

Applying projection-based methods to the asymmetric traffic assignment problem¹

Codina Sancho, Esteve*

*Statistics and Operations Research Department, Universitat Politècnica de Catalunya,
Campus Nord, Building C5, Office 216*

&

Ibáñez Marí, Gemma

*Center for Innovation in Transport, CENIT, Universitat Politècnica de Catalunya,
Campus Nord, Building B5*

&

Barceló Bugeda, Jaume

*Center for Innovation in Transport, CENIT,
Statistics and Operations Research Department, Universitat Politècnica de Catalunya,
Campus Nord, Building C5, Office 215*

Abstract: *This paper examines the application of a path-based algorithm, to the static and fixed demand asymmetric traffic assignment problem. The algorithm is of the simplicial decomposition type and it solves the equilibration or master problem step by means of five existing projection methods for variational inequality problems in order to evaluate their performance on real traffic networks. The projection methods evaluated are: a) a cost approximation based method for minimizing the Fukushima's gap function, b) the modified descent method of Zhu and Marcotte (1998), c) the double projection method of Khobotov (1987) and three of its recently developed variants (Nadezhkina and Takhashi (2006), Wang et al. (2010), He et al. (2012)); d) the method of Solodov and Svaiter (1999); and e) the method of Solodov and Tseng (1996). These projection methods do not require evaluation of the jacobians of the path cost functions. The source for asymmetries are link costs with interactions, as in the case of priority ruled junctions. The path-based algorithm has been computationally tested using the previous projection methods on three medium to large networks under different levels of congestion and the computational results are presented and discussed. Comparisons are also made with the basic projection algorithm for the fixed demand asymmetric traffic assignment problem. Despite the lack of monotonicity properties of the test problems, the only method that failed to converge under heavy congestion levels was the basic projection algorithm. The fastest convergence was obtained in all cases solving the master*

problem step using the method of He et al (2012), which is a variant of Khobotov's method.

. E-mail: esteve.codina@upc.edu

This is the peer reviewed version of the following article: Codina, E. Ibáñez, G. And Barceló, J. "Applying projection-based methods to the asymmetric traffic assignment problem", which has been published in final form at *Computer-Aided Civil and Infrastructure Engineering*. First published online: 4 FEB 2014. DOI: 10.1111/mice.12069.

¹Research supported under Spanish Research Projects SIMETRIA (Ref. P 63/08, 27.11.2008) and TRA2008-06782-C02-02

1 INTRODUCTION

Traffic assignment models are a key part of urban transportation planning methodologies. Their aim is to predict link flows and travel times on a traffic network, which themselves are a result of route choices made by travelers from their origin to their destination. In one of its most simple statements, these models assume that the number of trips from each origin destination pair (OD pair) is fixed and does not depend on the level of congestion in the traffic network (fixed or inelastic demand models). In these models, link travel times are provided by the so-called volume delay functions that express the relationship between travel times on links and traffic volumes in the network. Traveler behaviour is assumed to follow the

Wardrop equilibrium principle (Wardrop (1952)) which can be formulated mathematically as a variational inequality (V.I.) problem and, under some simplifying assumptions, as an optimization problem.

V.I. problems in finite dimension spaces appear very often in economic and engineering models and are the means by which an equilibrium principle is stated mathematically. To describe these problems very briefly: given a closed and convex set $X \subset \mathfrak{R}^n$ of candidate solutions on which a continuous functional $F(\cdot): X \rightarrow \mathfrak{R}^n$ is defined, a special point x is searched so that the projection of $x - \alpha F(x)$, for any $\alpha > 0$, onto X results in the point x itself. In other words, if $P_X[\cdot]$ is the projection operator on X , defined by $P_X[z] = \operatorname{argmin} \{ \|z - x\|_2^2 \mid x \in X \}$, then a solution x of the V.I. problem verifies the fixed-point relationship $x = P_X[x - \alpha F(x)]$, which is the basis of projection algorithms for V.I. problems. This is equivalent to saying that the solutions x of the V.I. problem verify the condition $F(x)^\top (y - x) \geq 0, \forall y \in X$, which is the classical way in which V.I. problems are presented. Through this condition it is clear that, in the case of $F = \nabla f$, then the solutions of the V.I. problem are also local minimizers of the optimization problem $\operatorname{Min}_{x \in X} f(x)$. A convenient way for referring to a V.I. problem defined by a functional F and a closed and convex set X is by $VI(F, X)$. In the case of the traffic assignment problem, its formulation can be under link flows or on a formulation under flows on paths.

For reasons expressed in the State of the Art below, this paper focuses on using recently developed projection algorithms for monotone variational inequalities to solve the static asymmetric traffic assignment problem (ATA problem in the following) under a path flow formulation, while additionally discussing its performance for real traffic networks.

Short State of the Art of algorithms for the ATA problem and projection methods. For reasons that will be explained in section 2, the traffic assignment problem is referred to as an asymmetric traffic assignment (ATA) problem when expressed as a V.I. problem (or also, as problems with link interactions); and it is referred to as a diagonal traffic assignment problem when expressed as an optimization problem.

The formulation of the diagonal problem dates back to Beckman *et al.* (1956). Path-based formulations for solving the diagonal traffic assignment problem were advocated, for instance in Van Vliet and Kupiszewska (1999), not only for computational purposes but also for their utility when path analysis is required as part of the planning tasks. Recent algorithmic advances have made it possible to develop new methods for the fixed demand traffic assignment problem in its diagonal form, thus achieving user equilibrium with

more speed and precision than is possible with the traditional Frank-Wolfe algorithm (Frank and Wolfe (1956)) and other methods that rely on a link-based formulation. These advances have been motivated by increasingly larger network models, as well as by the select-link analysis requirements, which need great precision in calculating equilibrium flows. The evolution of algorithmic developments for solving the diagonal traffic assignment problem can be found in surveys such as Chen *et al.* (2002) or monographs such as Patriksson (1994) and Florian and Hearn (1995) and will not be explained here. Marcotte and Patriksson (2007) is another very relevant monograph capturing recent contributions to the modeling and algorithmic aspects of the static traffic assignment problem.

Also, as the modeling requirements have been more demanding, the separability assumptions in the volume delay functions (upon which diagonal formulations rely), have been found to lead quite frequently to modeling inaccuracies due to the over simplifications that they represent. This may happen, for instance, when it is recommendable to deal explicitly with delays at unsignalized intersections with link interactions using, for instance Harders formula (Harders (1968)). The modeling inaccuracies may also happen when dealing with generalized costs in complex multiclass-multimode planning models, Wu and Florian (1993), Florian and He (2002). Thus, in contrast to the the classical problem of traffic assignment with fixed demand, the ATA problem has received much less attention and the real life large-scale applications of the ATA problem are very seldom found (an example of one may be seen in Berka and Boyce (1996)). The application of the ATA model to a small realistic network can be found in Boyce *et al.* (1990).

Because of the theoretical similarity of diagonal and ATA problems, the approaches for solving the former models have influenced solution methods for the latter models. Relaxation methods for V.I. problems, such as the nonlinear Jacobi and Gauss-Seidel, are one of the oldest and most extended methods for solving the ATA problem, and several variants of the diagonalization methods have been tested by the academic community. A very good description of relaxation methods can be found in Patriksson (1999). A distinctive characteristic of the ATA problem is that real applications do not generally comply with the assumptions of monotonicity required by algorithms for V.I. problems for convergence (see, for instance Florian and Spiess (1982)). It is, however, common practice to apply these algorithms, regardless of this drawback, in hopes of a convergence to equilibrium solutions (see, for instance Berka and Boyce (1996)). Other methods for V.I. problems, such as the gap descent Newton method, have been tested only on small networks under a link-based formulation (see, for instance Marcotte and Guelat (1988)). This method does not seem to be properly suited to large applications and has not received further attention.

Projection algorithms (orthogonal projection or projection of the Goldstein-Levitin-Polyak type) for the ATA problem under a path-based formulation were first developed by Dafermos (1986) and by Bertsekas and Gafni (1982). A basic description of a path-based projection algorithm can be found in Florian and Hearn (1995) and is summarized in section 3 of this paper. At each iteration, the basic projection algorithm for the ATA problem uses a shortest path algorithm based on current link costs in order to enlarge the current set of paths. On this current set of paths, a single projection operation is performed to get a new path flow vector. This basic projection algorithm suggests a more general algorithmic framework in which -- instead of performing a single projection operation -- a vector of path flows (restricted to the current set of paths) is searched at each iteration so that their flows follow a Wardrop equilibrium. The algorithm stops when the current set of paths contains all the paths that can be equilibrated, or when the contribution to the equilibrium of new paths is very small. This schema, already used for diagonal problems by several authors and in Panicucci *et al.* (2007) for ATA problems, can be characterized as a simplicial decomposition (SD) path-based algorithm. The SD algorithm was developed by Hearn *et al.* (1987) for optimization problems and by Lawphongpanich and Hearn (1984) for ATA problems.

Projection methods for solving V.I. problems were studied intensively in the nineties as well as in recent years. Solving V.I. problems with differentiable gap functions (such as Fukushima's gap function and its natural extension: Marcotte's gap function) had been studied earlier by authors such as Fukushima (1986), Fukushima (1992) and Patriksson (1999), among others. Based on these ideas, several authors have developed methods in order to conveniently adjust the parameters of the projection method (self-adaptive methods) for a faster convergence of the algorithm (see, for instance Zhu and Marcotte (1998) and Han and Sun (2004)). The basic projection method can be interpreted as the minimization of a Fukushima gap function, performing just the initial iteration. In recent years, several projection-type methods for solving V.I. problems have been studied and enhanced by the research community. The first one of these methods is the Khobotov algorithm (1987). This algorithm basically consists of a double projection method, which is combined with an adaptive method for adjusting the parameter in the first projection step. This method is used by Marcotte (1991) to solve a link-based formulation of the ATA problem. Recently, Panicucci *et al.* (2007) adapted Khobotov's method by Marcotte in Marcotte (1991) for the ATA problem by using a path-based formulation, reportedly achieving computational results for small to medium-scale networks. To date, there are no other known uses of projection methods for variational inequalities to solve the ATA problem in large traffic networks.

