PROJECTION AND FUKUSHIMA'S GAP BASED METHODS FOR THE ASYMMETRIC TRAFFIC ASSIGNMENT PROBLEM

E. Codina[1], G. Ibáñez[2], J. Barceló[1,2]
[1]Department of Statistics and Operations Research and CENIT
[2]Center for Innovation in Transport
Universitat Politècnica de Catalunya

DR 2012/2

PROJECTION AND FUKUSHIMA'S GAP BASED METHODS FOR THE ASYMMETRIC TRAFFIC ASSIGNMENT PROBLEM

Authors:

E. Codina, G. Ibáñez and J. Barceló

Department of Statistics and Operations Research and CENIT
(Center for Innovation in Transport)
Technical University of Catalonia
Campus Nord, Building K2M, Office 306
Jordi Girona 29
08034 Barcelona
Spain

(esteve.codina,gemma.ibanez,jaume.barcelo)@upc.edu
Tel: +34 93 405 4659

The research reported in this paper has been funded by projects SIMETRIA (Ref. P 63/08, 27.11.2008), of the Spanish R+D National Programs and project TRA2008-06782-C02-02.

ABSTRACT

Traffic Assignment is the problem of assigning an Origin to Destination, OD matrix onto a network, under given conditions of using the links of the network, to determine the resulting traffic flows in the network. The underlying hypothesis is that travelers travel from origins to destinations in the network along the available routes connecting them. The characteristics of a traffic assignment procedure are determined by the hypothesis on how travelers use the routes. The main modelling hypothesis is based on the concept of user equilibrium which assumes that travelers try to minimize their individual travel times, that is, travelers chose the routes that they perceive as the shortest under the prevailing traffic conditions. The translation of these modelling hypotheses in terms of a mathematical model leads in the general case to a formulation in terms of a system of variational inequalities that has an equivalent convex optimization model when volume-delay functions are separable. However, the separability assumptions on the volume delay functions may lead quite frequently to modelling inaccuracies due to the over simplifications that they represent when dealing with generalized cost in complex multiclass-multimode planning models, or accounting for priorities at intersections, then the problems become asymmetric in terms of the Jacobian of the cost functions and the associated system of variational inequalities must be solved. Projection and Gap Function methods are among the most computationally efficient algorithms to solve the models. This paper explores a combination of a variant of Fukushima’s projection algorithm and gap Functions. The new algorithm is computationally tested for several large networks and the computational results are presented and discussed.

STATIC TRAFFIC ASSIGNMENT: EQUILIBRIUM ASSIGNMENT

Traffic Assignment is the problem of assigning an Origin to Destination, OD matrix onto a network, under given conditions of using the links of the network, to determine the resulting traffic flows in the network. The underlying hypothesis is that travelers travel from origins to destinations in the network along the available routes connecting them. The characteristics of a traffic assignment procedure are determined by the hypothesis on how travelers use the routes.

The main modelling hypothesis is based on the concept of user equilibrium which assumes that travelers try to minimize their individual travel times, that is, travelers chose the routes that they perceive as the shortest under the prevailing traffic conditions.

<u>User equilibrium modeling hypothesis</u>: the routes chosen by the travellers are those individually perceived as being the shortest under the prevailing traffic conditions. This hypothesis assumes that travellers try to minimize their individual travel times. It was formulated by Wardrop (1) in terms of what is now known as Wardrop's First Principle, or Wardrop's User Equilibrium: *The journey times on all the routes actually used are equal, and less than those which would be experienced by a single vehicle on any unused route.*

Consider a network defined in terms of a graph G=(N,A) with a set of nodes N representing either intersections or centroids, dummy nodes associated with the transportation zones, and a set A of arcs modeling the infrastructure and the connectors linking centroids to the networks. Consider also an OD matrix modeling the demand between transportation zones. The notation used through this paper is the following:

- $W = \{w = (p,q) | w \text{ is the OD pair for origin } p \text{ and destination } q\}$ = Set of all OD pairs
- $\Gamma = \underset{w \in W}{\otimes} \Gamma_w$; $\Gamma_w = \{r \mid r \text{ path joining the OD pair } w \in W\}$
- $A_\ell = \{a \in A \mid \exists w \in W,\ \exists r \in \Gamma_w^\ell,\ a \in r\ (\delta_{ar} = 1)\}$
- $\hat{A}_\ell = \{a \in A_\ell \mid \exists x(a) \text{ priority link over link } a\}$

  We say x(a) is a priority link over link a when x(a) and a are two incident links to the same intersection node, and flow arriving to the intersection from link x(a) has priority for passing through it.
- v_a : link flows, $a \in A$
- $s_a(v)$: cost on link $a \in A$
- h_r^w : path flows through the path $r \in \Gamma$, joining the OD pair $w \in W$
- s_r : cost on path $r \in \Gamma$
- $C(h)$: path costs
- $H = \underset{w \in W}{\otimes} H^w$; $H^w = \left\{ h^w \in \Re^{(n^w)} \mid \sum_{r \in \Gamma_w} h_r^w = g_w,\ h_r^w \geq 0 \right\}$
- $V = \Delta(H)$; $\Delta = (\delta_{ar})$ where $\delta_{ar} = \begin{cases} 1 \text{ if link } a \text{ belongs to path } r \\ 0 \text{ otherwise} \end{cases}$
- N: number of nodes; M: number of links

Wardrop's First Principle can be easily translated in terms of mathematical relationships, Florian and Hearn (2), flows on a network are in equilibrium that satisfies Wardrop's principle when for path flows h_k^* with costs s_k and shortest path costs u_w^* for OD pair w, satisfy:

$$\begin{aligned} &(s_k - u_w^*) h_k^* = 0,\ \forall k \in \Gamma_w,\ w \in W \\ &s_k - u_w^* \geq 0,\ \forall k \in \Gamma_w,\ w \in W \end{aligned} \quad (1)$$

$$\sum_{k\in\Gamma_w} h_k^* - g_w = 0,\ \forall w \in W$$
$$h^* \geq 0,\ u^* \geq 0 \qquad (2)$$

