SOLVING A DYNAMIC USER EQUILIBRIUM MODEL BASED ON SPLITTING RATES WITH GRADIENT PROJECTION ALGORITHMS

Guido Gentile, DICEA, Università di Roma "La Sapienza", Guido.Gentile@UniRoma1.it

Abstract

This article shows how Gradient Projection (GP) algorithms are capable of solving with high precision a Dynamic User Equilibrium (UE) model based on Splitting Rates, i.e. turning movements fractions by destination.

Dynamic Traffic Assignment (DTA) is formulated as a Variational Inequality problem defined on temporal profiles of arc conditional probabilities, that express a sequence of deterministic route choices taken at nodes by road users directed toward each destination.

Congestion is represented through a macroscopic traffic model capable to reproduce a range of phenomena having increasing complexity, from links with bottleneck to intersections with spillback. Different time discretizations, from few seconds to few minutes, are also possible, which allows a range of applications from planning to operation.

This assignment model, which is fully link based, is proved to be equivalent to a path based formulation. It also allows for the computation of a handy gap function for analysing convergence to equilibrium.

Numerical experiments on test networks are presented, showing that the proposed GP algorithms converge to dynamic equilibrium in a reasonable number of iterations, outperforming the Method of Successive Averages (MSA).

Keywords: *Dynamic Traffic Assignment; deterministic and sequential route choice; bottleneck and spillback congestion; Variational Inequalities; implicit path enumeration.*

1. INTRODUCTION

1.1 MOTIVATIONS

Although DTA models that are applicable to large road networks can be found in the literature since 20 years, dynamic assignment is only now receiving around the world a greater attention from transport planners and operators of traffic management centres, for two main reasons. First, it is now clear that in many cases static models are not adequate for simulating congestion on transport networks: where queuing is a relevant phenomenon, they are just not able of reproducing real traffic data for speeds and flows at the same time. Second, the primary levers of daily traffic management, such as driver information and signal setting, act on anticipatory rerouting and vehicle accumulation, whereas these phenomena have an intrinsic dynamic nature.

However, static traffic assignment is still considered the reference approach, if the main purpose of the analysis is not the simulation of a given situation, but rather the comparison among different design scenarios in terms of aggregated Key Performance Indicators. The main reason for this is, in our view, the lack of good algorithms for solving Dynamic User Equilibrium problems with a sufficient level of precision. It is not

just a matter of faster convergence; the truth is that the most common solution approach, that is the Method of Successive Averages applied to a micro or meso-scopic model, is often not capable at all of converging in practice to a stable solution.

1.2 LITERATURE REVIEW

Despite the intensive efforts produced by the transportation research community (starting from the early models of, e.g.: Jayakrisham et al., 1994; Ben-Akiva et al., 1997; Adamo et al., 1999), Dynamic Traffic Assignment is still one of the most challenging issues in network modelling. A satisfactory mathematical framework that ensures the existence and uniqueness of a Dynamic User Equilibrium, along with a convergent algorithm that can rapidly compute its solution, are goals to be achieved yet. Some consistent formulations are available (e.g.: Friesz et al., 1993; Ran and Boyce, 1994; Heydecker and Addison, 1998); but one shall be ready to pay a high price in terms of model realism on the supply side (e.g. simple point-queue models with no flow conflicts at nodes and no spillback congestion) and/or on the demand side (e.g. instantaneous shortest paths or System Optimum vs. User Equilibrium).

Thus, Fixed-Point-like formulations, with their poor mathematical properties, and MSA algorithms, with their slow convergence patterns, still dominate the panorama of DTA models today. An up-to-date review of some academic and commercial software available for real-life applications can be found in Barcelo (2010).

In traffic management applications the essential property of a DTA model is its sensitivity (sound reaction) to local changes of the main supply parameters, accidental, such as the capacity reductions due to traffic events, and intentional, such as the signal settings at node intersections due to traffic control. While coping with this requirement, the dynamic models applied in real-time (e.g.: Mahmassani, 2001; Ben-Akiva et al., 2002; Gentile and Meschini, 2011) introduce some simplifying assumption compared to off-line models wrt route choice: given the lack of information and/or experience, users are normally assumed to follow their habitual paths, which leads to the Dynamic Network Loading. On the contrary, in transport planning some simplification to enhance the model properties is possible wrt the representation of network congestion, because pathological conditions, such as spillback and gridlock, are moderated by equilibrium route choices and should be anyhow avoided by proper design.