Recently, Wang *et al.* (2010), propose adjusting the parameter for the second projection for the Khobotov algorithm and Nadezhkina and Takahashi (2006), propose the method of successive average (MSA) steps, also following the second projection step. Another method is that developed by Solodov and Svaiter (1999). He *et al.* (2012a, 2012b) develop a wide algorithmic framework for solving V.I. problems by proximal point methods, which contain as special cases the extragradient or double projection method of Khobotov and of Solodov and Svaiter. In these papers, and as a result of their analysis, He *et al.* propose another variant of the Khobotov method consisting simply of recalculating the step parameter of the second projection, which can be performed very efficiently. Finally, the method of Solodov and Tseng (1996) has also been taken into account. A common characteristic of all these methods is that they do not require the evaluation of the jacobian of the path costs, making it possible to apply them to very large networks. The methods based on the so-called D-gap function developed in Solodov and Tseng (2000) and in Peng and Fukushima (1999) require implicit or explicit use of the jacobian of path costs and will not be considered in this paper.

Contribution and layout of the paper. In this paper the performance of a SD path-based algorithm for the fixed ATA problem is tested using a selected set of projection-type methods for V.I. problems as solvers of the equilibration step. None of the selected methods requires the evaluation of jacobians. The selected set of methods are well representative of the state of the art and include, among others, several recently developed variants of the Khobotov extragradient method. No comparative tests of this kind for the ATA problem are known in the literature for medium to large realistic networks and, to our knowledge, only results for the basic Khobotov extragradient method have been reported. The tests show that the new variant in He *et al.* (2012a), has performed better in all instances. The tests have been carried out using three medium scale traffic networks with different levels of congestion. The source for asymmetries in the test networks have been the cost functions of links which depend on flows from other links entering at a common junction, a situation typically arising in priority ruled junctions.

The list of projection-type methods for V.I. problems considered in this paper are: a) the minimization of the Fukushima gap function by the cost approximation algorithm, b) the algorithm of Zhu and Marcotte, c) three variants of the Khobotov's double projection algorithm d) the Solodov and Svaiter algorithm and e) the method of Solodov and Tseng. Results for the basic projection algorithm for the ATA problem are also reported. The outline of the paper is as follows: section 2, next, contains a description of the ATA problem formulated as a V.I. problem, and it also introduces the required notation. Section 3 describes the basic projection algorithm for the

path flow formulation of the ATA problem and section 4 describes the SD path-based algorithm as an extension of it. Section 5 describes the projection-based algorithms for V.I. problems used for solving the equilibration step and, finally, section 6 presents and discusses the computational tests.

2 NOTATION AND BRIEF DESCRIPTION OF THE USER EQUILIBRIUM TRAFFIC ASSIGNMENT PROBLEM

In order to formulate mathematically the user equilibrium traffic assignment problem the following notation needs to be introduced. Consider a network defined in terms of a graph $G = (N, A)$ with a set of nodes N representing either intersections or dummy nodes associated with the transportation zones, usually referred to as centroids, and a set A of arcs used to model the infrastructure and the connections between centroids to the network. Consider also a trip table, or OD trip matrix, modeling the demand between transportation zones. The notation used through this paper is the following:

- W : the set of OD pairs (p, q)
- $\Gamma = \bigotimes_{w \in W} \Gamma_w$; $\Gamma_w = \{r \mid \text{path joining the OD pair } w \in W\}$
- g_w : trip demand for OD pair $w \in W$
- $A_\ell = \{a \in A \mid \exists w \in W, \exists r \in \Gamma_w^\ell, a \in r (\delta_{ar} = 1)\}$
- $\hat{A}_\ell = \{a \in A_\ell \mid \exists x(a) \text{ priority link over link } a\}$
- h_r^w : path flow through the path $r \in \Gamma_w$, joining the OD pair $w \in W$. If by the context, it is clear which OD pair w the path belongs to, the superscript will be omitted. Also, a vector of flows on paths will be denoted by h^w for OD pair $w \in W$.

$$\bullet H = \bigotimes_{w \in W} H^w; H^w = \left\{ h^w \in \mathfrak{R}^{n^w} \mid \sum_{r \in \Gamma_w} h_r^w = g_w, h_r^w \geq 0 \right\}$$

- If Δ is the link-path incidence matrix given by $\Delta = (\delta_{ar})$ being $\delta_{ar} = 1$ if link a belongs to path r and 0 otherwise, then the polytope of link flows V can be expressed as the image of H by the linear operator defined by matrix Δ , i.e.: $V = \Delta(H)$;

- v_a : link flows, $a \in A$. The vector of flows will be denoted by $v \in V$ on links of the network.

- $c_a(v)$: cost on link $a \in A$, which is assumed to depend on link flows v . The link cost vector function will be denoted by $c(\cdot): V \rightarrow \mathfrak{R}^{|A|}$.

- C_r : cost on path r ; the vector of path costs is denoted by $C(\cdot): H \rightarrow \mathfrak{R}^{n^W}$, where $n^W = \sum_{w \in W} n^w$. It is usually assumed that costs on paths can be calculated additively using costs on links.

User equilibrium modeling hypothesis: 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 (1952) 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.*

Wardrop's First Principle can be easily translated in terms of mathematical relationships as follows: flows on a network are in an equilibrium that satisfies Wardrop's principle when non-negative and feasible path flows h_r^* with costs C_r and shortest path costs u_w^* for OD pair w , satisfy:

$$\left. \begin{aligned} h_r(C_r - u_w^*) &= 0, \\ C_r - u_w^* &\geq 0 \end{aligned} \right\} \forall r \in \Gamma_w, w \in W \quad (1)$$

Relationships (1) are a direct translation of Wardrop's principle in mathematical terms as a complementarity condition. If path r carries flow, that is $h_r^* > 0$, then the complementarity equation (1) is satisfied if and only if $C_r - u_w^* = 0$; that is, the cost C_r of using path r for OD pair w is equal to u_w^* , the cost of the shortest path for the OD pair w . While if $C_r - u_w^* > 0$, that is the cost C_r of using path r is higher than the cost of the shortest path, then to satisfy the complementarity equation $h_r^* = 0$, path r does not carry 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 OD pair $w \in W$ then, the sum of flows on all paths for OD pair w must equal the demand g_w , and flows h_r must be non-negative. Taking into account the definitional constraints relating flows on arcs $a \in A$, with flows on paths $r \in \Gamma_w, w \in W$:

$$v_a = \sum_{w \in W} \sum_{r \in \Gamma_w} h_r \delta_{ar} \text{ where } \delta_{ar} = \begin{cases} 1 & \text{if arc } a \text{ belongs to path } r \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

then relationships (1) result in the V.I. formulated in the space of path flows:

$$\text{Find } h^* \in H \text{ so that: } C(h^*)^T (h' - h^*) \geq 0, \forall h' \in H \quad (3)$$

This V.I. was initially reformulated by Smith (1979) in the space of link flows as:

$$\text{Find } v^* \in V \text{ so that: } c(v^*)^T (v - v^*) \geq 0, \forall v \in V \quad (4)$$

Problem (3) can be reduced to an optimization problem under the simplifying assumption of a symmetric jacobian ∇C . This conditions is fulfilled when, for instance, the jacobian of volume delay functions is also symmetrical or if it is diagonal (or, equivalently, the volume delay function of a link depends only on the volume on that link: $c_a(v) = c_a(v_a), \forall a \in A$) (See, for instance, Florian and Hearn (1995)). Thus, in the literature, traffic assignment

problems are commonly referred to as a) diagonal problems or b) problems with link interactions. As no practical instances of traffic assignment models with link interactions and symmetric jacobians are known, then traffic assignment models formulated as an optimization model are simply termed as diagonal models; and those formulated as a V.I. problem are referred to as models with link interactions. Classically, the fixed demand diagonal traffic assignment model is formulated as the following equivalent convex optimization problem, (see, for instance Patriksson (1994)):

$$\begin{aligned} \text{Min}_v \quad & \sum_{a \in A} \int_0^{v_a} c_a(x) dx \\ \text{s.t.} \quad & v \in V \end{aligned} \quad (5)$$

which is also known as the fixed demand Beckmann problem (see Beckmann *et al.* (1956)).

Another approach for solving V.I. problems is by means of the so called gap functions. A function G is a gap function $G: V \rightarrow \mathfrak{R}$, for a V.I. problem (4) if it is nonnegative on V and it vanishes only at points $v^* \in V$ which are solutions of the V.I. Descent methods for V.I. problems are based on various types of gap functions. The most commonly used is the primal gap function G_p of Auslender (1976):

$$G_p(v) = \text{Sup}_{u \in V} c(v)^T(v - u) \quad (6)$$

3 THE BASIC PROJECTION METHOD

For variational inequality such as the one in (4), projection algorithms, at each iteration ℓ , project on the polyhedron V the point $\hat{v}^\ell = v^\ell - \delta Qc(v^\ell)$, where Q is a symmetric definite positive matrix and $\delta > 0$ is a suitable scale parameter. The projection of point \hat{v}^ℓ on the polyhedron V under the norm $\|\cdot\|_{Q^{-1}}$ is equivalent to solving the quadratic problem:

$$\begin{aligned} \text{Min}_y \quad & \frac{1}{2}(y - v^\ell)^T Q^{-1}(y - v^\ell) + \delta c(v^\ell)^T(y - v^\ell) \\ \text{s.t.} \quad & y \in V \end{aligned} \quad (7)$$

For the traffic assignment problem formulated in the space of path flows, the previous problem (7) has a separable structure for each O-D pair $w \in W$. The matrix Q is adopted as block-diagonal, $Q = \text{diag}(\dots Q^w \dots; w \in W)$, being each of the submatrices $Q^w = \text{diag}(\dots, q_r, \dots; r \in \Gamma_w)$. A common coefficient $\alpha = \alpha_r = \alpha_{r'} = \dots$ is usually adopted for all the paths r in the network, or choose as $\alpha_r = \partial C_r / \partial h_r$, the diagonal element of the jacobian ∇C . Thus, each of the subproblems has the following structure:

$$\begin{aligned} \text{Min}_h \quad & \sum_{r \in \Gamma_w} C_r(h^{\ell-1})(h_r - h_r^{\ell-1}) + 1/2 \alpha_r^{-1} (h_r - h_r^{\ell-1})^2 \\ \text{s.t.} \quad & \sum_{r \in \Gamma_w} h_r = g_w, h_r \geq 0 \end{aligned} \quad (8)$$

with $\alpha_r = \delta / q_r$ and δ a scaling parameter. In the case that a common coefficient α applies for all paths, this coefficient will be referred to as the parameter of the projection.

When applied to traffic assignment problems, it cannot be expected that all paths joining each OD pair are known a priori. The projection version of the traffic assignment problem generates new paths based on the link costs of the current solution while the algorithm is running. At iteration ℓ -th, the algorithm will work with a polytope H_ℓ of path flows on given subsets Γ_w^ℓ of n_w^ℓ paths for OD pairs $w \in W$, which have been identified as candidates up to iteration ℓ for each of the origin destination pairs, i.e: $H_\ell = \bigotimes_{w \in W} H_\ell^w$ with $H_\ell^w = \{h^w \in \mathfrak{R}^{n_w} \mid \sum_{r \in \Gamma_w^\ell} h_r^w = g_w, h_r^w \geq 0\}$. Γ_w^ℓ will be referred to as the working set of paths for OD pair $w \in W$.

