LINEAR USER COST EQUILIBRIUM: THE NEW ALGORITHM FOR TRAFFIC ASSIGNMENT IN VISUM

Guido Gentile
DITS, Sapienza Università di Roma, guido.gentile@uniroma1.it

Klaus Noekel
PTV AG, klaus.noekel@ptv.de

ABSTRACT

LUCE (Linear User Cost Equilibrium) is the new algorithm to solve traffic assignment with deterministic route choice available in VISUM.

The main idea is to seek at every node a user equilibrium for the local route choice of drivers 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. The solution to such linear program in terms of destination flows, recursively applied for each node, provides a descent direction with respect to the classical sum-integral objective function. The network loading is then performed through the corresponding splitting rates, thus avoiding explicit path enumeration.

Exploiting the inexpensive information provided by the derivatives of the link costs with respect to link flows, LUCE achieves a very high convergence speed that compares favourably to the other methods, while it assigns the demand flow of each o-d couple on several paths at once.

Keywords: deterministic traffic assignment, bush-based approach, linear user equilibrium, destination splitting rates, link cost derivatives, fast algorithm convergence, multiple path loading, implicit path enumeration.

1 INTRODUCTION

Although traffic assignment is a rather mature issue in transport modelling, to find a precise equilibrium on real networks is still a difficult problem to be solved. In practice, the algorithms currently available are not truly satisfactory for many applications: they simply don’t converge in reasonable time fine enough to allow consistent comparisons between design scenarios. Indeed, apparently small errors in the iterative procedure do not allow to appreciate the real differences among equilibria and may lead to false conclusions in relevant projects, thus vanishing any modelling effort.

This is a well recognized problem by the recent literature (see, for example, Boyce *et al.*, 2004), but is not really acceptable by practitioners, who are asking researchers and developers to enhance the algorithm convergence. To satisfy this necessity and overcome such a drawback, the main software producers in the world are in these days changing their traffic assignment procedures (e.g. VISUM, TRANSCAD, EMME), thus animating the international debate and the competition on this topic.

In this context, LUCE gives a noticeable contribution: it converges between 10 and 100 times faster than other recent algorithms and allows on large networks to reach in few minutes and iterations the relative gap of 10E-8, that is considered enough for any application. This result conveys a considerable advance in modelling practice, more than in algorithm theory.

After 30 years of Frank-Wolfe domain (LeBlanc, 1973; Nguyen, 1973), the seminal work of Bar-Gera (2002) opened the way to a new generation of solution methods for deterministic assignment, that exploit the natural partition of the Beckmann's mathematical program (1956) into sub-problems, one for each origin. The proposed LUCE algorithm revisits this approach by searching along a descent direction obtained through cost equilibration at nodes, for each destination – we find more intuitive and informative to consider destinations, instead of origins, since these are the aims of travellers.

The main idea is to seek at every node a user equilibrium (Wardrop, 1952) for the local route choice of divers directed toward the 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 the link at hand, 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 destination flows, applied at each node of the bush in topological order, provides a descent direction with respect to the classical sum-integral objective function of the original "non-linear" assignment problem. This postulate is proved only with reference to any given bush of the destination. To ensure the convergence of the procedure towards an equilibrium where all paths of the graph satisfy Wardrop's conditions, the current bush of the destination is changed at the beginning of each iteration, 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.

At each iteration, the proposed algorithm requires no shortest path but two visits of the bush links for each destination, that is equal to the complexity of the STOCH single pass procedure (Dial, 1971) for the Logit network loading. Moreover, 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. In conclusion, exploiting the inexpensive information provided by the derivatives of the arc costs with respect to link flows, LUCE presents a higher convergence speed than the existing methods, both in terms of runtime and iterations.

2 MATHEMATICAL FORMULATION

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.

We adopt the following notation:

f_{ij} total flow on link $ij \in A$, generic element of the $(|A| \times 1)$ vector $\mathbf{f}$;

c_{ij} cost of link $ij \in A$, generic element of the $(|A| \times 1)$ vector $\mathbf{c}$;

$c_{ij}(f_{ij})$ cost function of link $ij \in A$,

$Z \subseteq N$ set of the zone centroids ;

D_{od} demand flow between origin $o \in Z$ and destination $d \in Z$, generic element of the $(|Z|^2 \times 1)$ vector $\mathbf{D}$, that is the o-d matrix in row major order ;

K_{id} set of the acyclic paths between node $i \in N$ and destination $d \in Z$;

$K = \cup_{o \in Z} \cup_{d \in Z} K_{od}$ is the set of paths available to users ;

δ_{ij}^{k} is 1, if link $ij \in A$ belongs to path k, and 0, otherwise – for $k \in K$, this is the generic element of the $(|A| \times |K|)$ matrix $\mathbf{\Delta}$;

λ_{od}^{k} is 1, if path $k \in K$ connects origin $o \in Z$ to destination $d \in Z$ (i.e. $k \in K_{od}$), and 0, otherwise – this is the generic element of the $(|Z|^2 \times |K|)$ matrix $\mathbf{\Lambda}$;

F_k flow on path $k \in K$, generic element of the $(|K| \times 1)$ vector $\mathbf{F}$;

C_k cost of path k – for $k \in K$ this is the generic element of the $(|K| \times 1)$ vector $\mathbf{C}$;

W_i^d minimum cost to reach destination $d \in Z$ from node $i \in N$;

$|S|$ cardinality of the generic set S ;

There are two fundamental relations between flow variables. The flow on link $ij \in A$ is the sum of the flows on the paths that include it:

$$f_{ij} = \sum_{k \in K} \delta_{ij}^{k} \cdot F_k \; ; \tag{1}$$

the travel demand between origin $o \in Z$ and destination $d \in Z$ must be equal to the sum of the flows on the paths that connect them:

$$\sum_{k \in K_{od}} F_k = D_{od} \; ; \tag{2}$$

moreover, all path flows must satisfy non-negativity constraints: $F_k \geq 0$, $k \in K$.

