Transportation Research Part B 44 (2010) 1022–1046

Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

Traffic assignment by paired alternative segments

Hillel Bar-Gera

Ben-Gurion University, P.O. Box 653, 84105 Beer-Sheva, Israel

ARTICLE INFO

Article history:
Received 10 June 2009
Received in revised form 25 November 2009
Accepted 25 November 2009

Keywords:
User-equilibrium
Traffic assignment
Quick-precision
Route flows
Proportionality

ABSTRACT

The static user-equilibrium (UE) traffic assignment model is widely used in practice. One main computational challenge in this model is to obtain sufficiently precise solutions suitable for scenario comparisons, as quickly as possible. An additional computational challenge stems from the need in practice to perform analyses based on route flows, which are not uniquely determined by the UE condition. Past research focused mainly on the first aspect. The purpose of this paper is to describe an algorithm that addresses both issues. The traffic assignment by paired alternative segments (TAPAS) algorithm, focuses on pairs of alternative segments as the key building block to the UE solution. A condition of proportionality, which is practically equivalent to entropy maximization, is used to choose one stable route flow solution. Numerical results for five publicly available networks, including two large-scale realistic networks, show that the algorithm can identify highly precise solutions that maintain proportionality in relatively short computation times.

1. Introduction

The purpose of traffic assignment is to predict how travelers choose routes over a road network. The main assumption, known as the user-equilibrium (UE) assumption, is that travelers choose routes to minimize their own travel times (or another related generalized cost measure) in view of congestion in the network, where congestion and travel times are a function of the flows that result from the choices of all travelers combined. The basic UE model, which further assumes separable additive deterministic costs, can be formulated mathematically as a convex-optimization problem (Beckmann et al., 1956). Practical applications involve many challenges, including the collection of field data, particularly about travel times; the preparation of model inputs; model calibration and validation; and interpretation and analysis of model outputs. The computational process of producing outputs from given inputs presents several additional challenges. Two principal ones are quick precision and reasonable route flows.

Precision has been a focus of research on the UE model for many years, with significant improvements in the last two decades. Larsson and Patriksson (1992) presented a route-based algorithm known as disaggregate simplicial decomposition (DSD). Jayakrishnan et al. (1994) presented a route-based algorithm with gradient projection. An origin-based assignment (OBA) algorithm that focuses on average costs of alternative approaches at merge nodes was presented by Bar-Gera (2002). Dial (2006) demonstrated quick-precision performance for his Algorithm B, that uses an origin-based data representation to avoid route enumeration and focuses on flow shifts from maximum cost route segments to minimum cost route segments. (The main design components of Algorithm B were documented earlier in Dial (1999).) A cutting plane method was proposed by Babonneau et al. (2006). Nie (2007, 2010) identified a mistake in the derivation of second order derivatives used in the OBA algorithm, and offered a solution that improves its computational efficiency. Florian et al. (2009) proposed a projected gradient route-based algorithm that considers all routes for a single origin–destination pair simultaneously. Gentile (2009) presented the LUCE algorithm, which uses an origin-based data structure. All of these algorithms can achieve the

E-mail address: bargera@bgu.ac.il

0191-2615/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.trb.2009.11.004

precision levels needed for scenario comparisons. Substantial improvements in computation times were demonstrated, especially in the last five years.

The basic UE model has a unique solution in terms of total link flows, but multiple solutions in terms of route flows. The current state-of-practice is to use a single route flow solution, chosen arbitrarily from all UE solutions. Instead of an arbitrary choice, conditions of route set consistency and proportionality are proposed here (Section 3). These conditions are practically equivalent to the condition of entropy maximization proposed by Rossi et al. (1989), which ensures stability (Lu and Nie, 2009).

Only few attempts have been made at computing the maximum entropy user-equilibrium (MEUE) solution. Janson (1993) presented a link-based computational procedure using successive stochastic user-equilibrium (SUE) approximations, providing fairly modest levels of precision. Related origin-based derivations, including a correction to Janson's derivation, were presented by Akamatsu (1997). Bell and Iida (1997) discussed a balancing method to address this problem. Larsson et al. (1998) suggested a dual post-process, including all routes that are within a predetermined threshold from minimal cost. This choice of routes may lead to undesirable results, as shown in Bar-Gera (2006), where an alternative computationally-efficient primal post-process is proposed. The main problem of the latter is that it requires extremely precise UE solutions, in some cases beyond the limitations of double precision arithmetic.

The main contribution of this paper is a new algorithm for the static deterministic user-equilibrium traffic assignment, with a formal proof of convergence, that achieves quick precision, route set consistency and proportionality (i.e., an approximation of entropy maximization). New practical measures are proposed for consistency and proportionality, to allow quantitative performance evaluation of the proposed algorithm in these respects. Results on precision performance by computation time are presented for two large scale and three additional networks. Direct comparisons with two previous methods, Frank-Wolfe (LeBlanc et al., 1975) and origin-based assignment (Bar-Gera, 2002), as well as indirect comparisons with three more recent methods, Dial (2006), Florian et al. (2009) and Gentile (2009), show that the new method is among the leading algorithms in terms of quick precision. (To clarify: indirect comparisons conducted on different computing platforms can only provide an order of magnitude assessment of relative performance. Direct comparison of advanced algorithms is highly sensitive to their implementation. When performance is within the same order of magnitude, the comparison may also be sensitive to the specific precision target, the proportion of origin–destination pairs with non-zero flow, and many other features of the specific test problem. It is not clear what would constitute a comprehensive direct comparison that may enable a proper ranking of leading algorithms, and an attempt to propose such ranking is thus avoided here.)

The basic idea of the proposed algorithm is shifting flow between pairs of alternative segments, for a set of stored pairs which is updated dynamically. The experience of the author suggests that in many cases simple implementation of this concept performs very well, but there are special cases that need to be addressed. This paper presents the key findings from the research, the difficulties that have been discovered, and the treatments identified for them.

The rest of the paper is organized as follows. The need for the proposed algorithm is elaborated in Section 2. Section 3 presents the conditions of consistency and proportionality, used to identify reasonable route flows. Section 4 provides a conceptual overview of the TAPAS algorithm, and explains the connection to negative cost cycle algorithms for network flow problems. A mathematical framework is presented in Section 5, including notation, optimization problems, review of previous derivations, and proposed performance measures. Critical issues related to algorithm convergence, including a formal proof, are given in Section 6. Section 7 focuses on how to achieve proportionality. Numerical results for 5 test networks are presented in Section 8, including direct and indirect comparisons of convergence performance to other methods. Conclusions and suggestions for future research are summarized in Section 9.

2. Motivation

The static deterministic user-equilibrium (UE) traffic assignment model has been studied for more than 50 years. Questions are occasionally raised about the need for additional research on this model, in the presence of many alternative models. Addressing these questions is the first goal of this section. The second goal is to discuss precision needs for scenario comparisons. The third goal is to elaborate on the need for reasonable route flows.

2.1. The usefulness of the user-equilibrium model

Among mathematical operations research models, traffic assignment is a remarkable practical success story. It is widely used today throughout the entire world, as part of the evaluation of almost every major project of investment in transportation infrastructure improvement, often as a central component within the travel forecasting process. The UE model simplifies reality considerably. One common criticism refers to the representation of traffic as continuous flows, while vehicle counts during a specific time interval are integer by their nature. Travel forecasts start with estimates of base year traffic, where an average over all (work) days may be more appropriate then a single count. In a future year forecast, probabilities can be associated with different integer counts on any particular route (or link). In the UE model these distributions are represented concisely by their expected values, as opposed to a single or few realizations depicted by integer flow models. So

while it may be common to consider integer flows as "more realistic," in fact continuous flows are better for concise representation of future forecasts. Additional advantages of continuous flow models are solution stability and formulation rigor.

Moreover, important critics of the UE model refer to other well studied phenomena, including dynamic changes in traffic conditions during the day such as queue accumulation and dissipation, numerous stochastic aspects of transportation systems, and many subtle local effects of traffic control devices and interactions between vehicles. Dynamic traffic assignment, introduced by Merchant and Nemhauser (1978), has evolved considerably as can be seen in the review by Peeta and Ziliaskopoulos (2001). Stochastic traffic assignment models, e.g., Daganzo and Sheffi (1977), usually focus on one stochastic aspect related to travel time perception/knowledge. Effects of traffic control devices and interactions between vehicles are addressed directly in microscopic simulation models, a very active area of research as is clearly evident from the 57 different simulation models reviewed by the project team of SMARTEST (2000).

There is no doubt about the value of studying and modeling aspects of the transportation system that cannot be captured by the basic UE model. However, in practice we see that the basic UE model is very widely used, and one may wonder why that is so. To answer this question we need to discuss the relationship between model realism and model usefulness. Clearly, these traits are closely related, since the main purpose of models is to represent reality. Yet realism is not the only consideration that makes models useful. Other considerations may be: proper sensitivity to policy decisions; ability to obtain inputs, as well as validation and calibration data; stability, repeatability and consistency, which are important particularly in comparisons between scenarios; computational efficiency; insights, understandability and accessibility; the confidence of the modeler in the results and the ability to convince others that the results provide a solid basis for making decisions; and more. In view of this multitude of considerations it seems that in modeling we cannot hope for "one size fits all." For any given system there are likely to be several different useful models, depending on the questions and analyses to be performed.

The question of model realism itself is not a trivial matter as well. The discussion should distinguish between predictive realism or accuracy and internal structure realism. The interest in accuracy may be at a very aggregate level, like the total vehicular distance and time traveled, or at a disaggregated level of a single link, a single origin–destination pair, or even a single route. Accuracy of flows and accuracy of travel times are also two separate issues. Accuracy of nominal predictions is quite different from accuracy in estimates of changes which is the typical goal of scenario analysis. The latter is particularly challenging since reality happens only once, so validation data can be available only for one scenario.

Since accuracy is difficult to demonstrate, arguments about realism often refer to the model's internal structure, assuming that more realistic internal structure guarantees a more realistic model, which is not necessarily so. For example, does the obvious fact that transportation systems are stochastic in many different ways imply that any model that contains a stochastic component is necessarily more realistic than a model that does not consider stochasticity at all? What if the incorporated stochastic component is not the most important stochastic component in reality? Are there any specific assumptions about the stochastic component and can we verify whether those are realistic or not? Given the challenges of obtaining sufficient data to estimate mean values in the real system properly, how much trust can we have in estimates of the magnitude of stochasticity? Is it possible that the magnitude of stochasticity is overestimated to the extent that it is in fact further away from reality than the zero stochasticity option? Similar questions can be asked about other improvements to internal structure realism.

The purpose of raising these questions is not to criticize any particular model, or to dispute the value of certain lines of research. The point is that internal structural realism is difficult to demonstrate; even when demonstrated it does not ensure better accuracy of nominal predictions, and on top of that better accuracy of nominal predictions does not necessarily imply better accuracy in scenario comparisons. Ultimately model usefulness is determined primarily by its ability to assist decision making processes, and as much as accuracy is important, it is only one of various considerations that determine model usefulness.

The basic static deterministic user-equilibrium model is widely used mainly because it has proven to be very useful. It is computationally efficient making it practical even for very large-scale networks. Data needs are relatively modest. It is simple to understand, allowing clear insights and making it relatively convincing. Last but not least, solutions are stable with respect to input data, which is an essential property for scenario comparisons. It is evident that alternative useful models exist, more are likely to appear, and their utilization will probably increase. Yet, in my humble opinion, the basic UE model will continue to be useful for many years, and the related computational challenges addressed in this paper will remain important.

2.2. User-equilibrium precision needs

The UE model is formulated as a convex optimization problem, and as such it is relatively well behaved. Still, the nonlinear nature and the large scale of practical UE models are known factors that make UE solutions a non-trivial challenge. Less often discussed are the particular challenges that arise from the predictive usage of the model, as opposed to the prescriptive usage in regular optimization problems. In regular prescriptive context of optimization problems the objective function reflects the true goal of the decision maker. Precision beyond input data accuracy is not needed, as the latter governs the merit of solution comparisons. (Clarification: accuracy is defined as the difference between a value in the model and the corresponding value in reality, where as precision is defined as the difference between the value in a given solution and the value in a perfect solution to a given mathematical problem.)

Precision needs in a predictive context are determined by the need to compare scenarios. As a convex optimization model, stability is ensured; see recent discussion by Lu (2008). Stability means that small changes to inputs lead to small changes in values of the exact solution, which is an essential property to allow scenario comparisons. To benefit from the theoretical stability, solutions must be sufficiently precise so that computed differences between scenarios will reflect differences between the exact solutions, rather than random artifacts that result from the differences between computed solutions and the exact solutions.

This idea is illustrated in Fig. 1, showing two performance measures of interest X and Y (e.g., total flow on two links) in several evaluations. In this figure there are two scenarios, A and B. A_r and B_r are the realities in both scenarios. The exact model solutions, A_e and B_e, are different from the realities, due to model inaccuracies. Since most model inaccuracies affect both scenarios in the same way, the difference between A_e and B_e is similar in direction and in magnitude to the difference between A_r and B_r. Precision errors randomly affect computed solutions in a different way for each scenario. If sufficiently precise solutions are used, as in A_p and B_p, scenario comparison is still valid. However, the comparison may become meaningless if the solutions are too approximated relative to the difference between the scenarios, as in A_a and B_a, even though the precision is still better than the accuracy.

Boyce et al. (2004) examined a practical case study from Philadelphia and concluded that the traffic assignment precision needed in the particular case is a relative gap of 1E-4 (see Eq. (8) there for the specific definition of relative gap). Slavin et al. (2009) examined another case study, using a model for the Washington DC metropolitan region, further emphasizing the need in high levels of precision, which can be quite challenging to achieve with traditional methods.

2.3. Choosing route flow solutions

In addition to precision, another major computational challenge is created by the non-uniqueness of route flows in the UE model. Again, in a prescriptive optimization this is not a critical concern, since all solutions with the same objective function value are equivalently good, and any arbitrary choice among optimal solutions would be satisfactory. In predictive models, however, arbitrary choices are problematic. Ideally one might hope to choose the route flow solution that is as close as possible to reality. Unfortunately, existing empirical data on route flows are too limited to determine whether any particular model reflects reality more accurately than others, which does not necessarily mean that the choice does not matter at all.

Section 3 presents conditions of route set consistency and proportionality for choosing route flow solutions. The proposed conditions are based on a simple behavioral assumption that many modelers are likely to consider as plausible. The effect of these conditions on the results is relatively easy to understand, making the results easier to interpret. Solutions satisfying these conditions are good approximations of the entropy maximizing solution, which is stable. Compared to arbitrary choice, the proposed conditions provide a more useful model, even if empirical evaluation of its realism is not available yet.