The resulting projection algorithm is:

Initialization: Find an initial set of acyclic paths for each OD pair $w \in W$, and load on them the demand g_w . Let Γ_w^0 be the initial working set of paths and let $h^0 \in H_0$ be an initial feasible flow vector; $\ell = 1$.

At iteration ℓ -th:

- Find the shortest paths $\tilde{r} \in \Gamma_w$, $w \in W$ with costs on links $C(h^{\ell-1})$; let \hat{h}_{Sp}^ℓ be the flows on the shortest paths.
- Add the new paths detected in step a): $\Gamma_w^\ell = \Gamma_w^{\ell-1} \cup \{\tilde{r}\}$, $\forall w \in W$
- If $G_{rel}(h^{\ell-1}) = C(h^{\ell-1})^T(h^{\ell-1} - \hat{h}_{Sp}^\ell) / C(h^{\ell-1})^T h^{\ell-1} \leq \varepsilon$ STOP.
- For each $w \in W$, solve previous quadratic problem [Q] in (8) over the known set of paths Γ_w^ℓ and let the vector of path flows \hat{h}^ℓ be its solution.
- Line search (optional): $\text{Min}_{0 \leq \theta \leq 1} G_p(h^{\ell-1} + \theta(\hat{h}^\ell - h^{\ell-1}))$. Let $\tilde{\theta}$ be the solution of the line search ($\tilde{\theta} = 1$ if the line search is not performed);
- Update flows: $h^{\ell+1} = h^{\ell-1} + \tilde{\theta}(\hat{h}^\ell - h^{\ell-1})$; $\ell \leftarrow \ell + 1$. Go to a)

The convergence of the algorithm is ensured if the link cost vector function is Lipschitz continuous and strongly monotone. The second condition is in general not verified by network models in practical applications. It must be remarked that the working set of paths Γ_w^ℓ always increases as the algorithm proceeds. The computational performance of the previous algorithm strongly depends on the solution of the quadratic problem [Q] at step d). A good choice for solving it is by means of the algorithm of Michelot (1986), which is also similar in essence to the one proposed by Wu (1991). A description of the algorithm of Michelot used to solve problem [Q] is given in appendix A.

4 AN SD PATH-BASED ALGORITHM FOR SOLVING TRAFFIC ASSIGNMENT PROBLEMS

The projection algorithm described in the previous section precedes a more general schema for solving traffic assignment problems. New paths are found and added iteratively to the ones already used by the algorithm in previous iterations. At each major iteration, and until a stopping criterion is satisfied, the current set of paths is then equilibrated, i.e., the traffic assignment problem is solved *as if* the set of paths in the working set were the only ones available on the network. Additionally, paths that have not been loaded with any flow may be dropped from the working set.

For a given vector $h^w \in H^w$ of path flows for origin-destination pair $w \in W$, let $\Gamma_w^+(h^w)$ denote the subset of paths r with positive flow, $h_r > 0, r \in \Gamma_w^+$, for OD pair $w \in W$. $\Gamma_w^+(h^w)$ will be referred to as the set of active paths for $w \in W$.

The SD path-based algorithm can be stated as follows:

Initialization: Find an initial set of feasible acyclic paths Γ_w^0 for each OD pair $w \in W$, g_w and load on them the demand so that a feasible path flow vector $h^0 \in H_0$ is obtained; set iteration counter $\ell = 1$.

At iteration ℓ -th:

- a) Optionally, for each OD pair $w \in W$ update the working set of paths by dropping those with null flow in previous iterations. If just the $\ell - 1$ -th iteration is considered, then $\Gamma_w^{\ell-1} = \Gamma_w^+(h^{w, \ell-1})$.
- b) [Subproblem Step] Increase the working set of paths by finding the shortest paths $\tilde{r} \in \Gamma_w^+$, $w \in W$, using costs $C(h^{\ell-1})$: $\Gamma_w^\ell = \Gamma_w^{\ell-1} \cup \{\tilde{r}\}$, $w \in W$. Let \hat{h}_{SP}^ℓ be the path flow vector on the shortest paths.
- c) If $G_{rel}^\ell(h^{\ell-1}) = C(h^{\ell-1})^T(h^{\ell-1} - \hat{h}_{SP}^\ell) / C(h^{\ell-1})^T h^{\ell-1} \leq \varepsilon$ then STOP. $h^{\ell-1}$ is an approximate solution.
- d) [Equilibration Step/Master Problem] Equilibrate the paths in the working sets Γ_w^ℓ , $w \in W$: obtain a new vector of path flows h^ℓ by solving the variational inequality subproblem:

$$\text{Find } h^* \in H_\ell \text{ so that: } C(h^*)^T(h' - h^*) \geq 0, \forall h' \in H_\ell \quad (9)$$

Take $h^\ell = h^*$; $\ell \leftarrow \ell + 1$.

The previous algorithm appears in many of the works that have recently been developed for solving diagonal traffic assignment problems under a path based approach in order to obtain very accurate solutions for large traffic networks. When paths with null flow are not dropped from the working set of paths, this algorithm can be considered as an instance of the SD of Lawphongpanich and Hearn (1984) applied to the ATA problem formulated as V.I. (3) in path flows. Because of this, this algorithm will be

referred to in this paper as a SD path-based algorithm. In Lawphongpanich and Hearn (1984), the SD algorithm is shown to converge under strict monotonicity and continuity of the link cost function vector, provided that in step a) unused paths are not dropped from the working set. Among others, authors who have already used it for the diagonal traffic assignment problem are, for instance, Chen *et al.* (2002), Jayakrishnan *et al.* (1994), Florian *et al.* (2009) and Dial (2006). However, instead of a quasi-exact equilibration of path flows in step d), these methods use simple step size evaluation methods for updating path flows, which result in some disequilibrium reduction. Also, Larsson and Patriksson (1992) develop path-biased methods for diagonal problems. Panucci *et al.* (2007), use this SD path-based algorithm for the ATA problem.

For the V.I. problem (9) defined in the equilibration step, it makes sense to define its own Auslender's gap function G_P^ℓ :

$$G_P^\ell(h) = \text{Sup}_{f \in H_\ell} C(h)^T(h - f) \quad (10)$$

and its corresponding relative gap G_{rel}^ℓ as:

$$G_{rel}^\ell(h) = G_P^\ell(h) / C(h)^T h \quad (11)$$

5 USING PROJECTION ALGORITHMS TO SOLVE THE EQUILIBRATION STEP

In this section a description is given of the projection algorithms for V.I. problems that have been used to solve the equilibration step d) of the SD path-based algorithm described in section 4. Subsection 5.1 describes the minimization of the Fukushima gap function and the application of the Zhu and Marcotte algorithm (1998) under the scope of the cost approximation algorithm of Patriksson. In subsection 5.2 the Khobotov algorithm used in Marcotte (1991) and three variants of it are described. In subsection 5.3 the method of Solodov and Svaiter (1999) is described and in subsection 5.4 the method of Solodov and Tseng (1996) is described. All these methods permit solving the ATA problem without the need for evaluating the jacobian ∇C of path costs.

In the description of the projection methods in the following subsections, elements with index ℓ will be assumed fixed by the outer loop in the SD path-based algorithm, whereas index k will be used for the iterations of the algorithm that solves the equilibration step. These iterations will be referred to as minor; iterations of the outer loop will be referred to as major.

5.1 Solving the equilibration step by cost approximation

For finding a solution $x \in X$ of a V.I. problem $VI(F, X)$, Patriksson (1999) introduces the idea of a cost approximation algorithm. It basically consists of replacing the cost functional $F(\cdot)$ at each iteration with a local

approximation to F , $\Psi_k(x)$, at the point x^k , defined by $\Psi_k(x) = \Phi_k(x, x^k) + F(x^k) - \Phi_k(x^k, x^k)$, where Φ_k is a suitably chosen function. At iteration k -th, the approximate V.I., $VI(\Psi_k, X)$ is solved and a point y^k is obtained so that the direction $d_k = y^k - x^k$ is a descent direction for a family of gap functions. In our case, $x \equiv h$, $F \equiv C$ and $\Phi_k(x, x^k) = \alpha_k^{-1}(x - x^k)$. The cost functional $C(h)$ is then approximated by:

$$\bar{C}_k(h, h^k; \alpha_k) = C(h^k) + \alpha_k^{-1}(h - h^k) \quad (12)$$

and the cost approximation algorithm for solving V.I. in (9) could be defined as solving at each iteration ℓ a sequence of approximate V.I.'s, $VI(\bar{C}_k(\cdot, h^k; \alpha_k), H_\ell)$. If α_k is held constant from iteration to iteration, then the directions generated by the cost approximation algorithm are descent directions for the Fukushima's gap function. Based on this schema, two methods described in subsections 5.1.1 and 5.1.2 have been tested in order to solve the equilibration step.

5.1.1 Minimization of a Fukushima gap function

The differentiable gap function of Fukushima (Fukushima (1986), (1992)) is a particular case of Marcotte's gap function and it is defined for the ATA problem under a path-based formulation as follows:

$$G^F(h) = -\text{Min}_{f \in H} C(h)^T (f - h) + 1/2 \alpha^{-1} \|f - h\|_2^2 \quad (13)$$

Because of the structure of the problem (9) in the equilibration step d) to be solved at each iteration of the SD path-based algorithm described in section 4, the minimization of the Fukushima gap function appears to be an appealing candidate. This is equivalent to setting constant the projection parameter α_k of the cost approximation (12). Thus, solving problem (9) is done by minimizing the restriction of the Fukushima gap function G_ℓ^F on the subset of paths in the working set up to iteration ℓ : $\text{Min}_{h \in H_\ell} G_\ell^F(h)$. The following algorithm, which can be considered an instance of the general family of Patriksson's cost approximation algorithms (see, Patriksson (1999)), can be used to minimize G_ℓ^F :

Initialization: $h^{\ell,0} = h^{\ell-1}$; $k = 0$ (superscript ℓ suppressed for readability)

Iteration k :

- Let \tilde{f}^k solve $\text{Min}_{f \in H_\ell} C(h^k)^T (f - h^k) + 1/2 \alpha^{-1} \|f - h^k\|_2^2$ (14)
- Obtain by means of a suitable step length θ^* the next iterate $h^{k+1} = h^k + \theta^*(\tilde{f}^k - h^k)$ and evaluate costs $C(h^{k+1})$.
- If $G_{rel}^\ell(h^{k+1}) \leq \varepsilon'$ STOP. Get $h^* \approx h^{k+1}$.
- $k \leftarrow k + 1$. Go to a)

The stopping criterion at step c) uses the restriction of the primal gap function defined in the path flows space, which in turn is defined by the currently identified set of paths at iteration ℓ , this being the gap function denoted by G_ℓ . Again, the computational performance of the algorithm strongly depends on the solution of the quadratic problem (14). To solve it we use also the method of Michelot (1986). The step length calculation in b) can be done using an MSA step length, i.e., $\theta^* = 1/(k+1)$, or by accepting flows \tilde{f}^k as the new iterate (equivalent to fix $\theta^* = 1$). The performance of the algorithm strongly depends on the choice of the parameter α . Large values for α result in oscillating behaviour and large step values for $\|h^{\ell+1} - h^\ell\|_2$, and a line search is required in order to obtain convergence. When very large values of α are used, then the quadratic term of the projection problem [Q] can be omitted and this is equivalent to accepting as a solution of problem [Q] the flows on the shortest paths obtained in step b) of the SD path-based algorithm in section 4. In this case, this algorithm is equivalent to using the restricted SD of Hearn *et al.* (1987) with space only to store one vertex at each iteration. Small values for α result in small steps $\|h^{\ell+1} - h^\ell\|_2$ and, although a line search is not required, the convergence is also slow. For the algorithm to converge with an inexact line search, sufficient conditions are a differentiable and strongly monotone link cost vector function with Lipschitz continuous jacobian.

