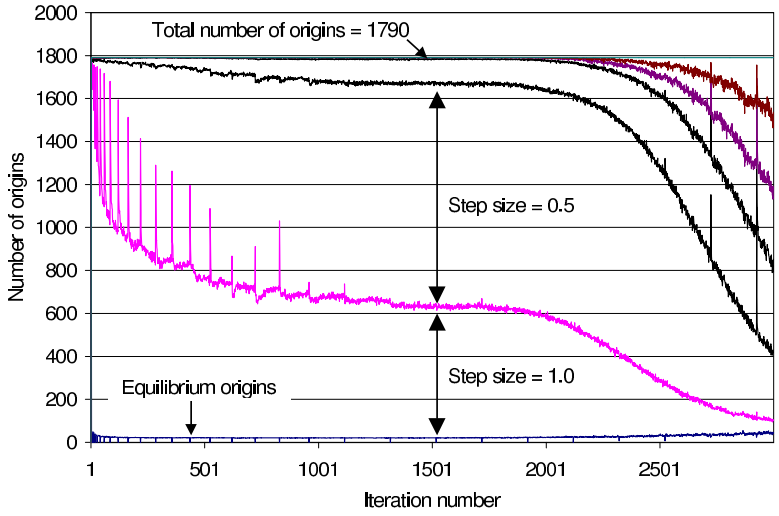


Figure 39. Origin-based method step size for the Chicago regional network

truncated to zero, and the shift is ignored. Origins where the step size was truncated consist the seventh category.

Figures 39, 40 and 41, show the number of origins in each category in every iteration for the three networks. These are "stacked" diagrams, where the lowest line represents the first category - equilibrium origins, the difference between the first and second lines represent the number of origins in the second category - step size 1.0, the difference between the second and third lines represent the number of origins in the third category - step size 0.5, and so on. The highest line represents the total number of origins in all categories which is of course constant for each network.

From these figures we see that throughout most of the iterative process for the vast majority of the origins, a step size of 1.0 or 0.5 is used. A step size smaller than 0.1 is almost never used, except for the last iterations of the larger networks. The relatively large step sizes imply that the proposed shifts are not too large. For many origins a step size of 0.5 is chosen, meaning that a step size of 1.0 is found to be too large. This implies that the proposed shifts are not too small. In short, the shift proposed by this method using link costs and their derivatives seems to be well estimated. These well estimated shifts provide an important contribution to the computational efficiency of the method.

The convergence of the origin-based method to a global optimum depends highly on the ability of the method to choose the correct restricting subnetworks. It is practi-

