Balanced solution of multiclass deterministic assignment through the LUCE algorithm

Guido Gentile*

1. Introduction

Although deterministic traffic assignment is a rather mature subject in transport modelling, to find an equilibrium on real networks is still a difficult problem to solve. In practice, none of the classical algorithms, link-based (e.g. LeBlanc et al., 1975; Florian et al. 1987) or path-based (e.g. Larsson and Patriksson, 1992; Jayakrishnan et al., 1994), converges at the solution in a reasonable time precisely enough to allow consistent comparisons between design scenarios. Indeed, apparently small errors due to early termination of the iterative solving procedure do not allow the analyst to appreciate the real differences among equilibria and may lead to false conclusions in relevant projects, thus vanishing any modelling effort: without precision in the calculation, all attempts to enhance the accuracy of the model, i.e. its ability to reproduce reality, may be frustrated (Boyce et al., 2004).
More recently, a new generation of bush-based algorithms has made possible the precise solution of the traffic assignment problem for very large networks, thus allowing an effective comparison of design scenarios (Bar-Gera, 2002; Dial, 2006; Gentile, 2009). Lately, also path-based algorithms improved considerably their convergence performance (Florian, 2009); but given the non uniqueness of path flows, they tend to provide very poor solutions with only few routes loaded per o-d pair, compared to the many more that have the same equilibrium cost.
It is well known that, while the solution in terms of total link volumes is unique under standard assumptions, the solution in terms of class/destination link flows in not, like for path flows. This arises the problem of selecting a most-likely solution among the many existing, such that, for example, the splitting rates for two identical class/destinations are equal, which corresponds to seeking a minimum entropy (Bar-Gera, 2006). Flow balancing is a relevant issue for a set of post processing tools, such as the critical link analysis or the o-d estimation from traffic counts, and in any case where the path flows are a desired output of the simulation, besides link volumes.
In this paper a new formulation of the multiclass user equilibrium with determinis-

* Dipartimento di Idraulica Trasporti e Strade, “Sapienza” Università di Roma

tic route choice is proposed, which contrary to the classical formulations of the problem (Dafermos, 1972; Sheffi, 1985; Van Vliet et al., 1986; Toint and Wynter, 1996), and similarly to its stochastic version (Bifulco, 1993), has a unique solution also in terms of class/destination link flows, thus overcoming from a different perspective the issue of flow balancing. This result is attainted at the small modelling price of introducing an additional positive term in the non-separable function of arc costs, which can be made as small as wanted, compatibly with numerical issues that could deteriorate the effectiveness of the approach.

To solve the proposed formulation the Linear User Cost Equilibrium (LUCE) algorithm (Gentile and Noekel, 2009) has been suitably extended, thus allowing to obtain an extremely precise estimate of the unique solution (e.g. a relative gap of 10E-8, that is considered enough for any application) in few minutes and iterations, also on large networks with many user classes.

The main idea is to seek at every node a user equilibrium (Wardrop, 1952) for the local route choice of travellers belonging to a same class and directed toward a same destination among the links of its forward star. The cost function associated to each one of these travel alternatives expresses the average impedance to reach the destination by continuing the trip with that link, linearized at the current flow pattern.

To allow recursive computations, only the links that belong to the current bush are included in the local choice set – a bush is an acyclic sub-graph that connects each origin to the destination at hand. The unique solution to the resulting linear user cost equilibrium in terms of class/destination flows, recursively applied at each node of the bush in a topological order, provides a descent direction with respect to the line-integral objective function of the original assignment problem. This postulate is proved with reference to any given bush of the class/destination at hand. To ensure the convergence of the procedure towards an equilibrium where all paths of the graph satisfy Wardrop's conditions, the current bush is changed before finding the search direction, by trying whether is possible (the resulting sub-graph must still be acyclic) to exclude unused links that bring away from the destination and to include links that improve shortest paths for that class.

The descent direction obtained for each class/destination is then exploited by rotation in a feasible direction method, where the line search follows the Armijo rule.

At each iteration, the proposed algorithm requires no shortest tree but two visits of the bush links for every class/destination, that is equal to the complexity of the STOCH single pass procedure (Dial, 1971) for the Logit network loading.

Contrary to the classical all-or-nothing assignment to shortest paths, the network loading map resulting from the application of the LUCE algorithm is a one-to-one function, that combined with the arc cost function yields a well-defined fixed point operator, thus offering both computational and theoretical advantages. Moreover, the LUCE network loading is performed consistently with the splitting rates resulting from the local equilibrium problems, thus assigning the demand flow to a whole bush of paths, which prevents the typical Frank Wolf zigzagging near the solution, and avoiding explicit path enumeration, which permits a fast computation and huge memory savings.

In the reminder of the paper we will present the mathematical formulation of the multiclass assignment problem and the main aspects of the LUCE algorithm to find a descent direction for this case, while the assignment algorithm, its proof of convergence and its numerical testing are addressed in our previous works.

2. Mathematical formulation

In this section we formulate the multiclass traffic assignment as a deterministic user equilibrium. The aim of the problem is then to find a cost-flow pattern which satisfies Wardrop's conditions: rational and perfectly informed users choose only *shortest* paths, in terms of costs, to travel on the network from their origin to their destination (the demand model); path costs are the sum of their link costs, which in turn are class specific and depend not only on the class flow but also on the link volume, thus yielding a non-separable function (the supply model).
The transport network is represented through a directed graph $G = (N, A)$, where N is the set of the nodes and $A \subseteq N\times N$ is the set of the links; $Z \subseteq N$ is the set of the zone centroids. The forward and backward stars of the generic node $i\in N$ are denoted, respectively: $FS(i) = \{j\in N: ij\in A\}$ and $BS(i) = \{j\in N: ji\in A\}$.
The demand is segmented into a set of user classes U. Without loss of generality, all users of the generic class $u\in U$ are directed toward a same destination $d(u)\in Z$, since this allows to save one index in the notation. Moreover, they share as usual the same cost attributes and perception, such as tolls and value of time, and make-up an homogeneous flow, with the same vehicle type and occupancy.