Complete disaggregation to route flows is rarely required in practice. On the other hand there are many applications where various levels of aggregation of route flows are needed, which are also not uniquely determined by the UE assumption. Typical applications include class-specific link flows in multi-class assignment; decomposition of total link flows by origin and destination, used for example, to determine the allocation of project investment funding among nearby communities; evaluation of average cost components, such as travel time and distance, from origin to destination (skims) in a generalized cost assignment; extraction of flows in a subarea, for example, as inputs to a microscopic simulation model; design of license plate surveys and model validation in view of their results; and more. Bar-Gera and Luzon (2007a,b) show that the set of all UE route flow solutions covers a very wide range, both at the route level as well as at various levels of

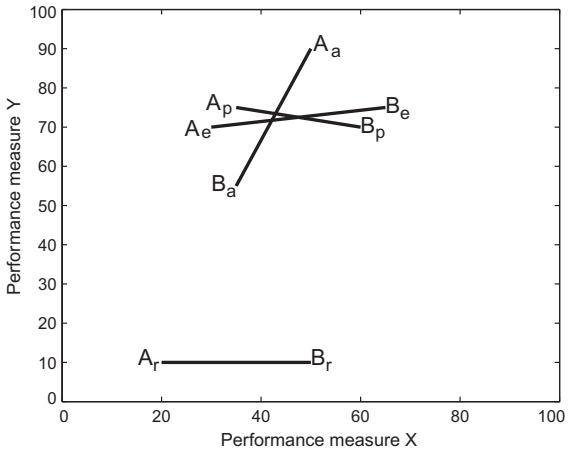

Fig. 1. Conceptual illustration of the effects of accuracy and precision on comparison between scenarios A and B in reality (r) and using exact, precise or approximate model results (e, p, and a, respectively). X and Y represent two performance measures of interest (e.g., total flow on two links).

aggregation. The practical implications of this issue for the use of existing commercial assignment software is a subject for on-going research.

3. Proportionality and route-set consistency

The main issue of non-uniqueness in route flows can be illustrated with a very simple example shown in Fig. 2. Suppose that the total link flows shown in the figure represent perfect UE solution, at which identical travel times are incurred on a pair of alternative segments, [2,3,5] and [2,4,5]. The key question is how many travelers from each origin use each segment. Three different solutions for the flow on each of the four routes in this network are given in Table 1. All three solutions correspond to the same total link flows exactly. If one wants to know how many travelers on link [2,3] come from which origin, or if one extracts subarea data to study the weaving pattern on link [1,2], the results of each route flow solution will lead to different answers. A consistent choice among the infinite possibilities of route flow solutions for a single UE model requires an additional behavioral assumption. One plausible assumption is *proportionality*, namely that the proportion of travelers on each of the two alternative segments should be the same regardless of their origin or their destination. Indeed, this is the case for the solution marked by $h*$ in Table 1, as a ratio of 1:3 is present for routes R1 and R2 (25:75) as well as for routes R3 and R4 (15:45).

Proportionality is fairly natural in a single class model, but it can also be used in multi-class models. If travelers from origin B belong to two different classes, for example, 40 passenger cars and 20 trucks (in passenger car equivalents), then there are two options. One option is that only one class can use both alternatives. For example, link [2,3] may be prohibited for truck use. In that case all trucks will use route R4 only, there is no need for any additional assumption on the way different classes use the two segments, and the assumption of proportionality will apply only to passenger cars from origins A and B. A related case is when the generalized cost of travel is equal on both segments for one class, but different for the other class. Since under the UE assumption travelers can use only the least cost route, this is in fact the same situation as the case of prohibited links, as far as determining route flows is concerned. The other option is that both alternative segments have equal costs for both classes of travelers. In that situation the assumption of proportionality can be applied to class flows as well, enabling to determine that on route R3 there will be 10 passenger cars and 5 trucks, and on route R4 there will be 30 passenger cars and 15 truck.

In general networks there may be many pairs of alternative segments with equal cost. For example, Fig. 3 shows a network with three such pairs, so there are eight different routes. The assumption of proportionality in this case simply means that the proportion of each route is the product of the proportions of the segments it consists of, so for example, the proportion of the route that uses the top alternative in each pair, namely route [1,2,3,5,6,7,9,10,11,13,14], is $(150/200) \times (40/200) \times (80/200) = 0.75 \times 0.2 \times 0.4 = 0.06$.

It is clear that under the assumption of proportionality all eight routes will be used, and "no route will be left behind." This property of the set of used routes is referred to as *consistency*. Of course this does not mean that all routes should be used, since under the UE assumption only least cost routes are used. So the condition of consistency is that "no route should remain unused, unless there is a good reason for it." A good reason is that using the route causes UE violation. In conclusion, a set of routes in a solution to UE problem is defined formally as consistent if it contains any route that can be used without changing total link flows. A consistency violation in a set of routes is considered "basic" if it can be demonstrated by consideration of one pair of alternative segments. Examples of non-basic consistency violations are discussed in Bar-Gera (2006).

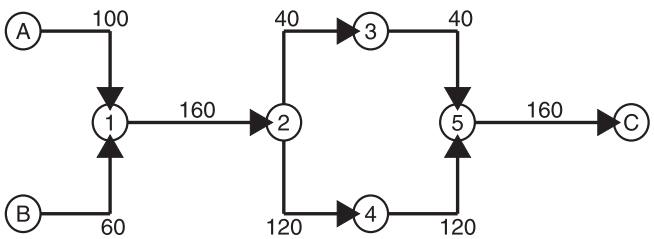

Fig. 2. Example network for non-unique route flow issue. Values along link are total flow in perfect equilibrium.

Table 1
Possible route flow solutions that match total link flows in Fig. 2.

O–D	Route	Description	$h*$	h_1	h_2
A–C	R1	A-1-2-3-5-C	25	40	0
A–C	R2	A-1-2-4-5-C	75	60	100
B–C	R3	B-1-2-3-5-C	15	0	40
B–C	R4	B-1-2-4-5-C	45	60	20

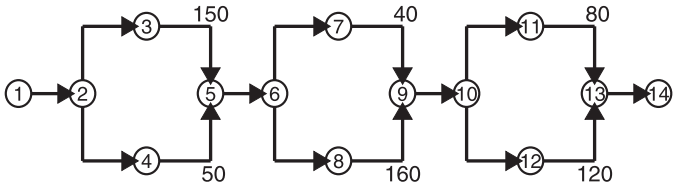

Fig. 3. Simple network to demonstrate proportionality and consistency.

The concept of proportionality can be extended to a formal definition for general networks as follows. A route flow solution maintains the condition of proportionality if for every pair of alternative segments, s_1 and s_2, the proportion of the flow on s_1 has a constant value, denoted by $\rho(s_1, s_2)$. This means that if r_p is a route segment from origin p to the diverge node of s_1 and s_2; r_q is a route segment from the merge node of s_1 and s_2 to destination q; $r_1 = r_p + s_1 + r_q$ is a route from p to q using segment s_1; $r_2 = r_p + s_2 + r_q$ is a similar route from p to q only that it uses s_2 instead of s_1; and h_{r_1}, h_{r_2} are the flows on the two routes, respectively; then the proportion of the flow on r_1 is the same as the proportion for any other pair of routes that differ by the same two segments, $h_{r_1}/(h_{r_1} + h_{r_2}) = \rho(s_1, s_2)$.

The assumption of proportionality was first introduced in Bar-Gera and Boyce (1999) as an interpretation for optimality conditions of route flow entropy maximization. Theoretically one can construct examples where route flows are not uniquely determined even under the condition of proportionality (Bar-Gera, 2006), while the entropy maximizing route flow solution is unique (Rossi et al., 1989). As entropy maximizing implies proportionality, it is a stricter condition. Bar-Gera (2006) showed that in realistic networks the difference between the two conditions is relatively small, for example, in the case of the Chicago Regional model there are 90,723,930 degrees of freedom (algebraic dimensions in the space of route flows) that are not determined by the UE model, and only 91 of them remain undetermined once the condition of proportionality is adopted. The practical significance of the difference between proportionality and entropy maximization is yet to be explored.

The goal of the research reported here, then, is to develop a computationally efficient algorithm for the basic UE model that will be capable of finding sufficiently precise solutions in terms of total link flows, and at the same time identifying a reasonably consistent set of routes and route flows that satisfy the condition of proportionality.

4. Algorithm background and overview

Studies of network flow models (e.g., Ahuja et al., 1993), focusing primarily on single commodity fixed link costs (linear objective), consider negative cost cycles as a key component. In these studies, cycles (or generalized cycles) refer to a sequence of nodes and links, such that the first node is the same as the last, and every two consecutive nodes are connected by a directed link either in the same direction as the cycle (forward link) or in the opposite direction (backward link). An example of such a cycle is shown in Fig. 4. Assuming that the direction of the cycle is clockwise, link [13,14] is a forward link and link [3,4] is a backward link. "Sending" x units of flow around the cycle means adding x units of flow to every

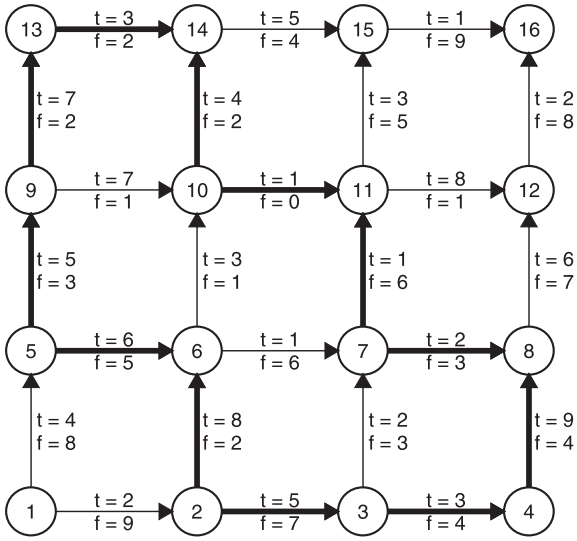

Fig. 4. Example of a generalized cycle, marked by thick lines. (f shows link flow and t shows link cost.)

forward link and subtracting x units of flow from every backward link. This operation can be performed only if the flow on all backward links is not less than x, and particularly if the flow on all backward links is strictly positive. In the example shown in Fig. 4, we cannot send flow around the cycle in the counterclockwise direction, because the flow on link [10, 11] is 0. When sending flow around generalized cycles, conservation of flow and therefore feasibility is maintained. The change in total cost as a result of sending one unit of flow around the cycle, referred to as the cycle cost, is the sum of forward link costs minus the sum of backward link costs. The cycle in Fig. 4 starting from node 5 clockwise has a cost of $5 + 7 + 3 - 4 + 1 - 1 + 2 - 9 - 3 - 5 + 8 - 6 = -2$. When the cost of a cycle is negative, as in this case, sending flow around it reduces total solution cost. Identifying negative cost cycles and sending flow around them are the basic steps in many network flow algorithms (e.g., Busaker and Gowen, 1961; Dantzig, 1963; Busaker and Saaty, 1965; Rockafellar, 1984).

Particularly relevant to this work is the multi-stage negative cost cycles algorithm proposed by Schneur and Orlin (1998) for the multi-commodity network flow problem with capacitated fixed cost links. A non-linear penalty component is added to the objective function to address capacity constraints, where the penalty weight is increased from stage to stage. In parallel, at every stage a fixed shift value is chosen, which is reduced from stage to stage. Fixed shifts simplify computations needed to identify negative cost cycles, and thus contribute to computational efficiency. In her PhD thesis, Schneur (1991) reports that a line search may lead to more precise solutions than fixed shifts, but argues that the additional precision is not as important as computational efficiency. Perhaps their perspective is due to the context of the problem that they have considered. An interesting feature of their algorithm is the storage of cycles identified previously for additional examination, which they find highly beneficial.

The algorithm proposed here considers only two types of negative cost cycles. One is a completely backward cycle, where we simply reduce the flow on all the links until one of them becomes zero. The more important type of generalized cycle on which we focus primarily is a Pair of Alternative Segments (PAS). The direction of the generalized cycle is chosen to match the direction of the lower cost segment. Sending flow around the generalized cycle associated with a PAS is simply equivalent to shifting flow from the higher cost segment to the lower cost segment, which is similar to the seminal idea of Dafermos and Sparrow (1969) of shifting flows from higher cost routes to lower cost routes. Since each segment may be incorporated in many routes, a single shift can take into consideration the total amount of flow available for shifting in all the routes that contain the higher cost segment.

Dial (2006) utilized this fact for routes from the same origin as follows. For every origin an a-cyclic set of links is chosen, referred to as a "bush." For every merge node in a bush two segments are traced backwards, one along the minimum cost tree and one along the maximum cost tree (within the bush), until they intersect. Origin-based flows are shifted between these distinct segments to equilibrate costs.

The basic element of the algorithm proposed here, of shifting flows between distinct segments, is therefore closely related to Dial's approach. The main differences from Dial's algorithm are: the procedure to identify PASs which is based on Bar-Gera (2006); there are no restrictions to a specific bush; PASs are stored from iteration to iteration; and all relevant origins for each PAS are considered.

Intuition about the potential of PASs can be gained from the example grid network shown in Fig. 5. In this network, from origin A to destination F there are 252 routes, two of them (r and r') are shown in the figure in solid and dashed thick lines. The two routes are different in the middle segment from diverge node 8 to merge node 29. We can view the PAS $\{[8,14,20,21,22,28,29],[8,9,10,16,17,23,29]\}$ as the distinguishing component between the two routes.

Overall there are $252 \times 251/2 = 31{,}626$ pairs of routes, and each pair of routes has a distinguishing component PAS. Not all resulting PASs are different, but many of them are. Fortunately it is not necessary to consider all possible PASs in the network. It is sufficient to choose a basic subset of PASs. In the case of the grid network in Fig. 5 we can choose the set of 25

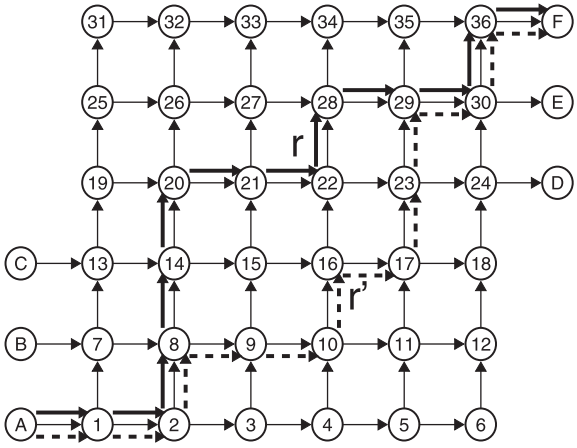

Fig. 5. Example of a pair of alternative segments in a grid network.

Table 2
Sequence of routes using basic PAS shifts.

Route ID	Description	Distinguishing PAS (to next route)
$r = r_0$	A,1,2,8,14,20,21,22,28,29,30,36,F	{[14,20,21],[14,15,21]}
r_1	A,1,2,8,14,15,21,22,28,29,30,36,F	{[15,21,22],[15,16,22]}
r_2	A,1,2,8,14,15,16,22,28,29,30,36,F	{[22,28,29],[22,23,29]}
r_3	A,1,2,8,14,15,16,22,23,29,30,36,F	{[8,14,15],[8,9,15]}
r_4	A,1,2,8,9,15,16,22,23,29,30,36,F	{[16,22,23],[16,17,23]}
r_5	A,1,2,8,9,15,16,17,23,29,30,36,F	{[9,15,16],[9,10,16]}
$r' = r_6$	A,1,2,8,9,10,16,17,23,29,30,36,F	

PASs around each block of the grid, such as {[5,6,12],[5,11,12]}. These 25 PASs are sufficient because a shift of flow between any pair of routes can be represented as a sequence of shifts of flow between pairs of routes, such that the intermediate distinguishing components are PASs in the chosen set. For example, a shift of flow from route r to r' in Fig. 5 can be represented as a sequence of shifts between the routes in Table 2, from $r = r_0$ to r_1; from r_1 to r_2; ...; and from r_5 to $r_6 = r'$. The distinguishing components shown in the table are all members of the set of 25 "around the block" PASs.