5.1.2 The modified descent algorithm of Zhu and Marcotte

The algorithm of Zhu and Marcotte (1998) for minimizing G^F shown below adjusts the parameter of the projection α of the Fukushima gap function in order to accelerate the convergence of the equilibration step.

Initialization: Set values for $a = a_0 = 1/\rho$, r , $0 < \gamma < 1$, $h^{\ell,0} = h^{\ell-1}$; $k = 0$ (superscript ℓ suppressed for readability).

Iteration k :

- Let $\tilde{f}^k \equiv f^*(h^k)$ solve: (15)

$$\text{Min}_{f \in H_\ell} C(h^k)^T (f - h^k) + 1/2 \alpha \|f - h^k\|_2^2$$

- If $C(h^k)^T (\tilde{f}^k - h^k) \leq \frac{2+\gamma}{2\alpha(1-\gamma)} \|\tilde{f}^k - h^k\|_2^2$ then

$$a = \min\{a + \Delta a, r^{-1}\}; \text{Repeat step a)}$$

- Take $h^{k+1} = \tilde{f}^k$
- If $G_{rel}^\ell(h^{k+1}) \leq \varepsilon'$, STOP; adopt as an approximate solution h^{k+1} .
- $k \leftarrow k + 1$. Go to a)

5.2 Variants of the Khobotov double projection algorithm

The basic method of Khobotov (1987) is an extension for variational inequality problems of Korpelevich's extragradient method. In its current form, it was first used by Marcotte (1991) to solve network equilibrium problems formulated by using primal-dual conditions and it was then tested on small networks. For solving path-based ATA problems, it has been used recently by Panicucci *et al.* (2007).

A Khobotov-type projection algorithm for solving the equilibration step (9) can be described as follows:

Initialization: Set values for $\beta, \xi \in]0, 1[$, $\bar{\alpha} > 0$, $k = 0$.

Iteration k:

- a) If $G_{rel}^{(\xi)}(h^k) \leq \varepsilon'$ STOP.
 - b) $v = 0$;
 - c) While $\alpha_k > \beta / L_k^v$ or $v = 0$ do
 - If $v > 0$ then $\alpha_k = \min\{\xi\alpha_k, \beta / L_k^v\}$;
 - Let \tilde{f}^k solve: $\text{Min}_{f \in H_\ell} C(h^k)^\top (f - h^k) + \frac{1}{2}\alpha_k^{-1} \|f - h^k\|_2^2$
 - $L_k^v = \|C(\tilde{f}^k) - C(h^k)\|_2^2 / \|\tilde{f}^k - h^k\|_2^2$; $v \leftarrow v + 1$
 - d) Find suitable δ_k for a second projection. Let $\hat{f}(\delta_k)$ solve:

$$\text{Min}_{f \in H_\ell} C(\tilde{f}^k)^\top (f - h^k) + \frac{1}{2}\alpha_k^{-1} \delta_k^{-1} \|f - h^k\|_2^2 \quad (16)$$
 - e) Take as new iterate $h^{k+1} = \hat{f}(\delta_k)$; $\alpha_{k+1} = \min\{\bar{\alpha}, \beta / L_k^v\}$;
- $k \leftarrow k + 1$; Goto a)

The algorithm of Khobotov used in Panicucci *et al.* (2007) simply adopts $\delta_k = 1$. In the following, adopting $\delta_k = 1$ will be referred to as the basic Khobotov algorithm. Recently, several variants of the previous algorithm have been developed, which enhance the performance of the basic algorithm on the test problems examined by their authors. Subsections 5.2.1 and 5.2.2 describe these variants and, in section 6 (where computational results are presented), applying these variants to the ATA problem is discussed.

5.2.1 Proximal-like methods of He, Liao and Wang

These authors propose two variants of the Khobotov algorithm. He *et al.* (2012a) develop a family of approximate proximal point algorithms which contain as special cases the methods of Khobotov and also the method of Solodov and Svaiter (1999) for variational the inequality problems described in section 5.3. In He *et al.* (2012b) a set of test problems are solved by these authors and they propose to evaluate the parameter δ_k in the second projection, using:

$$\delta_k = \gamma \frac{\|\tilde{f}^k - h^k\|_2^2 - \alpha_k (h^k - \tilde{f}^k)^\top (C(h^k) - C(\tilde{f}^k))}{\|\tilde{f}^k - h^k - \alpha_k (C(h^k) - C(\tilde{f}^k))\|_2^2} \quad (17)$$

The authors show in He *et al.* (2012b) that $\delta_k \geq \gamma/2$, where γ is a fixed parameter which should be adopted as 1.8. As

the computational tests will show in section 6, evaluating δ_k using formula (17) is the best option and provides greater efficiency to the Khobotov algorithm.

Wang *et al.* (2010) propose another method for the calculation of the parameter δ_k at k -th iteration of the second projection. It consists of finding a zero δ^* of the function

$$K_k(\delta) = \delta^{-1} \|\hat{f}(\delta) - h^k\|_2^2 + 2\alpha_k C(\tilde{f}^k)^\top (\hat{f}(\delta) - h^k) \quad (18)$$

where $\hat{f}(\delta)$ is the solution of the quadratic problem (16). The value of δ^* can be calculated with few iterations of a secant method. Wang *et al.* (2010) provide an initial guess to start the search. A disadvantage of this method is that each iteration in the secant method requires the solution of a quadratic program (16).

5.2.2 Hybrid Nadezhkina-Khobotov modified projection method

The algorithm considered for solving the V.I. in the equilibration step is an hybridization of the adjustment method for the α parameter of Khobotov and the method in Nadezhkina and Takahashi (2006). The combination of these methods, along with the adaptation of the parameter α_k for the projections used by Khobotov consists in performing an MSA step length right after the second projection (step d)) of the Khobotov method with $\delta_k = 1$ described in section 5.2 (equation (16)). Thus, the next iterate h^{k+1} in step e) of the generic Khobotov method is taken as:

$$h^{k+1} = h^k + \gamma'_k (\hat{f} - h^k) \quad (19)$$

In our implementation, step length γ'_k has been chosen as the Nagurney-Zhang MSA step (1996) after some initial tests. Other MSA methods that have been tested, and which result in shorter step lengths, provided a worse performance.

For completeness, there follows in next subsections 5.3 and 5.4 a description of the methods of Solodov and Svaiter (1999) and Solodov and Tseng (1996) adapted to solving the equilibration/master problem step of the SD path-based algorithm.

5.3 The method of Solodov and Svaiter

Initialization: Set values for $\theta > 1$, $0 < \sigma, r < 1$, $k = 0$.

Iteration k:

- a) If $G_{rel}^{(\xi)}(h^k) \leq \varepsilon'$ STOP.
- b) Set $\alpha_k = \min\{\theta\beta_k, 1\}$; let \tilde{f}^k solve:

$$\text{Min}_{f \in H_\ell} C(h^k)^\top (f - h^k) + \frac{1}{2}\alpha_k^{-1} \|f - h^k\|_2^2 \quad (20)$$
- c) Find the smallest integer $n \geq 0$ so that

$$C(h^k + r^n \alpha_k (\tilde{f}^k - h^k)) \geq \sigma \alpha_k^{-1} \|\tilde{f}^k - h^k\|_2^2$$
- d) Let $\beta_{k+1} = r^n \alpha_k$; $\tilde{f}^k = h^k + \beta_{k+1} (\tilde{f}^k - h^k)$

e) Let h^{k+1} solve:

$$\text{Min}_{f \in H_\ell} 1/2 \|f - h^k\|_2^2 \text{ s.t. } C(\tilde{f}^k)^\top (f - \tilde{f}^k) \leq 0 \quad (21)$$

The last quadratic problem (21) can be solved rather efficiently by lagrangian relaxation and basically searches the projection parameter for a second projection. The dual lagrangian function φ of problem (21) is:

$$\varphi(\mu) = \text{Min}_{f \in H_\ell} \mu C(\tilde{f}^k)^\top (f - \tilde{f}^k) + 1/2 \|f - h^k\|_2^2 \quad (22)$$

which is differentiable and, very approximately, linear. Let $f^*(\mu)$ solve (22). Then, dual lagrangian problem $\text{Max}_{\mu \geq 0} \varphi(\mu)$ can be solved with very few iterations by a newton method using the derivative of φ , which is $\frac{d\varphi}{d\mu} = C(\tilde{f}^k)^\top (f^*(\mu) - \tilde{f}^k)$.

5.4 The method of Solodov and Tseng

Initialization: Set values for $\beta, \xi \in]0, 1[$, $\bar{\alpha} > 0$, $\theta \in]0, 2[$, $k=0$.