For flows and costs we adopt the following notation:

f_{ij}^{u} flow on link $ij\in A$ of class $u\in U$ directed to destination $d(u)\in Z$, generic element of the $(|A|\cdot|U|\times 1)$ vector $\mathbf{f}$;

v_{ij} volume on link $ij\in A$, that is the total flow of all classes

$v_{ij} = \sum_{u\in U} f_{ij}^{u}$; (1)

Q_{o}^{u} demand flow of class $u\in U$ between origin $o\in Z$ and destination $d(u)\in Z$;

c_{ij}^{u} cost of link $ij\in A$ for users of class $u\in U$ directed to destination $d(u)\in Z$, generic element of the $(|A|\cdot|U|\times 1)$ vector $\mathbf{c}$;

$\mathrm{c}(\mathbf{f})$ non-separable arc cost function

$\mathbf{c} = \mathrm{c}(\mathbf{f})$; (2)

s_{ij} impedance of link $ij\in A$;

$\mathrm{s}_{ij}(v_{ij})$ separable impedance function of link $ij\in A$

$s_{ij} = \mathrm{s}_{ij}(v_{ij})$; (3)

g_{ij} impedance derivative of link $ij\in A$

$g_{ij} = \partial \mathrm{s}_{ij}(v_{ij}) / \partial v_{ij}$. (4)

In the case of fixed demand and symmetric arc cost functions, the deterministic user equilibrium with multiple classes can be formalized through the following non-linear optimization problem, which involves no path variables:

find $\mathbf{f}^*\in\Omega^* \equiv \arg\min \Phi(\mathbf{f}) = \int_{\mathbf{0}}^{\mathbf{f}} c(\mathbf{x})^T\cdot d\mathbf{x}$ (5.1)

subject to:

$0 \le f_{ij}^{\,u} \le \sum_{o\in Z} Q_o^{d(u)}$, $\forall ij\in A$, $\forall u\in U$; (5.2)

$\sum_{j\in FS(i)} f_{ij}^{\,u} - \sum_{j\in BS(i)} f_{ji}^{\,u} = \{0$, if $i\notin Z$;
$-\sum_{o\in Z} Q_o^{d(u)}$, if $i = d(u)$;
$Q_i^{d(u)}$, otherwise$\}$, $\forall i\in N$, $\forall u\in U$. (5.3)

This particular formulation in terms of class link flows is different from Beckman's mathematical program (Beckman et al., 1956) in terms of link volumes, where the consistency of the link flows with the given travel demand is ensured by explicitly introducing the path flows. Indeed, the conservation of class flows at nodes, stated by constraints (5.3), assures implicitly that each trip traverses a path connecting its origin to its destination: the demand flows of a given class $u\in U$ spring from the various origins and circulate on the network until they eventually dwell into their common destination $d(u)$. The upper bound defined by constraint (5.2) can be violated only if some user passes through the same link for more than one time, which will be excluded, given that paths with cycles cannot be shortest in the case of positive link costs.

To characterize problem (5) we introduce the following assumptions:
H1. additive path costs, i.e. the cost associated by users of a certain class to a given path on the network is the sum of the link costs that belong to it;
H2. fixed and non-negative demand, i.e. $Q_o^{\,u} \ge 0$;
H3. centroids connected among each other, i.e. there exist a path on G to travel from every origin to any destination.
The link impedance function $s_{ij}(v_{ij})$ is assumed to be, as usual:
H4. separable, meaning that the impedance of a link depends only on its volume;
H5. continuous;
H6. strictly monotone increasing;
H7. nonnegative.
For the development of the LUCE algorithm, the link impedance function is further assumed to be:
H8. continuously differentiable;
H9. with a strictly positive derivative, i.e. $g_{ij} > 0$ also at a null volume;
H10. strictly positive, i.e. $s_{ij} > 0$.
For the definition of our multiclass assignment model, the arc cost function is related to the corresponding link impedance function, and is assumed to be:
H11. spatially separable, i.e. the congestion occurs only among the flows that are travelling on the same link;
H12. with a symmetric Jacobian.
In particular, we will consider the form:

$c_{ij}^{\,u}(\mathbf{f}) = s_{ij}(v_{ij}) + \chi_{ij} \cdot f_{ij}^{\,u} + \sigma_{ij}^{\,u}$, (6)

whose Jacobian $\nabla c(\mathbf{f})$ is a symmetric matrix with diagonal blocks, each one relative to a single link $ij\in A$; the generic elements for classes $u\in U$ and $z\in U$ are:

$\partial c_{ij}^{\,u}(\mathbf{f}) / \partial f_{ij}^{\,u} = \partial s_{ij}(v_{ij}) / \partial v_{ij} + \chi_{ij} = g_{ij} + \chi_{ij}$, $\partial c_{ij}^{\,u}(\mathbf{f}_{ij}) / \partial f_{ij}^{\,z} = \partial s_{ij}(v_{ij}) / \partial v_{ij} = g_{ij}$. (7)

Each block can be reduced to a matrix of ones with a "reinforced" diagonal pre-multiplied by a positive coefficient, which is positive definite, given that: $g_{ij} > 0$ and $\chi_{ij} > 0$. Therefore, the entire Jacobian $\nabla c(\mathbf{f})$ is also positive definite.

For example, the following specification of (6) can be adopted:

$$\chi_{ij} = \varepsilon \cdot s_{ij}(0) / \Phi_{ij} \,, \tag{8}$$

$$\sigma_{ij}^{u} = m_{ij} / \gamma_u \,, \tag{9}$$

where Φ_{ij} , $s_{ij}(0)$ and m_{ij} are, respectively, the capacity, the free flow impedance and the toll of link $ij\in A$, γ_u is the value of time (in this case impedances and costs are measured in terms of times) while ε is a small positive number (e.g. $\varepsilon = 10^{-4}$).
Note that the link volume is a straight sum of the class link flows, without weights, since this ensures that (6) has a symmetric Jacobian. If the vehicles used by the different classes have a different impact on the joint congestion, here expressed by the impedance function, then the demand flows shall be consistently scaled by an equivalency coefficient. In these case, however, $\mathbf{c}^{T}\cdot\mathbf{f}$ is not equal to the total cost of the network. Analogously, the link impedance is perceived by all user classes in the same way.
For practical applications we can relax hypothesis H6 and H7, allowing the impedance function on some links to be constant and/or null, but assuming that there is no more than one path connecting any two nodes on G constituted exclusively by such links. Furthermore, hypothesis H8 can be relaxed by assuming that the differentiability holds almost everywhere, thus allowing to consider piecewise linear functions. Finally, hypothesis H10 can be removed, thus allowing to consider the most used polynomial functions, such as the BPR.