As usual, we assume additive path costs, i.e. the impedance C_k associated by users to a given path k is the sum of the costs on the links that belong to it:

$$C_k = \sum_{ij \in A} \delta_{ij}^{k} \cdot c_{ij} \; . \tag{3}$$

By definition, the minimum cost to reach destination $d \in Z$ from node $i \in N$ is the cost of any shortest path that connects them:

$$W_i^d = \min\{C_k : k \in K_{id}\} \; . \tag{4}$$

In the classical case of separable arc cost functions, fixed demand and deterministic route choice, the traffic assignment problem can be formalized through the following program:

$$\min\{\omega(\mathbf{f}) = \sum_{ij\in A} \int_0^{f_{ij}} c_{ij}(f) \cdot d f : \mathbf{f}\in\Theta\}, \quad (5)$$

where:

$\Theta = \{\mathbf{f}\in\Re^{|A|}: \mathbf{f} = \mathbf{\Delta}\cdot\mathbf{F}, \mathbf{F}\in\Omega\}$ is the set of feasible link flows, and

$\Omega = \{\mathbf{F}\in\Re^{|K|}: \mathbf{F} \geq \mathbf{0}, \mathbf{\Lambda}\cdot\mathbf{F} = \mathbf{D}\}$ is the set of feasible path flows.

To ensure the existence and uniqueness of the solution to problem (5) we assume that:

$c_{ij}(f_{ij})$ is non-negative, continuous, strictly monotone increasing ;

K_{od} is non-empty ;

D_{od} is non-negative .

Problem (5), which is convex, can also be expressed in terms of path flows as follows:

$$\min\{\Phi(\mathbf{F}) = \sum_{ij\in A} \int_0^{\sum_{k\in K} \delta_{ij}^k \cdot F_k} c_{ij}(f) \cdot d f : \mathbf{F}\in\Omega\} , \quad (6)$$

where, although the solution uniqueness does not hold anymore, the convexity of the mathematical program is preserved, implying that any descent algorithm in the space of path flows will provide one of the global solutions, which then make up a convex set.

The relevance of program (6) to traffic assignment stands from the fact that, in the case of additive path costs, its first order (necessary) conditions coincide with the following formulation of the deterministic user equilibrium based on Wardrop's Principles, for each $o\in Z$ and $d\in Z$:

$$F_k \cdot (C_k - W_o^d) = 0 , \quad \forall k\in K_{od} , \quad (7.1)$$

$$C_k \geq W_o^d , \quad \forall k\in K_{od} , \quad (7.2)$$

$$F_k \geq 0 , \quad \forall k\in K_{od} , \quad (7.3)$$

$$\sum_{k\in K_{od}} F_k = D_{od} . \quad (7.4)$$

Based on (7):

- all used paths ($F_k > 0$) have minimum cost ($C_k = W_o^d$) ;

- any unused path ($F_k = 0$) has not a lower cost ($C_k \geq W_o^d$) .

We have a user equilibrium if conditions (7) hold jointly for each o-d couple, under the condition that each path cost C_k is a function (potentially) of all the path flows **F** through the arc cost function:

$$C_k = \sum_{ij\in A} \delta_{ij}^k \cdot c_{ij}(\sum_{k\in K} \delta_{ij}^k \cdot F_k) \text{ , in compact form } \mathbf{C} = \Delta^T\cdot c(\Delta\cdot\mathbf{F}) . \quad (8)$$

Since the gradient of $\Phi(\mathbf{F})$ is $\mathbf{C} = \Delta^T\cdot c(\Delta\cdot\mathbf{F})$, by linearizing the objective function of problem (6) at a given a point $\mathbf{F}\in\Omega$, for $\mathbf{X} \to \mathbf{F}$ we obtain:

$$\Phi(\mathbf{X}) = \Phi(\mathbf{F}) + \mathbf{C}^{T}\cdot(\mathbf{X}\text{-}\mathbf{F}) + o(||\mathbf{X}\text{-}\mathbf{F}||). \tag{9}$$

From equation (9) we recognize that a direction **E**-**F** is descent if and only if:

$$\mathbf{C}^{T}\cdot(\mathbf{E}\text{-}\mathbf{F}) < \mathbf{0}. \tag{10}$$

In other words, to decrease the objective function and maintain feasibility we necessarily have to *shift path flows getting a lower total cost with respect to the current cost pattern*, i.e. move the current solution from **F** towards an $\mathbf{E}\in\Omega$, such that $\mathbf{C}^{T}\cdot\mathbf{E} < \mathbf{C}^{T}\cdot\mathbf{F}$, where $\mathbf{C} = \Delta^{T}\cdot c(\Delta\cdot\mathbf{F})$; the necessity derives from the convexity of the problem, since in this case at any point **X** such that $\mathbf{C}^{T}\cdot(\mathbf{X}\text{-}\mathbf{F}) > \mathbf{0}$ we have: $\Phi(\mathbf{X}) > \Phi(\mathbf{F})$.