In this article Gradient Projection algorithms are applied to dynamic assignment. This approach has been successfully applied to solve static traffic assignment, initially with explicit path enumeration (e.g. Jayakrishnan et al., 1994). Lately, similar approaches have been used to develop bush-based algorithm with implicit path enumeration (e.g. Dial, 2006), that allowed a relevant improvement in the state-of-the-practice for the case of stationary equilibrium. For the case of DTA, GP algorithms based on explicit path enumeration are used in Mahut and Florian (2008) in the assignment tool Dynameq, and also the route swapping mechanism proposed by Smith and Mounce (2011) in their more theoretical analysis can be seen as a variant of this method.

1.3 OBJECTIVES AND CONTRIBUTIONS

The main contribution of the present paper is twofold:

- to show how gradient projection can effectively be applied also to sequential local choices at nodes for users directed toward a given destination, and

- to prove that, in the case of deterministic route choices, the Dynamic User Equilibrium based on arc conditional probabilities formulated as a Variational Inequality problem is equivalent to that based on path probabilities.

This paper builds up on the research (Bellei et al., 2005; Gentile et al., 2005; Bellei et al., 2006; Gentile et al., 2007) that brought to the model called DUE (Dynamic User Equilibrium), which is the tool for macroscopic dynamic assignment available in the VISUM software package (Gentile et al., 2006), with the aim of improving convergence to equilibrium wrt MSA.

The sub-models of the supply side (arc model, node model) can be combined in two different ways to produce arc travel times starting from arc flows. The first approach is to consider short time intervals of a few seconds and to process all nodes for each temporal layer in chronological order; this is the classical way of dealing with the network simulation in DTA macro and meso-scopic models (e.g.: Papageorgiou, 1990; Lo and Stezo, 2012). The second approach allows to consider long time interval of few minutes and to process relations among whole temporal profiles, but requires iterating to reproduce spillback (e.g.: Gentile et al., 2007; Himpe et al., 2013). A through comparison between the two approaches has been recently developed in Gentile (2015); in this paper we limit our attention to the second approach.

Hypocritical congestion is here modelled through an arc performance function for whole links, namely the Average Kinematic Wave (AKW) model proposed in Gentile et al. (2005), while hypercritical congestion is modelled through a transmission model for intersections with spillback, namely the Network Performance Function (NPF) proposed in Gentile (2015), which is based on the GLTM (Gentile, 2010). Actually, also the latter is capable of reproducing hypocritical congestion, but relies on the shift of cumulative inflows and outflows to obtain the arc travel time, which is too coarse when the time intervals are long.

As in many other papers on DTA, for economy of space, the focus is here on the user equilibrium and not on the simulation or congestion model. The reader is thus referred to the above papers which specifically deal with the cost function.

The proposed assignment model is based on a continuous representation of time, where each variable is a temporal profile (a continuous function of time). Route choice and flow propagation are performed for each destination separately; temporal layers with (possibly) long time intervals (up to 10-15 minutes) are processed in chronological order (reversed, for route choice). This means that here it is not possible to exploit the acyclicity of the space-time network for the computation of dynamic shortest paths and network loading, like could instead be done if the time was discretized with short time intervals (1-5 seconds). This complication pays back with the possibility of simulating on any modern machine and in a practical time (few hours) large scale networks with many thousands of arcs and hundreds of zones. The original version of the model was based on Logit route choices (Bellei et al., 2005), but in this paper we will refer to its deterministic version.

In this paper, the Dynamic User Equilibrium is sought through Gradient Projection methods, which are capable to solve models of different complexity (depending on the prevailing congestion conditions) with high precision in a reasonable number of iterations. The main contribution of this article is then the proposal and test of a family of GP algorithms for our link-based DTA model.

In deterministic static assignment, the gradient of the objective function (given by the sum of arc cost integrals, originally proposed by Beckmann et al., 1956) wrt path flows is the vector of path costs for each OD pair. This also the operator of the corresponding

Variational Inequality formulation. Thus, in case of explicit path enumeration Gradient Projection is easy. The main novelty of the proposed algorithms lays, however, in the application of the GP method to the local route choice at a node among the arcs of its forward star for users directed toward a given destination, thus adopting implicit path enumeration.

In general, the search direction is in this case obtained by applying a flow shift, from alternatives whose cost is above the average to alternatives whose cost is below the average, that is proportional to the difference between the cost of the alternative and the average cost.