If travel times are equal within each of these 25 PASs, then travel times are equal between all routes, and UE is ensured. The UE problem can therefore be viewed as a problem with 25 conditions and 25 solution variables, compared with 252 variables and 251 conditions if all routes are considered. If there are three origins, A, B, C and three destinations D, E, F, then there are 720 routes, but the same set of 25 PASs is sufficient, as certain PASs are relevant to more than one O–D pair. For example, the PAS {[15,16,22], [15,21,22]} is relevant to all 9 O–D pairs. Note that if proportionality is satisfied for the chosen 25 basic PASs, then it holds for all other PASs as well, thus further emphasizing the advantages of consideration of PASs.

The traffic assignment by paired alternative segments (TAPAS) algorithm proposed here is based on an origin-based solution representation, that is, storing an array of the flow from every origin through each link, aggregated over all destinations. The origin-based representation has an immediate route flow interpretation, to be presented in Section 5 by Eq. (11), that ensures within-origin consistency and proportionality. In addition to origin-based link flows, the algorithm maintains a set of active PASs and a list of relevant origins to each PAS. The key operations of the algorithm are: shifting flows between segments of an existing PAS in order to equalize the costs; PAS set management including identification of new PASs, elimination of inactive PASs, and updating the lists of relevant origins for PASs; and redistribution of PAS flows between origins according to the condition of proportionality. In addition to the basic structure of the algorithm, special situations and their treatments are discussed in Section 6.

Flow shifts are in fact the simplest component of the algorithm. First we determine which segment has higher cost. For example, given the link costs in Table 3 segment 1 has a higher cost. Next we consider the list of relevant origins. In this example there are four relevant origins in the list, namely: 11, 73, 102, and 137. For each of these origins we determine the maximum feasible shift from the higher cost segment as the minimum of all origin-based link flows. For example, the maximum flow from origin 73 that can be shifted from segment 1 to segment 2 is the origin-based flow on link 45 which is 4. Adding the feasible shifts from all relevant origins determines the total feasible shift, in this case 16. The maximum possible shift from segment 2 to segment 1 in this example is 11, but this information is not needed for any subsequent computations.

If performing the maximum feasible shift still leaves the same segment as the higher cost segment, then that shift is implemented. Otherwise, a line search is used to determine what amount of shift leads to equal segment costs. Note that the line search deals only with changes to total link flows, and only to the relatively small number of links involved in the PAS. As such it can be computed fairly quickly. Once the total shift is determined by the line search, it is divided between the origins in proportion to the feasible shift for each origin. In the example above if the total shift is 4 then the shifts for origins 11, 73, 102, and 137 are 0.5, 1, 0.5, and 2, respectively.

Table 3
Example for flow shift operation

Link	Cost	Origin 11	Origin 73	Origin 102	Origin 137	Total flow
Segment 1	20					
Link 23	5	4	9	4	15	32
Link 45	3	2	4	7	32	45
Link 53	12	7	6	2	8	23
Minimum		2	4	2	8	16
Segment 2	16					
Link 11	5	7	3	7	8	25
Link 103	2	5	2	13	6	26
Link 57	3	5	4	0	9	18
Link 88	6	12	1	23	5	41
Minimum		5	1	0	5	11

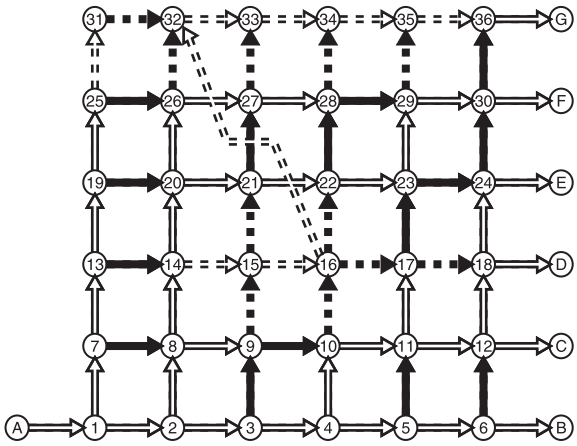

Fig. 6. Identifying a new pair of alternative segments. Used links (by origin A) are solid; unused links are dashed; links along minimum cost routes are hollow.

Managing the list of active PASs is essential to the success of the algorithm, and is partly science and partly art. The following process is performed for every origin p separately, in cyclic order. First the subnetwork of links used by the origin is inspected for directed cycles. If cyclic flows are found, they are removed immediately. Next the tree of least cost routes is constructed. If link a is used by origin p, that is $f_{pa} > 0$, but a is not part of the tree of least cost routes, a PAS is needed.

For every such link the alternative approach in the minimum cost tree $a'(a'_h = a_h)$ is denoted, where a_t and a_h are the link tail and head, respectively. The goal is to ensure the possibility of flow shifts between these two merging links, a and a'. The first step in the inspection is to check if there is an active PAS whose segments end with the merging links. If so, we make sure that the current origin p is included in the list of relevant origins for that PAS. Special cases where an active PAS exist, the origin is listed as relevant, but the PAS appears to be ineffective are discussed in Section 6, as well as some potential treatments. If there is no existing PAS, a new PAS is constructed in which segment 1 consists of non-zero origin-based flow links and segment 2 is part of the least cost tree. In general PASs with shorter segments are preferred, since they are likely to be relevant to more origins, and flow shift computations are faster. A breadth first search over used links from a_t backwards towards the origin is used until the first node on the least cost route to a_h is encountered. That node is taken as a diverge node for the PAS, and the two segments are traced.

This procedure is illustrated using the example in Fig. 6. Used links by travelers from origin A are marked by solid thick lines, while other links are dashed. The least cost tree is marked by hollow links (solid or dashed). In many cases finding a PAS is a relatively simple task. For example, link $a = [6, 12]$ is used but it is not part of any minimum cost route. The predecessor of node 12 in the least cost tree is $a' = [11, 12]$. The conditions when choosing a PAS are that one segment will consist of used links and will end with link a; the other segment will be part of the least cost tree and will end with link a'; and the only common nodes for the two segments are the first (diverge) node and the last (merge) node. Under these conditions, there is only one possibility to choose a PAS for links $a = [6, 12]$ and $a' = [11, 12]$, namely the used segment is $s_1 = [4, 5, 6, 12]$ and the least cost segment is $s_2 = [4, 10, 11, 12]$. The breadth first search procedure (as many other possible search procedures) identifies this PAS by starting from node 6 and scanning nodes 5 and 4.

Another common situation which is slightly more complicated occurs if link $a = [23, 24]$ is chosen for inspection. In this case possible diverge nodes for legitimate PASs are 1, 2 and 11. Breadth first search starts from node 23 and the order by which nodes are scanned is: 22, 17, 21, and 11. The search stops at node 11 and the resulting PAS includes used segment $s_1 = [11, 17, 23, 24]$ and least cost segment $s_2 = [11, 12, 18, 24]$. This example shows the key advantage of the breadth first search procedure, which usually leads to "short" or "local" PASs.

Occasionally the situation can be more complicated. For example, if the link chosen for inspection is $a = [30, 36]$, the construction of a PAS is not trivial. The alternative least cost approach is $a' = [35, 36]$. The least cost route to $a_h = 36$ is $[A, 1, 2, 8, 14, 15, 16, 32, 33, 34, 35, 36]$. A breadth first search from $a_t = 30$ examines nodes in the following order: 29, 24, 28, 23, 18, 27, 22, 17, 12, 26, 21, 11, 6, 25, 20, 10, 5, 19, 14. The search stops at node 14, since it is part of the least cost route to $a_h = 36$. The PAS diverge node is therefore 14, the used segment is $s_1 = [14, 20, 26, 27, 28, 29, 30, 36]$ and the least cost segment $s_2 = [14, 15, 16, 32, 33, 34, 35, 36]$. In this case, in fact, there are several possible choices for used segment s_1 from diverge node 14 to merge node 36. The current algorithm chooses arbitrarily the segment that is implied from the backward search. In complicated situations like this the question of PAS efficiency may arise, an issue that is discussed in further detail in Section 6.

At the end of every iteration all active PASs are examined for elimination. If all the flow was shifted from one segment to the other and there has been no shifts for two consecutive iterations the PAS is eliminated. Redistribution of flows between origins by the condition of proportionality is a relatively complicated element of the algorithm. The general scheme is to

```
Find initial solution using all or nothing assignment
Repeat iteratively:
  For every origin
    Remove all cyclic flows
    Find tree of least cost routes
    For every link used by the origin which is not part of the tree
      If there is an existing effective PAS
        Make sure the origin is listed as relevant
      Else
        Construct a new PAS
    Choose a random subset of active PASs
      Shift flow within each chosen PAS

  For every active PAS
    Check if it should be eliminated
    Perform flow shift to equilibrate costs
    Redistribute flows between origins by the proportionality condition


Final proportionality iterations:
  For every active PAS
    Redistribute flows between origins by the proportionality condition
```

Fig. 7. TAPAS algorithm – general structure.

make an adjustment to each PAS once every iteration. Further details about this procedure are given in Section 7. Balancing the three components of the algorithm is a delicate task that is based mainly on experience. The limited experience obtained so far suggest that flow shifts for active PASs should be performed more often than searches for new PASs. Therefore, in the current implementation, every iteration consists of one search for new PASs for every origin, and several flow shifts for every active PAS. The number of PAS shifts per iteration is a parameter that can be determined by the user. In the results reported here 20 shifts per iteration are used. The overall structure of the algorithm is described in Fig. 7.

5. Formulations and performance measures

Consider a transportation network consisting of a set of nodes N and a set of links A. A (simple) route segment is a sequence of (distinct) nodes $[v_1,\ldots,v_k]$ such that $[v_l, v_{l+1}] \in A \quad \forall 1 \leqslant l \leqslant k-1$. 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.) The same notation is used for a single-node route segment $[v]$, which is the empty route segment at v; i.e., the route segment that starts from v, ends at v, and does not contain any links. The first node of route segment r is considered its *tail* and denoted by r_t, and the last node is considered the route's *head* denoted by r_h. In particular by definition $a \equiv [a_t, a_h]$ for every link $a \in A$ and $a_t, a_h \in N$. The set of all route segments from node i to node j is denoted by R_{ij}. If route segment $r = [i = v_1,\ldots,v_n = j] \in R_{ij}$ is followed by route segment $s = [j = u_1,\ldots,u_m = k] \in R_{jk}$ then the combination (concatenation) of the two segments is denoted by $(r+s) = [i = v_1,\ldots v_{n-1}, v_n = j = u_1, u_2 \ldots, u_m = k]$. The statement $s \subseteq r$ means that route segment s is part of route segment r.

The set of possible origins is denoted by $N_o \subseteq N$, and the set of possible destinations for each origin $p \in N_o$ is denoted by $N_d(p) \subseteq N$. The set of all routes that connect an origin to a destination is denoted by $\mathbf{R} = \bigcup_{p\in N_o}\bigcup_{q\in N_d(p)} R_{pq}$. The flow of travelers (also called demand) in units of vehicles per hour (vph) from each origin $p \in N_o$ to every destination $q \in N_d(p)$ is denoted by d_{pq}; $\mathbf{d}$ denotes the array of O–D flows. The flow along route $r \in R_{pq}$ from origin p to destination q is denoted by h_r, and $\mathbf{h}$ denotes the vector of route flows. Aggregating route flows through a link over all destinations results in origin-based link flows $f_{pa}(\mathbf{h}) = \sum_{q\in N_d(p)}\sum_{r\in R_{pq}:\ r\supseteq a} h_r$. The $|A|$ by $|N_o|$ array of origin-based link flows is denoted by $\mathbf{f}$. Similarly, for a general segment s, the origin-based segment flow is $g_{ps}(\mathbf{h}) = \sum_{q\in N_d(p)}\sum_{r\in R_{pq}:\ r\supseteq s} h_r$, and particularly for any node v the origin-based node flow is denoted by $g_{p[v]}$. Aggregating origin-based link flows over all origins results in total link flows $f_{\bullet a}(\mathbf{h}) = \sum_{p\in N_o} f_{pa}(\mathbf{h})$. The vector of total link flows is denoted by $\mathbf{f}_\bullet$. Link costs $t_a(f_{\bullet a})$ are separable, strictly positive, monotonically increasing and uniformly Lipschitz continuous functions of total link flows. Uniform Lipschitz continuity means that there exist a global constant Λ (the modulus) such that $|t_a(f_{\bullet a}+\delta) - t_a(f_{\bullet a})| \leqslant \Lambda\cdot|\delta|$. Since the flow on each link is bounded by the total flow on all O–Ds, any finite continuous piecewise differentiable function has bounded derivatives, and thus satisfies uniform Lipschitz continuity. In that sense it is a relatively mild condition. The UE traffic assignment problem (TAP) can be formulated mathematically as

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

See Patriksson (1994) for further details. Note: in some cases it is convenient to represent a network with zero cost links. Usually these are "permanent" zero cost links, regardless of the flow. Zero cost links might become a concern if they form a cycle; however, nodes along a permanent zero cost cycle may in fact be substituted by a single node. Therefore, we use here the convenient assumption that all link costs are strictly positive (non-zero).