The equations (1) are a direct translation of Wardrop's principle in mathematical terms as a complementarity condition. If path k carries flow, that is $h_k^* \geq 0$, then the complementarity equation (1) is satisfied if and only if $s_k - u_w^* = 0$, that is the cost s_k of using path k for the w-th OD pair is equal to u_w^* the cost of the shortest path for the w-th OD pair, while if $s_k - u_w^* > 0$, that is the cost s_k of using path k is higher than the cost of the shortest path, then to satisfy the complementarity equation $h_k^* = 0$, that is path k doesn't carries any flow, as expected from Wardrop's principle for paths whose costs are not minimal. Constraints state when a flow is feasible or not in terms of flow balance. If Γ_w is the set of all paths for the w-th OD pair then the sum of flows on all paths for the w-th OD pair must equal the demand g_w for w-th OD pair and flows h_k must be non-negative. After some algebra, Florian and Hearn (2), and taking into account the definitional constraint relating flows on arcs a∈ A, where A is the set of arcs in the network, with flows on paths k:

$$v_a = \sum_{w\in W}\sum_{k\in\Gamma_w} h_k\delta_{ak} \quad \text{where} \quad \delta_{ak} = \begin{cases} 1 & \text{if arc a belongs to path k} \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

results

$$s\left(v^*\right)^T\left(v - v^*\right) \geq 0, v \in V \qquad (4)$$
$$V = \left\{ v{:}v_a = \sum_{w\in W}\sum_{k\in\Gamma_w} h_k\delta_{ak}; \sum_{k\in\Gamma_w} h_k = g_w, \forall w \in W \right\}$$

That is Smith's (3) variational inequality. It can be probed that there is no equivalent convex optimization problem unless the cost functions s(v) are separable, that is, their Jacobian is symmetric, Florian and Hearn (2). The simpler separability condition holds when they depend only on the flow in the link: $s_a(v) = s_a(v_a)\ \forall a \in A$, and demands g_w are considered constant, independent of travel costs, the variational inequality formulation has the following equivalent convex optimisation problem, Patriksson (4), Florian and Hearn (2):

$$Min\ S(v) = \sum_{a\in A}\int_0^{v_a} s_a(x)dx$$
$$s.t.\ \sum_{k\in\Gamma_w} h_k = g_w, \forall w \in W \qquad (5)$$
$$h_k \geq 0, k \in \Gamma_w, w \in W$$

and the definitional constraint of v_a (3). Although the traffic assignment problem is a special case of non-linear multi-commodity network flows problem, and may be solved by any of the methods used to solve this problem, more efficient algorithms, based on an adaptation of the linear approximation method of Frank and Wolfe, Frank and Wolfe (5) have been developed, Leblanc at al. (6), Florian (7). Other efficient algorithms based on the restricted simplicial approach have been developed, exploiting the properties of the convex polyhedron of feasible solutions defined by constraints (2), Hearn et al. (8), Lawphongpanich and Hearn (9), or on an adaptation of the parallel tangents method (PARTAN), Florian et al. (10).

However, the separability assumptions on the volume delay functions may lead quite frequently to modeling inaccuracies due to the over simplifications that they represent when dealing with generalized cost in complex multiclass-multimode planning models, Wu and Florian (11), Florian and He (12), or when it is recommendable to deal explicitly with delays at intersections. For example in the case for an unsignalized intersection shown in Figure 1, if we consider x(a) as the link with priority over link a, the delay at the intersection can be formulated as Harders (13):

$$d_a(v) = t_f + 3600h \frac{1 - e^{-\left(t_c \frac{v_{x(a)}}{3600h} + t_f \cdot \frac{v_a}{3600h}\right)}}{c_a(v) - v_a} \tag{6}$$

Where t_c is the critical gap time on the x(a) movement, and t_f is the joining time on the main link. Constant h is the time period in hours, and $c_a(v)$ are the intersection capacities which must satisfy $c_a(v) - v_a \geq 0$. These constrictions are convex and can be aproximated by $c_a(v_{x(a)}) \approx c_a^0 - k_{x(a)} v_{x(a)}$, using $k_{x(a)} = \frac{c_a^0}{c_{x(a)}}$, where $c_{x(a)}$ is the capacity on link x(a) and c_a^0 is a general capacity for an intersection. Thus, the constriction can be rewritten as $k_{x(a)} v_{x(a)} + v_a \leq c_a^0$.

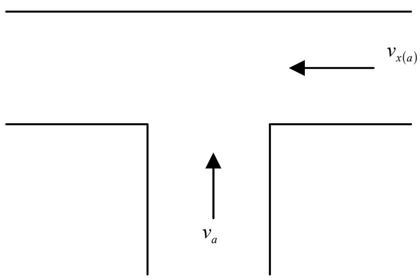

Figure 1.

In consequence, it is possible to reformulate Harders delay expression as:

$$d_a(v) \approx t_f + 3600h \frac{1 - e^{-\left(t_c \frac{v_{x(a)}}{3600h} + t_f \cdot \frac{v_a}{3600h}\right)}}{c_a^0 - v_a - k_{x(a)} v_{x(a)}} \tag{7}$$

However, this is an asymptotic expression whose use could carry numerical problems. For this reason we considered the option of using the following related expression for asymmetrical delays that does not contain any asymptotic relationship:

$$d_a(v) = t_f + \frac{1}{\theta} \ln\left(1 + e^{\theta \cdot b \cdot (x(v) - 1)}\right) \tag{8}$$

Where $x(v) = \frac{v_a + k_{x(a)} v_{x(a)}}{c_a^0 h}$ and b, θ are parameters properly chosen. In the developments that follow we have used this asymmetric cost formulation for links without priority while for the priority links we use the symmetrical expression of cost:

$$s_a(v) = t_0 \cdot \left(1 + \alpha\left(\frac{v_a}{c_a \cdot h}\right)^{\beta}\right) \tag{9}$$

Where c_a is the capacity on link a and t_0 is the link travel time under free flow conditions; α and β are variable parameters.

When the intersection has more than one priority link, as shown in Figure 2, then $k_{x(a)}v_{x(a)}$ is replaced by $\sum_{\substack{\forall x(a)\text{ priority}\\ \text{link over link } a}} k_{x(a)}v_{x(a)}$ in (7) for all priority links x(a).

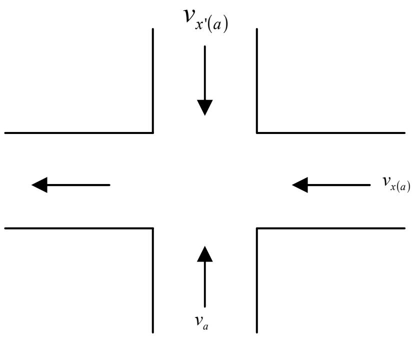

Figure 2.