The first tested algorithm is a direct application of the Exact Gradient Projection (EGP) method (Bertsekas, 1976), where the search direction is given by the geometrical projection of the gradient to the polytope (given by the non-negativity flow constraints and the consistency demand constraint) that forms the space of feasible solutions. This implies the solution of a quadratic program, which we propose to address at low computation cost using the Greedy approach, similarly to what is done in LUCE (Gentile, 2014).

The second tested algorithm considers an approximated projection method (Rosen, 1960), here referred to as Quasi Gradient Projection (QGP), where only non-null flows are taken into account in the determination of the search direction; if the new iterate falls out of the space of feasible solutions, then the search direction is shortened so as to satisfy all negativity constraints.

The third tested algorithm is an application of the Reduced Gradient Projection (RGP), which has been extensively applied in static assignment (e.g. Jayakrishnan et al., 1994). The consistency constraint (i.e., the sum of all path flows is equal to the demand flow) is eliminated from the optimization problem by extracting from it the flow of the current shortest path and by substituting in the objective function the latter with the resulting (linear) expression. The key idea is to get the search direction by applying a flow reduction to each non-minimal travel alternative that is proportional to the difference between the cost of the alternative and the cost of the best alternative.

If these projections are scaled by some approximation of the cost Hessian inverse the method provides a Quasi-Newton search. Because the differentiation of path costs in a dynamic framework is not at all trivial (due to the concatenation of travel times and the propagation of flows), then we tested several scaling of the gradient based on the costs themselves.

All such algorithms require to store in memory the (in)flow of each arc for each time interval per destination. Relevant savings can be achieved by recording only the positive flows and reconstructing the missing arc probabilities a posteriori, based on the current solution of dynamic shortest trees.

A further innovation that was introduced in the proposed algorithms concerns the propagation on the network of demand flows travelling towards a given destination based on given travel times and arc conditional probabilities. In our original DTA model (Bellei et al., 2005) this operation requires that only the conditional probabilities of efficient arcs (i.e. arcs that bring the user closer to the destination with respect to some topological order) are positive, since this restrictive condition ensures the possibility of propagating the flows in topological order. However, assuming a fixed (i.e. constant during the day) topological order for each destination (for example resulting from some reference cost pattern) may be not an easy and satisfactory choice; indeed, congestion can evolve into several directions, so that a fixed set of efficient arcs may cut out good paths (in terms of congested cost). To tackle this issue, static assignment algorithms

introduce a (somewhat cumbersome) bush management scheme, where an acyclic sub-graph containing all arcs with positive flow for each destination is updated to include whenever possible optimal arcs (i.e. arcs belonging to the current shortest tree). This process may considerably slow down convergence to an equilibrium on the whole graph. Moreover, in a dynamic context the definition of bushes presents further problems, connected to their temporal dimension; an attempt in this direction can be found in Ramadurai and Ukkusuri (2010).

To overcome these difficulties, we successfully tested here a different approach. This consists in formulating and solving the Flow Propagation Model as a sequence of square linear systems, one for each temporal layer, where each equation represents the flow conservation at a node during the current interval and the unknowns are the flows exiting from each node during the same interval. These systems are rather sparse and almost triangular; can be easily solved through the Gauss-Seidel method or through more specific methods, such as the BICGSTAB.

Casting the dynamic traffic assignment in terms of arc variables with implicit path enumeration and using a macroscopic congestion model (a variant of the Link Transmission Model) is a key factor for the robustness and consistency of the algorithm, on one side, as well as for its numerical tractability and computational efficiency, on the other side.

Indeed, Yang and Jayakrishnan (2012) who applied Gradient Projection to a path based DTA model, explicitly mention that "the method does not converge to the perfect dynamic user equilibrium state due to the use of a finite set of path flow vectors and due to the stochastic nature of the employed microscopic simulation model". Instead, our model implicitly consider all paths of the network and uses a macroscopic traffic model (which allows anyhow to represent spatial queues and spillback). These are probably the reasons why we did not find so far cases where our method did not converge (with a sufficient number of iterations), which leads us to state that *in practice* it is capable of reaching equilibrium. It shall however be highlighted that no proof is provided for this result, but only practical evidence.