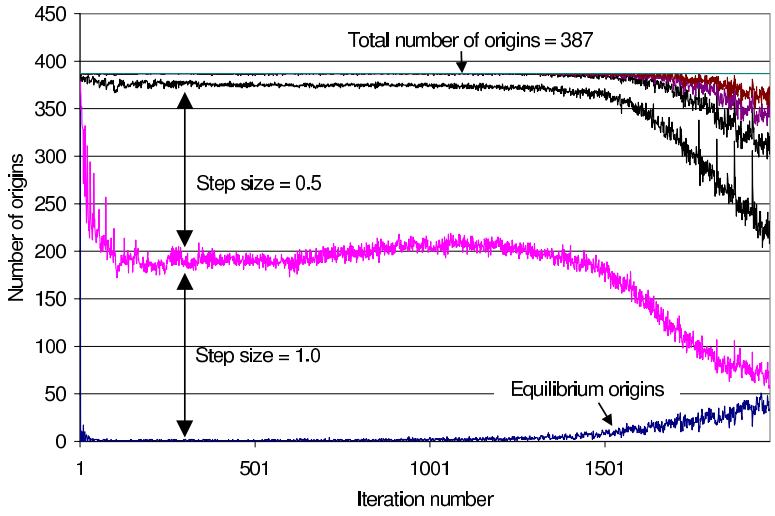


Figure 40. Origin-based method step size for the Chicago sketch network

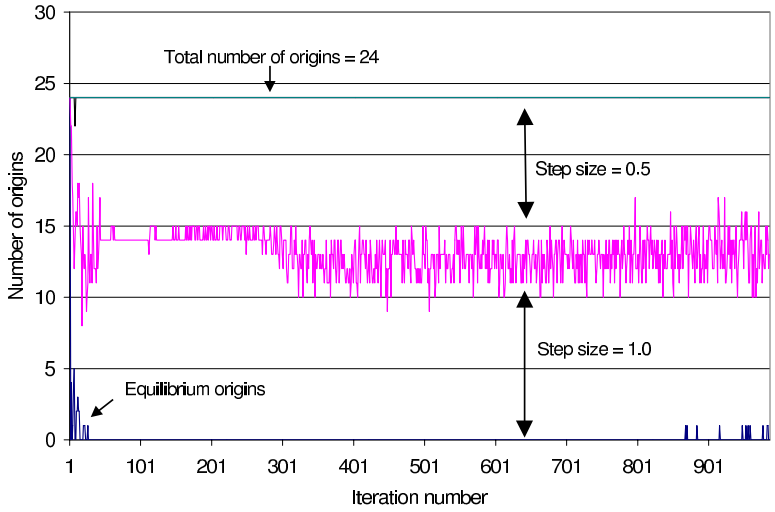


Figure 41. Origin-based method step size for the Sioux-Falls network

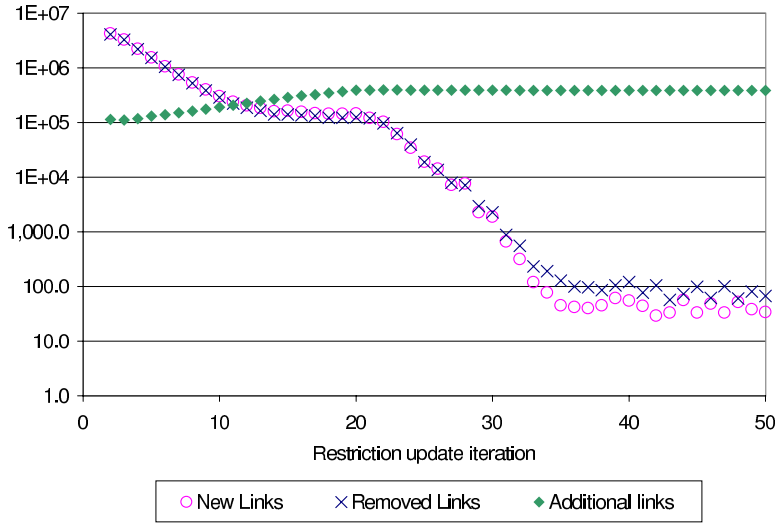


Figure 42. Origin-based structure progress for the Chicago regional network

cally impossible to monitor the development of each restricting subnetwork throughout the iterations, especially for a large-scale problem. Figures 42, 43 and 44 describe the development of the restricting subnetworks using three key parameters. The first is the total number of additional (non-basic) links in the subnetworks resulting from the restriction update procedure. The second is the number of empty links removed from the restricting subnetworks, and the third is the number of new basic links added to the restricting subnetworks.

Comment: Links added by the restriction update procedure described in section 2.8 carry flow only if they are basic, in which case the new link is stored. Links that satisfy the condition $u_i < u_j$, but are not basic, do not carry flow at the end of the full iteration, and therefore in order to save memory they are not stored in the new restricting subnetwork.

We can see that the number of additional links increases gradually in the first few iterations, and then stabilizes. The number of links added and removed decreases continuously, although some changes continue to occur until the last iteration, especially in the Chicago regional network. These changes suggest that although the solutions obtained are extremely accurate, as discussed in section 4.1, there is still some uncertainty regarding the exact structure of the equilibrium routes. The number of links added and removed in the last iterations is quite small, especially in comparison with the size of the problem; therefore, we may conclude that the resulting uncertainty is not of substantial significance.