ALGORITHMIC APPROACHES TO SOLVE THE FORMULATION OF USER EQUILIBRIUM ASSIGNMENT IN TERMS OF VARIATIONAL INEQUALITIES

The above examples justify the practical interest in finding efficient computational solutions to transport planning problems modeled in terms of asymmetric assignment models to aexplicitly account for interactions as the described and referenced. From a research point of view the most appealing approach is the search for efficient algorithms to solve the Smith's variational inequalities formulation of the asymmetric user equilibrium as formulated in (4). We will reformulate it as:

$$\begin{aligned} &\textit{Find } v^* \in V \textit{ s.t.}: \\ &\qquad s(v^*)^{\mathrm{T}}(v - v^*) \geq 0, \forall\, v \in V \end{aligned} \tag{10}$$

Or in its variational form:

$$\begin{aligned} &\textit{Find } v^* \in V \textit{ s.t.}: \\ &\qquad 0 \in s(v^*) + N_V(v^*) \end{aligned} \tag{11}$$

Where $N_V(v^*)$ is the normal set to set V at point v^*.

Following García and Marín (14) and Patriksson (15), algorithms to solve this variational problem can be classified in:

- Relaxation methods
- Projection methods
- Methods based on gap functions for variational inequalities :
    - Newton descent methods based on primal gap

- Primal descent algorithms based on the generalized primal gap function of Patriksson
- Algorithms for minimizing Marcotte's gap Function

This paper addresses algorithmically a combination of projection and gap function methods for asymmetric problems.

Gap functions

A function G is a gap function $G : V \to \mathrm{R}$, for variational inequalities (6) if:

a) Is nonnegative
b) It vanishes only at points $v^* \in V$ that are solution of the variational inequality.

Descent methods for variational inequalities are based on various types of gap functions, the most commonly used are the primal gap function G_P:

$$G_P(v) = Sup_{u \in V}\ s(v)^{\mathrm{T}}(v-u) \tag{12}$$

The dual gap function G_D:

$$G_D(v) = Inf_{u \in V}\ s(u)^{\mathrm{T}}(u-v) \tag{13}$$

The Marcotte's gap function $\tilde{G}_M$, defined after a function $\varphi(u,v) : V \times V \to \mathrm{R}$, continuously differentiable and strictly convex in u, $\forall v \in V$ and a parameter $\rho > 0$.

$$\tilde{G}_M(u) \overset{\Delta}{=} Max_{v \in V}\{c(v)^{\mathrm{T}}(v-u) - \rho\varphi(u,v)\} \tag{14}$$

The Smith's gap function G_S, defined from an integer $p \geq 0$ needs the set of vertices of set V, assumed to be polyhedral .

$$G_S(v) = \left(\sum_{\hat{v} \in V}[s(v)^{\mathrm{T}}(v-\hat{v})]_+^p\right)^{\frac{1}{p}} \tag{15}$$

The proposed algorithm uses the gap function with the method of cost approximations developed by Patriksson (14) which includes as a particular case the following descent direction algorithm for the primal gap G_P, García and Marin (13).

Descent direction algorithm for the primal gap: Newton Method

<u>k-th iterative step</u>

1. Build the linear approximation s_L^k to s at point v^k:

$$s_L^k(v) = s(v^k) + \nabla s(v^k)^T (v - v^k)$$

2. Solve the variational inequality with the linear approximation $s_L^k(.)$:

Find $v*$, s.t.: $s_L^k(v*)^T(v - v*) \geq 0, \forall v \in \upsilon \to \hat{v}^k$

3. If $G_P(\hat{v}^k) \leq \frac{1}{2} G_P(v^k)$ then make λ^k=1. Otherwise perform a linear search: $Min_{0 \leq \lambda \leq 1} G_P[v^k + \lambda(\hat{v}^k - v^k)] \to \lambda^k$

4. $v^{k+1} = v^k + \lambda^k(\hat{v}^k - v^k)$

Projection methods

For variational inequalities (10) projection algorithms project at each iteration ℓ, on the polyhedron V the point $\hat{v}^{\ell} = v^{\ell} + \rho Q s(v^{\ell})$, where Q is a symmetric matrix definite positive and $\rho > 0$ is a suitable scale parameter. The projection of point $\hat{v}^{\ell}$ on the polyhedron V under the norm $\| \cdot \|_{Q^{-1}}$ is equivalent to solve the quadratic problem:

$$\underset{y \in V}{Min} \quad \frac{1}{2}(y - \hat{v}^{\ell})^{\mathrm{T}} Q^{-1} (y - \hat{v}^{\ell}) \quad s.t. \tag{16}$$

Or equivalently:

$$\underset{y \in V}{Min} \quad \frac{1}{2}(y - v^{\ell})^{\mathrm{T}} Q^{-1} (y - v^{\ell}) + \rho s(v^{\ell})^{\mathrm{T}} (y - v^{\ell}) \quad s.t. \tag{17}$$

Or solving the variational inequality:

$$\begin{aligned} &Find\ v^{*} \in V\ s.t.: \\ &0 \in \hat{s}_{\ell}(v^{*}) + N_V(v^{*}) \end{aligned} \tag{18}$$

where $\hat{s}_{\ell}(\cdot)$ is the functional given by $\hat{s}_{\ell}(y) \stackrel{\Delta}{=} s(v^{\ell}) + \frac{1}{\rho} Q^{-1}(y - v^{\ell})$.

When applied to traffic assignment problems the separable structure of the set of feasible flows in the paths and the resulting projection algorithm is:

Initialization: Find an initial set of feasible acyclic paths for each OD pair $w \in W$, and load on them the demand g_w → Γ_w^0 and $h^0 \in H^0$; $\ell = 1$.