This approach to determine a descent direction can be applied to each o-d pair separately, to each destination, or to the whole network jointly. Based on the above general rule, setting the flow pattern **E** 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 is applied along it, we obtain the well known Frank-Wolfe algorithm. However, at equilibrium each o-d pair typically uses several paths, implying that any descent direction that loads a single path is intrinsically myopic; in fact such algorithms tail badly.

Once we get a feasible descent direction **E**-**F**, since Ω is convex, we can move the current solution along the segment $\mathbf{F}+\alpha\cdot(\mathbf{E}\text{-}\mathbf{F})$ and take a step $\alpha\in(0,1]$ such that the objective function of problem (6), redefined as $\phi(\alpha) = \Phi(\mathbf{F}+\alpha\cdot(\mathbf{E}\text{-}\mathbf{F}))$, is sufficiently lowered. In this respect, knowing that Φ is C^1 and convex, and thus also ϕ is such, we can consider the following alternative approaches:

- minimize ϕ through a line search along the segment, e.g. by means of the bisection method;

- minimize an approximation of ϕ along the segment, e.g. the quadratic interpolation driven by the derivative at $\alpha = 0$ and $\alpha = 1$, i.e.
$\alpha = \min\{1, 1 / (1 - \partial\phi(1)/\partial\alpha / \partial\phi(0)/\partial\alpha)\}$;

- determine the largest step $\alpha = 0.5^{k}$, for any non-negative integer *k*, such that: $\partial\phi(0.5^{k})/\partial\alpha < 0$, i.e. by an Armijo-like search.

All the three methods require to compute the directional derivative of the objective function:

$$\partial\phi(\alpha)/\partial\alpha = c(\Delta\cdot(\mathbf{F}+\alpha\cdot(\mathbf{E}\text{-}\mathbf{F})))^{T} \cdot \Delta\cdot(\mathbf{E}\text{-}\mathbf{F}) , \tag{11}$$

which implies to evaluate the arc costs at the candidate flows $\mathbf{F}+\alpha\cdot(\mathbf{E}\text{-}\mathbf{F})$, and then the difference between the corresponding total costs obtained with the flows **E** and **F**; if such total costs with **E** are smaller than those with **F**, then $\partial\phi(\alpha)/\partial\alpha$ is negative so that the optimal solution is more toward **E**, and vice versa.

3 LOCAL USER EQUILIBRIUM

In this section we present a new method to determine a descent direction, which is based on local shifts of flows that satisfy the total cost lowering rule, by exploiting the inexpensive information provided by the derivatives of the arc costs with respect to link flows.

In the following we will focus on users directed to destination $d \in Z$ and in particular on their local route choice at a generic node $i \in N$.

At equilibrium only shortest paths are utilized to reach d. Since the arc cost functions are strictly monotone increasing (i.e. an arc cost can be null only if its flow is such), these paths make up an acyclic sub-graph of G, that is a (reverse) bush rooted at d. On this base, when seeking a descent direction, in the following we can limit our attention to the current bush $B(d) \subseteq A$ and introduce a updating mechanism to make sure that eventually any shortest path will be included into it.

For the topology of the bush we will use the following notation:

$FSB(i, d) = \{j \in N: ij \in B(d)\}$ the forward star of node $i \in N$ made-up by nodes that can be reached from it through links belonging to the current bush $B(d)$ of destination $d \in Z$;

$BSB(i, d) = \{j \in N: ji \in B(d)\}$ the backward star of node $i \in N$ made-up by nodes that can reach it through links belonging to the current bush $B(d)$ of destination $d \in Z$.

For the flow pattern we will use the following notation:

f_{ij}^d current flow on link $ij \in A$ directed to destination $d \in Z$; by construction it is $f_{ij}^d = 0$ for each $j \notin FSB(i, d)$; moreover it clearly is: $f_{ij} = \sum_{d \in Z} f_{ij}^d$;

$f_i^d = \sum_{j \in FSB(i, d)} f_{ij}^d$ current flow leaving node $i \in N$ directed to destination $d \in Z$;

$y_{ij}^d = f_{ij}^d / f_i^d$ current flow proportion on link $ij \in A$ directed to destination $d \in Z$, if $f_i^d > 0$; $y_{ij}^d = 0$, otherwise ;

e_{ij}^d descent direction, in terms of flow on link $ij \in A$ directed to destination $d \in Z$;

e_i^d descent direction, in terms of flow leaving node $i \in N$ directed to destination $d \in Z$;

$x_{ij}^d = e_{ij}^d / e_i^d$ descent direction, in terms of flow proportion on link $ij \in A$ directed to destination $d \in Z$.

For the cost pattern we will use the following notation:

C_i^d average cost to reach destination $d \in Z$ from node $i \in N$;

g_{ij} cost derivative of link $ij \in A$;

G_i^d derivative of the average cost to reach destination $d \in Z$ from node $i \in N$.