Because the arc cost function is spatially separable, the line integral (5.1) becomes:

$$\Phi(\mathbf{f}) = {}_{\mathbf{0}}\int^{\mathbf{f}} c(\mathbf{x})^{T}\cdot d\mathbf{x} = \sum_{ij\in A} \sum_{u\in U} {}_{0}\int^{f_{ij}^{u}} c_{ij}^{u}(\, \dots f_{ij}^{z<u} \dots , x_{ij}^{u} , \dots 0 \dots)\cdot dx_{ij}^{u} \,, \tag{10}$$

which based on (6) reduced to:

$$\Phi(\mathbf{f}) = \sum_{ij\in A} {}_{0}\int^{v_{ij}} s_{ij}(x_{ij})\cdot dx_{ij} + \sum_{ij\in A} \sum_{u\in U} ½ \cdot \chi_{ij} \cdot (\, f_{ij}^{u})^{2} + \sum_{ij\in A} \sum_{u\in U} \sigma_{ij}^{u} \cdot f_{ij}^{u} \,. \tag{11}$$

The gradient and the Hessian of the objective function (11) at $\mathbf{f}\in\Omega$ are:

$$\partial\Phi(\mathbf{f}) / \partial f_{ij}^{u} = s_{ij}(v_{ij}) + \chi_{ij} \cdot f_{ij}^{u} + \sigma_{ij}^{u} = c_{ij}^{u}(\mathbf{f}) \text{ , in compact form } \nabla\Phi(\mathbf{f}) = c(\mathbf{f}) \,, \tag{12}$$

$$\nabla^{2}\Phi(\mathbf{f}) = \nabla c(\mathbf{f}) \,. \tag{13}$$

The line-integral (11) is continuously differentiable, as a consequence of hypothesis H5, and strictly convex (as any level set) in terms of class flows, as a consequence of hypothesis H6 (the integral of a monotone increasing function is convex). Indeed, the Hessian of the objective function is positive definite. Note that strict convexity is due to the second term, which is quadratic; otherwise strict convexity for the classical sum-integral objective function holds only in terms of link volumes, since these are linear (non strictly convex) functions of the class flows, but not in terms of class flows.
The linear constraints (5.2)-(5.3) make up a feasible set of class link flows Ω that is nonempty, as a consequence of hypothesis H3 (one feasible point can be obtained by loading each demand flow on the path whose existence is explicitly assumed),

compact and convex. Therefore, the continuity of the objective function, ensured by hypothesis H5, guarantees the existence of at least one solution of the optimization problem. Moreover, given the strict convexity of the objective function, there is only one global minimum **f***, which represents with the only stationary point.
Therefore, problem (5) admits a unique solution in terms of class link flows, as well as in terms of link volumes and cost pattern. Moreover, the uniqueness of **f*** allows to choose a balanced solution in terms of path flows, although they remain not univocal. For this purpose, it is useful to cast the assignment model into the framework of sequential route choices, where the probability P_k^u for class $u \in U$ of using a certain path k is given by the product among the flow proportions y_{ij}^u (formally introduced later on) of its links $A(k) \subseteq A$:

$$P_k^u = \prod_{ij \in A(k)} y_{ij}^u . \quad (14)$$

Theorem 1 – equivalence between stationary point and equilibrium
The stationary point of problem (5) is a user equilibrium, and vice versa.
Proof.
Problem (5) consists in minimizing the continuously differentiable strictly convex function $\Phi(\mathbf{f})$, subject to linear constraints that make up the nonempty, compact and convex set Ω. In this case, the optimality condition for the solution **f***, i.e. the necessary and sufficient condition for the one local and global minimum, are expressed by the following variational inequality (see, for example, Bertsekas, 1995, page 176):

$$\nabla\Phi(\mathbf{f}^*)^T \cdot (\mathbf{f} - \mathbf{f}^*) \geq 0 , \ \forall \mathbf{f} \in \Omega , \quad (15)$$

stating that at point **f*** any feasible direction must have a nonnegative directional derivative.
Based on (12), it is easy to observe that (15) coincides with Wardrop's equilibrium condition; indeed, at **f*** no other feasible flow pattern can yield lower total costs, implying that each user is travelling along a shortest path (in the case of scaled demand the same condition must be considered separately for each class):

$$c(\mathbf{f}^*)^T \cdot \mathbf{f} \geq c(\mathbf{f}^*)^T \cdot \mathbf{f}^* , \ \forall \mathbf{f} \in \Omega . \quad (16)$$

On this basis, any stationary point, and therefore any solution, to problem (5) is a user equilibrium, and vice versa. This also implies that the equilibrium is unique. ■

To solve this kind of convex optimization problem we can rely on the *feasible direction method* (Bertsekas, 1995, pages 192). The method starts with a feasible point $\mathbf{f}_1 \in \Omega$ and generates a sequence of feasible vectors $\{\mathbf{f}_k\}$ for which the objective function does not increase:

$$\Phi(\mathbf{f}_{k+1}) \leq \Phi(\mathbf{f}_k) , \ k = 1, 2, \ldots . \quad (17)$$

At each k the new iterate is found along a search direction $\mathbf{e}_k - \mathbf{f}_k$ based on a new feasible point $\mathbf{e}_k \in \Omega$ by making a step $\alpha_k \in [0,1]$, according to the algorithm:

$$\mathbf{f}_{k+1} = \mathbf{f}_k + \alpha_k \cdot (\mathbf{e}_k - \mathbf{f}_k) , \ k = 1, 2, \ldots . \quad (18)$$

The feasible direction method then requires a function to provide a search direction and an algorithm to perform the line search.
At each iteration k of the method we consider a specific class, by rotating u among

U. If only efficient links are used, we modify the current bush $B(u)$ to improve the scope of the search. Then, we determine the LUCE search direction $\mathbf{e}_k - \mathbf{f}_k \neq \mathbf{0}$, with $\mathbf{e}_k^u = \omega^u(\mathbf{f}_k)$ and $\mathbf{e}_k^z = \mathbf{f}_k^z$ for each $z \in U$ such that $z \neq u$, based on the current link flows $\mathbf{f}_k \in \Omega$ and the possibly new bush $B(u)$. Since the feasible set Ω is convex, we can finally move the current solution along the segment $\mathbf{f}_k + \alpha_k \cdot (\mathbf{e}_k - \mathbf{f}_k)$ and take a step $\alpha_k \in [0,1]$ such that the objective function Φ is sufficiently lowered.
To this end, we consider the following backtracking line search: determine the largest step $\alpha_k = 0.5^h$, for any non-negative integer h, such that the objective function satisfies, for a fixed $0 < \gamma < 0.5$ (e.g. $\gamma = 10^{-4}$), the well-known Armijo rule

$$\Phi(\mathbf{f}_k + \alpha \cdot (\mathbf{e}_k - \mathbf{f}_k)) \leq \Phi(\mathbf{f}_k) + \gamma \cdot \alpha_k \cdot \nabla\Phi(\mathbf{f}_k)^{\mathrm{T}} \cdot (\mathbf{e}_k - \mathbf{f}_k) \,. \tag{19}$$