At iteration ℓ **:**

1. For each OD pair $w \in W$ determine the set of paths Γ_w^{+} used on iteration $\ell - 1$
2. Increase the set of paths finding the shortest paths $\breve{k} \in \Gamma_w,\ w \in W$ with costs $s\left(\Delta h^{\ell-1}\right)$ → solution flows $\hat{h}_{SP}^{\ell}$.
3. Add new paths detected: $\Gamma_w^{+} = \Gamma_w^{+} \bigcup \{\breve{k}\},\ \forall w \in W$
4. If $G_{rel}\left(h^{\ell-1}\right) = \dfrac{C\left(h^{\ell-1}\right)^{T}\left(h^{\ell-1} - \hat{h}_{SP}^{\ell}\right)}{C\left(h^{\ell-1}\right)^{T}\hat{h}_{SP}^{\ell}} \le \varepsilon$ STOP. (19)
5. For each $w \in W$, solve the quadratic problem:
$$[Q] \quad \underset{h \in H^w}{Min} \sum_{k \in \Gamma_w^{+}} \left((h_k - h_k^{(\ell-1)}) s_k(h^{(\ell-1)}) + {q_k}\big/{2\rho}\,(h_k - h_k^{(\ell-1)})^2 \right) \quad s.to: \sum_{k \in \Gamma_w^{+}} h_k = g_w,\ h_k \ge 0 \quad \to \text{flows } \hat{h}^{\ell} \tag{20}$$
6. Linear Exploration (optional): $\underset{0 \le \theta \le 1}{\text{Min}}\ G_{\ell}(h(\theta))$ → $\tilde{\theta}$; where $h(\theta) = h^{\ell-1} + \theta\left(\hat{h}^{\ell} - h^{\ell-1}\right)$
   Update flows: $h^{\ell+1} = h^{\ell-1} + \tilde{\theta}\left(\hat{h}^{\ell} - h^{\ell-1}\right)$.
7. $\ell \leftarrow \ell + 1$. Go to 1)

Whose computational performance strongly depends on the solution of the quadratic problem [Q] (20) at step 3. To solve it we used an adaptation of the algorithm proposed by Wu (16) for which the problem is strictly convex and therefore Karush Kuhn Tucker (KKT) conditions are necessary and sufficient for optimality, and by applying them it is easy to probe that solutions for path flows are:

$$h_k(\lambda) = \max\left\{0; \frac{\lambda - s_k\left(h^\ell\right)}{q_k/\rho} + h_k^\ell\right\} \tag{21}$$

Where λ is the parameter which satisfies $\sum_{k\in K} h_k(\lambda) = g_w$, and it is found by an iterative process over the function $\varphi(\lambda) = \sum_{k\in K} h_k(\lambda)$, that is a monotonous undecreasing and piecewise linear function.

To compute the value of parameter λ we propose the following variant of Wu's algorithm. Let λ^0 be the initial value defined by:

$$\lambda^0 = \frac{g_w + \sum_{h_k>0}\left(\rho\frac{s_k(h)}{q_k} - h_k\right)}{\sum_{h_k>0}\rho/q_k} \tag{22}$$

With this initial value for λ^0, update flows $h_k\left(\lambda^0\right)\ \forall k \in \Gamma_w$ as described above in (21), evaluate $\varphi\left(\lambda^0\right) = \sum_{i\in I_0} h_i\left(\lambda^0\right)$ and register set $I_\ell = I\left(\lambda^\ell\right) = \left\{\, i \in \Gamma_w^\ell \,\middle|\, h_i\left(\lambda^\ell\right) > 0\right\}$ for $\ell = 0$. The rest of the algorithm is:

1. If $\varphi\left(\lambda^\ell\right) = g_w$ STOP.
2. Calculate $\lambda^{\ell+1} = \dfrac{g_w + \sum_{k\in I_\ell}\left(\rho\dfrac{s_k(h)}{q_k} - h_k\right)}{\sum_{k\in I_\ell}\rho/q_k}$ (23)
3. Update flows and evaluate $\varphi\left(\lambda^{\ell+1}\right) = \sum_{i\in I_{\ell+1}} h_i\left(\lambda^{\ell+1}\right)$
4. Register set $I_{\ell+1} = I\left(\lambda^{\ell+1}\right)$
5. $\ell \leftarrow \ell + 1$ and go to 1).

<u>Line Search</u>

In order to speed up the projection algorithm we have introduced an optional module of linear exploration which tries to find a new and better point using the information relating to both the current iteration and the previous one.

Consider the directional derivative on the descent direction $\hat{v}^\ell - v^\ell$:

$$G_p'\left(v^\ell; \hat{v}^\ell - v^\ell\right) = \left[s\left(v^\ell\right) + \nabla s\left(v^\ell\right)^T\left(v^\ell - v^*\left(v^\ell\right)\right)\right]^T\left(\hat{v}^\ell - v^\ell\right) \tag{24}$$

Where $v_a^\ell\ \ \forall a \in A$ are the link flows obtained on the last iteration, $\hat{v}_a^\ell\ \ \forall a \in A$ are the new link flows obtained from the quadratic problem, and $v_a^*\left(v^\ell\right)\ \ \forall a \in A$ are the link flows that we would obtain solving the shortest path problem with costs $s\left(v^\ell\right)$.

The linear exploration implemented in this case is the following:

0. Initialization: $v^{\ell,0}=\hat{v}^{\ell}$, $G_1=G_2=G_p\left(v^{\ell,0}\right)=G_p\left(\hat{v}^{\ell}\right)$, $G_0=\tilde{G}=G_p\left(v^{\ell}\right)$, $\tilde{\theta}=1$, $\theta_1=\theta_2=1$, p=0. Improved=false.
1. If $G_p\left(v^{\ell,p}\right)\le G_p\left(v^{\ell}\right)+\rho^p\eta\ G'_p\left(v^{\ell};\hat{v}^{\ell}-v^{\ell}\right)$ or p=n STOP.
2. Update flows: $v^{\ell,p+1}=v^{\ell}+\rho^{p+1}\left(\hat{v}^{\ell}-v^{\ell}\right)$; $p\leftarrow p+1$
3. $G_2=G_1$, $\theta_2=\theta_1$, $\theta_1=\rho^p$
4. Calculate $G_1=G_p\left(v^{\ell,p}\right)$
5. If $G_p\left(v^{\ell,p}\right)<\tilde{G}$ then: $\tilde{G}=G_p\left(v^{\ell,p}\right)$, $\tilde{v}=v^{\ell,p}$, $\tilde{\theta}=\theta_1=\rho^p$, improved=true.
6. Go to 1.