Iteration k :

a) If $G_{rel}^\ell(h^k) \leq \varepsilon'$ STOP.

b) A first projection point \tilde{f}^k and a projection parameter α_k is searched as in step c) of the method of Khobotov but

with L_k^v defined as $L_k^v = \frac{(C(\tilde{f}^k) - C(h^k))^\top (f^k - h^k)}{\|\tilde{f}^k - h^k\|_2^2}$

c) Find \tilde{h} as:

$$\tilde{h} = h^k - \gamma_k (h^k - \tilde{f}^k - \alpha_k (C(h^k) - C(\tilde{f}^k))), \text{ where } \gamma_k \text{ is} \quad (23)$$

$$\gamma_k = \theta(1 - \beta) \|\tilde{f}^k - h^k\|_2^2 / \|h^k - \tilde{f}^k - \alpha_k (C(h^k) - C(\tilde{f}^k))\|_2^2$$

d) Find the projection of the point \tilde{h} on H_ℓ and adopt it as the next iterate $f^* = h^{k+1}$:

$$\text{Min}_{f \in H_\ell} 1/2 \|f - h^k\|_2^2 \quad (24)$$

e) $\alpha_{k+1} = \min\{\bar{\alpha}, \beta/L_k^v\}$; $k \leftarrow k+1$; Goto a)

Convergence of the algorithms described in this section can be guaranteed only under monotonicity and Lipschitz continuity of the link cost vector function, excluding the following cases in which the link cost vector function must be:

- continuous and monotone for the method of Solodov and Tseng in subsection 5.4.
- differentiable and monotone for the method of Zhu and Marcotte. (1998) in section 5.1.2
- for the method of Solodov-Svaiter in subsection 5.3, convergence can be guaranteed under continuity of the link cost vector function and if any solution of the problem is also a solution of the associated Minty variational inequality. This condition is rather difficult to be verified for practical models and it can be guaranteed if the link cost vector function is monotone.

Again these conditions are not fulfilled by network models used in practical applications, because realistic link cost functions with link interactions have no monotonicity properties of any kind.

6 COMPUTATIONAL RESULTS

Three different networks were used to conduct the computational experiments (whose results are reported in this paper): the networks of Winnipeg (Canada), Terrassa (Spain) and Hesse (Germany). These test networks can be downloaded from Bar-Ghera's web page in Bar-Ghera (2013). Table 1, below, shows the main characteristics of the networks. Column #centr. shows the number of centroids or zones, and column #turns shows the number of forbidden or penalized turnings. Columns #AsyJun and #AsyJun(%) show, respectively, the number of junctions at which there are link interactions and the percentage of these junctions relative to the total number of junctions in the network where there are 2 or more incoming links.

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

Network	N	A	#centr.	W	#turns	#AsyJun	#AsyJun(%)
Winnipeg	1057	2535	154	4345	0	275	31.39%
Terrassa	1609	3264	55	2215	1103	177	15.39%
Hesse	4660	6674	245	17213	7054	348	27.55%

The computational experiments were performed on a PC with a Quad 3.20 GHz Intel Core, 650 with 4 Gb RAM and Windows 7 Service Pack 1. The algorithm used for shortest paths is an adaptation of the algorithm in Ziliaskopoulos (1995) for penalized turnings, using proper data structures for an L-deque strategy.

In all the networks, a given subset of intersections are considered to be ruled by priorities. For simplicity, costs on turning movements are constant, which is typical in practice in many real traffic assignment network model applications. Also, for simplicity, flow interactions in the test examples correspond only to interactions between links and not between internal movements of an intersection. A detailed description of the modeling of interactions between movements of an intersection and the its implementation in a user optimal model can be found in Meneguzzo (1995) and in Boyce *et al.* (1990). The delay formula for non-priority links that enter a priority ruled intersection that has been used in the tests is:

$$c_a(v) = t_\phi + \theta^{-1} \log\left(1 + e^{\theta \cdot b \cdot (\xi(v) - 1)}\right) \quad (25)$$

Formula (25) is an approximation to

$$c_a(v) = \max\{t_\phi, t_\phi + b(\xi(v) - 1)\} \quad (26)$$

In the above formula (25), t_ϕ is the free flow travel time of the link. The function $\xi(v)$ is given by $\xi(v) = (v_a + k_{x(a)} v_{x(a)}) / s_a^0 \mathcal{Z}$, and $x(a)$ is the link with

priority on link a . Parameters b and θ have been set to $b=4$ and $\theta=0.2$ for all the test networks and χ is the length of the modeling time period. It has been set to $\chi=21.5$ for the network of Hesse, $\chi=5$ for the network of Terrassa and to $\chi=7$ for the network of Winnipeg. The capacity s_a^0 of non-priority links has been set to $s_a^0=25000$ for the network of Hesse, $s_a^0=4000$ for the network of Terrassa and $s_a^0=400$ for the network of Winnipeg. By using the 'synthetic' delay formula (25), all the test networks have the same type of formula for link interactions instead of each one having its own, which could be the source of heterogeneities. Delay formula (25) does not implicitly impose any strict constraint on link flows; but for high values of b , it acts as a penalty that prevents flows from excessively exceeding capacity $s_a^0\chi$. The effect of the parameter θ is that travel time on links increases earlier for small values of θ and, thus, the network can be congested even for moderate values of the demand. There is no guarantee that, with these cost functions, the jacobian of path costs could be semi-definite positive, not is this the case for other delay cost functions in junctions ruled by priorities. However, there exists empirical evidence that existing algorithms for monotone variational inequalities converge for the ATA problem (see for instance Berka and Boyce (1996)). When the intersection has more than one priority link, as shown in Figure 1, then the term $k_{x(a)}v_{x(a)}$ in the expression for $\xi(v)$ is replaced by the summation $\sum_{\forall a' \text{ priority link over } a} k_{a'}v_{a'}$. For the priority links, we use the classical BPR function for the delay:

$$c_a(v_a) = t_0 \cdot (1 + \alpha (v_a / (s_a \chi))^\beta) \quad (27)$$

where s_a is the capacity on link a which is link dependent. t_0 is the link travel time under free flow conditions, which has been set to 0.75; α and β are fixed parameters set to $\alpha=0.1$ and $\beta=1.5$ for all the test networks.

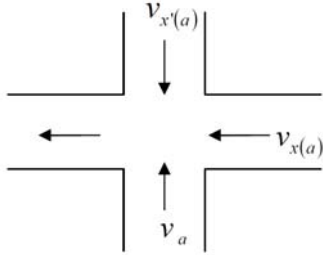


Figure 1. A junction ruled by priorities. Link a is a non-priority link and $x(a)$, $x'(a)$ are links with priority on a .

In order to measure the efficiency of the algorithms, two different gap formulae have been used. The first one ($RGap_1$) is the typical primal gap shown in (6) in its relative form. It is also the measure used in both algorithms to decide whether to continue at the end of every iteration

or to stop if the last solution found is rather good. $RGap_1$ is defined as follows:

$$RGap_1(v) \stackrel{\Delta}{=} \frac{G_p(v)}{c(v)^T v} = \frac{c(v)^T (v-u)}{c(v)^T v} \quad (28)$$

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

A second relative gap $RGap_2$ is provided by formula (29). It is also a gap measure, but in this case it is calculated using the space of path flows and is weighted by the demand between OD pairs. $RGap_2$ is defined as follows:

$$RGap_2(h) = \frac{\sum_{w \in W} (g_w^{-1} \sum_{r \in \Gamma_w} C_r(h) h_r - C_{rSP}(h))}{\sum_{w \in W} C_{rSP}(h)} \quad (29)$$

where $C_{rSP}(h)$ is the cost corresponding to the shortest path k_{SP} ; h is the path flow vector found at the current iteration; $C(h)$ is the corresponding vector of path costs and g_w is the demand corresponding to the OD pair $w \in W$. In order to have an absolute measure of the network's degree of congestion, the following gap measure, Gap_3 , (formula (30)) is also reported for the starting feasible point:

$$Gap_3(h) = |W|^{-1} \sum_{w \in W} (g_w^{-1} \sum_{r \in \Gamma_w} C_r(h) h_r - C_{rSP}(h)) \quad (30)$$

The computational experiments consist of executing the general algorithm described in section 4. For solving the equilibration step, the projection algorithms referred to in subsection 6.1 have been tested on the three networks of table 1. For a fair comparison of the methods, the algorithms use the same feasible initial point for the runs on a given network with a given level of congestion. In order to take into account the effect that different levels of demand have on how the algorithms perform for these networks, an origin destination trip table for each of the networks is divided by three factors: 1, 2 and 5. Factor 1 indicates a very high level of demand and the network is seriously congested. Factor 2 (O-D flows are 1/2 of O-D flows for factor 1) indicates that the level of demand causes simple congestion in the network and factor 5 (O-D flows are 1/5 of O-D flows for factor 1) means the congestion is not very high. The level of congestion for each of the networks is reflected in table 2, below, in the column AvgTT, which reports the average trip time in minutes at the final iteration. The average trip time in the network can be calculated by $Gap_3/RGap_2$. The superscript (1, 2 or 5, in the network acronym in table 2) stands for: 1=very high, 2= high, 5= moderately high. It must be remarked that 5= moderately high is, approximately, the usual congestion level for these traffic networks and this is reflected by the corresponding average trip time. In order to know the average trip time on the three networks at equilibrium as a

measure of the network congestion, the corresponding ATA problems have been solved using the basic Khobotov double projection algorithm described in section 5.2 with $\delta_k = 1$.

It must be noticed that no monotonicity properties of any kind exist for the test problems. This, however, has not been an obstacle for algorithms' convergence with the only exception discussed in 6.1 for congestion levels 1 and 2.

Table 2. Average trip times (minutes) at equilibrium for the three levels of congestion 1, 2, 5 in each of the networks (column AvgTT). The results have been obtained by using the basic algorithm of Khobotov ($\delta_k = 1$).

	RGap1	RGap2	Gap3(min)	AvgTT
Hess ¹	5.85E-04	5.26E-05	8.20E-03	156.00
Hess ²	3.75E-04	2.89E-05	2.42E-03	83.70
Hess ⁵	4.68E-05	1.06E-05	5.45E-05	51.36
Terr ¹	1.65E-03	1.05E-04	7.56E-03	72.04
Terr ²	1.44E-03	5.99E-05	2.52E-03	41.75
Terr ⁵	1.28E-03	1.86E-05	5.34E-04	28.68
Winni ¹	7.58E-06	6.51E-06	4.94E-04	75.89
Winni ²	3.27E-06	3.18E-06	1.33E-04	41.89
Winni ⁵	8.43E-07	9.64E-07	2.44E-05	25.31

6.1 Tested algorithms

The following is a list of the projection algorithms that were evaluated:

- The basic projection algorithm as described in section 3. Three variants have been tested:

- Algorithm PJBa without the optional line search of step e) in the basic projection algorithm of section 3. The projection parameter is chosen as $\alpha_r = \partial C_r / \partial h_r$, the diagonal element of the jacobian corresponding to path $r \in \Gamma_w$.

- Algorithm PJLS, which includes the line search at step e) described in the appendix B. The projection parameter is chosen in the same way that in previous algorithm PJBa.