Because the objective function is continuously differentiable, by Taylor's theorem a direction is descent (i.e. small steps along it guarantee that Φ is reduced) if and only if its directional derivative is negative; the necessity derives from the convexity of the problem. Based on (12), the direction **e-f** is descent and only if:

$$\mathrm{c}(\mathbf{f}_k)^{\mathrm{T}} \cdot (\mathbf{e}_k - \mathbf{f}_k) < 0 \,. \tag{20}$$

In other words, to decrease the objective function and maintain feasibility we necessarily have to "shift flows" getting a lower total cost with respect to the current cost pattern. This approach to determine a descent direction can also be applied to each o-d pair separately, to each origin, or to the whole network jointly. Here, since the proposed formulation is based on class link flows, the assignment problem will be partitioned by classes.
Based on the above general rule, setting the flow pattern $\mathbf{e}_k$ by means of an all-or-nothing assignment to shortest paths clearly provides a descent direction. If such a direction is adopted for all o-d pairs of the network jointly, and a line search with minimization rule is applied along it, we obtain the classical Frank-Wolfe algorithm. However, at equilibrium each o-d pair typically uses several paths, implying that any descent direction that loads a single path per o-d is intrinsically myopic; in fact it is well known that the FW algorithm tails badly.

3. Linear User Cost Equilibrium

In this section we introduce a new approach to determine a descent direction, which is based on shifts of flows that lower the total cost while loading multiple paths. This is obtained by exploiting the inexpensive information provided by the derivatives of the arc costs with respect to link flows, in the context of local equilibria at nodes based on linear cost approximations, for users of a given class directed toward a same destination.

To grasp immediately the underlying idea, we can refer to the simplest network where one o-d pair with demand Q is connected by two links with arc cost function $c_1(f_1)$ and $c_2(f_2)$, respectively. At the current flow pattern $\mathbf{f}' = (Q/2, Q/2)$, it is $c_1' < c_2'$ (see Figure 1, below), so that an all-or-nothing approach would lead to a descent direction (Q, 0), which is far away from the equilibrium **f*** (black circle in the Figure).

The LUCE approach, instead, is to consider the first order approximations of the arc cost functions at the current flow pattern, i.e. $c_a' + \partial c_a(f_a)/\partial f_a \cdot (f_a - f_a')$, with $a\in\{1,2\}$, and determine a user equilibrium **e** among these lines (white circle in the Figure): this descent direction efficiently approaches the equilibrium **f***, and in most cases can be taken as the new iterate with a step one.

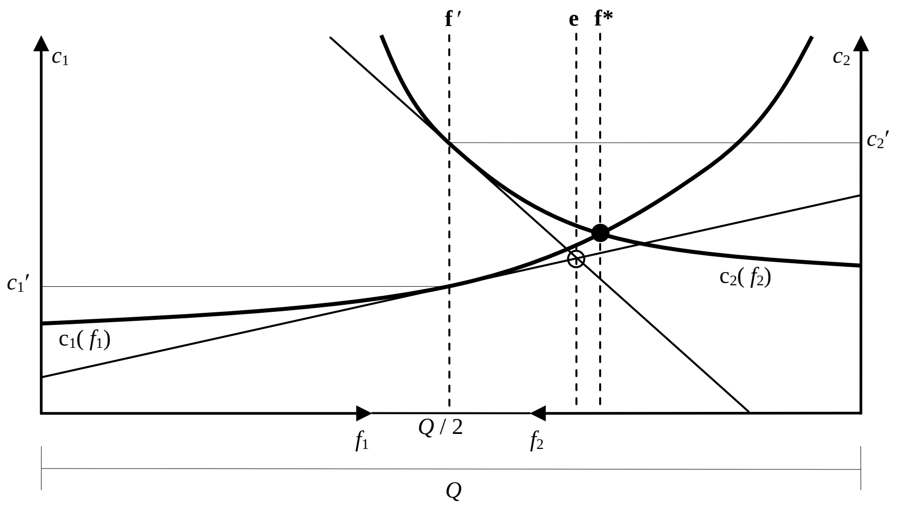

Fig. 1 – Linear Cost User Equilibrium between two paths.

From an optimization point of view, the linearization of the arc cost functions leads to a quadratic approximation of the sum-integral objective function, so that LUCE seems to resemble a gradient projection or a Newton-like method. However, this is true only for the above trivial example, where paths (the travel alternatives) and links (the congested elements) coincide. Indeed, for a general network where path costs are non separable functions of the path flows, the resulting quadratic program would not be very much easier to solve than the original one.

The LUCE approach is then translated, as we will explain in reminder of the paper, in a dynamic programming procedure, where the focus is on local choices at nodes for users directed toward a given destination. In that framework, we will introduce all the necessary assumptions to keep the cost functions associated to the travel alternatives separable. Finally, LUCE is not conceived as an approximation of some nonlinear optimization method but as a behaviourally driven approach.

A (reverse spanning) *bush* (Dial, 2006) rooted at the generic destination $d\in Z$ is a subset of links that make up an acyclic sub-graph of G, comprising one or more paths from each node $i\in N$ to d (if any, since a node could be not connected to d).

Referring to a given class of users $u\in U$, since the class link costs are strictly positive, the shortest paths to $d(u)\in Z$ form a bush. On this base, when seeking a user equilibrium where only shortest paths are utilized, instead of dealing with the entire graph G, we can limit our attention to a current bush for each $u\in U$, denoted $B(u) \subseteq A$, and introduce an updating mechanism to make sure that eventually any shortest path is included into it, i.e., the bush is *optimal*. Indeed, only in this case we can guarantee that an *equilibrated* bush represents also a user equilibrium on the whole graph, for that class of users. Following this approach, no link out of the current bush can carry destination flow:

$f_{ij}^u = 0$, $\forall ij\notin B(u)$. (21)

Because the bush is an acyclic graph, it's possible to define a topological order of its nodes, i.e. a positive integer $O(i, u)$ for every $i\in N$, such that for each link $ij\in B(u)$ the following relation holds true: $O(i, u) < O(j, u)$.