Where ρ,θ,n are parameters properly chosen. After this first phase of linear exploration, there is another phase which gives a second opportunity to reduce primal gap:

7. If improved=false then: $\tilde{\theta}=\dfrac{1}{\mu}\left[\dfrac{G_0-G_1+\beta_1\theta_1}{\beta_1+|\beta_0|}\right]$ (25)

Where $\beta_0=\eta\ G'_p\left(v^{\ell};\hat{v}^{\ell}-v^{\ell}\right)$ and $\beta_1=\dfrac{G_2-G_1}{\theta_2-\theta_1}$; μ is a variable parameter. After the linear exploration we take as a new point the updating of flows:

$h^{\ell+1}=h^{\ell}+\tilde{\theta}\left(\hat{h}^{\ell}-h^{\ell}\right)$ in the paths space (26)

$v^{\ell+1}=v^{\ell}+\tilde{\theta}\left(\hat{v}^{\ell}-v^{\ell}\right)$ in the links space (27)

<u>Finding initial feasible paths</u>

The way of finding an initial set of feasible acyclic paths for each OD pair is the algorithm described below:

Consider the function: $f(v)=\dfrac{1}{2}\sum_{a\in\hat{A}_{\ell}}\max^2\left\{0,v_a+k_a v_{x(a)}-c_a^0\right\};\quad v\in V_{\Gamma^{\ell}}$ (28)

Whose gradient is: $\nabla f(v)=\begin{cases}0 & \text{if } a\notin\hat{A}_{\ell}\\ v_a+k_a v_{x(a)}-c_a^0 & \text{if } v_a+k_a v_{x(a)}\ge c_a^0\\ 0 & \text{if } v_a+k_a v_{x(a)}<c_a^0\end{cases}$ (29)

<u>Algorithm</u>:

0) Determine a set of initial paths Γ^0 so that $\left|\Gamma_w^0\right|>1\ \ \forall w\in W$

<u>Iteration k</u>:

1) Solve $\nabla f\left(v^*\right)^T\left(v-v^*\right)\ge 0\ \ v\in V_{\Gamma^k}$
2) - If $f\left(v^*\right)=0$ then STOP (we have an initial set of feasible paths)
   - If $f\left(v^*\right)>0$ then update costs $s_a\left(v^*\right)=\nabla f\left(v^*\right)$
3) Find the shortest paths with new costs: $\breve{k}\in\Gamma_w$, $w\in W$. Add new paths detected: $\Gamma_w^k=\Gamma_w^{k-1}\bigcup\{\breve{k}\}$, $\forall w\in W$

4) $k \leftarrow k+1$. Go to 1)

Shortest paths with turnings

The above description of the various components of the proposed algorithm relay iteratively on the computation of shortest paths, however, taking into account the causes of the asymmetries it is obvious that practical realistic approaches must find the way of computationally dealing with them, that is dealing explicitly with the information about forbidden or penalized turnings in the computation of shortest path. Our approach uses an ad hoc version of the algorithm proposed by Ziliaskopoulos and Mahmassani (17) which incorporates information about turnings in graphs and suggests structures to store this information.

The specific notation for this adapted algorithm is the following:

$E[i]$ - set of emerging nodes from node i

$M[i, j]$ - set of possible nodes k from a turning (i, j, k)

$L(i, j)$ - ordinal of link (i, j) in the data structure

$p_A()$ - link predecessor vector. At the end the algorithm shows the path tree over the expanded graph

$p_D(d)$ - pointer to the link predecessor vector which indicates the last link in the path from origin r to destination d.

D - set of destinations

I - set of nodes where there is a penalized or forbidden turning

γ_{ijk} - cost of turn (i, j, k).

γ^{def} - cost of undeclared and permitted turnings (tipically zero)

λ_{ij} - cost from origin r until exiting node i, when going out in direction to node j

$\tilde{\lambda}_j$ - cost from origin r until node j (without entering into it)

The adaptation for the shortest path algorithm is described below. Links are added to set S according to criteria of an L-Deque shortest path algorithm, Gallo and Pallotino (18).

$$
\begin{array}{ll}
1- & \underline{Initialization}: \textit{for the origin } r, \textit{consider } S = \{ (r,j) \mid j \in E[r] \} \\
 & \lambda_{rj} = 0, \forall (r,j) \in S;\ \lambda_{i,\ell} = \infty, \forall (i,\ell) \in A, i \neq r \\
2- & \underline{While}\ \ S \neq \varnothing\ \ \underline{do}: \\
 & \quad Select\ (i,j) \in S;\ S = S \setminus \{(i,j)\}; \\
 & \quad \underline{If}\ \ j \in D\ \&\ \tilde{\lambda}_j > c_{ij} + \lambda_{ij}\ \underline{then}: \\
 & \qquad \tilde{\lambda}_j = c_{ij} + \lambda_{ij};\ p_D(j) = L(i,j) \\
 & \quad \underline{ElseIf}\ \ j \in I\ \underline{then}: \\
 & \qquad For\ k \in M[(i,j)]\ do: \\
 & \qquad\quad If\ \ \lambda_{jk} > \lambda_{ij} + c_{ij} + \gamma_{ijk}\ then: \\
 & \qquad\qquad \lambda_{jk} = \lambda_{ij} + c_{ij} + \gamma_{ijk};\ p_A(L(j,k)) = L(i,j); S = S \cup \{(j,k)\} \\
 & \qquad\quad EndIf \\
 & \qquad EndFor \\
 & \quad \underline{Else}\ \underline{then}: \\
 & \qquad For\ k \in E[j]\ do: \\
 & \qquad\quad If\ \ \lambda_{jk} > \lambda_{ij} + c_{ij} + \gamma^{def}\ then: \\
 & \qquad\qquad \lambda_{jk} = \lambda_{ij} + c_{ij} + \gamma^{def};\ p_A(L(j,k)) = L(i,j); S = S \cup \{(j,k)\} \\
 & \qquad\quad EndIf \\
 & \qquad EndFor \\
 & \quad \underline{EndIf} \\
 & \underline{EndWhile}
\end{array}
$$

A FUKUSHIMA'S GAP BASED ALGORITHM

The projection algorithm described in the previous section has been the basis for a modified version of Fukushima's projection algorithm using the Fukushima's Gap, Fukushima (19), (20), defined as follows:

$$G^F(h) = -\underset{f \in H}{\mathrm{Min}} \left[C(h)^T (f-h) + \rho/_2 \| f-h \|^2 \right] \tag{30}$$

Based on this new measure of the gap, and using a similar structure to the previous projection algorithm, we have developed the following modified projection algorithm:

<u>Initialization</u>: Find an initial set of feasible acyclic paths for each OD pair $w \in W$, and load on them the demand g_w → Γ_w^0 and $h^0 \in H^0$; $\ell = 1$.