To measure convergence of TAP solutions define route excess cost as, $ec_r = c_r - C^*_{pq}$ for every route $r \in R_{pq}$, where $C^*_{pq} = \min\{c_r : r \in R_{pq}\}$ is the minimum O–D costs. The total excess cost (also referred to as "the gap" or the "absolute gap") is defined as

$$TEC = \sum_{p\in N_o} \sum_{q\in N_d(p)} \sum_{r\in R_{pq}} h_r \cdot ec_r \tag{2}$$

TEC is often normalized by dividing it into various denominators. The main measure presented in this paper is the Average Excess Cost (AEC), which is the TEC divided by the total O–D flow (interzonal plus intrazonal), $TOD = \sum_{p\in N_o} \sum_{q\in N_d(p)} d_{pq}$, that is

$$AEC = TEC/TOD \tag{3}$$

Given the unique vector of UE total link flows $\mathbf{f}^*_\bullet$ we are particularly interested in the maximum entropy user-equilibrium (MEUE) route flow vector, i.e., the optimal solution of

$$\begin{aligned} [\text{MEUE}] \quad \max \quad & \mathbb{E}(\mathbf{h}) = -\sum_{p\in N_o} \sum_{q\in N_d(p)} \sum_{r\in R_{pq}} h_r \cdot \log(h_r/d_{pq}) \\ \text{s.t.} \quad & \sum_{r\in R_{pq}} h_r = d_{pq} \quad \forall p \in N_o; \forall q \in N_d(p) \\ & \sum_{r\in \mathbf{R}:\ r\supseteq a} h_r = f^*_{\bullet a} \quad \forall a \in A \\ & \mathbf{h} \geqslant 0 \end{aligned} \tag{4}$$

Lu and Nie (2009) showed that the unique optimal solution of this problem is stable with respect to the constraints on total O–D flows and total link flows; since optimal UE total link flows are stable with respect to O–D flows and link cost function parameters, the solution to (1) and (4) is stable. We refer to any route that may have a non-zero flow in at least one perfect UE solution as a "UE route," and denote the set of all such routes by $\mathbf{R}^*$. The set of routes considered as UE routes in a particular solution is denoted by R^0. As shown in Bar-Gera (2006), assuming that the set of considered routes is given we can write the Lagrangian as follows:

$$\begin{aligned} \max \quad & \mathbb{L}(\mathbf{h}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = -\sum_{p\in N_o} \sum_{q\in N_d(p)} \sum_{r\in R^0_{pq}} h_r \cdot \log(h_r/d_{pq}) + \sum_{p\in N_o} \sum_{q\in N_d(p)} \gamma_{pq} \cdot \left[\sum_{r\in R^0_{pq}} h_r - d_{pq}\right] \\ & \quad + \sum_{a\in A} \beta_a \cdot \left[\sum_{r\in \mathbf{R}^0:\ r\supseteq a} h_r - f^*_{\bullet a}\right] \\ \text{s.t.} \quad & \mathbf{h} \geqslant 0 \end{aligned} \tag{5}$$

The optimality conditions show that:

$$\begin{aligned} & h_r = d_{pq} \cdot \exp\left(-1 + \gamma_{pq} + \sum_{a\subseteq r} \beta_a\right) \quad \forall r \in R^0_{pq}; p \in N_o; q \in N_d(p) \\ \Rightarrow \quad & d_{pq} = \sum_{r\in R^0_{pq}} h_r = d_{pq} \cdot \exp(-1 + \gamma_{pq}) \cdot \sum_{r\in R^0_{pq}} \exp\left(\sum_{a\subseteq r} \beta_a\right) \quad \forall p \in N_o; q \in N_d(p) \\ \Rightarrow \quad & \gamma_{pq} = 1 - \ln\left(\sum_{r\in R^0_{pq}} \exp\left(\sum_{a\subseteq r} \beta_a\right)\right) \quad \forall p \in N_o; q \in N_d(p) \\ \Rightarrow \quad & h_r(\boldsymbol{\beta}) = d_{pq} \cdot \frac{\exp\left(\sum_{a\subseteq r} \beta_a\right)}{\sum_{r'\in R^0_{pq}} \exp\left(\sum_{a\subseteq r'} \beta_a\right)} \quad \forall r \in R^0_{pq}; p \in N_o; q \in N_d(p) \end{aligned} \tag{6}$$

The condition of proportionality holds for the MEUE solution since for every pair of UE routes $r_1, r_2 \in R^0_{pq}$ with a distinguishing PAS $s_1, s_2 \in R_{ij}$, meaning that $r_{1,2} = r + s_{1,2} + r'$ where $r \in R_{pi}$ and $r' \in R_{jq}$, the proportion of flow on r_1 according to (6) is

$$h_{r_1}/(h_{r_1}+h_{r_2}) = [1 + h_{r_2}/h_{r_1}]^{-1} = \left[1 + \exp\left(\sum_{a\subseteq s_2}\beta_a - \sum_{a\subseteq s_1}\beta_a\right)\right]^{-1} = \rho(s_1, s_2) \tag{7}$$

which depends only on the PAS and not on the routes.

The condition of proportionality might be violated if one route in such a pair is overlooked. In some cases it is possible to identify mistakes of this type. The basic case is when $s_1,\ s_2 \in S^0$, where $S^0 = \{s : \exists r \in R^0,\ s \subseteq r\}$ is the set of all segments of UE routes, and there is a pair of routes $r_{1,2} = r + s_{1,2} + r' \in R_{pq}$ such that only one of them is considered as a UE route, i.e., $r_1 \in R^0;\ r_2 \notin R^0$. To see that this must be a mistake notice that by the definition of S^0 there exists a UE route $\hat{r}_2 \in R^0$ such that $s_2 \subseteq \hat{r}_2$; both r_1 and $\hat{r}_2$ have non-zero flow in the optimal solution to (6); shifting a sufficiently small amount of flow, $\delta < \min(h_{r_1}, h_{\hat{r}_2})$, from r_1 to r_2 and from $\hat{r}_2$ to $\hat{r}_1$ demonstrates that route r_2 can have a non-zero flow in a solution with exactly the same total link flows. Therefore, excluding r_2 from R^0 is inconsistent. Other cases of mistakenly excluded routes can be identified, but they are relatively rare (Bar-Gera, 2006).

In general a route is considered to be mistakenly excluded if it is possible to assign non-zero flow to it by performing some flow "switch" while maintaining exactly the same total flow on all links. A set of routes is considered as consistent if "no route is left behind," i.e., if there are no mistakenly excluded routes. The perfect set of UE routes, R^*, is always consistent. Ensuring that the set of considered routes R^0 is consistent does not necessarily mean that it is identical to R^*, but it is clearly a good sign, and a natural condition when attempting to satisfy the condition of proportionality and/or entropy maximization.

In the context of a single origin, consistency and proportionality are relatively easy to establish. For a given origin p and a-cyclic origin-based link flows $\mathbf{f}_p$ the set of all routes that contain only used links ($f_{pa} > 0$) is perfectly consistent. Suppose that $\mathbf{h}$ is entropy maximizing and that $\boldsymbol{\beta}$ is the corresponding dual solution in (6). Define the node-to-node factor $y_{ij} = \sum_{s\in S^0_{ij}} \exp(\sum_{a\subseteq s}\beta_a)$. We can compute origin-based node flow, link flow and approach proportion (α_{pa}) as follows:

$$g_{p[v]}(\mathbf{h}) = y_{pv} \cdot \sum_{q\in N_d(p)} y_{vq} \cdot d_{pq} \cdot \exp(-1+\gamma_{pq}) \tag{8}$$

$$f_{pa}(\mathbf{h}) = y_{pa_t} \cdot \exp(\beta_a) \cdot \sum_{q\in N_d(p)} y_{a_h q} \cdot d_{pq} \cdot \exp(-1+\gamma_{pq}) \tag{9}$$

$$\alpha_{pa}(\mathbf{h}) = f_{pa}(\mathbf{h})/g_{p[a_h]}(\mathbf{h}) = \exp(\beta_a) \cdot y_{pa_t}/y_{pa_h} \tag{10}$$

Note that $y_{pp} = 1$ and that for every destination q, $y_{pq} = \exp(1-\gamma_{pq})$, hence for any route $r \in R^0_{pq}$

$$\prod_{a\subseteq r}\alpha_{pa} = \prod_{a\subseteq r}[\exp(\beta_a)\cdot y_{pa_t}/y_{pa_h}] = [y_{pp}/y_{pq}]\cdot\prod_{a\subseteq r}\exp(\beta_a) = \exp\left(-1+\gamma_{pq}+\sum_{a\subseteq r}\beta_a\right)$$

In combination with (6) we find that

$$h_r = d_{pq}\cdot\exp\left(-1+\gamma_{pq}+\sum_{a\subseteq r}\beta_a\right) = d_{pq}\cdot\prod_{a\subseteq r}\alpha_{pa} \tag{11}$$

Since origin-based approach proportions can be computed directly from origin-based link flows, (11) provides a direct and immediate route flow interpretation to any a-cyclic origin-based solution. Detailed discussions of this route flow interpretation can be found in Bar-Gera (2002) and Nie (2007), including alternative derivations and demonstrations of its validity. With this interpretation (Bar-Gera, 2002; (13)) shows that origin-based segment flows can be conveniently represented as:

$$g_{ps}(\mathbf{f}) = \sum_{q\in N_d(p)}\ \sum_{r\in R_{pq}:\ r\supseteq s} h_r(\mathbf{f}) = g_{ps_h}(\mathbf{f})\cdot\prod_{a\subseteq s}\alpha_{pa} \tag{12}$$

As a result, the origin-specific proportion on any PAS $\{s_1,\ s_2\}$ can be determined by:

$$g_{ps_1}/(g_{ps_1}+g_{ps_2}) = [1+g_{ps_2}/g_{ps_1}]^{-1} = \left[1+\frac{\prod_{a\in s_2}\alpha_{pa}}{\prod_{a\in s_1}\alpha_{pa}}\right]^{-1} = \rho_p(s_1,s_2) \tag{13}$$

and we can immediately verify within origin proportionality, since for any pair of UE routes $r_1, r_2 \in R^0_{pq}$ with a distinguishing PAS $s_1,\ s_2 \in R_{ij}$ by (11) the proportion of flow on r_1 is

$$h_{r_1}/(h_{r_1}+h_{r_2}) = [1+h_{r_2}/h_{r_1}]^{-1} = \left[1+\frac{\prod_{a\in s_2}\alpha_{pa}}{\prod_{a\in s_1}\alpha_{pa}}\right]^{-1} = \rho_p(s_1,s_2) \tag{14}$$

The route flow solution according to (11) not only maintains proportionality, but is in fact the entropy maximizing solution given origin-based link flows (e.g., Bar-Gera and Boyce, 1999). Furthermore, using this route flow interpretation, as demonstrated by Akamatsu (1997), the expression for route flow entropy can be computed directly from origin-based link flows by:

$$\begin{aligned}
E(\mathbf{f}) = E(\mathbf{h}(\mathbf{f})) &= -\sum_{p\in N_o}\sum_{q\in N_d(p)}\sum_{r\in R^0_{pq}} h_r \cdot \log\left(\frac{h_r}{d_{pq}}\right) \\
&= -\sum_{p\in N_o}\sum_{q\in N_d(p)}\sum_{r\in R^0_{pq}} h_r \cdot \log\left(\prod_{a\subseteq r}\alpha_{pa}\right) \\
&= -\sum_{p\in N_o}\sum_{q\in N_d(p)}\sum_{r\in R^0_{pq}}\sum_{a\subseteq r} h_r \cdot \log\left(\alpha_{pa}\right) \\
&= -\sum_{p\in N_o}\sum_{a\in A}\log(\alpha_{pa})\cdot\left[\sum_{q\in N_d(p)}\ \sum_{r\in R^0_{pq};r\supseteq a} h_r\right] \\
&= -\sum_{p\in N_o}\sum_{a\in A} f_{pa}\cdot\log(\alpha_{pa}) = -\sum_{p\in N_o}\sum_{a\in A} f_{pa}\cdot\log\left(\frac{f_{pa}}{g_{pa_h}}\right)
\end{aligned} \tag{15}$$

Consistency and proportionality between origins are a bit more challenging, and usually cannot be perfectly satisfied. Deviations from these conditions should therefore be measured. Deviations from proportionality are easier to measure by examining the proportions $\rho_p(s_1,s_2)$ for all origins relevant to PAS $\{s_1,s_2\}$. For every PAS a target proportion is chosen, either by the algorithm or by $\rho(s_1,s_2)=\sum_{p\in N_o} g_{ps_1}/\sum_{p\in N_o}[g_{ps_1}+g_{ps_2}]$; then, for every origin we compute the absolute flow deviation from proportionality (in vph) as $\zeta_{s_1s_2p}=|g_{ps_1}-\rho(s_1,s_2)\cdot(g_{ps_1}+g_{ps_2})|$. Flow deviations from proportionality can be aggregated into a global measure of performance in several different ways, for example, by considering the maximum flow deviation from proportionality (MFDP) over all PASs and origins, which is the value reported in this paper.

To measure consistency one might be interested in knowing how many more routes should be added to satisfy consistency. Even theoretically this is not a valid measure, since routes may be mistakenly excluded or mistakenly included, so to satisfy consistency we may need to omit existing routes and at the same time add others. In some cases, as shown by Gallager (1977), if the original set of routes is not "completely loop-free," adding routes until the set becomes consistent may be an infinite process. Even if the set of routes is completely loop-free, enumerating the missing routes is a difficult combinatorial problem. As an alternative it is suggested that the level of consistency can be measured by examining the distributions of reduced costs, defined as

$$rc_{pa} = c^*_{pa_t} + t_a - c^*_{pa_h} \tag{16}$$

where c^*_{pn} is the minimum cost from origin p to node n. Notice that the reduced cost is always non-negative, and it is zero if and only if link a is part of a least cost route.

Table 4 shows reduced cost distributions of four hypothetical solutions for a network with 40 links and 22 nodes of which 5 are origins, so there are 200 origin-link combinations (OLCs) in the array of origin-based link flows. Logarithmic bins of reduced costs are considered as they enable better inspection of values near zero, while covering the entire range of values with a modest number of bins. In each solution every OLC is counted once, depending on whether it is used or not ($f_{pa}>0$) and on its reduced cost value, rc_{pa}. For example, in solution 1 there are 120 OLCs with positive flow (used) and zero reduced cost; and 15 OLCs with zero flow (unused) in bin 10, i.e., reduced cost in the range of $(2^{-10},2^{-5}]$. In solution 1 all used OLCs have zero reduced cost. Recall that rc_{pa} is zero if and only if link a is included in a least cost route from origin p. Therefore,

Table 4
Reduced cost distributions of used/unused origin-link combinations in four hypothetical solutions: (1) perfectly consistent UE solution; (2) super-consistent solution; (3) mildly sub-consistent solution; (4) severely sub-consistent solution.

Bin	Reduced cost	Solution 1	Solution 2	Solution 3	Solution 4
1	0	120/0	110/0	108/2	105/5
2	$(0,2^{-45}]$	0/0	8/0	8/0	1/0
3	$(2^{-45},2^{-40}]$	0/0	1/0	0/1	0/3
4	$(2^{-40},2^{-35}]$	0/0	1/0	1/0	3/2
5	$(2^{-35},2^{-30}]$	0/0	0/0	0/0	0/0
6	$(2^{-30},2^{-25}]$	0/0	0/0	0/0	4/0
7	$(2^{-25},2^{-20}]$	0/5	0/5	0/5	0/5
8	$(2^{-20},2^{-15}]$	0/20	0/20	0/20	2/10
9	$(2^{-15},2^{-10}]$	0/30	0/29	0/29	0/33
10	$(2^{-10},2^{-5}]$	0/15	0/16	0/16	0/17
11	$(2^{-5},2^{0}]$	0/10	0/10	0/10	0/10

solution 1 perfectly satisfies the UE conditions. In this solution all OLCs with zero reduced cost are used, which means that all least cost routes are used, so the solution is also perfectly consistent.

Perfect solutions for practical problems are rather rare. Even if we manage to identify the exact set of UE routes, link flows and link costs can only be as close to the perfect UE solution as possible by computational precision limits. Approximate link costs lead to approximate route costs and approximate reduced costs. Reduced cost distributions for a solution that uses the exact set of UE routes but nearly precise link flows may look like solution 2 in Table 4. Any tiny precision error in reduced cost may cause used OLCs to shift from bin 1 to bin 2. An OLC may shift from bin 1 to bin 4 as a result of reduced cost precision error as small as 10^{-12}. Precision errors may cause shifts between other bins, in both directions, but due to the logarithmic scale such shifts occur less often. In this example one unused OLC shifted from bin 9 to bin 10.