In the following we will focus on the local route choice at a generic node $i\in N$ of users $u\in U$ directed to destination $d(u)\in Z$. To this end, it is useful to define the forward and backward star of the node i on the current bush, denoted, respectively, $FSB(i, u) = \{j\in N: ij\in B(u)\}$ and $BSB(i, u) = \{j\in N: ji\in B(u)\}$.

For the flow pattern we will use the following notation:

Ω_u feasible set of link flows of class $u\in U$ satisfying (5.2), (5.3) and (21) ;

f_i^u current flow of class $u\in U$ leaving node $i\in N$ directed to destination $d(u)\in Z$, that is obtained by summing up the link flows only on the bush forward star, since by construction it is $f_{ij}^u = 0$ for each $j\notin FSB(i, u)$ (to handle non connected nodes we set: $\sum_\varnothing = 0$)

$f_i^u = \sum_{j\in FSB(i, u)} f_{ij}^u$; (22)

y_{ij}^u current flow proportion for class $u\in U$ on link $ij\in A$, defined as

$y_{ij}^u = f_{ij}^u / f_i^u$, if $f_i^u > 0$, (23.1)

$y_{ij}^u = 0$, otherwise ; (23.2)

e_{ij}^u auxiliary flow of class $u\in U$ on link $ij\in A$, for search direction;

e_i^u auxiliary flow of class $u\in U$ leaving node $i\in N$, for search direction;

x_{ij}^u auxiliary flow proportion for class $u\in U$ on link $ij\in A$, for search direction;

$x_{ij}^u = e_{ij}^u / e_i^u$, if $e_i^u > 0$, (24)(24.1)

$x_{ij}^u = 0$, otherwise . (24.2)

For the cost pattern we will use the following notation:

C_i^u average cost for class $u\in U$ to reach destination $d(u)\in Z$ from node $i\in N$;

G_i^u derivative of the average cost for users of class $u\in U$ to reach destination $d(u)\in Z$ from node $i\in N$, with respect to the node flow, i.e. conceptually

$G_i^u = \partial C_i^u / \partial f_i^u$. (25)

The node average cost C_i^u is the expected impedance that a user encounters by travelling from node $i\in N$ to destination $d(u)\in N$. In (26.1) it is defined recursively, as if travellers choose paths accordingly with the current flow proportions; while (26.2) defines the iterate based on the locally best choice, for the case where the node is not used (to handle non connected nodes we set: $\min\{\varnothing\} = \infty$):

$C_i^u = \sum_{j\in FSB(i,\,u)} y_{ij}^u \cdot (c_{ij}^u + C_j^u)$, if $f_i^u > 0$, $C_i^u = 0$, if $i = d(u)u$, (26.1)

$C_i^u = \min\{c_{ij}^u + C_j^u : j\in FSB(i,\,u)\}$, otherwise . (26.2)

Under the assumption that an infinitesimal increment of flow of class $u\in N$ leaving node $i\in N$ directed towards destination $d(u)\in Z$ would diverge accordingly with the current flow proportions y_{ij}^u toward each $j\in FSB(i, u)$, we have:

$G_i^u = \sum_{j\in FSB(i,\,u)} (\, y_{ij}^u)^2 \cdot (g_{ij} + \chi_{ij} + G_j^u)$, if $f_i^u > 0$, $G_i^u = 0$, if $d(u) = u$, (27.1)

$G_i^u = \sum_{j\in FSB(i,\,u)} [C_i^u = c_{ij}^u + C_j^u] \cdot (g_{ij} + \chi_{ij} + G_j^u)$ /

/ $\max\{1, \sum_{j\in FSB(i,\,u)} [C_i^u = c_{ij}^u + C_j^u]\}$, otherwise . (27.2)

In (27.1) the derivatives $\partial(c_{ij}^u + C_j^u) / \partial f_{ij}^u = g_{ij} + \chi_{ij} + G_j^u$ are scaled by the share y_{ij}^u of ∂f_i^u utilizing link ij and then passing through node j, that jointly with the flow proportion involved in the weighted average (26.1) yields the square $(\, y_{ij}^u)^2$.

If otherwise the node is not used, (27.2) expresses the mean of all the derivatives $g_{ij} + \chi_{ij} + G_j^u$ for which link ij represents a locally best choice; for this purpose, we used the notation: [TRUE] = 1, [FALSE] = 0 (the lower bound of 1 in (27.2) is needed to avoid indeterminateness when some node is not connected and $FSB(i, u)$ can be empty).

The node average costs and their derivatives can be computed by processing on the bush each node $i\in N$ in a reverse topological order $O(i, u)$, starting from the destination $d(u)$, where $C_{d(u)}^u = G_{d(u)}^u = 0$. Indeed, the computation of equations (26) and (27) for node $i\in N/\{d(u)\}$ requires, respectively, the node average cost C_j^u and its derivative G_j^u, for each $j\in FSB(i, u)$, which will have been already calculated, given that $ij\in B(u)$ and thus $O(i, u) < O(\, j, u)$.

Now all the elements that are needed to address the linear user cost equilibrium for the e_i^u travellers of class $u\in U$ leaving node $i\in N$ directed to destination $d(u)\in Z$ have been introduced. The available alternatives in this particular assignment problem with deterministic choice model are the links of the bush exiting from node i. To each travel alternative we associate the following cost function:

$v_{ij}^u(e_{ij}^u) = (c_{ij}^u + C_j^u) + (g_{ij} + \chi_{ij} + G_j^u) \cdot (e_{ij}^u - y_{ij}^u \cdot e_i^u)$, (28)

resulting from a linearization at the current flow pattern of the average cost encountered by a user choosing the generic link ij, with $j\in FSB(i, u)$. Note that the latter is in general a function, not only of the flows on link ij, but also of the flows using the other links exiting from node i, since the paths on the bush leaving from the nodes of the forward star can partially overlap and merge before reaching the destination; however, this "non-separable" dependence will be ignored in the following.