<u>Iteration</u> ℓ :

1. For each OD pair $w \in W$ determine the set of paths Γ_w^+ used on iteration $\ell - 1$
2. Increase the set of paths finding the shortest paths $\breve{k} \in \Gamma_w$, $w \in W$ with costs $s(\Delta h^{\ell-1})$ → solution flows $\hat{h}_{SP}^{\ell}$.
3. Add new paths detected: $\Gamma_w^+ = \Gamma_w^+ \bigcup \{\breve{k}\}, \forall w \in W$
4. If $G_{rel}(h^{\ell-1}) = \dfrac{C(h^{\ell-1})^T (h^{\ell-1} - \hat{h}_{SP}^{\ell})}{C(h^{\ell-1})^T \hat{h}_{SP}^{\ell}} \leq \varepsilon$ then STOP. (31)
5. Equilibrate the existing paths → (G.F.) problem: $\underset{h \in H^\ell}{\mathrm{Min}}\ \mathrm{G}_\ell^{\mathrm{F}}(h)$ → new flows h^ℓ

(G.F.) problem

Initialization: $y^0 = h^{\ell-1}$; $k = 0$
Iteration k:
a) Solve:

$$\underset{f\in H^{\ell}}{\text{Min}}\ C(y^k)^T (f - y^k) + \rho/2 \left\| f - y^k \right\|^2 \tag{32}$$

New flows: $\tilde{f}^k \equiv f^*(y^k)$

b) Linear Exploration (optional): $\underset{0\le\alpha\le1}{\text{Min}}\ G_{\ell}^{F}(y(\alpha)) \rightarrow \alpha^*$; where $y(\alpha) = y^k + \alpha^*(\tilde{f}^k - y^k)$.

c) Update flows $y^{k+1} := y^k + \alpha^*(\tilde{f}^k - y^k)$ and evaluate costs $C(y^{k+1})$

d) If $\left| \dfrac{G_{\ell}(y^{k+1})}{C(y^{k+1})^T y^{k+1}} \right| \le \varepsilon$ STOP. Get $y^* \approx y^{k+1}$.

e) $k \leftarrow k+1$. Go to a)

6. $\ell \leftarrow \ell + 1$. Go to 1.

Again, the computational performance of the algorithm strongly depends on the solution of the quadratic problem (32). To solve it we use the same method described in the previous section, an adaptation of the algorithm proposed by Wu (16). In this case it can be probed that solutions for path flows are:

$$h_k(\lambda) = \max\left\{0; \frac{\lambda - s_k(h^{\ell})}{\rho} + h_k^{\ell}\right\} \tag{33}$$

And λ is the parameter which make flows satisfy $\sum_{k\in K} h_k(\lambda) = g_w$, and it is found by the iterative process over the function $\varphi(\lambda) = \sum_{k\in K} h_k(\lambda)$ described in the previous section. In this case we have:

$$\lambda^{\ell+1} = \frac{g_w + \sum_{k\in I_{\ell}} \left(\dfrac{s_k(h)}{\rho} - h_k \right)}{\sum_{k\in I_{\ell}} 1/\rho} \tag{34}$$

In this case the linear search can be simpler that the one used in the previous projection algorithm, the reason is that now the differentiability of Fukushima's gap function can be explicitly used in the linear search. Let's consider Fukushima's gap function evaluated on the point $y(\alpha) = y^k + \alpha(\tilde{f}^k - y^k)$:

$$C_k^T\left(y^k + \alpha(\tilde{f}^k - y^k)\right) + \rho/2 \left\| y^k + \alpha(\tilde{f}^k - y^k) \right\|^2 \tag{35}$$

Then, manipulating this expression we obtain a quadratical function on α:

$$\alpha^2 \rho/2 \left\| \tilde{f}^k - y^k \right\|^2 + \alpha\left[C_k^T(\tilde{f}^k - y^k) + \rho y^{kT}(\tilde{f}^k - y^k) \right] + C_k^T(y^k) + \rho/2 \left\| y^k \right\|^2$$

Whose minimum is on $\alpha^* = \dfrac{-\left[C_k^T(\tilde{f}^k - y^k) + \rho y^{kT}(\tilde{f}^k - y^k) \right]}{\rho \left\| \tilde{f}^k - y^k \right\|^2}$ (36)

Taking $\alpha^* = Min\{\alpha^*, 1\}$ and updating flows:

$$y^{k+1} := y^k + \alpha^*(\tilde{f}^k - y^k) \tag{37}$$

At the beginning of the algorithm, the process to find the initial set of feasible acyclic paths for each OD pair is exactly the same described before for the projection algorithm, as well as the adaptation for the shortest path problem in a graph with turning penalties and forbidden turns. Again, the method used to solve shortest paths has been L-Deque.

COMPUTATIONAL RESULTS

We have used three different networks to conduct the computational experiments whose results are reported in this paper, the networks of Winnipeg (Canada), Terrassa (Spain) and Hessen (Germany). Winnipeg is the smallest one in terms of nodes and links, and in spite of not having any turning restriction it is the one that has a higher quantity of non priority links (more than 15%). Consequently, Winnipeg is the network which uses more asymmetric delay functions on its links. Terrassa is bigger than Winnipeg, it has restrictions at some turnings and near a 7% of non priority links that use asymmetric delay functions. Hessen is the biggest one, it also has turning restrictions and almost a 6% of non priority links. Table 1 shows the main characteristics of the networks:

Network	nodes	links	centroids	OD pairs	forbidden or penalized turnings	non priority links (%)
Winnipeg	1057	2535	154	4345	0	15,58%
Terrassa	1609	3264	55	2215	1103	7,05%
Hessen	4660	6674	245	17213	7054	5,75%

Table 1: characteristics of the networks included in this paper.

To measure the efficiency of the algorithms we define two different formulae to calculate the gap. The first one ($RGap_1$) is the typical primal gap on its relative form. It is also the measure used in both algorithms to decide at the end of every iteration whether to continue or stop because of the last solution found is rather good. $RGap_1$ is defined as follows:

$$RGap_1(v) = \frac{s(v)^{\mathrm{T}}(v-u)}{s(v)^{\mathrm{T}}u} \qquad (38)$$

Where v are the link flows found at the current iteration, $s(v)$ are the corresponding costs and u the minimum link flows resulting from the shortest path calculation when link costs are $s(v)$.