If we examine solution 2 on its own, in the lack of the perfect solution 1 for comparison, it is not possible to determine definitely what happens. For example, it is also possible that the used OLC in bin 4 is not part of any exact UE route, so it should not have been used. In any case, reduced cost of all used links in solution 2 is not greater than $2^{-35} \approx 3E-11$ while reduced cost of all unused OLCs is greater than $2^{-25} \approx 3E-8$, so there is a clear separation between the two groups. We refer to a solution with such separation as *super-consistent*. The gap between the two groups on the logarithmic scale, which is the ratio between the thresholds, $2^{10} = 1024$ in the case of solution 2, is considered as the super-consistency level.

$$\text{super-consistency level} = \frac{\min\{rc_{pa} : f_{pa} = 0\}}{\max\{rc_{pa} : f_{pa} > 0\}} \tag{17}$$

A connection between super-consistency level and the minimal complexity of possible consistency violations is established in Bar-Gera (2006), showing in particular that any level of super-consistency above 2 ensures the basic condition of consistency discussed in Section 3 which is the most important one and the critical pre-requisite for proportionality. In that respect super-consistent solutions are very likely to be perfectly consistent.

Solution 3 is not super-consistent because there is one used OLC with reduced cost above 2^{-40} and 3 unused OLCs with reduced costs below 2^{-40}. It is possible that in the perfect solution these OLCs are part of used UE routes, as implied from the comparison to solutions 1 and 2. However, considering solution 3 by itself, we cannot know for sure, especially regarding the OLC in bin 4. We consider solutions without separation in reduced cost distributions as sub-consistent, as they are quite unlikely to be perfectly consistent. Both solutions 3 and 4 are sub-consistent, but to very different extents. In particular, solution 4 has used OLCs in bin 8, which most probably should not be used at all, so the solution appears to be not as close to satisfying the UE conditions as solution 3.

To measure the degree of sub-consistency we quantify how mixed are the distributions of used and unused OLCs. The key value is the number of potentially mistakenly excluded OLCs, i.e., unused OLCs whose reduced cost is lower than the maximum reduced cost of all used OLCs. In the case of solution 3 there are three potentially mistakenly excluded OLCs. This number should be considered in relation to the size of the problem. The size could be captured simply by the number of used links, but this could be slightly misleading. The number of links in a tree of least cost routes from a single origin equals the number of nodes minus one. Typically all of these links will be used, so in our case there will be at least 105 used OLCs. In the perfect solution 1 there are 120 used OLCs. In solution 4 there are 115 used OLCs and 25 potentially mistakenly excluded OLCs. This solution in fact contains only 10 additional OLCs, above the needed minimum. It seems that the level of sub-consistency in this case is better captured by comparing the 25 missing OLCs to the 10 captured additional OLCs, rather than to all 115 used OLCs. Therefore we define the sub-consistency level as the ratio of the number of potentially mistakenly excluded OLCs to the number of additional OLCs. The levels of sub-consistency of solutions 3 and 4 are 3/12 = 0.25 and 25/10 = 2.5, respectively, clearly quite different from each other. Formally the sub-consistency level is defined as:

$$SubconsistencyLevel = \frac{\left|\left\{p \in N_o, a \in A : f_{pa} = 0, rc_{pa} < \max\left\{rc_{p'a'} : f_{p'a'} > 0\right\}\right\}\right|}{\left|\left\{p \in N_o, a \in A : f_{pa} > 0\right\}\right| - |N_o| \cdot (|A| - 1)} \tag{18}$$

Comment: in practical applications there may be nodes that are not used at all by certain origins. The definitions of consistency measures should be adjusted to exclude those cases from the evaluation. Details of these adjustment are omitted here as they are not essential, in order to avoid further complication of the exposition.

6. Effective PAS shifts and convergence persistence

In most cases a relatively simple implementation of the concepts discussed in Section 4 works reasonably well, but in some cases flow shifts may be somewhat ineffective, leading to various behaviors that either prevent convergence or at least appear as persistence against convergence. This section starts with conditions to ensure PAS effectiveness. In some cases effective PAS cannot be found, and an alternative branch shift is proposed. A proof of convergence for a general algorithm structure that combines effective PAS shifts and branch shifts follows. Towards the end of the section peculiar cases of super coupling are presented and the resulting persistence is discussed. The effectiveness issues discussed here apply not only to the TAPAS algorithm, but possibly to other algorithmic approaches as well, as described herein.

6.1. Cost-effective PAS

Section 4 shows that typically all possible shifts of flow between routes can be accomplished by a relatively small set of PASs. Such a set of PASs is considered as *covering* or *spanning*. In the case of a single-origin a-cyclic network it is relatively easy to test whether a set of PASs is covering. In particular if at most two links merge at any node, the set of PASs is covering if and only if there is at least one PAS that merges at any merge node of the network. A generalization of this condition to multiple-link merge nodes is offered in Bar-Gera (2006).

Finding a covering set of PASs is important to proportionality, but less useful when discussing convergence, as illustrated by the simple network in Fig. 8. There are three routes in this network from origin 1 to destination 7. There are three PASs, and each two of them form a covering set. Suppose we identified in previous iterations the PASs A:{[1,2,3,5,7],[1,6,7]} and *B*:{[2,3,5],[2,4,5]}. Every possible shift of flow among routes can be accomplished by a combination of flow shifts that use these two PASs.

In each PAS, when constructed, one segment was a least cost segment, but this is not necessarily the case in subsequent iterations. Suppose that at some iteration, denoted as case I, the flows are: $f_{[1,6]} = f_{[6,7]} = 10$ and zero on all other links, while costs are: $t_{[1,6]} = t_{[6,7]} = 2$; $t_{[1,2]} = t_{[2,3]} = t_{[3,5]} = t_{[5,7]} = 1$; and $t_{[2,4]} = t_{[4,5]} = 0.5$. This solution is clearly not an equilibrium since all the flow is on route [1,6,7] with cost 4 while the cost of route [1,2,4,5,7] is only 3. Nevertheless, a line search considering PAS A will make zero shift because the costs are equal, and a line search considering PAS B will make zero shift because the high cost segment has zero flow. Being an extreme case this situation can possibly be detected and treated by a special procedure. More difficult to identify are similar situations where, for example, $t_{[2,3]} = 0.99 + f_{[2,3]}$ (case II), so that the current cost of the first segment of PAS B is 1.99. In this case a line search with PAS A will suggest a shift of 0.01 vph, and a following line search with PAS B will shift this flow of 0.01 vph to the shortest route. The process will continue 1000 times until all flow is shifted. This behavior, which can be described as "dripping" or "cascading," is clearly not effective. A related problem may occur in certain other origin-based algorithms (Bar-Gera, 2002; Nie, 2007), but does not occur with algorithm B (Dial, 2006).

One way to ensure that PASs are "cost-effective" is to require that the lower cost segment is included in a least cost route, which might be a relatively severe requirement. If we change the above example so that $t_{[2,3]} = 0.51 + f_{[2,3]}$; $t_{[3,5]} = 0.5$ (case III), the first segment of PAS A is not part of a least cost route, even though it would be quite effective. The result of enforcing the least cost requirement would be to generate many unnecessary PASs with computational efforts both when generating them and when shifting flows for each of them.

An alternative approach is to compare the PAS cost difference with the reduced costs at the merge. In the above example the reduced cost is $rc_{1[6,7]} = 1$. The PAS cost differences in cases I–III are 0, 0.01, and 0.99, respectively. We choose a factor, say $\mu = 0.5$, and consider a PAS as cost-effective if the PAS cost difference is at least half of the reduced cost. Formally PAS $\{s_1, s_2\}$ is considered cost-effective for origin p if $c_{s_1} - c_{s_2} \geqslant \mu \cdot rc_{pa}$, where segments s_1, s_2 end with links a, a', respectively, and only a' is part of a least cost route. Notice that when a new PAS is constructed the PAS cost difference is at least as high as the reduced cost, so if none of the existing PASs is cost-effective we can always identify a PAS that is cost-effective.

6.2. Flow-effective PAS

In the construction of a new PAS, as presented in Section 4, the backwards breadth first search over used links ensures that the high cost segment has a strictly positive (non-zero) flow. Similar to the cost-effectiveness issue, this property may change over the iterations. Using Fig. 8 for illustration, consider case IV where the flows are: $f_{[1,2]} = f_{[2,4]} = f_{[4,5]} = f_{[5,7]} = 10$ and zero on all other links, while costs are: $t_{[1,6]} = t_{[6,7]} = t_{[2,3]} = t_{[2,4]} = 2$; $t_{[1,2]} = t_{[3,5]} = t_{[4,5]} = t_{[5,7]} = 1$. This solution is clearly not an equilibrium since all the flow is on route [1,2,4,5,7] with cost 5 while the cost of route [1,6,7] is only 4. A line search with PAS B will lead to zero shift because the costs of the two segments are equal to 5. A line search with PAS A will lead to zero shift because the flow on the high cost segment is zero.

Again case IV is an extreme situation that could possibly be identified as a special case, but similar situations may lead to cascading effects. Consider, for example, what happens if we modify case IV so that $t_{[2,3]} = 1.99 + f_{[2,3]}$ (case V). Considering PAS B we shift 0.01 vph to equalize the costs, and then with PAS A we shift this flow to route [1,6,7]. Again 1000 repetitions will be needed to reach equilibrium.

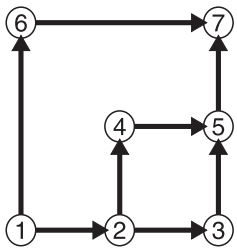

Fig. 8. Example network for basic PAS effectiveness issues.

The proposed solution is to choose a factor, say $v = 0.25$, and to consider a PAS as flow-effective if the ratio between the minimum flow of all links along the high cost segment to the flow on the last link of that segment is not less than the chosen factor. Formally PAS $\{s_1, s_2\}$is considered flow-effective for origin p if $\min(f_{p\hat{a}} : \hat{a} \subseteq s_1) \geqslant v \cdot f_{pa}$, assuming that $c_{s_1} > c_{s_2}$ and a is the last link of segment s_1.

When examining high reduced cost link a and the least cost alternative approach a', if all existing PASs with the same merging links are not flow-effective, we search for a new PAS, but this time with an additional restriction on the backward search to consider only links with flow greater or equal to $v \cdot f_{pa}$, rather than considering all used links. Such a search is not guaranteed to work, due to flow spreading, which is discussed in the next subsection.

6.3. Flow spreading and branch shifts

Consider the grid network in Fig. 6. Suppose that in early iterations flows from other origins use the segment [32,33,34,35,36] extensively, so the flows from origin A do not use any route containing this segment. Suppose that as costs among other routes are equilibrated, flows from A to F are spread over all of these other routes, and the cost of all of them is 30. In subsequent iterations the usage of segment [32, 33, 34, 35, 36] by other origins reduces, until it becomes part of the least cost route, and a new PAS will be constructed. Since current flows from A to node 30 are highly spread it is quite possible that the new PAS will not be flow-effective. It is in fact possible that there will be no flow-effective PAS. This would be the case if, for example, $f_{[30,36]} = 60$ and $f_{[25,26]} = f_{[20,26]} = f_{[20,21]} = f_{[10,11]} = f_{[5,11]} = f_{[5,6]} = 10$.

The breadth-first-search method will choose a local PAS of arbitrary flow-effectiveness. In each subsequent iteration a small amount of flow will be shifted using the new PAS that merge at 36, and then the other PASs will be used to shift flows in a way that re-equilibrates the costs to node 30. The resulting process may be quite persistent. One solution is to add more PASs that merge at 36, ignoring the flow-effectiveness condition, to allow direct shifting from other routes to the least cost route. Depending on how widely spread are the flows, the number of PASs needed could be large, and the additional computational burden may be significant.

Flow spreading is particularly likely in an algorithm that attempts to maintain conditions of proportionality and consistency, since these conditions suggest that as many routes as possible should be used. It is possible to achieve any desired level of UE precision with a very small number of routes, bounded by the number of O–D pairs plus the number of links. Some route-based methods rely on this property and use only a few alternative routes, thus reducing the likelihood of flow spreading, although not necessarily avoiding it completely. A clear drawback of using few routes is that the resulting set of used routes is often highly inconsistent and the flows are very far from satisfying proportionality. In-depth examination of flow spreading effects and their significance in various algorithms might reveal interesting findings, but it is beyond the scope of this paper.

It appears that a different treatment may be needed. Following the terminology proposed by Dial (1971, 2006), referring to an a-cyclic network from a single origin as a "bush," we consider the set of segments from an origin ending with a single link as a "branch." When approach proportions at a single node are changed, the implied effect from (11) on route flows and origin-based link flows can be viewed as a shift from one branch to another. To address the issue of spread flows it is proposed to shift flow from the branch that ends at the high reduced cost link to the least cost alternative segment. The proportions of flow on segments within the branch remain constant, so the shift reduces the objective function as long as the average cost of the branch is higher than the cost of the alternative segment. A simple line search can therefore be used to determine exactly how much flow should be shifted. This procedure is referred to as a *branch shift*, and it is applied to any case in which a flow-effective PAS cannot be found.

6.4. Convergence

The proof of convergence is based on the effectiveness of PAS shifts and branch shifts. It is sufficient to ensure that one of the two shifts is performed only for problematic origin link combinations, which are used OLCs (non-zero origin-based flow) with high reduced cost above a certain iteration-specific threshold, where the sequence of thresholds, Φ_k, converges to zero. (In the computational experiments reported here $\Phi_k = AEC_0 \cdot 10^{-k}$, where AEC_0 is the average excess cost as defined in (3) for an initial all-or-nothing solution.) We show first that there is a constant strictly positive lower bound on the reduction of the objective function in effective PAS shifts and in branch shifts. Since infinite repetitions of such reductions is not possible, this result is used to prove convergence by contradiction.