It is probably the first time that the convergence results of a DTA model are plotted on a logarithmic scale, like it is custom for static assignment algorithms. We will show how in practice convergence (measured by the relative gap) is reached in an acceptable number of iterations (100) to a good level (e.g. 10^{-4}) for moderate congestion (without spillback) and to a fair level (e.g. 10^{-2}) for high congestion (with spillback).

In the seminal paper by Friesz et al. (1993) the DTA model is cast in the framework of functional analysis where travel time and flow variables are temporal profiles, and a VI formulation is proposed based on path flows. We adopt a similar approach, but our VI formulation is based on arc and node variables only. Although the proposed solution algorithm implies time discretization, we keep the concept of temporal profiles by assuming piecewise linear functions of time for travel times and cumulative flows. To do so on a fixed temporal discretization some (minor) approximation is however necessary.

In this framework, we can consider large time interval of several minutes, which is an extremely relevant feature if computing times are an issue, like in operation.

As clearly pointed out by Friesz and Mookherjee (2006), "the fundamental properties of DUE cannot be ignored simply for the sake of computability, while two features have been particularly over-simplified in many computational studies: the intrinsically nested nature of path delays, and the time shifted natures of arc inflows/outflows needed to construct a rigorous model of flow propagation". We completely agree on this assertion and, based on our experience, we can confirm that any violation of this principles hinder

not only the consistency of the model, but also the practical possibility of converging to equilibrium.

To these requirements we would add that the congestion model shall be realistic and include the representation of spillback and capacity drop. Indeed, the spatial back-propagation of queues to adjacent links is a critical aspect of transport networks which is capable of severely deteriorating traffic conditions and is thus crucial for planning as well as for operation. By the way, in traffic management the local regulation policies, such as ramp metering, signal setting and dynamic speed control, try exactly to avoid the occurrence of these phenomenon. In this context, a DTA model which is not capable of reproducing them would be of little, if no, help.

To support the validity of the proposed algorithms we present some numerical experiments on an elementary network, where it is possible to have expectations on the solution, as well as on larger test networks, to check how the proposed algorithms scale when several destinations interact on longer paths. Different time discretizations with intervals of 6, 60 and 600 sec. are considered. The outcome is very encouraging.

For the first time we were able to achieve in all our experiments on the elementary networks the same level of convergence (in terms of gap function) that we nowadays expect to see in static assignment (e.g. 10^{-5}). Actually the proposed methods are often able to reach nearly double precision.

On the larger network results are still satisfactory, as the proposed methods are always able to reach a minimal convergence (e.g. 10^{-2}) in few iterations, while this objective could not be reached through MSA. Even better results are obtained if the level of congestion is not too high. After all, when spillback occurs the problem becomes strongly non-separable, also in space and not only in time. Thus, we can expect slower convergence than in separable static models.

The model resulting from this research has been implemented in a new version of the software DUE for VISUM, called TRE, which is the simulation engine of PTV-OPTIMA (Gentile and Meschini, 2011), a comprehensive solution for real-time traffic prediction and decision support system.

The paper is organized as follows. Section 2 presents the mathematical background, including the general formulation of User Equilibrium and its solution through Gradient Projection methods. Section 3 presents the Dynamic Traffic Assignment problem and its arc formulation for implicit path enumeration. Section 4 presents at the pseudo-code level the solution algorithms based on GP applied to local choices. Section 5 presents the numerical experiments on both elementary networks and larger test networks. No conclusion section is provided, as this last paragraph of this introduction serves well at this scope.

2. GRADIENT PROJECTION ALGORITHMS FOR THE USER EQUILIBRIUM PROBLEM

2.1 FORMULATIONS OF EQUILIBRIUM

Let's consider a generic equilibrium problem where each user of *group i* must choose one among a non-empty set A_i of available *alternatives* (e.g. the paths connecting a given O-D pair, or the arcs exiting a given node to continue the journey toward a given destination). Let I be the set of homogeneous groups (wrt to the choice) among which

users are partitioned and A be the set of all alternatives: $A = \sqcup_{i \in I} A_i$ ($\sqcup$ denotes the union of disjoint sets).