The deterministic user equilibrium under consideration, such that each alternative used by class $u\in U$ has the same cost V_i^u of reaching the destination $d(u)$ from node i, can be formulated by the following system of inequalities where the choice set is the bush forward star, in analogy to Wardrop's Principles where the travel alternatives are instead the paths connecting a same o-d couple (Sheffi, 1985):

$e_{ij}^u \cdot (v_{ij}^u(e_{ij}^u) - V_i^u) = 0$, $\forall j\in FSB(i, u)$, (29.1)

$v_{ij}^u(e_{ij}^u) \geq V_i^u$, $\forall j\in FSB(i, u)$, (29.2)

$e_{ij}{}^u \geq 0$, $\forall j \in FSB(i, u)$, (29.3)
$\sum_{j \in FSB(i, u)} e_{ij}{}^u = e_i{}^u$. (29.4)
If the node flow is null, i.e. $e_i{}^u = 0$, the solution to the above problem is trivially: $e_{ij}{}^u = 0$, for each $j \in FSB(i, u)$, while V_i^u can be arbitrarily set equal to 0, if $i = d(u)$, and to $\min\{c_{ij}{}^u + C_j{}^u : j \in FSB(i, u)\}$, otherwise. In the reminder of this section, let us then consider the case where $e_i{}^u > 0$.
Based on the main equation (29.1), either the auxiliary link flow $e_{ij}{}^u$ is null, or the cost $v_{ij}{}^u(e_{ij}{}^u)$ of travelling via $j \in FSB(i, u)$ is equal to the *linear equilibrium cost* V_i^u. Since $e_i{}^u > 0$, then the flow $e_{ij}{}^u$ on some link ij shall be positive and the corresponding travel cost via $j \in FSB(i, u)$ must be equal to V_i^u. Therefore, the inequalities (29.2) ensure that V_i^u is the minimum among the costs via each $j \in FSB(i, u)$. Constraint (29.3) requires the solution link flows to be nonnegative. Finally, constraint (29.4) requires that all and only the node flow shall be assigned to the available links.
To improve readability, based on (28) problem (29) can be rewritten as:
$x_j \cdot (a_j + b_j \cdot x_j - v) = 0$, $\forall j \in J$, (30.1)
$a_j + b_j \cdot x_j \geq v$, $\forall j \in J$, (30.2)
$x_j \geq 0$, $\forall j \in J$, (30.3)
$\sum_{j \in J} x_j = 1$, (30.4)
where:
$J = \{(i, j, u): j \in FSB(i, u)\}$;
$a_j = (c_{ij}{}^u + C_j{}^u) - (g_{ij} + \chi_{ij} + G_j{}^u) \cdot e_i{}^u \cdot y_{ij}{}^u$;
$b_j = (g_{ij} + \chi_{ij} + G_j{}^u) \cdot e_i{}^u$;
$x_j = e_{ij}{}^u / e_i{}^u$;
$v = V_i^u$.
Because the cost derivatives are always positive, given that $e_i{}^u > 0$, each term b_j is positive; instead, some term a_j may be null or negative.

Applying the usual Beckmann approach we can reformulate the assignment problem (30) as the following quadratic program with linear constraints:
$\min\{\sum_{j \in J} \int_0^{x_j} (a_j + b_j \cdot x) \cdot dx: \mathbf{x} \in X\} = \min\{\sum_{j \in J} a_j \cdot x_j + 0.5 \cdot b_j \cdot x_j^2: \mathbf{x} \in X\}$, (31)
where X is the nonempty, compact and convex set of all vectors satisfying the feasibility conditions (30.3) and (30.4). The continuity of the objective function ensures the existence of a solution. The objective function is strictly convex, being the sum of integrals of increasing functions. The gradient of the objective function is a vector with generic entry $a_j + b_j \cdot x_j$, and then the Hessian of the objective function is a diagonal matrix with generic entry b_j . Since all entries b_j are strictly positive, the Hessian is positive definite and problem (31) has a unique solution. This is a desired property, given that the solution of the problem under consideration is the back bone of the proposed assignment algorithm. Moreover, because the constraints that constitute the feasible set are linear and satisfy the independence qualification, we can also asses (Still, 2006) that the solution to the LUCE basic problem (29) is a continuous one-valued function of the current flow pattern.

To solve problem (30) we propose the following simple method. In order to satisfy condition (30.1), either it is $x_j = 0$, or it is $a_j + b_j \cdot x_j = v$. Let $H \subset J$ be the set of alternatives with positive flow, that is $H = \{ j \in J: x_j > 0\}$.
For any given H, the solution in terms of flow proportions is immediate, since from (30.4) it is $\sum_{j\in H} (v - a_j) / b_j = 1$; therefore we have:

$$v = (1 + \sum_{j\in H} a_j / b_j) / (\sum_{j\in H} 1 / b_j) \ , \tag{32.1}$$

$$x_j = (v - a_j) / b_j \ , \qquad \forall j \in H \ , \tag{32.2}$$

$$x_j = 0 \ , \qquad \forall j \in J \setminus H \ . \tag{32.3}$$

The flow proportions provided by (32) implicitly satisfy (30.4), but to state that the chosen H yields the actual solution of problem (30), we still must ensure the two conditions: $a_j < v$, for each $j \in H$ (as required by (30.3), since $x_j = (v - a_j) / b_j > 0$), and $a_j \geq v$, for each $j \in J \setminus H$ (required by (30.2), since $x_j = 0$). This implies that at the solution the value of v resulting form (32.1) must partition the set J into two sub-sets: the set H, made up by the alternatives j such that $a_j < v$, and its complement $J \setminus H$, made up by the alternatives j such that $a_j \geq v$.
At a first glance the problem to determine the set H of alternatives with positive flow may seem to be combinatorial; however, this is not the case. Indeed, equation (32.1) can be rewritten as a recursive formula, thus showing the effect of adding an alternative k to the set H:

$$\mathrm{v}(H \cup \{k\}) = (\mathrm{v}(H) \cdot \sum_{j\in H} 1 / b_j + a_k / b_k) / (\sum_{j\in H} 1 / b_j + 1 / b_k) \ . \tag{33}$$

The right hand side of (33) can be interpreted as an average between $\mathrm{v}(H)$ and a_k with positive weights $\sum_{j\in H} 1 / b_j$ and $1 / b_k$. Therefore, the linear equilibrium cost increases by adding to H any alternative k for which a_k is higher than the current value $\mathrm{v}(H)$, and vice versa it decreases by removing from H such alternatives. Consequently, the correct partition set H can be simply obtained by removing iteratively to an initially complete set each alternative $j \in H$ such that $a_j > v$, i.e. each alternative for which (32.2) yields a negative flow proportion.
The proposed greedy algorithm terminates in a finite number of steps (at the most $|J|$-1 iterations are required) yielding the unique solution of the basic LUCE problem (29).

Incidentally, it's worth noticing the perfect analogy of (32.1) with the expression of the total travel time at a stop in hyperpath-based transit assignment, given by the waiting time plus the riding time, in the case of independent exponential headways whose mean is b_j and riding times once boarded equal to a_j for each line j. The above greedy algorithm is proved to be valid also to determine the attractive lines, although in that case we start from an empty set (Pallottino and Nguyen, 1988).