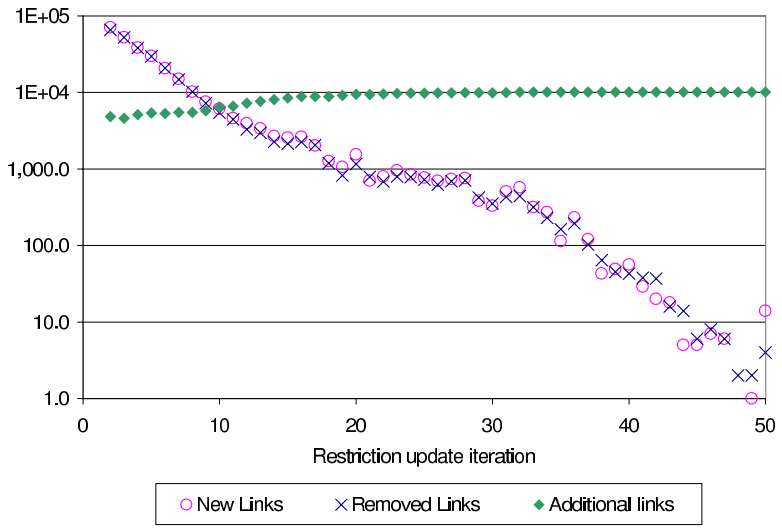


Figure 43. Origin-based structure progress for the Chicago sketch network

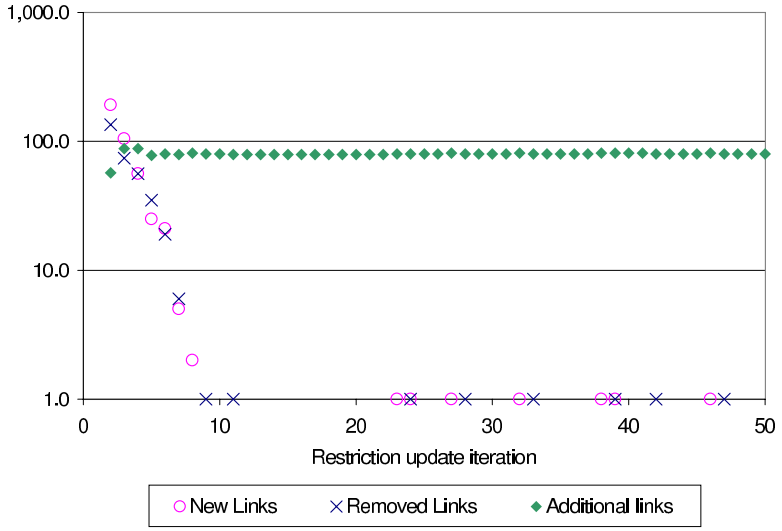


Figure 44. Origin-based structure progress for the Sioux-Falls network

5. CONCLUSIONS

The main contribution of this research is the introduction of a computationally efficient, memory conserving, origin-based solution method for the user equilibrium traffic assignment problem that provides highly accurate solutions. The second main contribution of this work is the introduction of the behavioral concept of bypass proportionality, and the study of its relationship to entropy maximization.

There are two main reasons to prefer origin-based algorithms over the state-of-practice Frank-Wolfe method for practical applications: detailed solution, and substantially lower computation time when higher accuracy is required. The origin-based method allows one to obtain extremely accurate solutions, up to the machine accuracy, which is effectively impossible using the Frank-Wolfe method. The amount of computation time required to obtain such highly accurate solutions may be considered fairly reasonable for practical purposes, at least for small and medium size networks. As for large-scale networks, achieving the machine accuracy limit is probably not cost-effective, at least not using currently available computers.

Accuracy requirements in practice depend on computer technology. The number of Frank-Wolfe iterations that can be computed in a practical sense for large-scale networks has increased from 5-10 in 1980 to 20-40 presently. As computer speed and memory increase further, additional iterations will likely be performed, as more highly converged solutions are desired. At that time we suggest that origin-based algorithms will become the preferred method.

There are several reasons for the computational efficiency of the origin-based method. The choice of different flow shifts for each pair of alternative approaches, taking costs and cost derivatives into account, yields a well-estimated search direction that needs only a small adjustment using a scaling step size. The boundary search procedure provides an efficient tool to eliminate residual flows, thus allowing the addition of new routes while refraining from the introduction of cycles. The restriction to a-cyclic origin-based subnetworks allows the definition of topological order. Using the topological order, the computation time of the minimum, maximum, and average cost from the origin to all destinations is linear in the number of links in the subnetwork. The computation time of the search direction per origin in a quick iteration is also on the order of the number of links in the subnetwork of that origin. The topological order of an a-cyclic network can also be found in a time which is a linear function of the total number of links. Another possible contribution to the method's efficiency

is the relatively moderate optimization complexity, as measured by the number of independent variables, especially in comparison with the route-based approach.

Origin-based solutions have an immediate route flow interpretation. This interpretation can be justified either by bypass proportionality, or by entropy maximization. In that sense the detail provided by an origin-based solution is practically equivalent to the detail of a route-based solution. Such detail is not provided by a link-based solution. As noted in section 1.5, such detail is needed for several important practical applications like impact fee assessment, emission estimation, "window" models, and more. Detailed solutions also allow one to adjust a feasible solution when demand or network topology are modified. This feature makes origin-based algorithms highly suitable in cases where the traffic assignment problem is one component of a larger transportation modeling problem. Route-based methods also provide detailed solutions; however, origin-based methods are more suitable for practical large-scale applications because of their reasonable memory requirements.