The average cost C_i^d is the expected impedance that a user encounters by travelling from node $i \in N$ to destination $d \in N$. In (12.1) it is defined recursively, as if drivers utilize paths accordingly with the current flow proportions; while (12.2) defines the iterate for the idle case as the locally best choice:

if $f_i^d > 0$, then $C_i^d = \sum_{j \in FSB(i, d)} y_{ij}^d \cdot (c_{ij} + C_j^d)$, else (12.1)

$C_i^d = \min\{c_{ij} + C_j^d: j \in FSB(i, d)\}$. (12.2)

In the following we assume that the cost function $c_{ij}(f_{ij})$ is differentiable for each link $ij \in A$:

$g_{ij} = \partial c_{ij}(f_{ij}) / \partial f_{ij}$. (13)

Under the assumption that an infinitesimal increment of flow leaving node $i \in N$ directed towards destination $d \in Z$ would diverge accordingly with the current flow proportions, we have :

if $f_i^d > 0$, then $G_i^d = \partial C_i^d / \partial f_i^d = \sum_{j \in FSB(i, d)} y_{ij}^{d\ 2} \cdot (g_{ij} + G_j^d)$, else (14.1)

$G_i^d = \sum_{j \in FSB(i, d)} [C_i^d = c_{ij} + C_j^d] \cdot (g_{ij} + G_j^d) / \sum_{j \in FSB(i, d)} [C_i^d = c_{ij} + C_j^d]$, (14.2)

where in (14.1) the derivatives $g_{ij} + G_j^d$ are scaled by the share y_{ij}^d of ∂f_i^d utilizing link ij and then passing through node j, that jointly with the flow proportion involved in the averaging yields the square $y_{ij}^{d\ 2}$; while (14.2) in the idle case expresses the mean of all the derivatives $g_{ij} + G_j^d$ for which link ij gives a locally best choice.

The average costs and their derivatives can be computed by processing the nodes of the bush in reverse topological order, starting from $C_d^d = G_d^d = 0$.

We now address the local user equilibrium for the e_i^d drivers directed to destination $d \in Z$, whose available alternatives are the links of the bush exiting from node $i \in N$. To each travel alternative we associate the cost function:

$v_{ij}^d(e_{ij}^d) = (c_{ij} + C_j^d) + (g_{ij} + G_j^d) \cdot (e_{ij}^d - y_{ij}^d \cdot e_i^d)$, (15)

resulting from a linearization at the current flow pattern of the average cost encountered by a user choosing the generic link ij, with $j \in FSB(i, d)$.

This problem can be formulated, in analogy to (7), by the following system of inequalities:

$e_{ij}^d \cdot (v_{ij}^d(e_{ij}^d) - V_i^d) = 0$, $\forall j \in FSB(i, d)$, (16.1)

$v_{ij}^d(e_{ij}^d) \geq V_i^d$, $\forall j \in FSB(i, d)$, (16.2)

$e_{ij}^d \geq 0$, $\forall j \in FSB(i, d)$, (16.3)

$\sum_{j \in FSB(i, d)} e_{ij}^d = e_i^d$, (16.4)

where we denote:

V_i^d local equilibrium cost to reach destination $d \in Z$ from node $i \in N$;

v_{ij}^d cost of the local alternative $j \in FSB(i, d)$ to reach destination $d \in Z$ from node $i \in N$.

If $e_i^d = 0$, the solution to the above problem is trivially: $e_{ij}^d = 0$, for each

$j \in FSB(i, d)$. Consider then the case where $e_i^d > 0$. To improve readability, problem (16) can be rewritten as:

$$x_j \cdot (a_j + b_j \cdot x_j - v) = 0 \;, \qquad \forall j \in J \;, \tag{17.1}$$

$$a_j + b_j \cdot x_j \geq v \;, \qquad \forall j \in J \;, \tag{17.2}$$

$$x_j \geq 0 \;, \qquad \forall j \in J \;, \tag{17.3}$$

$$\sum_{j \in J} x_j = 1 \;, \tag{17.4}$$

where:

$J = \{(i, j, d): j \in FSB(i, d)\}$;

$a_j = (c_{ij} + C_j^d) - (g_{ij} + G_j^d) \cdot e_i^d \cdot y_{ij}^d$;

$b_j = (g_{ij} + G_j^d) \cdot e_i^d$;

$x_j = e_{ij}^d / e_i^d$;

$v = V_i^d$.

Applying the usual Beckmann approach we can reformulate the equilibrium problem (17) as the following quadratic program:

$$\min\{\sum_{j \in J} \int_0^{x_j} (a_j + b_j \cdot x) \cdot dx: \mathbf{x} \in X\} = \min\{\sum_{j \in J} a_j \cdot x_j + 0.5 \cdot b_j \cdot x_j^2: \mathbf{x} \in X\} \;, \tag{18}$$

where X is the convex set of all vectors satisfying the feasibility conditions (17.3) and (17.4). 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 . Therefore, if all entries b_j are strictly positive, the Hessian is positive definite and problem (18) has a unique solution. In order to ensure such a desirable property we assume, without loss of generality, that the derivates g_{ij} are strictly positive for all links $ij \in A$. Indeed, since the arc cost functions are strictly monotone increasing, g_{ij} can be null only if also f_{ij}^d is null; therefore, at the equilibrium $b_j = 0$ implies $x_j = 0$. In practice we can substitute any $g_{ij} = 0$ with a small ε.