Lemma 1. *The reduction in objective function of an effective PAS shift for OLC p^0, a^0 is at least $\xi(f_{p^0a^0}, rc_{p^0a^0})$, which is strictly positive if $f_{p^0a^0} > 0, rc_{p^0a^0} > 0$.*

Proof. Denote the segment that ends at a^0 as s^0, and the other segment by s^1. Since the PAS is flow effective and cost effective the minimum flow along s^0 is at least $v \cdot f_{p^0a^0}$, and the cost difference between s^0 and s^1 is at least $\mu \cdot rc_{p^0a^0}$. Recall that link cost functions are uniformly Lipschitz continuous with modulus Λ, meaning that $|t_a(f_{\bullet a} + \delta) - t_a(f_{\bullet a})| \leqslant \Lambda \cdot |\delta|$. Therefore, when an amount of δ flow is shifted from s^0 to s^1, the change in the cost of every link in these segments is at most $\Lambda \cdot \delta$. The total number of links in each segment is less than $|A|$, hence the total change in cost difference is at most $\Lambda \cdot \delta \cdot 2|A|$. As long as the shift is less than $\delta \leqslant \mu \cdot rc_{p^0a^0}/4\Lambda|A|$, the cost difference remains at least $\mu \cdot rc_{p^0a^0} - 2\delta\Lambda|A| \geqslant \mu \cdot rc_{p^0a^0}/2$. If

$\mu \cdot rc_{p^0a^0}/4\Lambda|A| \geqslant v \cdot f_{p^0a^0}$, meaning that a shift of $\delta = \mu \cdot rc_{p^0a^0}/4\Lambda|A|$ is feasible, the reduction in objective function will be at least $(\mu \cdot rc_{p^0a^0}/2) \cdot (\mu \cdot rc_{p^0a^0}/4\Lambda|A|) = \mu^2 \cdot rc^2_{p^0a^0}/8\Lambda|A|$. Otherwise, a shift of $\delta = v \cdot f_{p^0a^0}$ or more is implemented, which reduces the objective function by at least $(v \cdot f_{p^0a^0}) \cdot (\mu \cdot rc_{p^0a^0}/2)$. In conclusion $\xi(f_{p^0a^0}, rc_{p^0a^0}) = \min(\mu^2 \cdot rc^2_{p^0a^0}/8\Lambda|A|, \mu \cdot v \cdot f_{p^0a^0} \cdot rc_{p^0a^0}/2)$ satisfies the requirements of the lemma. □

Lemma 2. *The reduction in objective function of a branch shift for OLC p^0, a^0 is at least $\xi(f_{p^0a^0}, rc_{p^0a^0})$.*

Proof. The cost of every segment in the branch from origin p^0 that ends at link a^0 is at least $rc_{p^0a^0}$ above the minimum cost alternative. Thus the difference between the average branch cost and the alternative segment is at least $rc_{p^0a^0}$. As in the previous proof, a shift of δ reduces the cost difference by at most $\Lambda \cdot \delta \cdot 2|A|$. If $\delta = rc_{p^0a^0}/4\Lambda|A| > f_{p^0a^0}$ all the flow will be shifted, and the reduction in objective function will be at least $(f_{p^0a^0}) \cdot (rc_{p^0a^0}/2) \geqslant \xi(f_{p^0a^0}, rc_{p^0a^0})$. Otherwise, the amount of flow shifted is at least $\delta = rc_{p^0a^0}/4\Lambda|A|$ and the reduction in objective function is at least $rc^2_{p^0a^0}/8\Lambda|A| \geqslant \xi(f_{p^0a^0}, rc_{p^0a^0})$. □

Theorem 1. *if $\Phi_k \to 0$ and in every iteration k for every problematic OLC with $f_{pa} > 0$ and $rc_{pa}(\mathbf{f}^k) > \Phi_k$ either effective PAS shift or a branch shift is applied, then the sequence of objective function values $T(\mathbf{f}^k)$ converges to equilibrium.*

Proof. The sequence $T(\mathbf{f}^k)$ is monotonically non-increasing and bounded below by the optimal value T^*. The set of feasible origin-based solutions is compact, therefore the sequence of solutions produced by the iterations has a converging subsequence $\mathbf{f}^{k_l} \to \bar{\mathbf{f}}$. Suppose by contradiction that $\bar{\mathbf{f}}$ is not an equilibrium solution, so that there exist origin p^0 and link a^0 such that $\bar{f}_{p^0a^0} = \delta > 0$ and $rc_{p^0a^0}(\bar{\mathbf{f}}) = \epsilon > 0$. There exist l_0 such that $\Phi_{k_l} < \epsilon/2,\ f^{k_l}_{p^0a^0} > \delta/2$ and $rc_{p^0a^0}(\mathbf{f}^{k_l}) > \epsilon/2$ for all $l > l_0$. Therefore, in each of these iterations the OLC p^0a^0 is considered problematic, leading to either an effective PAS shift or a branch shift. In both cases the reduction in objective function is at least $\xi(\delta/2, \epsilon/2) > 0$. Infinite repetitions of such reduction leads to an infinite reduction in the objective function, which is a contradiction. □

6.5. Extreme coupling

Theoretical convergence does not necessarily imply efficient convergence. Numerical experiments on large-scale networks revealed a peculiar behavior, in which a cost difference of a certain magnitude ϵ is identified on a particular PAS, a flow shift on the PAS effectively equalizes the costs of the two segments, and the costs remain equal for a while. Nevertheless, it does not take very long before the cost difference returns almost exactly to the same cost difference ϵ. In-depth examination suggested that there are several possible mechanisms that may cause such behavior. A relatively simple one of these can be illustrated by the example in Table 5. O–D flow from A to B is 100 and from C to D is 100. The cost of all the links is 10, with the following exceptions: $t_{[1,6]} = 9.99$ and $t_{[1,2]} = 9 + 0.01 * f_{[1,2]}$. The equilibrium solution is that $h_b = h_c = 100$ and $h_a = h_d = 0$. As a result $f_{[1,2]} = h_c = 100,\ t_{[1,2]} = 10$, and $c_a = 60;\ c_b = 59.99;\ c_c = c_d = 70$.

Suppose that an example of this type is embedded in a much larger network, and as a result of other routes we get a solution where $h_a = h_d = 100$ and $h_b = h_c = 0$. In this case, $f_{[1,2]} = h_a = 100$, so the costs are $t_{[1,2]} = 10$ and $c_a = 60;\ c_b = 59.99;\ c_c = c_d = 7$, as before. But this time the solution is not at equilibrium because route *a* is being used even though its cost is higher than the cost of route *b*. Most efficient TAP algorithms are disaggregated (by route, origin, or destination) so flow shifts for each of the two O–D pairs (A–B/C–D) are considered separately. Considering travelers from A to B, flow should be shifted from route *a* to route *b*. Shifting just one unit of flow reduces the cost of route *a* to 59.99, thus equalizing the cost of the two routes. Once this unit of flow is shifted, the cost of route *c* reduces to 69.99, so now the flow on route *d* is using a higher cost route and some of it should be shifted to route *c*. Again it is enough to shift one unit of flow to equalize the cost of the two routes. To reach the equilibrium solution this process needs to be repeated 100 times.

The existence of interactions between PASs is clear from the outset, and justifies the need for an iterative process. The strong interaction demonstrated here may be described as "coupling." One can easily construct even more extreme coupling examples where 1000 or 1,000,000 repetitions would be needed. It seems that larger number of repetitions are more likely to be needed when cost differences are smaller, so that the impact on any global measure of convergence (AEC, TEC) would be relatively small. As a result, this type of problem is likely to be encountered only when very high precision solutions are computed, probably beyond the practical needs for scenario comparisons. High precision does make a difference when consistency is evaluated using the measures proposed in Section 5, as demonstrated in the results presented in Section 8.

Experience so far suggests that extreme coupling is a relatively rare situation. So the most important thing is to make sure that the treatment of this problem does not ruin the general performance of the algorithm. Unfortunately, a good solution for

Table 5
Extreme coupling example.

PAS index	Route name	Route nodes
1	*a*	[A,1,2,3,4,5,B]
1	*b*	[A,1,6,7,8,5,B]
2	*c*	[C,9,10,1,2,11,12,D]
2	*d*	[C,9,13,14,15,16,12,D]

extreme coupling has not been identified so far in this research. As a tentative remedy "enhanced iterations" are used, in which most computation time is dedicated to flow shifts on PASs, rather than to searching for new PASs.

7. Equalizing proportions between origins

In this section we focus on the equalization of proportions between origins for a single PAS, $\{s_1, s_2\}$ with relevant origins $P \subseteq N_o$. It is understood that the procedure will be applied repeatedly to all PASs identified by the algorithm, as described in Section 4. For an isolated PAS the adjustment is rather straight forward. Consider the example in Table 6. Clearly the origin-specific proportions are not the same. The deviation of flow on segment 1 is computed as $\delta_p = \rho(s_1, s_2) \cdot (g_{ps_1} + g_{ps_2}) - g_{ps_1}$, where the overall proportion is computed from the sum of segment flows over all origins by $\rho(s_1, s_2) = \sum_{p \in N_o} g_{ps_1} / \sum_{p \in N_o} [g_{ps_1} + g_{ps_2}] = 0.8$. If these deviations are applied as adjustments to origin-based segment flows, i.e., $g_{ps_1} = g_{ps_1} + \delta_p$ and $g_{ps_2} = g_{ps_2} - \delta_p$ the resulting flows are (20,5) for origin 3, (12,3) for origin 25, and (48,12) for origin 43, and the PAS proportion for all origins are the same (0.8).

This naive approach, proposed mistakenly as a general method in Bar-Gera (2006), is valid only if segments are "isolated" in the sense that they do not contain intermediate merge nodes. Consider, for example, the situation in Fig. 9, with two origins A and B, and two destinations C and D. Suppose that proportionality is sought for the PAS {[3,4,6],[3,5,6]}. In case I, origin-based link flows are $f_{A[A,1]} = f_{A[1,3]} = 109$; $f_{A[3,4]} = 100$; $f_{A[4,C]} = 90$; $f_{A[4,6]} = 10$; $f_{A[3,5]} = f_{A[5,6]} = 9$; $f_{A[6,D]} = 19$; $f_{B[3,4]} = f_{B[4,6]} = 10$; $f_{B[3,5]} = f_{B[5,6]} = 990$; $f_{B[B,3]} = f_{B[6,D]} = 1000$; all other flows are zero. Origin-based segment flows are $g_{A[3,4,6]} = 10$; $g_{A[3,5,6]} = 9$; $g_{B[3,4,6]} = 10$; $g_{B[3,5,6]} = 990$. Origin-specific PAS proportions are $\rho_A(s_1, s_2) = 10/19$ and $\rho_B(s_1, s_2) = 10/1000$. In this case, since the segments are in fact isolated, the naive method provides the correct adjustment. The overall ratio is $\rho(s_1, s_2) = 20/1019$, the adjustments are $\delta_A = -9.627$; $\delta_B = 9.627$, the modified origin-based link flows are $f_{A[3,4]} = 90.373$; $f_{A[4,6]} = 0.373$; $f_{A[3,5]} = f_{A[5,6]} = 18.627$; $f_{B[3,4]} = f_{B[4,6]} = 19.627$; $f_{B[3,5]} = f_{B[5,6]} = 980.373$, flows on links outside the PAS are not adjusted, and the new origin-specific PAS ratios are both equal to 0.01963. Perfectly precise proportionality (limited by rounding only) is obtained in a single adjustment.

Consider now case II, by changing case I so that $f_{A[A,1]} = 1009$; $f_{A[1,2]} = f_{A[2,4]} = 900$; $f_{A[4,C]} = 990$. Due to the merging flows from origin A at node 4 the origin-based segment flow according to (12) is $g_{A[3,4,6]} = 1$. In this case the naive adjustments are $\delta_A = -0.891$; $\delta_B = 0.891$, the new origin-based link flows are $f_{A[3,4]} = 99.109$; $f_{A[4,6]} = 9.109$; $f_{A[3,5]} = f_{A[5,6]} = 9.891$; $f_{B[3,4]} = f_{B[4,6]} = 10.891$; $f_{B[3,5]} = f_{B[5,6]} = 989.109$, and the new origin-specific proportions are $\rho_A(s_1, s_2) = 0.08371$ and $\rho_B(s_1, s_2) = 0.01089$, which are quite different. The new deviation from proportionality is 0.777, which is approximately 13% lower than the original deviation from proportionality. In this case repeated application of the same adjustment process provides a reduction of deviation from proportionality of 13% per repetition. Depending on the desired level of precision, the number of necessary repetitions could be quite substantial, even with this very simple example.

The key issue is that adjustments are made at the level of origin-based link flows, influencing origin-based segment flows in a non-linear manner, which is ignored in the naive approach. In fact, it is not quite clear whether the naive adjustments guarantee convergence to proportionality. Therefore, a more formal evaluation is necessary.

Table 6
Equalizing proportions on isolated PAS.

Origin (p)	Flow 1 (g_{ps_1})	Flow 2 (g_{ps_2})	Proportion ($\rho_p(s_1, s_2)$)	Deviation/adjustment (δ_p)
3	15	10	0.6	+5
25	15	0	1	−3
43	50	10	5/6	−2
Total	80	20	0.8	

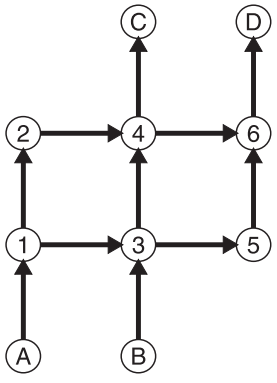

Fig. 9. The limitations of naive proportionality adjustments.

Consider a vector of origin-based PAS adjustments, $\boldsymbol{\delta}$, such that $\delta_p = 0 \quad \forall p \notin P$. The associated array of adjustments to origin-based link flows is $\Delta\mathbf{f}(\boldsymbol{\delta})$ defined by:

$$\Delta f_{pa} = \begin{cases} \delta_p & a \subseteq s_1 \\ -\delta_p & a \subseteq s_2 \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

Total link flows remain the same if the sum of adjustments is zero. Using (15) the entropy maximizing adjustments for a single PAS are the solution to the following optimization problem:

$$\begin{aligned} \max \; & E(\mathbf{f} + \Delta\mathbf{f}(\boldsymbol{\delta})) = -\sum_{p \in N_o} \sum_{a \in A} (f_{pa} + \Delta f_{pa}(\boldsymbol{\delta})) \cdot \log\left(\frac{f_{pa} + \Delta f_{pa}(\boldsymbol{\delta})}{g_{pa_h}(\mathbf{f} + \Delta\mathbf{f}(\boldsymbol{\delta}))}\right) \\ \text{s.t.} \quad & \sum_p \delta_p = 0 \\ & \mathbf{f} + \Delta\mathbf{f}(\boldsymbol{\delta}) \geqslant 0 \end{aligned} \tag{20}$$

and the Lagrangian is

$$\begin{aligned} \max \; & \mathscr{L} = E(\mathbf{f} + \Delta\mathbf{f}(\boldsymbol{\delta})) + \lambda \cdot \left(\sum_p \delta_p\right) \\ \text{s.t.} \quad & \mathbf{f} + \Delta\mathbf{f}(\boldsymbol{\delta}) \geqslant 0 \end{aligned} \tag{21}$$

To derive optimality conditions we consider each origin-link component of the entropy separately, denoting

$$E_{pa}(\mathbf{f} + \Delta\mathbf{f}(\boldsymbol{\delta})) = -(f_{pa} + \Delta f_{pa}(\boldsymbol{\delta})) \cdot log\left(\frac{f_{pa} + \Delta f_{pa}(\boldsymbol{\delta})}{g_{pa_h}(\mathbf{f} + \Delta\mathbf{f}(\boldsymbol{\delta}))}\right)$$

$$E(\mathbf{f} + \Delta\mathbf{f}(\boldsymbol{\delta})) = \sum_{p \in N_o} \sum_{a \in A} E_{pa}(\mathbf{f} + \Delta\mathbf{f}(\boldsymbol{\delta}))$$