CITED LITERATURE

Akamatsu, T. (1996). Cyclic flows, Markov process and stochastic traffic assignment. *Transportation Research*, **30B**, 369–386.

Akamatsu, T. (1997). Decomposition of path choice entropy in general transport networks. *Transportation Science*, **31**, 349–362.

Bar-Gera, H., and D. Boyce (1999). Route flow entropy maximization in origin-based traffic assignment. In *Transportation and Traffic Theory, Proceedings of the 14th International Symposium on Transportation and Traffic Theory, Jerusalem, Israel, 1999*, A. Ceder, ed., Elsevier Science, Oxford, UK, 397–415.

Beckmann, M., C. B. McGuire, and 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. In *Proceedings of the International Symposium on Systems Optimization and Analysis*, A. Bensoussan and J. L. Lions, eds., Springer-Verlag, Berlin, 210–224.

Bertsekas, D. P., E. M. Gafni, and K. S. Vastola (1979). Validation of algorithms for optimal routing of flow in networks. In *Proceedings of the 1979 IEEE Conference on Decision and Control, San Diego, CA, January 10-12, 1979*, 220–227.

Bertsekas, D. P., E. M. Gafni, and R. G. Gallager (1984). Second derivative algorithms for minimum delay distributed routing in networks. *IEEE Transactions on Communications*, **COM-32**, 911–919.

Bertsekas, D. P. (1998) *Network Optimization - continuous and discrete models*, Athena Scientific, Belmont, MA.

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

Dafermos, S. C. (1968). *Traffic Assignment and Resource Allocation in Transportation Networks*, Ph.D. thesis, Johns Hopkins University, Baltimore, MD.

Dafermos, S. C., and F. T. Sparrow (1969). The traffic assignment problem for a general network. *Journal of Research of the National Bureau of Standards*, **73B**, 91–118.

Evans, S. P. (1976). Derivation and analysis of some models for combining trip distribution and asignment. *Transportation Research*, **10**, 37–57.

Florian, M., and D. Hearn (1995). Network equilibrium models and algorithms. In *Network Routing, Handbooks in OR & MS, Vol. 8*, M. O. Ball et al., eds., Elsevier Science, Oxford, UK, 485–549.

Florian, M., and H. Spiess (1983). Transport networks in practice. In *Proceedings of the Conference of the Operations Research Society of Italy, Napoli*, 29–52.

Frank, M., and P. Wolfe (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, **3**, 95-110.

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

Gallager, R. G. (1977). A minimum delay routing algorithm using distributed computation. *IEEE Transactions on Communications*, **COM-25**, 73–85.

Gibert, A. (1968). A method for the traffic assignment problem. *Report LBS-TNT-95*, Transportation Network Theory Unit, London Business School, London, UK.

Hagstrom J. N. (1997). Computing tolls and checking equilibrium for traffic flows. University of Illinois at Chicago, Department of Information and Decision Sciences, working paper.

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

Hearn, D. W. (1984). Practical and theoretical aspects of aggregation problems in transportation planning models. In *Transportation Planning Models*, M. Florian, ed., North-Holland, Amsterdam, 257-287.

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

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

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

Larsson, T., M. Patriksson, and C. Rydergren (1998). Application of simplicial decomposition with nonlinear column generation to nonlinear network flows. In *Licentiate Thesis*, Clas Rydergren, *Thesis No. 702*, Linköping Institute of Technology, Linköping, Sweden.

Larsson, T., J. Lundgren, M. Patriksson, and C. Rydergren (1999). Most likely traffic equilibrium route flows - analysis and computation. *Report LiTH-MAT-R-1999-05*, Department of Mathematics, Linköping Institute of Technology, Linköping, Sweden.

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

LeBlanc, L. J., E. K. Morlok, and W. P. Pierskalla (1975). An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research*, **9**, 309–318.

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

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

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

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

Rossi, T. F., S. McNeil and C. Hendrickson (1989). Entropy model for consistent impact fee assessment. *Journal of Urban Planning and Development/ASCE*, **115**, 51-63.

Wardrop J.G. (1952). Some theoretical aspects of road traffic research. In *Proceedings of the Institution of Civil Engineers*, Part II, **1**, 325-378.