To solve problem (17) we propose the following simple method. In order to satisfy condition (17.1), 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 (17.4) it is $\sum_{j \in H} (v - a_j) / b_j = 1$; therefore we have:

$$v = (1 + \sum_{j \in H} a_j / b_j) / (\sum_{j \in H} 1 / b_j) \;, \tag{19.1}$$

$$x_j = (v - a_j) / b_j \;, \qquad \forall j \in H \;, \tag{19.2}$$

$$x_j = 0 \;, \qquad \forall j \in J \setminus H \;. \tag{19.3}$$

The flow proportions provided by (19) implicitly satisfy (17.4), but to state that the chosen H yields the actual solution of problem (17), we still must ensure the following conditions: $a_j < v$, for each $j \in H$ (as required by (17.3), since $x_j = (v - a_j) / b_j > 0$), and $a_j \geq v$, for each $j \in J \setminus H$ (as required by (17.2), since $x_j = 0$). This implies that at the solution the value of v resulting form (19.1) 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 seam to be combinatorial; however, this is not the case. Indeed, equation (19.1) can be rewritten as a recursive formula, thus showing the effect of adding an alternative k to the set H:

$$v(H \cup \{k\}) = (v(H) \cdot \sum_{j\in H} 1 / b_j + a_k / b_k) / (\sum_{j\in H} 1 / b_j + 1 / b_k) . \quad (20)$$

The right hand side of (20) can be interpreted as an average between $v(H)$ and a_k with positive weights $\sum_{j\in H} 1 / b_j$ and $1 / b_k$. Therefore, the local equilibrium cost increases by adding to H any alternative k for which a_k is higher than the current value $v(H)$, 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 (19.2) yields a negative flow proportion.

To obtain a complete pattern of link flows $\mathbf{e}^d$ for a given destination $d\in Z$ consistent with the local user equilibrium we simply have to solve problem (16) at each node $i\in N\setminus\{d\}$ proceeding in topological order, where the node flow is computed as follows:

$$e_i^d = \sum_{j\in BSB(i, d)} e_{ji}^d + D_{id} . \quad (21)$$

In section 2 it has been shown that a given direction is descent if, and only if, (10) holds true, which in terms of link flows directed to destination $d\in Z$ becomes:

$$\sum_{ij\in A} c_{ij} \cdot (e_{ij}^d - f_{ij}^d) < 0 , \quad (22)$$

expressing that the shift of flow from $\mathbf{f}^d$ to $\mathbf{e}^d$ must entail a decrease of total cost with respect to the current cost pattern. The proof that the proposed procedure provides a descent direction goes beyond the scope of this note and the interested reader is referred to Gentile G. (2009).

A detailed description of the assignment algorithm is also presented in the above reference. Below we present some numerical results assessing the performance of the method.

4 NUMERICAL RESULTS

In this section we analyze the convergence of the proposed LUCE algorithm.

Typically, the distance of the current flow patter $\mathbf{f}$ from the equilibrium is measured in terms of the relative gap $\rho(\mathbf{f}) = 1 - W(c(\mathbf{f}))^T\cdot\mathbf{D} / c(\mathbf{f})^T\cdot\mathbf{f}$, where $W(c(\mathbf{f}))^T\cdot\mathbf{D}$ are the total minimum costs, while $c(\mathbf{f})^T\cdot\mathbf{f}$ are the total equilibrium costs. Based on Wardrop's First Principle, the relative gap tends to zero at equilibrium, and is always smaller than one. As an alternative, the precision of convergence can be expressed by the maximum cost difference between used paths, which also tends to zero at equilibrium.

In the following, we present the behaviour of LUCE on the eight networks synthetically described in Table 1, showing the evolution of both the relative gap and the maximum gap in terms of iterations and run time (referred to a 2.0GHz cpu). These benchmarks, cover a wide range of possible cases, from small to very large (in terms of links and zones cardinality), from almost uncongested to heavily congested (ΔT denotes the relative increment of total costs from zero flow to equilibrium).

Table 1. Benchmarks considered in the convergence analysis.

network	links	zones	ΔT%
sioux falls	76	24	148
dial	80	4	135
cosenza	524	36	17
winnipeg	2836	138	17
lynnwood	3414	139	43
roma	9369	453	341
chicago	39018	1768	44
philadelpia	40003	1489	31

In all instances, LUCE converges to a satisfactory ρ = 10E-5 (many experts agree that beyond this level, no practical modifications of the solution are usually appreciated) with very few iterations (between 10 and 20), and with some more iterations (between 20 and 40) reaches its limit of numerical instability, that is ρ = 10E-8 for 16 digit computers. The decreasing pattern is smoother for the relative gap (Figure 1), that is an aggregated indicator of convergence, than for the maximum gap (Figure 2), that expresses a worst case distance from the equilibrium.

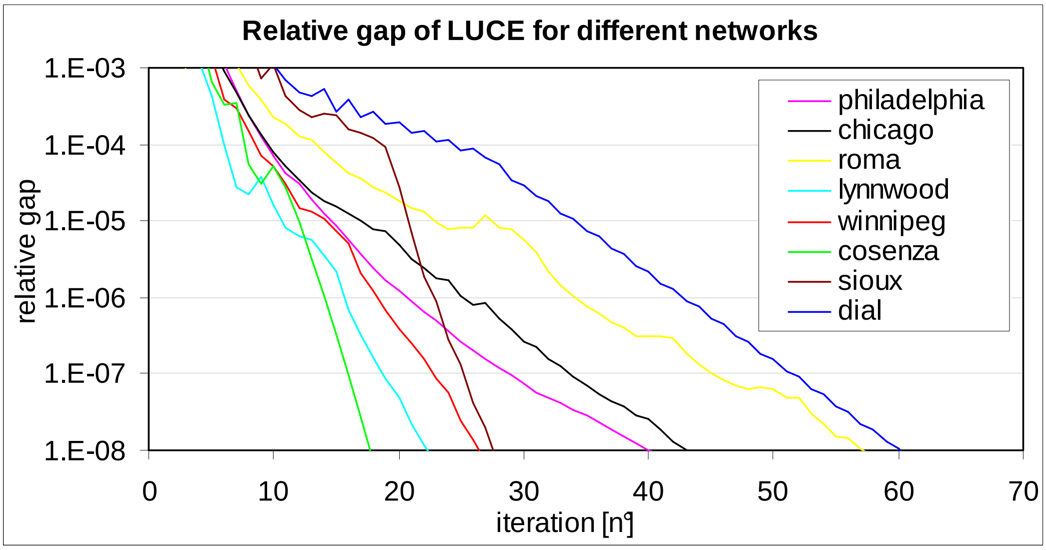

Figure 1. Convergence of LUCE in terms of relative gap.