Considering the last link of segment 1, $a_1 \subseteq s_1$ that ends at the merge node $a_{1_h} = s_{1_h} = n$, the change in origin-based link flow is $\Delta f_{pa_1}(\boldsymbol{\delta}) = \delta_p$, and the total node flow does not change, $g_{pn}(\mathbf{f} + \Delta\mathbf{f}(\boldsymbol{\delta})) = g_{pn}(\mathbf{f})$ because the adjustment on segment 2 cancels the adjustment on segment 1. Therefore:

$$\frac{\partial E_{pa_1}}{\partial \delta_p} = -\log\left(\frac{f_{pa_1} + \delta_p}{g_{pn}}\right) - (f_{pa_1} + \delta_p) \cdot \frac{1}{f_{pa_1} + \delta_p} = -\log\left(\frac{f_{pa_1} + \delta_p}{g_{pn}}\right) - 1 \tag{22}$$

Similarly, considering the last link of segment 2, $a_2 \subseteq s_2$; $a_{2_h} = s_{2_h} = n$, the change in origin-based link flow is $\Delta f_{pa_2}(\boldsymbol{\delta}) = -\delta_p$, and

$$\frac{\partial E_{pa_2}}{\partial \delta_p} = \log\left(\frac{f_{pa_2} - \delta_p}{g_{pn}}\right) + 1 \tag{23}$$

Entropy components for other links leading to the merge node do not change, so the contribution of the merge node to the derivative is:

$$\sum_{a:a_h=n} \frac{\partial E_{pa}}{\partial \delta_p} = -\log\left(\frac{f_{pa_1} + \delta_p}{g_{pn}}\right) + \log\left(\frac{f_{pa_2} - \delta_p}{g_{pn}}\right) \tag{24}$$

Considering any other link of segment 1, $a' \subseteq s_1;\ a'_h = j \neq n$, the change in origin-based link flow and in the total node flow are $g_{pj}(\mathbf{f} + \Delta\mathbf{f}(\boldsymbol{\delta})) - g_{pj}(\mathbf{f}) = \Delta f_{pa'}(\boldsymbol{\delta}) = \delta_p$. Hence the entropy component derivative is:

$$\frac{\partial E_{pa'}}{\partial \delta_p} = -\log\left(\frac{f_{pa'} + \delta_p}{g_{pj} + \delta_p}\right) - (f_{pa'} + \delta_p) \cdot \left(\frac{1}{f_{pa'} + \delta_p} - \frac{1}{g_{pj} + \delta_p}\right) = -\log\left(\frac{f_{pa'} + \delta_p}{g_{pj} + \delta_p}\right) - 1 + \frac{f_{pa'} + \delta_p}{g_{pj} + \delta_p} \tag{25}$$

For any other link $a \neq a'$ that ends at the same node, $a_h = j$, the entropy component derivative is:

$$\frac{\partial E_{pa}}{\partial \delta_p} = \frac{f_{pa}}{g_{pj} + \delta_p} \tag{26}$$

The sum over all the links ending at j is

$$\begin{aligned} \sum_{a_h=j} \frac{\partial E_{pa}}{\partial \delta_p} &= -\log\left(\frac{f_{pa'} + \delta_p}{g_{pj} + \delta_p}\right) - 1 + \frac{f_{pa'} + \delta_p}{g_{pj} + \delta_p} + \sum_{a_h=j; a \neq a'} \frac{f_{pa}}{g_{pj} + \delta_p} \\ &= -\log\left(\frac{f_{pa'} + \delta_p}{g_{pj} + \delta_p}\right) - 1 + \frac{\sum_{a_h=j} f_{pa} + \delta_p}{g_{pj} + \delta_p} = -\log\left(\frac{f_{pa'} + \delta_p}{g_{pj} + \delta_p}\right) \end{aligned} \tag{27}$$

where the last transition is due to the fact that the origin-based node flow is equal to the sum of arriving origin-based link flows, $g_{pj} = \sum_{a_h=j} f_{pa}$. An equivalent derivation shows that for any node in segment 2 other than the merge and the diverge nodes, $j \in s_2;\ j \neq s_{2_t};\ j \neq s_{2_h}$, the sum of derivative components is:

$$\sum_{a_h=j} \frac{\partial E_{pa}}{\partial \delta_p} = \log\left(\frac{f_{pa'} - \delta_p}{g_{pj} - \delta_p}\right) \tag{28}$$

where $a' \subseteq s_2;\ a'_h = j$. PAS adjustments for origin p do not influence any other origin-link components; therefore, combining (24), (27), and (28) we see that the overall entropy derivative is:

$$\frac{\partial E}{\partial \delta_p} = -\sum_{a \subseteq s_1} log\left(\frac{f_{pa} + \delta_p}{g_{pa_h}(\delta_p)}\right) + \sum_{a \subseteq s_2} log\left(\frac{f_{pa} - \delta_p}{g_{pa_h}(\delta_p)}\right)$$

This expression can be simplified using (12) to

$$\frac{\partial E}{\partial \delta_p} = -\log\left(\frac{g_{ps_1}(\delta_p)}{g_{ps_{1_h}}(\delta_p)}\right) + \log\left(\frac{g_{ps_2}(\delta_p)}{g_{ps_{2_h}}(\delta_p)}\right) = -\log\left(\frac{g_{ps_1}(\delta_p)}{g_{ps_2}(\delta_p)}\right)$$

The optimality conditions of (21) are therefore

$$\frac{\partial \mathscr{L}}{\partial \delta_p} = -\log\left(\frac{g_{ps_1}(\delta_p)}{g_{ps_2}(\delta_p)}\right) + \lambda = 0 \tag{29}$$

$$\frac{g_{ps_1}(\delta_p)}{g_{ps_2}(\delta_p)} = \exp(\lambda) \tag{30}$$

$$\rho_p(\delta_p) = \frac{g_{ps_1}(\delta_p)}{g_{ps_1}(\delta_p) + g_{ps_2}(\delta_p)} = (1 + \exp(-\lambda))^{-1} = \rho \quad \forall p \in P \tag{31}$$

simply saying that after the adjustment all origin-specific PAS proportions should be the same.

A nice trick to solve problems of this type is to consider the dual variable λ, or equivalently the overall proportion ρ, as the key solution variable. In order to do so, we need to consider the inverse functions $u_p = \rho_p^{-1}$, so that $u_p(\rho)$ provides the necessary adjustment in origin p to modify the PAS proportion to ρ. The adjustment for every origin δ_p is bounded to the range $[\delta_p^{\min}, \delta_p^{\max}]$, where $\delta_p^{\min} = -\min_{a \subseteq s_1} f_{pa}$ and $\delta_p^{\max} = \min_{a \subseteq s_2} f_{pa}$. The range of ratios covered by these adjustments is in fact full, since $\rho_p(\delta_p^{\min}) = 0$ and $\rho_p(\delta_p^{\max}) = 1$. As a result, the function u_p is defined on the range [0,1] for all the origins. The sum of adjustments is denoted by $U(\rho) = \sum_{p \in P} u_p(\rho)$. Our goal is therefore "simply" to find the root $U(\rho) = 0$. The thorn is that explicit expressions for the functions u_p are difficult to identify. In the results reported here quadratic approximations of u_p are used, leading to a quadratic form of $U(\rho)$, for which the root can be immediately identified.

One advantage of this approach is that if the PAS is in fact isolated, i.e., there are no merge nodes, $\rho_p(\delta_p)$ is a linear function and therefore its inverse u_p is also linear. If this is the case for all relevant origins the enhanced approach is equivalent to the naive method, and provides perfect proportionality in one adjustment. In other cases, due to the quadratic approximation, repetitions are necessary, but in most cases very few repetitions reach the limit of computing precision. Since proportionality adjustments for a specific PAS are embedded in an overall iterative scheme that considers all other PASs, as well as other computations, a single approximated adjustment per PAS per iteration is used in the current implementation.

8. Numerical results

Numerical results are presented for five test networks, all available from Bar-Gera (2001). (One modification was made to the test network of Philadelphia, in order to avoid zero costs, which is to set the distance parameter to 0.01 in the generalized cost function.) Basic characteristics of the test networks are given in Table 7. The new TAPAS method is directly compared with two other methods, the Frank-Wolfe (FW) method (LeBlanc et al., 1975) and the origin-based assignment (OBA) (Bar-Gera, 2002). Comparisons with more advanced methods are based on reports in the literature.

All computations were conducted on a set of regular PCs (Windows operating system, 2GB RAM) at the University of Wisconsin in Madison, through the Condor throughput computing environment (Condor, 2009). Among other advantages of this experimental arrangement is the ability to perform each test on several computers with similar properties, and thus obtain average computing times that are less sensitive to the properties of any individual machine. Computation times reported below are averages from ten tests. All codes were written in C. Common code implementations were used in all three algorithms as much as possible.

Information about the structure of the sets of PASs for each network is presented in Table 8. It shows that the number of PASs in each set is relatively small in comparison to the number of routes represented by these PASs. According to Bar-Gera (2006) the minimum numbers of PASs needed to achieve equilibrium for the Chicago sketch and regional networks are 266 and 5019, respectively. The actual numbers of PASs used by the new algorithm are in the same order of magnitude as these minimum values. Every PAS consists of two segments, each of them defined by a sequence of links, as well as a set of relevant

Table 7
Basic characteristics of test networks.

Network	Zones (origins)	Nodes	Links	O–D pairs
Sioux Falls	24	24	76	528
Chicago sketch	387	933	2950	93,513
Berlin center	865	12,981	28,376	49,688
Chicago regional	1790	12,982	39,018	2,297,945
Philadelphia	1525	13,389	40,003	1,151,166

Table 8
Structure of TAPAS solutions for different test networks.

Network	PASs	Origins per PAS	Links per segment	Used routes
Sioux Falls	54	2.56	3.24	759
Chicago sketch	343	20.91	3.81	127,248
Berlin center	404	5.10	5.42	56,887
Chicago regional	21,409	99.00	15.14	92,265,590
Philadelphia	17,818	66.19	22.63	370,108,042

origins. The table shows the average number of links per segment, over all segments in all PASs, as well as the average number of origins per PAS. The results show that segments are relatively short, indicating that PASs are mostly local. Typical PASs have a fairly large number of relevant origins, increasing the confidence that the proposed mechanism to detect PASs behaves properly.

Computation times needed to reach various levels of convergence by each of the three algorithms in all five networks are presented in Table 9. Missing values in the table represent levels of convergence that were not reached by the specific algorithm within the allocated time for computation. The standard deviations of CPU time over the ten runs of each of the cases were less than 10% of the presented average values. Computation times reported in Bar-Gera (2002) for the Chicago regional network are 84 and 354 minutes for relative gaps of 1E-3 and 1E-5 which are more or less equivalent to AEC values of 1E-2 and 1E-4, respectively, for which the presently reported computing times are 52 and 224 minutes. This reduction of approximately 40% in CPU times is due to newer computers.

The main finding from Table 9 is that even at relatively modest levels of precision of AEC=1E-2, TAPAS is about five times faster than FW. As the level of precision increases, the advantage of TAPAS becomes more substantial. Results for the Chicago regional network and the Philadelphia network are presented in greater detail in Fig. 10, further supporting the findings from Table 9. The issue of persistence, discussed in Section 6.5, appears in the Philadelphia network when AEC is approximately 1E-7, and causes a substantial slow down of convergence. Several experimental variants of the TAPAS algorithm enable faster resolution of the persistency issue in this case, but at the expense of degradation in performance for other test networks. Better treatment for persistent behaviors appears to be a very interesting challenge for future research.

Table 9
Computation times by convergence level.

Network	Algorithm	Average excess cost						
		10^{-2}	10^{-3}	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
Sioux falls	FW	0.1 s	0.7 s	8.3 s				
	OBA	0.0 s	0.0 s	0.1 s	0.3 s	0.4 s	0.6 s	0.8 s
	TAPAS	0.0 s	0.0 s	0.1 s	0.1 s	0.1 s	0.2 s	0.2 s
Chicago sketch	FW	2.3 s	9.5 s	83.2 s				
	OBA	8.1 s	10.6 s	16.8 s	28.6 s	48.1 s	69.3 s	91.7 s
	TAPAS	2.2 s	3.3 s	3.3 s	5.9 s	7.3 s	7.3 s	8.7 s
Berlin center	FW	2.1 m	14.3 m	135.0 m				
	OBA	7.1 m	7.9 m	9.3 m	12.1 m	23.2 m	25.5 m	27.3 m
	TAPAS	1.4 m	1.4 m	2.9 m	2.9 m	2.9 m	2.9 m	2.9 m
Chicago regional	FW	43.8 m	342.5 m					
	OBA	50.7 m	130.7 m	208.7 m				
	TAPAS	9.0 m	9.0 m	13.1 m	21.5 m	30.2 m	37.7 m	49.3 m
Philadelphia	FW	19.4 m	149.5 m					
	OBA	36.1 m	48.2 m	65.5 m	279.9 m			
	TAPAS	4.1 m	7.2 m	7.2 m	17.9 m	191.0 m	191.0 m	194.3 m

Note: s, seconds; m, minutes.

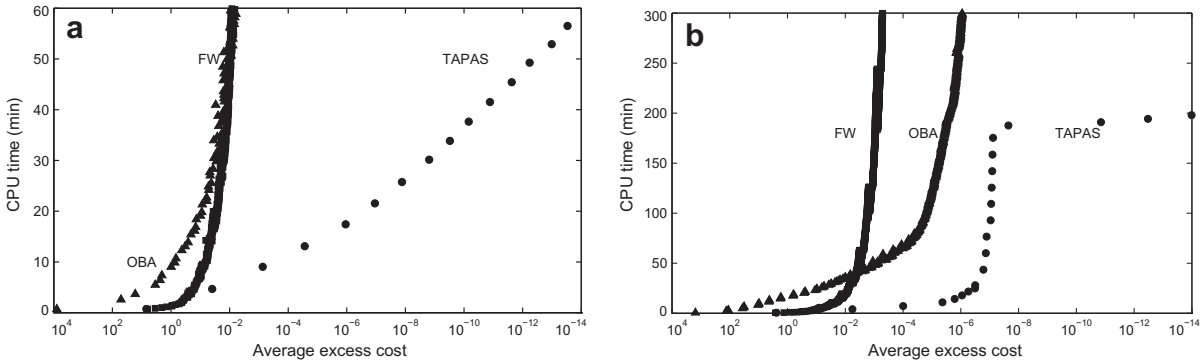

Fig. 10. CPU time required to achieve desired levels of convergence. (a) Chicago regional; (b) Philadelphia.

Indirect comparison with other methods, given that computations were performed on different machines, and possibly with other influencing factors, should be considered with caution. The following reports are for the Chicago regional network, and convergence is measured by "relative gap." Several definitions of relative gap appear in various documents, which is why this term is not used as a convergence measure in this paper. Under most definitions the relative gap for the Chicago

Table 10
Reduced cost distribution by iteration for the Chicago regional network.