The share (or fraction) p_a of users choosing the generic alternative $a \in A_i$ is here referred to as its *probability*. The set S_p of feasible probability vectors is a non-empty, convex polytope:

$$S_p = \{\mathbf{p} \in \Re^A : \sum_{a \in A_i} p_a = 1 \ , \forall i \in I; \ p_a \geq 0 \ , \forall a \in A\} \ . \quad (1)$$

The alternatives of different groups are interdependent. Specifically, the *cost* $c_a \geq 0$ of each alternative $a \in A$ is here assumed to be non-negative and to depend (jointly) on the entire choice pattern $\mathbf{p}$ (e.g. through travel demand and arc flows, as clarified in Section 3.3 for the case of DTA), and not (separately) on the probability p_a of that alternative only:

$$c_a = c_a\left(p_b, \forall b \in A\right) \ ; \text{ in compact form: } \mathbf{c} = c(\mathbf{p}) \ . \quad (2)$$

We assume that users are rational decision makers and are perfectly informed; they will then choose an alternative with minimum cost (deterministic behavior). On this base, no user finds convenient to unilaterally change alternative at equilibrium.

In other words, at equilibrium for each group all used alternatives have the same minimum cost, and no unused alternative has a lower cost. Therefore, the average cost among users is equal to the minimum cost. In the case of route choice on transport networks, the above statements are also referred to as Wardrop Principles (1952).

The classical way of formulating the deterministic equilibrium is to find a feasible choice pattern $\mathbf{p}^* \in S_p$ that satisfies the following Complementarity Conditions (CC):

$$\left(c_a\left(\mathbf{p}^*\right) - c_i^{min}\left(\mathbf{p}^*\right)\right) \cdot p_a^* = 0 \quad \forall a \in A_i, \forall i \in I \ , \quad (3)$$

where, for each group $i \in I$, it is:

$$c_i^{min} = Min\left(c_a, \forall a \in A_i\right) \ . \quad (4)$$

At equilibrium $\mathbf{p}^*$, if an alternative $a \in A_i$ is used, i.e. $p_a^* > 0$, then Equation (3) requires that its cost is minimum, i.e. $c_a(\mathbf{p}^*) = c_i^{min}(\mathbf{p}^*)$; if instead that alternative is unused, then its cost can be higher than the minimum, because Equation (3) is satisfied for $p_a^* = 0$.

The equilibrium problem can be also formalized as a (Finite-Dimensional) Variational Inequality (VI), where a feasible choice pattern $\mathbf{p}^* \in S_p$ is sought such that (the sum of) the average cost for all groups $c(\mathbf{p}^*)^T \cdot \mathbf{p}$, for $\mathbf{p} = \mathbf{p}^*$, is minimal wrt any (other) $\mathbf{p} \in S_p$:

$$\sum_{a \in A} c_a\left(\mathbf{p}^*\right) \cdot \left(p_a^* - p_a\right) \leq 0 \quad , \forall \mathbf{p} \in S_p \ ; \quad (5)$$

in compact form: $c(\mathbf{p}^*)^T \cdot (\mathbf{p}^* - \mathbf{p}) \leq 0 \ , \forall \mathbf{p} \in S_p$.

Note that the cost pattern $c(\mathbf{p}^*)$ is that corresponding to the candidate equilibrium $\mathbf{p}^*$. This is also called *descriptive* (or Nash) user equilibrium. Instead, the problem of finding the minimum average costs $c(\mathbf{p})^T \cdot \mathbf{p}$ is called *normative* system equilibrium (but it will not be addressed here).

The first application of VI to traffic assignment was proposed by Smith (1979) and Dafermos (1980) for the static case.

The above VI (5) reflects the definition of equilibrium: if at given choice pattern $\mathbf{p}^* \in S_p$ for some user it is possible to change the alternative to a better one, the resulting new choice pattern $\mathbf{p} \in S_p$ would imply a lower average cost $c(\mathbf{p}^*)^T \cdot \mathbf{p}$; then $\mathbf{p}^*$ is not a solution of the VI nor an equilibrium. If the contrary is true, then $\mathbf{p}^*$ is a solution of the VI and an equilibrium.