- Algorithm PJN. The parameter α of the projection is set to a very large value, which is equivalent to obviating the projection step. A line search at step e) is necessary for converging the algorithm. The one described in the appendix B has been used.

- Algorithms FK based on minimization of the Fukushima gap function by following the cost approximation schema (Patriksson (1999)) described in subsection 5.1. It solves equilibration step d) of the general algorithm with the following variants:

- Algorithm FKCA. The gap function of Fukushima is minimized by following the algorithm described in subsection 5.1.1 taking a step length $\theta^* = 1$ in step b).

- Algorithm FKMSA. The same algorithm as FKCA, but the step length is evaluated with the basic MSA method $\theta^* = 1/(k+1)$.

- Algorithm MCTT: the projection algorithm of Zhu and Marcotte (1998), as described in section 5.1.2, applied to solving the equilibration step d) of the SD path-based algorithm. Parameters have been set to $a=0.2, \Delta a = 0.001, r = 2, \gamma = 0.01$ for all test instances.

- Algorithms KH: the double projection Khobotov algorithm and the variants described in section 5.2. In all the tests of the variants related to below, the parameter $\bar{\alpha}$ was set to $\bar{\alpha} = 10$.

- Algorithm KHOB: the algorithm used by Marcotte (1991) and by Panicucci *et al.* (2007), with $\delta_k = 1$ in step d) (second projection). Parameters have been set to $\beta = 1/2, \xi = 1/2$ for all test instances.

- Algorithm KHPL: the algorithm proposed by He, *et al.* (2012a), (2012b) taking δ_k as given in expression (17). The following sets of values for the parameters of the algorithm have been tested: a) ($\beta = 0.9, \xi = 0.7, \gamma = 1.8$), b) ($\beta = 0.9, \xi = 0.7, \gamma = 1$), c) ($\beta = 1/2, \xi = 1/2, \gamma = 1$) and d) ($\beta = 1/2, \xi = 1/2, \gamma = 1.8$). The best results have been obtained with $\beta = 1/2, \xi = 1/2, \gamma = 1.8$.

- Algorithm KHOA: the algorithm proposed by Wang *et al.* (2010) as described in subsection 5.2.1. The following sets of values for the parameters of the algorithm have been tested: a) ($\beta = 1/2, \xi = 1/2$) and b) ($\beta = 0.9, \xi = 0.7$). A better performance was obtained with $\beta = 1/2, \xi = 1/2$.

- Algorithm KHNT: the algorithm by Nadezhkina and Takahashi (2006), as described in subsection 5.2.2. For this algorithm the only set of parameters tested was $\beta = 1/2, \xi = 1/2$.

- Algorithm SLSV: the double projection of Solodov and Svaiter (1999). The values of the parameters of the algorithm that have been tested are $\theta = 1.5, \sigma = 0.5$ and $r = 2$.

- Algorithm SLTS: the double projection of Solodov and Tseng (1996). The values of the parameters of the algorithm that have been tested are $\theta = 1.5, \beta = 0.9$ and $\xi = 0.3$.

As expected, the algorithms on the list which performed worst when tested were PJBa and PJLS. The level of congestion seriously affected performance of algorithm PJBa. For all networks, it converged slowly for the lowest level of congestion (level 5), but oscillated without converging for levels 2 and 1 in all test networks. All other algorithms converged for all the levels of congestion in the three test networks, despite the non-monotonicity in the test problems. When the line search described in the appendix B is included in algorithm PJBa, then the resulting algorithm (algorithm PJLS) converges for all the levels of congestion in all the networks. The following tables 3 and 4, below, show results for algorithms PJBa and PJLS for levels of congestion 5 and 1, respectively, after 400 seconds of CPU:

Table 3. Natural logarithm of RGap1 for algorithm PJBa after 400s of CPU time. Lowest level of congestion.

	Hess ⁵	Terr ⁵	Winni ⁵
PJBa	-4,74	-4,21	-10,17

Table 4. Natural logarithm of RGap1 for algorithm PJLS after 400s of CPU time. Highest level of congestion.

	Hess ¹	Terr ¹	Winni ¹
PJLS	-3,39	-2,06	-8,27

The criterion for stopping the master problem/equilibration step (at step d) of the SD path-based algorithm in section 4 was to setting a variable tolerance ε' depending on the relative gap G_{rel} achieved after the subproblem Step b). At a major iteration ℓ , \hat{k} iterations of the equilibration step are performed until a sufficiently small relative gap $G_{rel}^{(\ell)}(h^{\hat{k}}) \leq \varepsilon'_\ell$ is reached or $\hat{k} = N_{max}$. The value for ε'_ℓ has been set to:

$$\varepsilon'_\ell = \min\{\varepsilon_0, \beta G_{rel}(h^{\ell-1})\} \quad (31)$$

Values for ε_0 and β have been set to 0.05 and 0.05 respectively. Thus, in the initial major iterations, the master problem/equilibration step is solved until a reasonable gap ε_0 is satisfied, but as long as the algorithm proceeds and the relative gap for the major iteration becomes smaller, the subproblem step must be solved more accurately.

Algorithms FKCA and FKMSA also performed poorly, although a little better than algorithms PJBa and PJLS, because the number of iterations required to solve equilibration step d) was too high. Thus, they were discarded after the first tests.

Also, algorithms KHOA and SLSV performed worse than the remaining variants of the Khobotov double projection family of methods and than the SLTS algorithm. Their evaluation was also abandoned after the first tests. In the case of the KHOA algorithm, the calculation of the optimal step length δ^* for the second projection by a secant method required that the quadratic problem (16) had to be solved too many times per iteration.

After discarding the above-mentioned algorithms, algorithms KHOB, KHPL, KHNT, MCTT, PJN and SLTS were tested by setting a limit N_{max} of 30 and 100 minor iterations respectively to solve equilibration step d) of the SD path-based algorithm in section 4. They were applied to the three networks for the three levels of congestion and limited to a maximum CPU time of 400 seconds. The limit of 100 minor iterations per equilibration was not reached by the initial major iterations; but the remaining major iterations always reached the limit. The number of minor iterations for the equilibration step proves to be the most relevant factor in attaining high degrees of convergence. In all cases, results obtained by setting a limit of 100 iterations for solving the equilibration step outperform those obtained by setting only a maximum of 30. This suggests that using projection methods --such as the methods based on the D-gap function (Solodov and Tseng (2000)), which would require explicitly or implicitly evaluating the jacobian of

path cost functions-- could provide the SD path-based algorithm with very small relative gaps. Tables 5 and 6 summarize the results obtained by algorithms KHOB, KHPL, KHNT, MCTT, PJN and SLTS using 30 and 100 iterations respectively; whereas Figures 2, 3 and 4 illustrate their convergence for 100 iterations on the three test networks at the highest level of congestion. On the x -axis, the total CPU time in seconds versus the natural logarithm of the relative gap, $RGap_1$, defined in (28), is shown. These graphics show that the most advantageous algorithm is KHPL and that algorithm KHPL perform higher than algorithm KHOB, tested already in Panicucci *et al.* (2007), thus showing the effectiveness of the correction formula (ref{proximal-like-step}) for the projection parameter. Also, the MSA step length (19) used in the variant KHNT performed poorly making the Khobotov schema to loose its efficiency. Finally, algorithm PJN performs much better than algorithm PJLS, with results in the range of the remaining algorithms KH, MCTT and SOTS, although its performance is slightly lower. This comparable performance is apart from the Winnipeg network, in which PJN performs clearly worse.

Table 5. Natural logarithm of RGap1 for six algorithms for a fixed CPU time (400s) and 30 iterations for the equilibration step.

Net	KHOB	KHPL	KHNT	SLTS	MCTT	PJN
Hess ¹	-7.13	-7.79	-5.20	-5.82	-5.48	-6.38
Hess ²	-7.83	-8.44	-5.50	-7.41	-5.69	-6.24
Hess ⁵	-9.34	-9.67	-6.66	-8.68	-7.00	-8.10
Terr ¹	-6.30	-6.80	-4.32	-5.71	-4.83	-6.30
Terr ²	-6.28	-6.72	-4.71	-6.12	-5.16	-6.68
Terr ⁵	-6.51	-6.57	-5.46	-6.40	-5.79	-7.14
Winni ¹	-11.46	-11.90	-9.70	-9.67	-12.14	-6.36
Winni ²	-11.96	-12.58	-10.38	-10.80	-12.44	-7.20
Winni ⁵	-13.20	-13.44	-11.57	-11.72	-12.24	-8.53

Table 6. Natural logarithm of RGap1 for five algorithms for a fixed CPU time (400s) and 100 iterations for the equilibration step.

Net	KHOB	KHPL	KHNT	SLTS	MCTT
Hess ¹	-7.44	-7.78	-4.80	-6.39	-5.70
Hess ²	-7.89	-8.62	-5.03	-7.65	-6.32
Hess ⁵	-9.97	-11.34	-5.94	-9.28	-7.43
Terr ¹	-6.41	-7.38	-4.05	-6.12	-5.20
Terr ²	-6.54	-7.04	-4.44	-6.31	-5.67
Terr ⁵	-6.66	-6.62	-5.40	-6.60	-6.26
Winni ¹	-11.79	-12.50	-9.20	-9.97	-12.52
Winni ²	-12.63	-13.39	-10.07	-10.87	-12.91
Winni ⁵	-13.99	-14.56	-11.46	-11.86	-12.74

Table 7, below, shows the average oscillation between successive iterations of link flows for the same tests shown in table 6, at the final iteration, i.e.: $\sum_{a \in A} |v_a^\ell - v_a^{\ell-1}| / |A|$. As

it is shown in the table, the oscillation for algorithm for algorithm KHPL is slightly smaller

Table 7. Mean difference in link flows from the final two iterations for five algorithms with a fixed CPU time: $\sum_{a \in A} |v_a^t - v_a^{t-1}| / |A|$.