The second formula ($RGap_2$) is also a gap measure but calculated in this case on the path space and it is weighted by the demand between OD pairs. This measure is the one used in software EMME2 to evaluate whether a solution is good enough or still not. $RGap_2$ is defined as follows:

$$RGap_2(v) = \frac{\sum_{w\in W}\frac{1}{g_w}\left(\sum_{k\in\Gamma_w} C(h_k)h_k - C_{k_{SP}}(h)^T h_{SP}\right)}{\sum_{w\in W}\frac{1}{g_w} C_{k_{SP}}(h)^T h_{SP}} \qquad (39)$$

Where h are the path flows found at the current iteration, $C(h)$ the corresponding costs, h_{SP} the path flows resulting from the shortest path routine when link costs are the associated to the current path flows h and g_w is the demand corresponding to the OD pair $w \in W$. As flows h_{SP} correspond to the assignment of all the demand g_w through the minimum cost path and zero flow through the rest of paths joining the OD pair $w \in W$, it is possible to rewrite formula (39) as the following:

$$RGap_2(v) = \frac{\sum_{w \in W} \left(\frac{1}{g_w} \sum_{k \in \Gamma_w} C(h_k) h_k - C_{k_{SP}}(h) \right)}{\sum_{w \in W} C_{k_{SP}}(h)} \tag{40}$$

Where $C_{k_{SP}}(h)$ is the cost corresponding to shortest path k_{SP}.

The conducted computational experiments have consisted of executing the two algorithms, the projection and the modified projection, for the three networks. In order to compare the performance of both algorithms in terms of the quality of the achieved solutions either the number of iterations or the cpu time has been fixed getting the results achieved for the same number of iterations or the same computational time for each algorithm with each network.

The Figures 3 and 4 depict respectively how the measures $RGap_1$ and $RGap_2$ descend as a function of the number of iterations for the Modified Projection Algorithm. We show them for two different networks, Figure 3 shows the descent of $RGap_1$ in the Winnipeg network and Figure 4 shows the descent of $RGap_2$ in the Hessen network. Both graphs are in the logarithmic scale to show more clearly the behavior of the measure. We observe that the speed of descent is very fast at the first iterations and after is always descending but slowly.

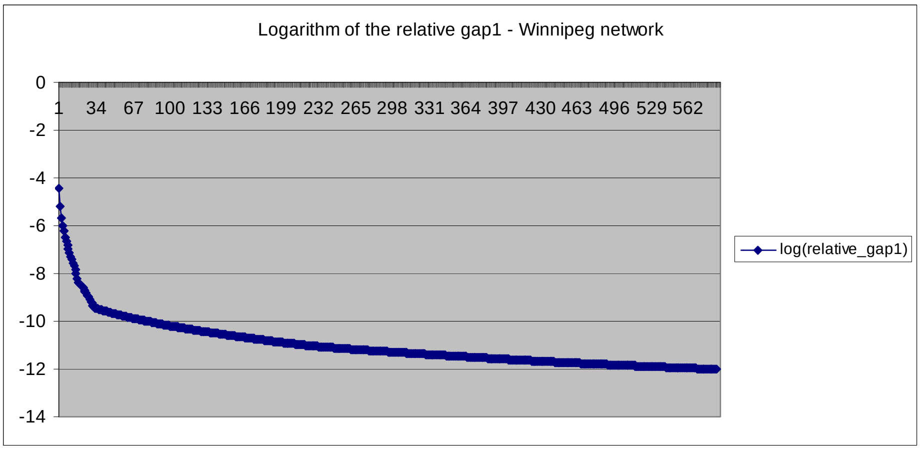

Figure 3

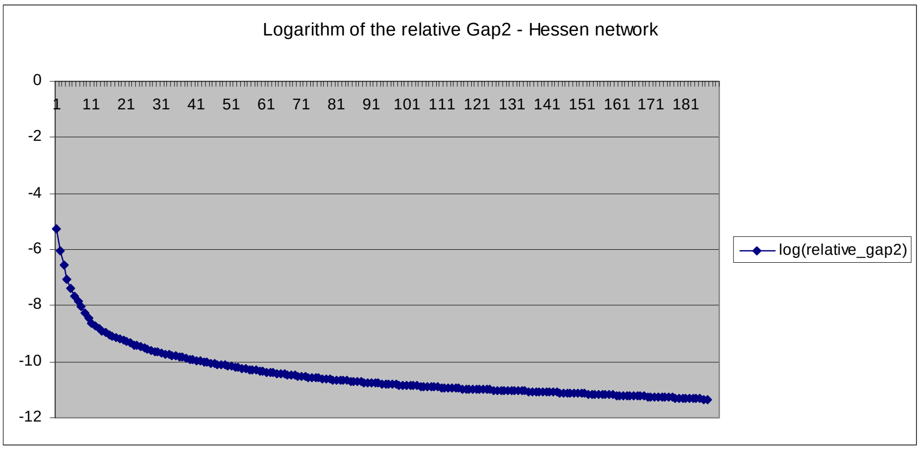

Figure 4

The comparison between the results for both algorithms is presented in tables 2, 3 and 4, which show for each algorithm and each network four different measures:

- number of iterations that the algorithm did until it reached the objective established in that table
- the value of the relative gap_1 reached at the end of all the iterations
- the value of the relative gap_2 reached at the end of all the iterations
- the CPU time in seconds that the main modules of the algorithm (the quadratic problem, the linear exploration and the shortest path problem) used until the last iteration

In Table 2 we fix a CPU time of 10 seconds and we compare the measures for the relative gaps obtained at the end of this CPU time. For all networks we can see that both the $RGap_1$ and the $RGap_2$ value are always lower using the new modified algorithm. These values are especially significant in the case of $RGap_1$ for Terrassa network, which improves from 3.93E-03 to 9.08E-04, thus gaining an order of magnitude using the new algorithm. The same happens in the case of $RGap_2$ for Hessen network, which also gains an order of magnitude with the new algorithm, improving from 1.40E-04 to 5.70E-05.

Algorithm	Network	Number of iterations	Relative Gap_1	Relative Gap_2	CPU time (s)
PROJECTION ALGORITHM	Winnipeg	63	3.93E-05	2.50E-05	10
	Terrassa	56	3.93E-03	3.69E-04	10
	Hessen	9	7.08E-04	1.40E-04	10

Algorithm	Network	Number of iterations	Relative Gap_1	Relative Gap_2	CPU time (s)
NEW ALGORITHM	Winnipeg	119	3.21E-05	2.00E-05	10
	Terrassa	216	**9.08E-04**	1.17E-04	10
	Hessen	34	3.65E-04	**5.70E-05**	10

Table 2: comparison of the computational results obtained for the same CPU time (10 seconds)

Table 3 is similar to Table 2. In this case we fix a CPU time of 15 seconds and we also compare the relative gaps obtained at the end of this CPU time. Again we can see for all networks that both the $RGap_1$ and the $RGap_2$ values are always lower using the new algorithm. Moreover, we obtain particularly interesting values in the case of Terrassa network, where it improves the result by an order of magnitude both for $RGap_1$ value and $RGap_2$ value.