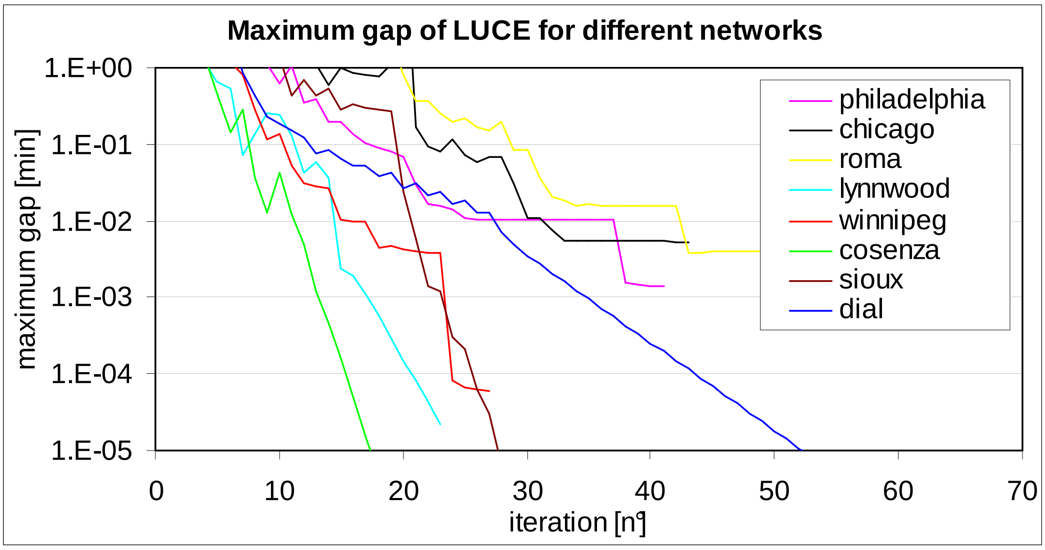

Figure 2. Convergence of LUCE in terms of maximum gap.

In general, LUCE appears to be a very robust method, because the relative computation time (run time dived by the number of links and by the number of zones) that is required to obtain either a precise or a loose solution does not vary considerably for the different networks, as depicted in Figure 3, although heavy congestion implies longer run times, as everyone can expect (see the case of the Rome). This reflects the linear complexity of the algorithm, and means that the modeller can profit of making a rough estimate on run times in advance. Thus LUCE matches the expectations of practitioners, who can well accept that computing times grow proportionally with the dimension of the problem in terms of supply (i.e. links) and demand (i.e. zones). Actually, with path based methods the problem complexity scales instead linearly in terms of od-couples and quadratically in terms of zones. Moreover, the overall convergence seems to be fairly linear, despite some convexity is practically unavoidable in any traffic assignment algorithm for large networks with several destinations.

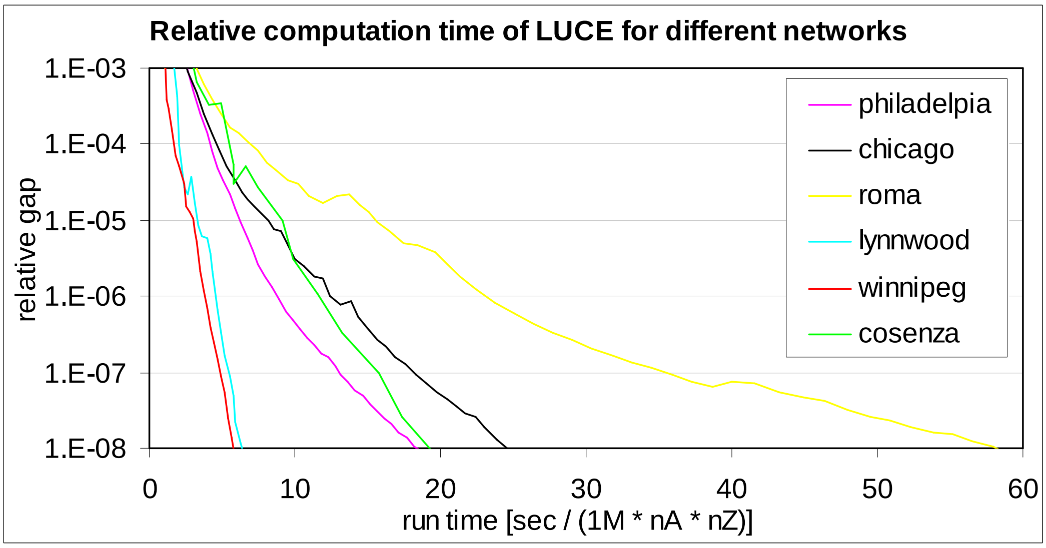

Figure 3. Relative computation time of LUCE for six real networks.