Net	KHOB	KHPL	KHNT	MCTT	PJN
Hess ¹	3.87E-02	1.47E-02	6.90E-02	5.72E-02	6.67E-02
Hess ²	1.90E-02	4.92E-03	2.63E-02	2.87E-02	1.56E-02
Hess ⁵	3.29E-03	1.03E-03	5.39E-03	6.82E-03	2.17E-04
Terr ¹	1.02E-02	4.70E-03	3.71E-02	1.04E-02	1.97E-01
Terr ²	4.26E-03	1.71E-03	1.38E-02	4.26E-03	7.38E-02
Terr ⁵	1.19E-03	4.54E-04	3.53E-03	1.16E-03	6.99E-03
Winni ¹	1.17E-04	3.14E-05	8.30E-04	4.41E-05	6.42E-03
Winni ²	3.17E-05	1.29E-05	3.92E-04	1.49E-05	2.52E-03
Winni ⁵	6.09E-06	3.08E-06	4.63E-05	7.76E-06	5.42E-04

Table 8, below, shows the distribution of CPU time among the six algorithms with better performance for the test networks with congestion level 1. Column #It shows the number of iterations performed in 400 CPU seconds; next columns show the percentage of time required for evaluation of path costs, for shortest path calculations, for the quadratic problems involved in the projection algorithms (excluding path-cost calculations) and finally the last column shows the remainder of CPU time. For the PJN algorithm, column EL shows the percentage of time spent in the line search. Algorithms KH and SLTS have a similar pattern, different from the pattern of algorithm MCTT and from the pattern of algorithm PJN. Also remarkable is the very small percentage of CPU time spent in shortest path calculations of algorithms KH and SLTS, which is a little bit higher for algorithm MCTT. The time spent in shortest path calculations for algorithm PJN is significantly higher and can be compared to the one spent by vertex generating algorithms for the ATA problem formulated in the space of link flows.

Table 8. Number of iterations and distribution of CPU time in % for algorithms KH, MCTT, SLTL and PJN

Winni1	#it	Path Costs (%)	Shortest paths (%)	Quadratic Problem (%)	Rest (%)
KHPL	64	41.2	0.3	35.7	22.7
KHOB	63	39.1	0.3	37.4	23.3
MCTT	137	54.7	0.7	44.4	0.2
SLTS	57	33.7	0.3	45.7	20.3
KHNT	62	40.8	0.3	35.6	23.3
PJN	226	7.4	8.3	0	84.3
<i>Terr1</i>					
KHPL	84	52.7	0.2	25.6	21.4
KHOB	83	51.7	0.3	27.5	20.5
MCTT	154	68.7	0.6	30.6	0.2
SLTS	82	47.4	0.2	32.9	19.5
KHNT	85	54.2	0.3	23.2	22.3
PJN	331	13.7	5.4	0	80.9

<i>Hess1</i>					
KHPL	14	60.4	0.4	22.1	17.1
KHOB	17	60.0	0.5	23.4	16.2
MCTT	33	73.9	0.8	25.1	0.1
SLTS	15	54.3	0.4	28.4	16.8
KHNT	17	60.1	0.4	22.1	17.4
PJN	63	11.9	7.8	0	80.4

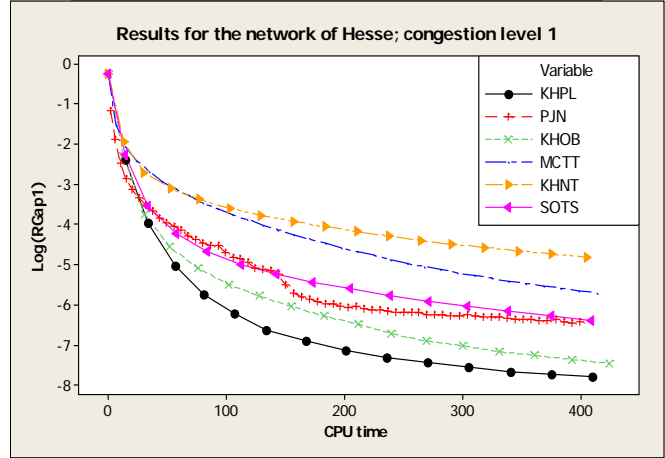


Figure 2. Natural logarithm of $Rgap_1$ versus CPU time (seconds) for the Hesse test network.

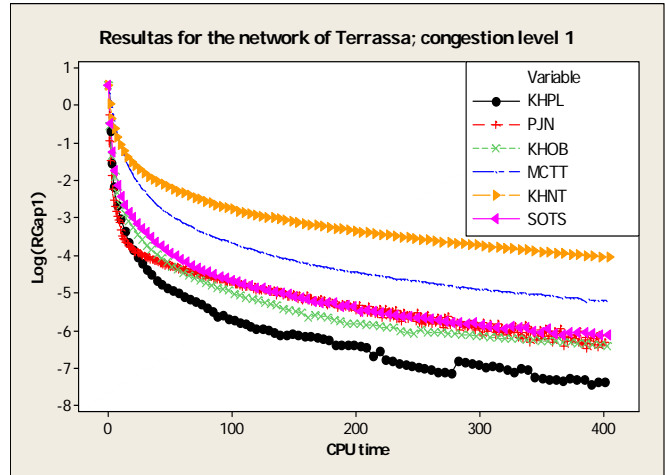


Figure 3. Natural logarithm of $Rgap_1$ versus CPU time (seconds) for the Terrassa test network.

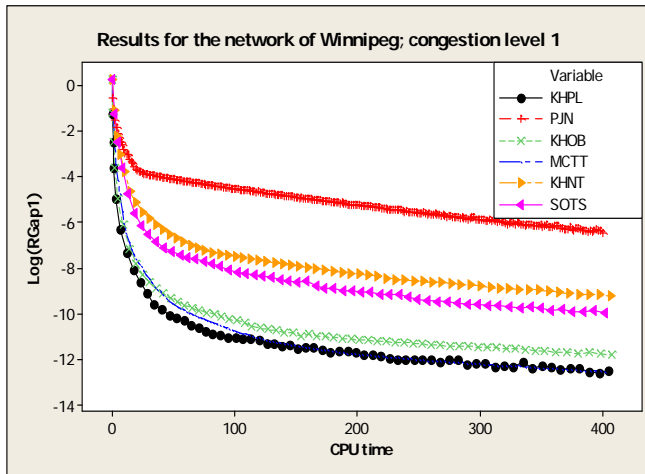


Figure 4. Natural logarithm of $Rgap_1$ versus CPU time (seconds) for the Winnipeg test network.

7 CONCLUSIONS AND FURTHER RESEARCH

This paper demonstrates the application of an SD path-based algorithm on the fixed demand asymmetric traffic assignment problem. It also solves the equilibration step by using five existing projection methods for variational inequality problems in order to evaluate their performance in three real traffic networks under different levels of congestion. The projection methods evaluated are: a) a cost approximation based method for minimizing the Fukushima gap function; b) the modified descent method of Zhu and Marcotte (1998); c) the double projection method of Khobotov (1987) and three of its recently developed variants (Nadezhkina and Takahashi (2006), Wang *et al.* (2010), He *et al.* (2012b)); d) the method of Solodov and Svaiter (1999); and e) the method of Solodov and Tseng (1996). Because of the similarities to the classical path-based projection algorithm for the ATA problem, the performance of this algorithm and two variants of it have also been tested. All the projection methods for V.I. problems considered in this paper have in common that the jacobians of the path cost functions do not require evaluation. Of the previously mentioned projection algorithms, only the basic Khobotov method has been previously used in Panicucci *et al.* (2007). The tests show that the performance of the SD path-based algorithm is always enhanced if the equilibration step is solved with high precision. This suggests that using projection methods that explicitly or implicitly require the jacobian of path cost functions to be evaluated (such as the methods based on the D-gap function in Solodov and Tseng (2000)) could obtain a very high degree of convergence. The oscillation of link flows in the final iterations is also examined; this is a relevant feature for path-based analysis. The tests also illustrate that convergence of the SD algorithm and stability in link flows are both best when using the recently developed variant KHPL of the Khobotov algorithm

described in He *et al.* (2012b), overcoming the results obtained with the basic Khobotov method and also, the results of the remaining algorithms tested in this paper. The tests also show that the classical projection method may not converge for very congested networks. Convergence was always attained for the remaining algorithms, despite the lack of monotonicity properties in the test problems.

REFERENCES

- Auslender, A. (1976), *Optimisation: méthodes numériques*. Mason.
- Bar-Ghera, H. *Transportation Network Test Problems* <http://www.bgu.ac.il/~barghera/tntp> (last access: October 9th 2013)
- Beckmann, M., McGuire, C.B. & Winsten, C.B. (1956), *Studies in the Economics of Transportation*, Yale Univ. Press, New Haven, Connecticut.
- Berka, S. & Boyce, D.E. (1996), Generating Highway travel times with a large-scale, asymmetric user equilibrium assignment problem. in *Advanced Methods in Transportation Analysis*, L. Bianco, P. Toth Ed. Springer, pp 29-62.
- Bertsekas, D.P. & Gafni, E.M. (1982), Projection methods for variational inequalities with application to the traffic assignment problem. *Math. Program. Study* 17, 139-159.
- Boyce, D., Meneguzzer, C., Roupail, N., Sen, A. & Lauritzen, T. (1990), An user-optimal route choice model with asymmetric cost functions Incorporating intersection related travel times. Final Report to Illinois Department of Transport. Illinois Universities Transportation Research Consortium. Report No. FHWA/IL/RC-004.
- Chen, A., Lee, D.-H. & Jayakrishnan, R. (2002), Computational study of state-of-the-art path-based traffic assignment algorithms. *Math. Comput. Simul.* 59, 509-518.
- Dafermos, S. (1986), An iterative scheme for variational inequalities. *Mathematical Programming* 26, 40-47
- Dial, R. (2006), A path-based user equilibrium traffic assignment that obviates path enumeration. *Transportation Research Part B* 40 917-936.
- Florian, M. & Hearn, D. (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.
- Florian, M. & He, S. (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.
- Florian, M., Constantin, I. & Florian, D. (2009), A new look at the projected gradient method for equilibrium assignment. *TRB Annual Meeting 2009*, Washington USA.
- Florian, M. & Spiess, H. (1982), The convergence of diagonalization algorithms for asymmetric network

equilibrium problems, *Transportation Research B* Vol 16 issue 6 pp 477-483.

Frank, M. & Wolfe, P. (1956), An Algorithm for Quadratic Programming, *Naval Research Logistic Quarterly* 3, pp 95-110.

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

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

Han, D. & Sun, W. (2004), A new modified Goldstein-Levitin-Polyak projection method for variational inequality problems *Computers and Mathematics with Applications* 47, pp. 1817-1825.

Harders, J. (1968), The capacity of unsignalized urban intersections. *Schriftenreihe "Strassenbau und Strassenverkehrstechnik"*, Vol 76.

He, B., Liao, L-Z. & Wang, X. (2012a), Proximal-like contraction methods for monotone variational inequalities in a unified framework I: Effective quadruplet and primary methods, *Computational Optimization Applications* 51:649-679.

He, B., Liao, L-Z. & Wang, X. (2012b), Proximal-like contraction methods for monotone variational inequalities in a unified framework II: general methods and numerical experiments, *Computational Optimization Applications* (2012) 51:681-708.

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

Jayakrishnan, R., Tsai, W.K., Prasker, J., & Rajadhyaksha, S. (1994), A faster path-based algorithm for traffic assignment. *Transportation Research Record*, 1443, 1994 pp.75-83.