The following statements are all equivalent, thus showing the coincidence of the two formulations (3) and (5):

$$\sum_{a\in A_i} c_a\cdot\left(p_a^*-p_a\right)\le 0,\forall \mathbf{p}\in S_p^{A_i} \Leftrightarrow \sum_{a\in A_i} c_a\cdot p_a^* \le Min\left(\sum_{a\in A_i} c_a\cdot p_a,\forall \mathbf{p}\in S_p^{A_i}\right)=c_i^{min}\cdot\left(1=\sum_{a\in A_i} p_a^*\right)\Leftrightarrow$$
$$\sum_{a\in A_i}\left(c_a-c_i^{min}\right)\cdot p_a^*\le 0\Leftrightarrow\left(c_a-c_i^{min}\right)\cdot p_a^*=0,\forall a\in A_i \quad S_p^{A_i}=\left\{\mathbf{p}\in\Re^{A_i}: p_a\ge 0,\forall a\in A_i;\sum_{a\in A_i} p_a=1\right\} \quad (6)$$

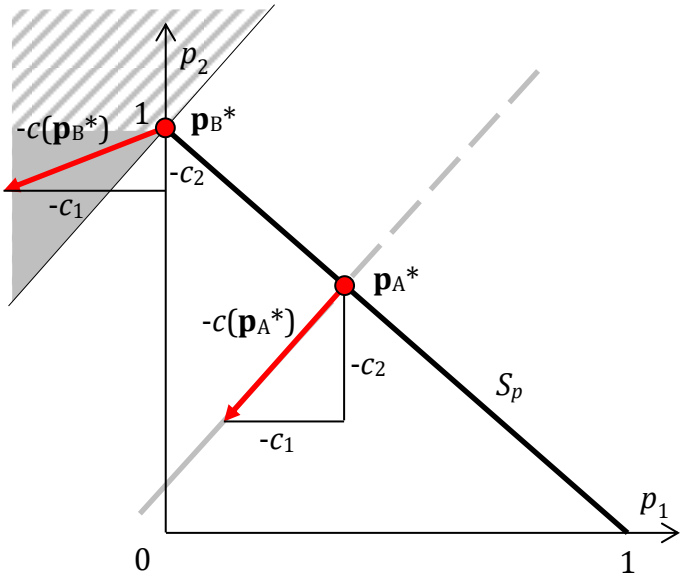

Figure 1. Graphic interpretation of VI, for the case of one group with two alternatives. In grey, the normal cones to the feasible set in points A and B; their unfeasible portion is dashed (costs are non-negative).

With reference to Figure 1, equilibrium can occur: in points like A, where both alternatives are used with the same cost and the cost vector is then perpendicular to the feasible set of probabilities S_p ; in points like B, where some alternative is not used, and the opposite of the cost vector belongs to the *normal cone* wrt the feasible set (the normal cone at point $\mathbf{p}^*\in S_p$ is the set of all directions with which no $\mathbf{p}$-$\mathbf{p}^*$ with $\mathbf{p}\in S_p$ makes an acute angle). The latter is actually the general case, because the normal cone in A is the perpendicular line to S_p ; this is exactly what stated by the VI. If the cost vector is the gradient of an objective function, the VI is the necessary first order condition for its minimization in the feasible set.

Let $\gamma(\mathbf{p})\in[0,1]$ be the *relative gap function*:

$$\gamma(\mathbf{p})=Max\left(1-\frac{c(\mathbf{p})^T\cdot\mathbf{x}}{c(\mathbf{p})^T\cdot\mathbf{p}},\text{st:}\,\mathbf{x}\in S_p\right). \quad (7)$$

Given a candidate equilibrium $\mathbf{p}\in S_p$, the relative gap $\gamma(\mathbf{p})$ yields the best improvement on the sum of average costs that is achievable by shifting choice probabilities to better alternatives. At equilibrium the gap function shall then be null, as by definition no improvement is achievable. The problem can then be formulated as the following non-differentiable Min-Max program (MM):

$$Min\left(\gamma(\mathbf{p}),\text{st:}\,\mathbf{p}\in S_p\right). \quad (8)$$

The relative gap function is also equivalent to one minus the ratio between the sums of minimum and average costs:

$$\gamma(\mathbf{p}) = 1 - \frac{\sum_{i \in I} c_i^{min}(\mathbf{p})}{\sum_{i \in I} c_i^{med}(\mathbf{p})} \ , \tag{9}$$

where, for each group $i \in I$, it is:

$$c_i^{med} = \sum_{a \in A_i} c_a \cdot p_a \ . \tag{10}$$

Indeed, given a cost pattern $c(\mathbf{p})$, the minimum average cost for all groups is obtained by assigning all users to minimum cost alternative(s):

$$Max\left(1 - \frac{c(\mathbf{p})^T \cdot \mathbf{x}}{c(\mathbf{p})^T \cdot \mathbf{p}}, \text{st: } \mathbf{x} \in S_p \right) = 1 - \frac{Min\left(c(\mathbf{p})^T \cdot \mathbf{x}, \text{st: } \mathbf{x} \in S_p \right)}{c(\mathbf{p})^T \cdot \mathbf{p}} = 1 - \frac{\sum_{i \in I} c_i^{min}(\mathbf{p})}{\sum_{i \in I} c_i^{med}(\mathbf{p})} \ . \tag{11}$$

The gap function can be taken as an indicator of how far we are from an equilibrium at the current choice pattern $\mathbf{p}$, and can thus be used in the stop criterion of the solution algorithm.