To choose the step method used in combination with the descent direction provided by LUCE we have compared on the above benchmarks the three algorithms proposed at the end of section 2: line search by bisection, analytic minimization of the quadratic interpolation, Armijo rule. Best performances are obtained through either the second or the third method, as depicted in Figure 4 with reference to the Chicago network. However, the quadratic interpolation benefits from the theoretical property of not requiring line search iterations within the step method, and in fact it is sometimes slightly faster than the Armijo rule, so that it was preferred to the latter in the specification of the assignment algorithm.

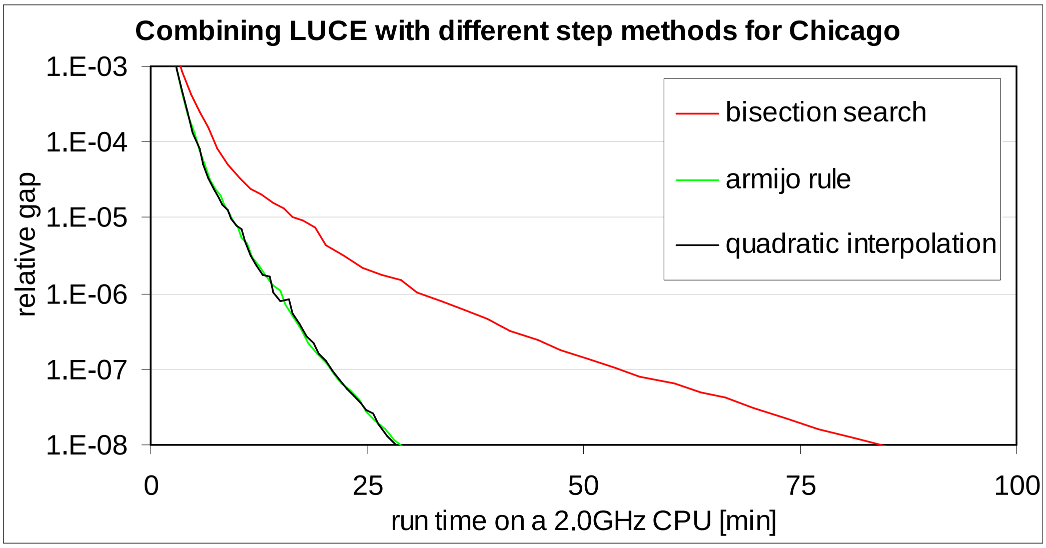

Figure 4. Convergence of LUCE in combination with three step methods on the Chicago network.

The substantial equivalence between the Armijo rule and the quadratic interpolation derives also from the fact that for most destinations at each iteration the unitary step provided by the LUCE direction is optimal, as depicted in Figure 5.

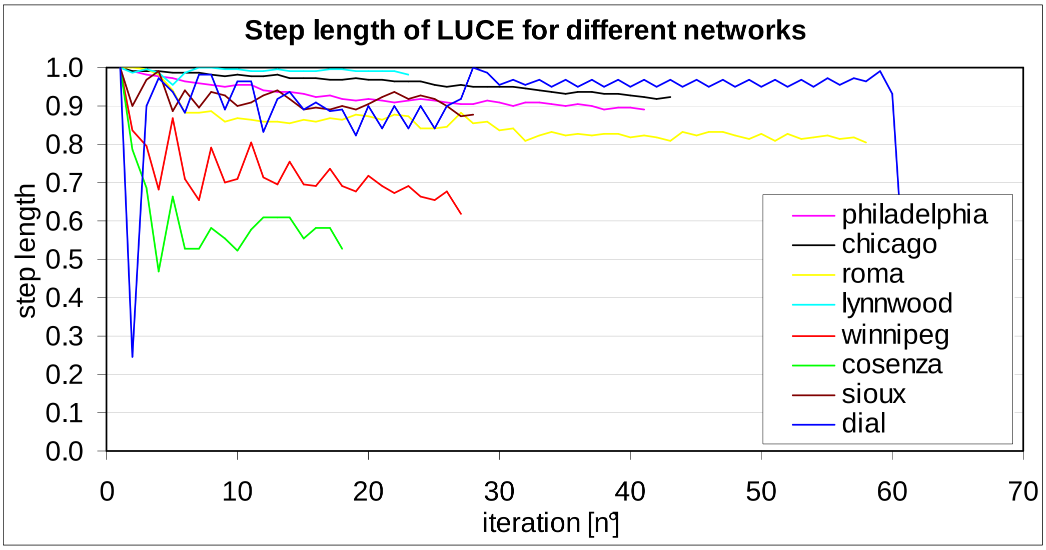

Figure 5. Average step length among all destinations during the iterations of LUCE.

In Figure 6 we plot the relative gap against the run time to analyze the convergence of LUCE on the Chicago network for four levels of demand, obtained scaling by 0.5, 1.0, 1.5 and 2.0 the original o-d matrix. As expected the speed of convergence decreases noticeably with the relevance of congestion, showing in this case an even proportionality: about 10 times the total cost increment (from 44% to 415%) requires about 10 times the computation effort (form 28 min to 214 min).