Khobotov, E.N. (1987), Modification of the extragradient method for solving variational inequalities and certain optimization problems. *USSR Comput. Math. Phys.* 27, pp. 120-127.

Larsson, T., & Patriksson, M. (1992), Simplicial decomposition with disaggregate representation for the traffic assignment problem. *Transportation Science*, 26, 1992. pp. 4-17.

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

Marcotte, P. (1991), Application of Khobotov's algorithm to variational inequalities and network equilibrium problems. *INFOR* vol 29, no 4, pp. 258-270

Marcotte, P. & Guelat, J. (1988), Adaptation of a modified Newton method for solving the asymmetric traffic assignment problem. *Transportation Science* 22, pp. 112-124.

Marcotte, P. & M. Patriksson (2007), Traffic equilibrium, in *Transportation*, volume 14, *Handbooks in Operations Research and Management Science*, C. Barnhart and G. Laporte (eds), Elsevier, Oxford, 623-713.

Meneguzzer, C. (1995), An equilibrium route choice model with explicit treatment of the effect of intersections, *Transportation Research Part B*, 29 (5), 329-356.

Michelot, C. (1986), A finite algorithm for finding the projection of a point onto the canonical simplex of R^n . *Journal of Optimization Theory and Applications*. 50, 195-200

Nadezhkina, N. & Takahashi, W. (2006), Weak convergence theorem by an extragradient method for nonexpansive mappings and monotone mappings, *Journal of Optimization theory and Applications* 128 (2006), pp 191-201.

Nagurney, A. & Zhang, D. (1996), *Projected Dynamical Systems and Variational Inequality with Applications*. Kluwer Academic Publishers, Dordrecht, Boston, MA.

Panicucci, B., Pappalardo, M. & Passacantando, M. (2007), A path-based double projection method for solving the asymmetric traffic network equilibrium problem *Optimization Letters* (2007)1 171-185

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

Patriksson, M. (1994), *The traffic Assignment Problem: Models and Methods*, VSP B.V.

Peng, J-M. & Fukushima, M. (1999), A hybrid Newton method for solving the variational inequality problem via de D-gap function. *Math. Program. Ser. A* 86, pp. 367-386.

Smith, M.J. (1979), Existence, Uniqueness and Stability of Traffic Equilibria, *Transportation Research B*, 1B, pp 295-304.

Solodov, M.V. & Svaiter, B.F. (1999), A new projection method for variational inequality problems, *SIAM J. Control. Optim.* Vol. 37, No 3, pp. 765-776.

Solodov, M.V. & Tseng, P. (1996), Modified projection type methods for monotone variational inequalities, *SIAM J. Control. Optim.* Vol. 34, No 5, pp. 1814-1830.

Solodov, M.V. & Tseng, P. (2000), Some methods based on the D-gap function for solving monotone variational inequalities. *Computational Optimization and Applications*, 17, pp 255-277.

Van Vliet, D. & Kupiszewska, D. (1999), 101 uses for path-based assignment. *Transportation Planning Methods. Proceedings of Seminar F, European Transport Conference, 27-29 September 1999, Cambridge, UK.*

Wang, X., He, B. & Liao, L-Z. (2010), Steplengths in the extragradient type methods. *Journal of Computational and Applied Mathematics* 233 (2010) pp. 2925-2939

Wardrop, J.G. (1952), Some Theoretical Aspects of Road Traffic Research, *Proceedings of the Institution of Civil Engineers, Part II* 1 (2), 325-378.

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

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

Zhu, D.L. & Marcotte, P. (1998), Modified descent methods for solving the monotone variational inequality problem. Operations Research Letters, Vol 14, Issue 2 pp. 111-120.

Ziliaskopoulos, A.K. & Mahmassani, H.S. (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.

Appendix A. Solving quadratic problem [Q] efficiently

For a given OD pair $w \in W$, problem [Q] can be conveniently re-scaled and expressed as the following problem [Q']:

$$\begin{aligned} \text{Min}_x \quad & 1/2 \sum_{k \in I} a_k x_k^2 + \sum_{k \in I} b_k x_k \\ \text{[Q']} \text{ s.t.:} \quad & \sum_{k \in I} x_k = 1 \quad |\lambda \\ & x_k \geq 0 \quad |\mu_k \end{aligned} \quad (32)$$

where for a simpler notation $I \equiv \Gamma_w^+$. Subscripts w have been omitted. λ and $\mu_k, k \in I$ are the corresponding Lagrange multipliers and coefficients a_k and b_k are related to those of problem [Q] by $a_k = \rho^{-1} q_k g_w^2 (> 0)$, $b_k = (C_k(h^\ell) - \delta^{-1} q_k h_k^\ell) g_w$. The quadratic problem [Q'] is strictly convex and therefore Karush Kuhn Tucker (KKT) conditions are necessary and sufficient for optimality, and by applying them it is easy to prove that solutions verify:

$$\lambda = \min_{k \in I} \{a_k x_k + b_k\} \quad (33)$$

and as a function of λ , the primal solutions of [Q'] must verify:

$$x_k(\lambda) = \max\{0, a_k^{-1}(\lambda - b_k)\} \quad (34)$$

Thus the following piecewise linear and increasing function $\varphi(\lambda) = \sum_{k \in I} x_k(\lambda)$ can be defined. It is not difficult to prove that there exists a unique solution λ^* to the equation $\varphi(\lambda) = 1$, imposing feasibility for $x_k(\lambda)$, $k \in I$ and that $x_k(\lambda^*), k \in I$ solve problem [Q] and also that $x_k(\lambda^*), \lambda^*, \mu_k^* = a_k x_k(\lambda^*) + b_k - \lambda^*, k \in I$ verify KKT conditions for problem [Q].

To compute λ^* , the following secant algorithm is used for solving problem [Q] in the projection algorithm. The algorithm converges in a finite number of iterations, which is linear in the size $|I|$ of problem [Q].

Initialization. Let $v = 0$ and $I_0 = I$.

Let λ^0 be the initial value defined by:

$$\lambda^0 = (1 + \sum_{k \in I_0} a_k^{-1} b_k) / \sum_{k \in I_0} a_k^{-1}$$

While $\phi(\lambda^v) > 1$ *do*: $\lambda^v = (1 + \sum_{k \in I_v} a_k^{-1} b_k) / \sum_{k \in I_v} a_k^{-1}$

–Evaluate $\phi(\lambda^{v+1})$; set $I_{v+1} = \{k \in I_v \mid a_k^{-1}(\lambda^{v+1} - b_k) > 0\}$; $v \leftarrow v+1$

Appendix B. A line search using the Auslander gap function

In order to speed up the projection algorithm we have introduced an optional module of line search using Auslander's primal gap function (6) at step e) of the basic projection algorithm described in section 3.

Consider at iteration ℓ -th the directional derivative of the Auslander gap function on the descent direction $\hat{v}^\ell - v^\ell$:

$$G'_p(v^\ell; \hat{v}^\ell - v^\ell) = [c(v^\ell) + \nabla c(v^\ell)^T (v^\ell - v^{SPT}(v^\ell))]^T (\hat{v}^\ell - v^\ell) \quad (35)$$

where $v_a^\ell, \forall a \in A$, are the link flows obtained in the last iteration, $\hat{v}_a^\ell, \forall a \in A$, are the new link flows obtained from the quadratic problem, and $v^{SPT}(v^\ell)$ is the link flow vector that would be obtained by solving the shortest path problem with costs $c(v^\ell)$. Note that the previous derivative (35) uses the jacobian of the costs in the space of link flows, which can be stored because of the sparse structure of the matrix. Notice that ∇c can be stored in a number of $|A|(1+x \cdot m)$ positions, x being the percentage of links whose cost depends on more than one link flow and m the average number of link interactions.

The line search implemented in this case is the following:

- Initialization: $v^{\ell,0} = \hat{v}^\ell, G_1 = G_2 = G_P(v^{\ell,0}) = G_P(\hat{v}^\ell), G_0 = \tilde{G} = G_P(v^\ell), \tilde{\theta} = 1, \theta_1 = \theta_2 = 1, p = 0$.
- If $G_P(v^{\ell,p}) \leq G_P(v^\ell) + \rho^p \eta G'_p(v^\ell; \hat{v}^\ell - v^\ell)$ or $p = n$ STOP.
- Update flows: $v^{\ell,p+1} = v^\ell + \rho^{p+1}(\hat{v}^\ell - v^\ell)$; $p \leftarrow p+1$
- $G_2 = G_1, \theta_2 = \theta_1, \theta_1 = \rho^p$
- Calculate $G_1 = G_P(v^{\ell,p})$
- If $G_P(v^{\ell,p}) < \tilde{G}$ then: $\tilde{G} = G_P(v^{\ell,p}), \tilde{v} = v^{\ell,p}, \tilde{\theta} = \theta_1 = \rho^p$.
- Go to b).

where $0 < \rho, \eta < 1$ are parameters that have been set to $\rho = 0.5$ and $\eta = 0.5$.

The previous algorithm is of the backward search type with an Armijo's rule of acceptance. A limited number of iterations must be imposed on the previous line search. If the initial guess $\tilde{\theta} = 1$ is not good, then the final value G_1 of the function may be far from the acceptance region and may not even be smaller than G_0 . In this case, the final value G_1 and the previous one G_2 may be used in order to

heuristically estimate a value for the step length $\tilde{\theta}$ that will probably provide an improvement of $G_p(\cdot)$. If $\beta_0 = \eta G'_p(v^\ell; \hat{v}^\ell - v^\ell)$ and $\beta_1 = (G_2 - G_1)/(\theta_2 - \theta_1)$, then the value for $\tilde{\theta}$ could be chosen as the one that verifies $G_0 + \tilde{\theta}\beta_0 = G_1 + \beta_1(\tilde{\theta} - \theta_1)$. $\tilde{\theta}$ would be then $\tilde{\theta} = (G_0 - G_1 + \beta_1\theta_1)/(\beta_1 - \beta_0)$. A more conservative value for $\tilde{\theta}$ is adopted as

$$\tilde{\theta} = \mu^{-1} (G_0 - G_1 + \beta_1\theta_1)/(\beta_1 + |\beta_0|) \quad (36)$$

where $\mu > 1$ is a parameter chosen as $\mu = 1.5$. After the line search, path and link flows are updated using the step length $\tilde{\theta}$: $h^{\ell+1} = h^\ell + \tilde{\theta}(\hat{h}^\ell - h^\ell)$ and $v^{\ell+1} = v^\ell + \tilde{\theta}(\hat{v}^\ell - v^\ell)$.