Iter	10	20	30	40	50	60	70	80
AEC	1.4E−04	2.8E−07	3.6E−09	6.9E−11	1.2E−12	2.9E−14	1.4E−14	1.3E−14
Reduced cost								
(A) Used origin-link combinations								
$< 2^{-60}$	165	2140	5896	6347	7504	37,314	121,917	142975
$(2^{-60}, 2^{-55}]$	0	0	0	0	0	0	0	0
$(2^{-55}, 2^{-50}]$	0	0	2	7	2	51	237	283
$(2^{-50}, 2^{-45}]$	549	5239	13,867	14,422	19,775	112,102	304,687	341,428
$(2^{-45}, 2^{-40}]$	159	2527	5779	11,772	115,984	417,481	267,677	211,004
$(2^{-40}, 2^{-35}]$	81	2400	4842	69,194	369,931	126,155	1103	4
$(2^{-35}, 12^{-30}]$	243	3539	40,724	330,415	176,489	2396	2	0
$(2^{-30}, 2^{-25}]$	365	5934	243,132	249,695	5447	4	0	0
$(2^{-25}, 2^{-20}]$	736	38,288	339,703	12,360	7	0	0	0
$(2^{-20}, 2^{-15}]$	2950	250,029	36,395	13	0	0	0	0
$(2^{-15}, 2^{-10}]$	14,159	339,865	34	0	0	0	0	0
$(2^{-10}, 2^{-5}]$	111,347	13,566	0	0	0	0	0	0
$(2^{-5}, 2^{0}]$	819,966	238	0	0	0	0	0	0
$(2^{0}, 2^{5}]$	352,588	0	0	0	0	0	0	0
$> 2^{5}$	0	0	0	0	0	0	0	0
(B) Unused origin-link combinations								
$< 2^{-60}$	1,258,345	12,371	863	151	40	17	2	1
$(2^{-60}, 2^{-55}]$	0	0	0	0	0	0	0	0
$(2^{-55}, 2^{-50}]$	1142	6	0	0	0	0	0	0
$(2^{-50}, 2^{-45}]$	44,244	405	22	6	3	3	1	129
$(2^{-45}, 2^{-40}]$	209	173	24	4	8	7	2	1008
$(2^{-40}, 2^{-35}]$	67	65	72	123	186	93	1206	0
$(2^{-35}, 2^{-30}]$	100	70	246	520	249	1216	0	0
$(2^{-30}, 2^{-25}]$	378	347	1274	748	864	135	135	135
$(2^{-25}, 2^{-20}]$	698	1210	2809	650	515	15	15	15
$(2^{-20}, 2^{-15}]$	2806	8884	967	879	198	198	198	198
$(2^{-15}, 2^{-10}]$	16,934	24,438	7555	6451	6481	6481	6481	6481
$(2^{-10}, 2^{-5}]$	163,786	159,651	158,201	158,337	158,343	158,345	158,348	158,348
$(2^{-5}, 2^{0}]$	4,684,886	5,821,319	5,869,858	5,875,230	5,876,170	5,876,437	5,876,549	5,876,557
$(2^{0}, 2^{5}]$	12,233,136	13,929,716	14,015,663	14,024,871	14,026,959	14,027,535	14,027,799	14,027,813
$> 2^{5}$	0	0	0	0	0	0	0	0

Table 11
TAPAS memory requirements for different test networks.

Network	Origin-based link flows	PAS set	Operating system report
Sioux falls	14.59 kB	12.96 kB	
Chicago sketch	187.27 MB	0.10 MB	
Berlin center	8.71 MB	0.15 MB	204.30 MB
Chicago regional	532.85 MB	12.29 MB	607.51 MB
Philadelphia	465.43 MB	10.62 MB	537.13 MB

regional network equals approximately to 0.05 times the AEC. Dial (2006) reports computation times of 9, 15 and 29 min for relative gaps of 1E-2, 1E-3 and 1E-4, respectively. The plots presented by Florian et al. (2009) suggest computation times of approximately 10, 20 and 50 minutes for relative gaps of 1E-3, 1E-4 and 1E-5, respectively. The plots presented by Gentile (2009) suggest computation times of approximately 5, 10 and 15 minutes for relative gaps of 1E-4, 1E-5 and 1E-6, respectively. This rough comparison shows that the ranges are similar, and that a very prudently structured experimental framework is needed to properly rank these methods.

The progress of solution consistency for the Chicago regional network is presented in Table 10, showing equivalent information to Table 4, that is the distributions of reduced costs for used and unused origin-link combinations (OLCs). Each column represents a single solution obtained after the indicated number of iterations. To improve clarity used OLCs that are part of the minimum cost tree are not shown, since in every iteration there are about 2E+7 of those. For the most converged solution, at iteration 80 with AEC=1.3E-14, all used OLCs have very low reduced costs, not exceeding 2^{-35}, while the vast majority of unused OLCs have reduced costs above 2^{-25}. This clear separation between the two groups suggest a fairly consistent solution. There are still more than 1000 unused OLCs with low reduced costs. The reader can detect that reduced costs of these OLCs decrease from iteration 70 to iteration 80, suggesting that these OLCs should in fact be used. The current algorithm does not contain the ability to make that observation, and does not utilize these OLCs. In that respect the solution is considered sub-consistent, and by (18) its sub-consistency level is evaluated as 0.001. It is clear that after 10 iterations when AEC=1.4E-4 the solution is not consistent at all: there are more than 1E+6 unused OLCs with reduced cost less than 2^{-60} (which is practically zero), and at the same time there are 1E+6 used OLCs with reduced costs above 2^{-10}. However, ten more iterations bring convergence to AEC=2.8E-7 and separation between used and unused OLCs begins to appear, as the main bulk of approximately 5E+5 used OLCs has reduced costs in the range of 2^{-20} to 2^{-10}, while the number of unused OLCs with reduced costs below 2^{-10} is about 5E+4, which is one order of magnitude smaller.

Improvement of consistency with convergence occurs in all other networks as well. In the following consistency evaluation of the most converged solution for each network is presented. The Philadelphia solution is also sub-consistent at level 4.2E-5, but the solutions for Chicago sketch and Berlin center are both super-consistent at levels 7.4E+8 and 9.0E+2, respectively. The congestion level in the Sioux-Falls network is relatively high; and as a result identifying a consistent solution is not easy. Seven unused OLCs with low reduced cost in the solution identified by TAPAS translate to sub-consistency level of 0.07. So while opportunity for improvements still exists, overall the solutions produced by TAPAS are considered reasonably consistent.

After 100 regular iterations no additional improvement to convergence in the Chicago regional network seems possible, probably due to limitations of double precision arithmetic. At that stage the maximum deviation from proportionality is 0.19 vph. Eleven additional proportionality iterations consume 2 minutes (117 seconds) of CPU time and reduce the maximum deviation from proportionality to 1.8E-10 vph, which appears to be the limit. The reduction in MDFP is approximately a factor of five in every iteration on this large network with many interactions; compared to 13% reduction for the naive method in the simple example presented in Section 7, this offers clear evidence that the quadratic approximation approach works very well.

The main memory requirement in the current implementation is for storing the array of origin-based link flows as double precision floats, as shown in Table 11. The table also shows estimated memory requirements for PAS information, based on data structure statistics, which is substantially lower than the memory needed for origin-based link flows. For larger networks the actual memory used is reported by the operating system, demonstrating that other memory requirements are not critical. The memory requirements for networks similar in size to those evaluated here can be accommodated by most computers presently available. Storing the array of origin-based link flows for certain larger networks that are currently used in practice could be problematic. Possibilities for algorithm adjustments to improve memory requirements are a subject for future exploration.

9. Conclusions

The static user-equilibrium (UE) traffic assignment model is used extensively in current travel forecasting practice, and seems likely to continue to be useful for the foreseeable future. A new algorithmic approach to solving this classic model has been presented, focusing on pairs of alternative segments as the key building block in the equilibration process. Numerical results for an implementation of this approach demonstrate the ability to find highly converged solutions in relatively short computing times.

Particular attention is given to route flows, which are often used in practice even though they are not uniquely determined by the UE condition. A simple and reasonable condition of proportionality is proposed to identify favorable route flow solutions. Proportionality is nearly equivalent to entropy maximization, which ensures unique and stable route flow solutions. By its definition the proportionality condition applies to pairs of alternative segments, and thus the proposed algorithm enables one to address proportionality rather effectively. This combination of quick-precision and proportionality is greatly needed in practice.

The algorithmic approach presented here can be further explored in many different ways, such as using the local nature of PASs for parallel computing, reducing memory requirements, and addressing convergence persistence.

Acknowledgments

The author expresses sincere gratitude to Miron Livni, Greg Tahin and their colleagues in the Condor team at the University of Wisconsin-Madison for providing access to the grid computing environment on which all computational experiments were conducted. Comments and suggestions from David Boyce, Marco Nie and Dirck Van Vliet on previous versions of this paper are greatly appreciated. The research was conducted in part during the author's sabbatical, hosted by the school of Civil Engineering at Purdue University.

References

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 1993. Networks Flows. Prentice-Hall, Upper Saddle River, NJ.
Akamatsu, T., 1997. Decomposition of path choice entropy in general transport networks. Transportation Science 31 (4), 349–362.
Babonneau, B., du Merle, O., Vial, J.P., 2006. Solving large scale linear multicommodity flow problems with an active set strategy and proximal-ACCPM. Operations Research 54 (1), 184–197.
Bar-Gera, H., 2001. Transportation Network Test Problems. <www.bgu.ac.il/~bargera/tntp>.
Bar-Gera, H., 2002. Origin-based algorithm for the traffic assignment problem. Transportation Science 36 (4), 398–417.
Bar-Gera, H., 2006. Primal method for determining the most likely route flows in large road networks. Transportation Science 40 (3), 269–286. doi:10.1287/trsc.1050.0142.
Bar-Gera, H., Boyce, D., 1999. Route flow entropy maximization in origin-based traffic assignment. In: Ceder, A. (Ed.), Proceedings of the 14th International Symposium on Transportation and Traffic Theory, Jerusalem, Israel, 1999. Elsevier Science, Oxford, UK, pp. 397–415.
Bar-Gera, H., Luzon, A., 2007a. Differences among route flow solutions for the user-equilibrium traffic assignment problem. Journal of Transportation Engineering 133 (4), 232–239.
Bar-Gera, H., Luzon, A., 2007b. Non-unique route flow solutions for user-equilibrium assignments. Traffic Engineering and Control 48 (9), 408–412.
Beckmann, M., McGuire, C.B., Winston, C.B., 1956. Studies in the Economics of Transportation. Yale University Press, New Haven, CT.
Bell, M., Iida, Y., 1997. Transportation Network Analysis. John Wiley and Sons.
Boyce, D., Ralevic-Dekic, B., Bar-Gera, H., 2004. Convergence of traffic assignment: how much is enough? Journal of Transportation Engineering 130 (1), 49–55.
Busaker, R.G., Gowen, T.L., 1961. A procedure for determining a family of minimal cost network flow patterns. O.R.O. Technical Report 15, Johns Hopkins University.
Busaker, R.G., Saaty, T.L., 1965. Finite Graphs and Networks. McGraw-Hill, New York.
Condor homepage. <http://www.cs.wisc.edu/condor> (accessed 04.05.09).
Dafermos, S.C., Sparrow, F.T., 1969. The traffic assignment problem for a general network. Journal of Research of the National Bureau of Standards 73B, 91–118.
Daganzo, C.F., Sheffi, Y., 1977. On stochastic models of traffic assignment. Transportation Science 11 (3), 253–274.
Dantzig, G., 1963. Linear Programming and Extensions. Princeton University Press, Princeton, NJ.
Dial, R.B., 1971. A probabilistic multipath traffic assignment model which obviates path enumeration. Transportation Research 5 (2), 83–111.
Dial, R.B., 1999. Accurate traffic equilibrium: how to bobtail Frank-Wolfe. Technical Report. Volpe National Transportation Research Center, Cambridge, MA.
Dial, R.B., 2006. A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. Transportation Research Part B 40 (10), 917–936.
Florian, M., Constantin, I., Florian, D., 2009. A new look at the projected gradient method for equilibrium assignment. Transportation Research Record No. 2090, pp. 10–16.
Gallager, R.G., 1977. Loops in multicommodity flows. Proceedings of the Tenth IEEE Conference on Decision and Control, New Orleans, LA, 819–825.
Gentile, G., 2009. Linear user cost equilibrium: a new algorithm for traffic assignment, working paper.
Janson, B.N., 1993. Most likely origin–destination link uses from equilibrium assignment. Transportation Research Part B 27 (5), 333–350.
Jayakrishnan, R., Tsai, W.K., Prashker, J.N., Rajadhyaksha, S., 1994. A faster path-based algorithm for traffic assignment. Transportation Research Record 1443, 75–83.
Larsson, T., Lundgren, J., Patriksson, M., Rydergren, C. 1998. Most likely traffic equilibrium route flows - analysis and computation, In: C. Rydergren, (Eds.), Optimization Methods for Analysis of Transportation Networks. Theses No. 702, Linkopings Universitet, Sweden.
Larsson, T., Patriksson, M., 1992. Simplicial decomposition with disaggregated representation for the traffic assignment problem. Transportation Science 26 (1), 4–17.
LeBlanc, L.J., Morlok, E.K., Pierskalla, W.P., 1975. An efficient approach to solving the road network equilibrium traffic assignment problem. Transportation Research 9 (5), 309–318.
Lu, S., 2008. Sensitivity of static traffic user equilibria with perturbations in arc cost function and travel demand. Transportation Science 42 (1), 105–123.
Lu, S., and Nie, Y. 2009. Stability of user-equilibrium route flow solutions for the traffic assignment problem. Transportation Research Part B, in press, doi:10.1016/j.trb.2009.09.003.
Merchant, D.K., Nemhauser, G.L., 1978. Model and an algorithm for the dynamic traffic assignment problems. Transportation Science 12 (3), 183–199.
Nie, Y. 2007. A Note on Bar-Gera's Algorithm for the Origin-Based Traffic Assignment Problem, submitted for publication in Transportaton Science.
Nie, Y., 2010. A class of bush-based algorithms for the traffic assignment problem. Transportation Research Part B 44 (1), 73–89.
Patriksson, M., 1994. The Traffic Assignment Problem – Models and Methods. VSP, Utrecht, Netherlands.
Peeta, S., Ziliaskopoulos, A., 2001. Foundations of dynamic traffic assignment: the past, the present and the future. Networks and Spatial Economics 1 (3-4), 233–266.
Rockafellar, R.T., 1984. Network Flows and Monotropic Optimization. John Wiley and Sons, New York.
Rossi, T.F., McNeil, S., Hendrickson, C., 1989. Entropy model for consistent impact fee assessment. Journal of Urban Planning and Development/ASCE 115 (2), 51–63.

Schneur, R.R. 1991. Scaling algorithms for multicommodity flow problems and network flow problems with side constraints. Ph.D. Thesis. Massachusetts Institute of Technology, Cambridge, MA.
Schneur, R.R., Orlin, J.B., 1998. A scaling algorithm for multicommodity flow problems. Operations Research 46 (2), 231–246.
Slavin, H., Brandon, J., Rabinowicz, A., Sundaram, S., 2009. Application of accelerated user equilibrium traffic assignments to regional planning models, Paper prepared for presentation at the 12th Transportation Research Board National Transportation Planning Applications Conference, Houston, Texas.
SMARTEST, 2000, Final report. European Commission, 4th Framework Programme, Transport RTD Programme, Contract No: RO-97-SC.1059.