4. Bush management

An important issue of any assignment algorithm based on bush equilibration is clearly how and when to modify the current bush. For this task LUCE resembles the approach of Algorithm B (Dial, 2006), but is rather innovative in both respects.

To state precisely our bush management scheme we need to introduce some further notation regarding node potentials:

W_i^u minimum cost for users of class $u \in U$ to reach their destination $d(u) \in Z$ from node $i \in N$ on graph G ;

$\mathrm{W}_i^u(\mathbf{c})$ least cost function of node $i \in N$ for users of class $u \in U$ to reach their destination $d(u) \in Z$, yielding the node minimum cost

$W_i^u = \mathrm{W}_i^u(\mathbf{c})$; (34)

$\hat{W}_i^u$ minimum cost for users of class $u \in U$ to reach their destination $d(u) \in Z$ from node $i \in N$ on the bush $B(u)$.

The least cost function $\mathrm{W}_i^u(\mathbf{c})$ is continuous, although non differentiable, and is typically computed jointly for all nodes with respect to a same destination $d(u) \in Z$ through a shortest tree algorithm on the whole graph (see, for example, Pallottino and Scutellà, 1998), whose map will be denoted as $\mathrm{W}^u(\mathbf{c})$.

The node minimum cost $\hat{W}_i^u$ is instead given by the cost of the shortest path on the current bush from $i \in N$ to the destination $d(u) \in Z$, which can be computed, similarly to C_i^u and G_i^u, by processing on the bush each node $i \in N$ in a reverse topological order $O(i, u)$, starting from the destination $d(u)$, where $\hat{W}_{d(u)}{}^u = 0$, through the following recursive equation:

$\hat{W}_i^u = \min\{c_{ij}{}^u + \hat{W}_j^u \colon j \in FSB(i, u)\}$, if $i \neq d(u)$, $\hat{W}_i^u = 0$, otherwise . (35)

The computation of equation (35) for node $i \in N/\{d(u)\}$ requires the node minimum cost $\hat{W}_j^u$, for each $j \in FSB(i, u)$, which will have been already calculated, given that $ij \in B(u)$ and thus $O(i, u) < O(j, u)$.

Each bush is initialized to the set of efficient links that bring closer to the destination on the graph:

$B(u) = \{ij \in A \colon W_i^u > W_j^u\}$. (36)

Since the link costs are strictly positive, the resulting initial bush, which is acyclic by construction, includes all the initially shortest paths to the destination, thus spanning all the nodes connected to $d(u)$.

However, at the generic iteration the current bush may include non efficient links that do not bring closer to the destination and, most important, may exclude shortest links that would improve its node minimum costs, thus leading to new shortest paths. Obviously we shall modify the current bush in order to ensure that eventually all shortest paths are available in the route choice.

To this end, if for each $ij \in A$ such that $f_{ij}{}^u > 0$ holds true that $\hat{W}_i^u > \hat{W}_j^u$, then we update the bush to:

$B(u) = \{ij \in A \colon \hat{W}_i^u > \hat{W}_j^u\}$. (37)

This way: a) the resulting bush is acyclic, by construction; b) the bush is spanning all the nodes connected to the destination; c) all class flows travel on the bush; d) all *non efficient* links ij such that $\hat{W}_i^u \leq \hat{W}_j^u$ carry zero flow and are removed; e) the bush is made only by *efficient* links ij such that $\hat{W}_i^u > \hat{W}_j^u$; f) each *shortest* link ij such that $\hat{W}_i^u \geq c_{ij}{}^u + \hat{W}_j^u$ is included into the bush. At equilibrium, since link costs are unique, the node minimum costs to reach the destination are also unique, implying that there is only one possible topology of the bush consistent with (37).

It may happen that some node (which for hypothesis H3 is not an origin) is not connected to the destination on the graph. Based on (36) and (37), these nodes with infinite minimum costs will not be included in the initial bush, nor in its further modifications.

The LUCE algorithm tends to equilibrate the current bush; thus, eventually the flow on non efficient paths disappears, and the bush can be properly modified. After the modification of the bush through (37), in general some shortest paths on the graph may still be partially out of the bush, since the links just added to it may be used to further improve the node minimum costs on the bush. We now prove that, if all the shortest links belong to the bush already before the update, this is impossible.

Theorem 2 – conditions for bush optimality
If (before the update) there is no shortest link out of the bush, i.e. $\{ij \in A \backslash B(u): \hat{W}_i^u \geq c_{ij}^u + \hat{W}_j^u\} = \emptyset$, then it includes all the shortest paths on the graph (proved in Gentile, 2009).

Theorem 2 can be seen as a strict version of the correctness proof for the Bellman-Ford algorithm (Bellman, 1958), originally referred to a spanning tree, which states that: if $\hat{W}_i^u \leq c_{ij}^u + \hat{W}_j^u$ for each link $ij \in A$ of the graph, then $\hat{W}_i^u = W_i^u$ for each node $i \in N$; moreover, this condition can be reached in a finite number of steps by iteratively updating $\hat{W}_i^u$ with $c_{ij}^u + \hat{W}_j^u$ where the relation is not satisfied.
Since by construction $\hat{W}_i^u \leq c_{ij}^u + \hat{W}_j^u$ for each link $ij \in B(u)$, then the condition actually required is: $\{ij \in A \backslash B(u): \hat{W}_i^u > c_{ij}^u + \hat{W}_j^u\} = \emptyset$. Based on Bellman's Theorem some shortest path may be partially out of the bush; although at equilibrium this is not critical, since if the path would be used its cost would increase. In any case, Theorem 2 overcomes this ambiguity.

The innovation proposed for the bush management scheme with respect to Algorithm B is twofold. First, to compute the node minimum costs we don't apply a shortest tree algorithm on the entire graph, but rely on a recursive procedure on the bush that requires a simple visit of such an acyclic sub-graph: the complexity is this way reduced from $\propto|A|\cdot\log(|A|)$ to $\propto|A|$.
Second, we don't wait for a perfectly equilibrated bush before updating it, but simply make sure that only efficient links are used: this accelerates the process of including in the bush any arising shortest path.

5. Search direction

In this section we first sketch the dynamic programming procedure to find the search direction for a given class $u \in U$ and then state that it is descent, if the current bush is not equilibrated. This important properties can be used to establish a convergent assignment algorithm.