Algorithm	Network	Number of iterations	Relative Gap_1	Relative Gap_2	CPU time (s)
PROJECTION ALGORITHM	Winnipeg	95	2.68E-05	1,59E-05	15
	Terrassa	84	2.19E-03	2,22E-04	15
	Hessen	14	4.3E-04	8,95E-05	15

Algorithm	Network	Number of iterations	Relative Gap_1	Relative Gap_2	CPU time (s)
NEW ALGORITHM	Winnipeg	156	2.47E-05	1.48E-05	15
	Terrassa	265	**6.62E-04**	**9.12E-05**	15
	Hessen	47	3.08E-04	4.16E-05	15

Table 3: comparison of the computational results obtained for the same CPU time (15 seconds)

Finally, in Table 4 what we fix is the $RGap_2$ value, which tends to be always smaller than $RGap_1$, and in this case the table shows the results from the first iteration where appears a

RGap$_2$ lower than 10^{-4}. We observe that in all cases the CPU time used to reach this value is smaller with the new algorithm. For Winnipeg network this value is reduced to a 54% of the time used by the projection algorithm, and for Terrassa and Hessen networks we obtain a CPU time lower than half the time used by the projection algorithm (reduction of 47.5% and 39.7% respectively). In terms of absolute value the most significant reduction is the obtained for the Terrassa network, with a difference of 14.56 seconds of CPU time.

Algorithm	Network	Number of iterations	Relative Gap$_1$	Relative Gap$_2$	CPU time (s)
PROJECTION ALGORITHM	Winnipeg	13	2.08E-04	9.81E-05	2.141
	Terrassa	153	6.82E-04	9.95E-05	27.735
	Hessen	13	5.15E-04	9.72E-05	14.612

Algorithm	Network	Number of iterations	Relative Gap$_1$	Relative Gap$_2$	CPU time (s)
NEW ALGORITHM	Winnipeg	20	2.07E-04	9.69E-05	**1.172**
	Terrassa	248	7.44E-04	9.96E-05	**13.175**
	Hessen	20	5.21E-04	9.69E-05	**5.803**

Table 4: comparison of the computational results obtained for the same relative gap ($<10^{-4}$)

CONCLUSIONS AND FUTURE RESEARCH

In conclusion, the new modified projection algorithm proposed in this paper gives good results solving the asymmetrical traffic assignment problem in rather large networks with many OD pairs and turning penalties. Further research will consist in improving the data structures which support all the information used by the algorithms. We suspect that in this way we could obtain even better results for the new algorithm based on the modification of Fukushima's method.

ACKNOWLEDGEMENTS

The research reported in this paper has been funded by projects SIMETRIA (Ref. P 63/08, 27.11.2008), of the Spanish R+D National Programs and project TRA2008-06782-C02-02.

REFERENCES

(1) J.G. Wardrop, (1952), Some Theoretical Aspects of Road Traffic Research, Proc. Inst. Civil Engineers, Part II, pp. 325-378.
(2) M. Florian and D. Hearn (1995), Network Equilibrium Models and Algorithms, Chapter 6 in: M.O. Ball et al., Eds., Handbooks in OR and MS, Vol.8, Elsevier Science B.V.
(3) M.J. Smith (1979), Existence, Uniqueness and Stability of Traffic Equilibria, Transportation Research B, 1B, pp 295-304.
(4) M. Patriksson (1994), The traffic Assignment Problem: Models and Methods, VSP B.V.
(5) M. Frank and P. Wolfe (1956), An Algorithm for Quadratic Programming, Naval Research Logistic Quarterly 3, pp 95-110.
(6) L.J. LeBlanc, E.K. Morlok and W.P. Pierskalla (1975), An Efficient Approach for Solving the Road Network Equilibrium Traffic Assignment Problem, Transportation Research 5, pp 309-318.
(7) M.Florian (1986), Nonlinear Cost Network Models in Transportation Analysis, Mathematical Programming Study, 26 pp. 167-196.

(8) D.W. Hearn, S.Lawphonpanich and J.A.Ventura (1987), Restricted Simplicial Decomposition: Computation and Extensions, Mathematical Programming Study, 31 pp. 99-118.

(9) S. Lawphongpanich and D.W.Hearn (1984), Simplicial Decomposition of the Asymmetric Traffic Assignment Problem, Transportation Research 18B pp. 123-133.

(10) M. Florian, J.Guelat and H.Spiess (1987), An Efficient Implementation of the PARTAN Variant of the Linear approximation Method for the Network Equilibrium Problem, Networks 17 pp. 319-339.

(11) J.H. Wu and M. Florian (1993), A simplicial decomposition method for the transit equilibrium assignment problem, Annals of Operations Research 44, 245-260

(12) M. Florian and S. He (2002), A Multi-Class Multi-Mode Variable Demand Network Equilibrium Model with Explicit Capacity Constraints in Transit Services, 16th International EMME/2 UGM, Albuquerque, New Mexico, USA

(13) J. Harders, (1968), The capacity of unsignalized urban intersections. Schriftenreihe “Strassenbau und Strassenverkehrstechnik”, Vol 76.

(14) R. García and A. Marín (2003), Network equilibrium with combined modes: models and solution algorithms. Transportation Research Part B 39 pp 223-254.

(15) M. Patriksson (1999), Nonlinear Programming and Variational Inequality problems. Kluwer Academic Publishers.

(16) J.H. Wu (1991), A Study of Monotone Variational Inequalities and their Application to Nertwork Equilibrium Problems, Ph. D. Thesis, Centre de Recherche sur les Transports, Université de Montréal, Publication #801.

(17) A.K. Ziliaskopoulos and H.S. Mahmassani, (1995). ”A note on least time path computations considering delays and prohibitions for intersection movements”. Transp. Res B Vol. 30, No. 5 pp. 359-367.

(18) G. Gallo and S. Pallotino (1988), "Shortest Path Algorithms" Annals of Operations Research, Vol. 13 (1988) pp. 3-79

(19) M. Fukushima (1986). A relaxed Projection Method for Variational Inequalities, Mathematical Programming, 35, pp. 58-70.

(20) M. Fukushima (1992). Equivalent differentiable optimization problems and descent methods for asymmetric variational inequalities. Mathematical Programming, *53*, pp. 99-110.