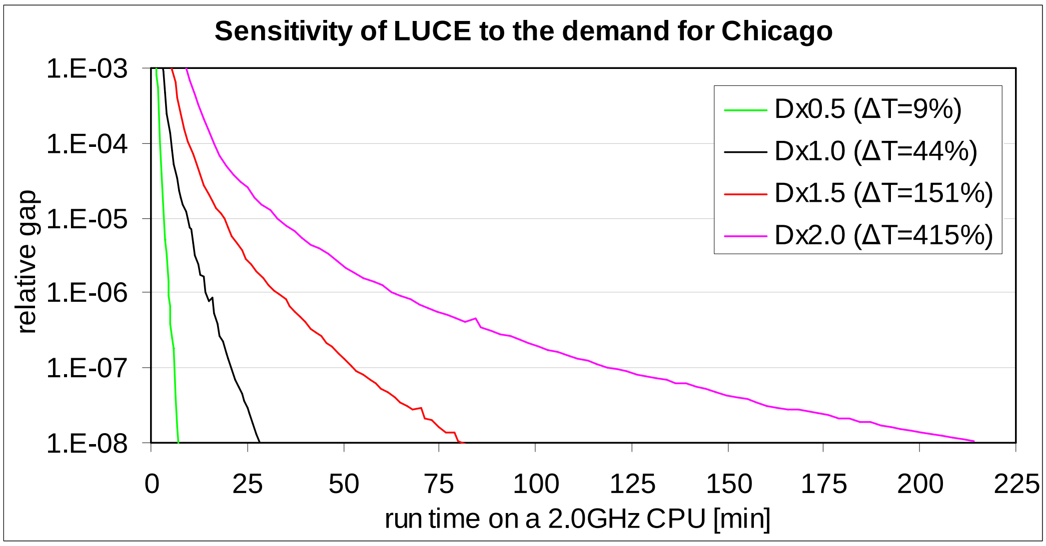

Figure 6. Convergence sensitivity of LUCE to congestion on the Chicago network.

Finally, in Figure 7 we compare the convergence performances of five different algorithms on the Chicago network. Frank Wolfe by LeBlanc (1973) and Nguyen (1973) is here reported just to recall that a precise solution of the assignment problem requires in practice more specialized procedures.

Bar-Gera's OBA (2002) takes into account link and node cost derivates with the aim of calculating Newton-type flow shifts among the paths of the bush, thus obtaining a descent direction in the space of link flows. The efficient computation of this second-order formulation requires to introduce an approximation of the exact analytical expressions. Moreover, the resulting flow shifts may lead to violations of non-negativity constraints, thus requiring some truncations. It's worth noting that LUCE exploits the same information, i.e. cost derivatives, but instead of seeking some parallelism to non-linear optimization techniques like the OBA, it adopts an ad-hoc approach founded on the intuitive ideas of local equilibrium and first-order approximation.

Algorithm B proposed by Dial (2006) is a link based procedure that shifts trips from max- to min-paths to make their cost difference minima.

The Projected Gradient algorithm, recently revised by Florian (2009), is a path based method that regards their costs as the gradient of the sum integral objective function.

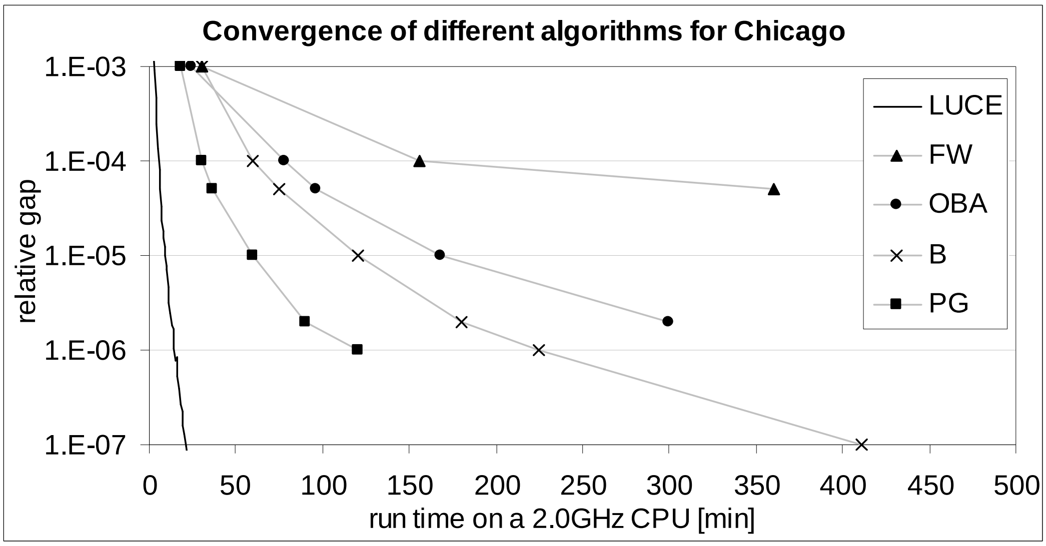

Figure 7. Performance comparison of LUCE to other four assignment algorithms on the Chicago network, whose convergence patterns are retrieved from papers and presentations.

ACKNOWLEDGEMENTS

The key concept of LUCE has been inspired by the conversations had with Hillel Bar-Gera during the First International Symposium on DTA, held in Leeds 2006. The idea of how dealing with bushes was elaborated from Dial's paper.

REFERENCES

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

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

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

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.

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

LeBlanc L. (1973) Mathematical programming algorithms for large scale network equilibrium and network design problems. Ph.D. Dissertation. Department of Industrial Engineering and Management Sciences, Northwestern University.

Nguyen S. (1973) A mathematical programming approach to equilibrium methods of traffic assignment with fixed demands. Publication 138, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Québec.

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