Interestingly, by summing up Equations (3) we get:

$$\sum_{i \in I} c_i^{med}(\mathbf{p}) - \sum_{i \in I} c_i^{min}(\mathbf{p}) = 0 \ . \tag{12}$$

Based on (9) this is another formulation of equilibrium, i.e. the problem of finding $\mathbf{p}^* \in S_p$ as the feasible Zeros of a Function (ZF):

$$\gamma(\mathbf{p}^*) = 0 \ . \tag{13}$$

Let $P(\mathbf{x}, X, \mathbf{G})$ denote in general the $\mathbf{G}$-norm projection of $\mathbf{x} \in \Re^A$ on a convex set $X \subseteq \Re^A$, i.e. the one to one map which provides the unique solution $\mathbf{y}^*$ of the following minimum distance problem:

$$Min\left(\|\mathbf{y} - \mathbf{x}\|_{\mathbf{G}} = \sqrt{(\mathbf{y} - \mathbf{x})^T \cdot \mathbf{G} \cdot (\mathbf{y} - \mathbf{x})}, \text{st: } \mathbf{y} \in X \right) \ , \tag{14}$$

where $\mathbf{G}$ is any symmetric positive definite matrix. Also based on Figure 1, VI (5) is equivalent to the following fixed-point problem based on (minus) Cost Projection (CP):

$$\mathbf{p}^* = P(\mathbf{p}^* - \mathbf{G}^{-1} \cdot c(\mathbf{p}^*), S_p, \mathbf{G}) \ , \tag{15}$$

where $\mathbf{G}$ is useful to scale the costs in the space of probabilities. Equilibrium is then attained when the projection of the cost vector opposite, scaled by the inverse of $\mathbf{G}$, coincides with the current probability vector.

The above fixed-point formulation of equilibrium is totally different from the classical Fixed-Point problem (FP) based on the composition of two functions, one for the demand and the other one for the supply (e.g. Cantarella, 1997):

$$\mathbf{p}^* \in p(c(\mathbf{p}^*)) \ ; \tag{16}$$

here the (one to many) map $p(\mathbf{c})$ expresses the deterministic choice model of Wardrop. This map yields all feasible probability vectors $\mathbf{p}$ that, for a given cost vector $\mathbf{c}$, satisfy the complementarity conditions (3).

The *choice map* $p(c(\mathbf{p}))$ and the *projection map* $P(\mathbf{p} - \mathbf{G}^{-1} \cdot c(\mathbf{p}), S_p, \mathbf{G})$ coincide only at equilibrium. Note that (a point of) the choice map is usually obtained as an all-or-nothing assignment to minimum cost alternatives, thus being intrinsically unstable and discontinuous when close to an equilibrium of type A; instead, the projection map is continuous, if the cost function is such, and shifts probabilities to better alternatives in inversely proportional manner to their cost (more details will be provided in Section 2.2), thus being more suited for equilibrium algorithms.

If the cost function is continuously differentiable and its Jacobian $\nabla c(\mathbf{p})$ is symmetric for all $\mathbf{p} \in S_p$ (not applicable in DTA models), then VI (5) is equivalent to the first order necessary conditions of the following Non-Linear optimization program (NL):

$$Min\left(\varphi(\mathbf{p}) = \int_{\mathbf{0}}^{\mathbf{p}} c(\mathbf{x})^T \cdot d\mathbf{x}, \text{st: } \mathbf{p} \in S_p \right) , \tag{17}$$

where the curve integral provides the same result following any path in the space of probabilities from **0** to **p**. Indeed, the *gradient* of the objective function is the cost vector:

$$\nabla\varphi(\mathbf{p}) = c(\mathbf{p}) . \tag{18}$$

The first application to traffic assignment of this approach can be found in the seminal work of Beckmann et al. (1956).