The analysis will be developed in the space of link flows of one class, while all other class flows are implicitly left unchanged, thus exploiting the natural partition of problem (5).

To obtain a complete pattern of link flows $\mathbf{e}^u \in \Omega_u$ for a given class $u \in U$ consistent with the LUCE approach we have to solve the basic problem (29) for each node $i \in N$ proceeding on the current bush $B(u)$ in a topological order $O(i, u)$. To this end, each time the node flow e_i^u must be preliminary determined as follows:

$$e_i^u = \sum_{j \in BSB(i, u)} e_{ji}^u + Q_i^u \text{ , if } i \neq d(u) \text{ , } e_i^u = 0 \text{ , otherwise .} \tag{38}$$

The computation of equation (38) for node $i \in N / \{d(u)\}$ requires the link flow e_{ji}^u, for each $j \in FSB(i, u)$, which will have been already calculated by solving the basic problem for node j, given that $ji \in B(u)$ and thus $O(j, u) < O(i, u)$.

In the end, to obtain the search direction we have to:
- compute the link costs, their derivatives and the current flow proportions;
- compute the node average costs and their derivatives in reverse topological order;
- solve the sequence of LUCE basic problems for each node in topological order.

This procedure can be seen as a continuous one-valued function $\omega^u(\mathbf{f}) \in \Omega_u$ of $\mathbf{f} \in \Omega$, from the space of link flows of all classes to the space of link flows of class u only: We will denote by $\mathbf{f}^u \in \Omega_u$ the projection of $\mathbf{f}$ in the latter space; actually $\omega^u(\mathbf{f})$ depends on $\mathbf{f}^u$ and on the link volumes, given the current bush $B(u)$.

In the following two propositions, we state that the direction $\omega^u(\mathbf{f})$-$\mathbf{f}^u$ is descent for any $\mathbf{f} \in \Omega$ and any class $u \in U$, unless $\mathbf{f}^u$ is an equilibrium on $B(u)$ with respect to the current cost pattern c($\mathbf{f}$), meaning that all paths used to reach destination $d(u)$ on the current bush have minimum cost. This is the cornerstone for the prove of convergence of the LUCE assignment algorithm, given in Gentile (2009).

Theorem 3 – LUCE direction is feasible and descent, or null.
The direction $\omega^u(\mathbf{f})$ - $\mathbf{f}^u$ is feasible and descent for any $\mathbf{f} \in \Omega$ and any class $u \in U$, or it is not null (proved in Gentile, 2009).

Theorem 4 – LUCE direction is null if and only if the current bush is equilibrated.
The direction $\omega^u(\mathbf{f})$ - $\mathbf{f}^u$ is (not) null for any $\mathbf{f} \in \Omega$ and any class $u \in U$, such that $\mathbf{f}^u$ is (not) an equilibrium on $B(u)$ with respect to c($\mathbf{f}$) (proved in Gentile, 2009).

Conclusions

In this paper we have extended the formulation of the LUCE algorithm to the case of multiple user classes. The key question addressed is the uniqueness of the model also in terms of class/destination flows, which allows to univocally determine a balanced solution for path flows. This result was accomplished by introducing an additional term in the non-separable arc cost function.

References

Bar-Gera H. (2002) Origin-based algorithm for the transportation assignment problem. *Transportation Science* 36, 398-417.

Bar-Gera H. (2006) Primal method for determining the most likely route flows in large road networks. *Transportation Science* 40, 269–286.

Beckmann M., McGuire C., Winston C. (1956) *Studies in the economics of transportation*. Yale University Press, New Haven, Connecticut.

Bellman R. (1958) On a Routing Problem. *Quarterly of Applied Mathematics* 16, 87-90.

Bertsekas D. (1995) *Non linear programming*. Athena Scientific, Belmont, Massachusetts.

Bifulco C. (1993) A multimodal and multiuser assignment model. In *Proceedings of the 2nd EURO Working Group Meeting on Transportation*, INRETS, Arcueil, France.

Boyce D., Ralevic-Dekic B., Bar-Gera H. (2004) Convergence of traffic assignments: how much is enough? *Journal of Transportation Engineering* 130, 49-55.

Dafermos S. (1972) The traffic assignment problem for multiclass-user transportation networks. *Transportation Science* 6, 73-87.

Dial R. (1971) A probabilistic multipath assignment model than obviates path enumeration. *Transport Research* 5, 83-111.

Dial R. (2006) A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. *Transport Research* B 40, 917-936.

Florian M. (2009) New look at projected gradient method for equilibrium assignment. *Transportation Research Board* Annual Meeting 2009, Paper #09-0852.

Florian M., Guelat J., Spiess H. (1987) An efficient implementation of the PARTAN variant of the linear approximation method for the network equilibrium problem. *Networks* 17, 319-339.

Gentile G. (2009) Linear User Cost Equilibrium: a new algorithm for traffic assignment. Submitted to *Transportation Research* B.

Gentile G., Noekel K. (2009) Linear User Cost Equilibrium: the new algorithm for traffic assignment in VISUM. In *Proceedings of European Transport Conference* 2009, Leeuwenhorst Conference Centre, The Netherlands.

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., Patriksson M. (1992) Simplicial decomposition with disaggregated representation for the traffic assignment problem. *Transportation Science* 26, 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, 309-318.

Nguyen S., Pallottino S. (1988) Equilibrium traffic assignment for large scale transit networks. *European Journal of Operational Research* 37, 176-186.

Pallottino S. and Scutellà M.G. (1998) Shortest path algorithms in transportation models: classical and innovative aspects. In *Equilibrium and Advanced Transportation Modelling*, ed.s P. Marcotte and S. Nguyen, Kluwer, 245-281.

Sheffi Y. (1985) Urban transportation networks. Prentice-Hall, New Jersey.

Still G. (2006) A lecture on parametric optimization: an introduction given at the Middle East Technical University of Ankara. University of Twente, the Netherlands.

Toint P. and Wynter L. (1996) Asymmetric multiclass traffic assignment: a coherent formulation. In *Proceedings of the 13th International Symposium on Transportation and Traffic Theory*, ed. J.-B. Lesort, Pergamon, Exeter, UK.

Van Vliet D., Bergman T., Scheltes W.H. (1986) Equilibrium traffic assignment with multiple user classes. In *Proceedings of the PTRC Summer Annual Meeting*.

Wardrop J. (1952) Some theoretical aspects of road traffic research. In *Proceedings of the Institution of Civil Engineers*, Part 2, 325-378.