Referring to (15), we will then talk of Gradient Projection also in the general (non-differentiable or asymmetric) case, where no objective function is available. In other (naïve) words, VI can be interpreted as the first order conditions of some optimization problem which may be impossible to formulate. The cost function of the VI is thus the gradient of this "phantom" NL.

The equilibrium can then be equivalently be formulated as CC, VI, MM, ZF, CP, FP and NL (only in case of symmetry). Existence of a solution is ensured if $c(\mathbf{p})$ is continuous and can be proved by applying Brower's theorem to the FP (16). Uniqueness of the solution is ensured if $c(\mathbf{p})$ is strictly monotone (increasing) over S_p:

$$(c(\mathbf{p}_2) - c(\mathbf{p}_1))^{\mathrm{T}} \cdot (\mathbf{p}_2 - \mathbf{p}_1) > 0 , \forall\ \mathbf{p}_2 \in S_p \neq \mathbf{p}_1 \in S_p ; \tag{19}$$

a proof by contradiction is provided in Cantarella (1997).

2.2 BACKGROUNDS ON PROJECTION ALGORITHMS

Inspired by the fixed-point formulation (15), one of the most immediate algorithms to solve VI (5) and then obtaining an equilibrium is the Gradient Projection (GP) method. Like other methods (e.g. Newton), this consists in solving at each iteration n a VI with an approximated cost function $c_n(\mathbf{p}) \cong c(\mathbf{p})$ to obtain the next iterate $\mathbf{p}_{n+1}$:

$$c_n(\mathbf{p}_{n+1})^{\mathrm{T}} \cdot (\mathbf{p}_{n+1} - \mathbf{p}) \leq 0 , \forall \mathbf{p} \in S_p . \tag{20}$$

In this case, the approximated cost function is defined as a deviation from the original cost function evaluated at the current iterate $\mathbf{p}_n$:

$$c_n(\mathbf{p}) = c(\mathbf{p}_n) + \mathbf{G} \cdot (\mathbf{p} - \mathbf{p}_n) . \tag{21}$$

By applying the CP formulation (15) to the VI (20) , based on (21) we have:

$$\mathbf{p}_{n+1} = P(\mathbf{p}_{n+1} - \mathbf{G}^{-1} \cdot (c_n(\mathbf{p}_{n+1}) = c(\mathbf{p}_n) + \mathbf{G} \cdot (\mathbf{p}_{n+1} - \mathbf{p}_n)) , S_p , \mathbf{G}) . \tag{22}$$

Thus, the solution of the VI relative to the generic iteration n is obtained by calculating the projection of the (suitably scaled) anti-gradient $-\mathbf{G}^{-1} \cdot c(\mathbf{p}_n)$ on S_p :

$$\mathbf{p}_{n+1} = P(\mathbf{p}_n - \mathbf{G}^{-1} \cdot c(\mathbf{p}_n) , S_q , \mathbf{G}) . \tag{23}$$

The algorithm starts with a feasible choice pattern $\mathbf{p}_1 \in S_p$ and for n = 1, 2, … obtains through (23) a sequence of feasible probability vectors $\{\mathbf{p}_n\}$.

The latter converges to the unique $\mathbf{p}^*$ which solves the original VI, under the following conditions: the cost function $c(\mathbf{p}_n)$ is Lipshitz continuous (i.e. there is a limit in how fast it can change) and strongly (which is more than strictly) monotone for each $\mathbf{p}_2 \in S_p$ and $\mathbf{p}_1 \in S_p$:

$$||c(\mathbf{p}_2) - c(\mathbf{p}_1)||_2 \leq \beta \cdot ||\mathbf{p}_2 - \mathbf{p}_1||_2 , \tag{24}$$

$$(c(\mathbf{p}_2) - c(\mathbf{p}_1))^{\mathrm{T}} \cdot (\mathbf{p}_2 - \mathbf{p}_1) \geq \gamma \cdot ||\mathbf{p}_2 - \mathbf{p}_1||_2^2 , \tag{25}$$

with constants β and γ such that:

$$(\beta/\lambda_{min}(\mathbf{G}))^2 < 2 \cdot (\gamma/\lambda_{max}(\mathbf{G})) , \tag{26}$$

where $\lambda_{min}(\mathbf{G})$ and $\lambda_{max}(\mathbf{G})$ denote, respectively, the smallest and largest eigenvalue of $\mathbf{G}$. The proof of the above (and more general) results can be found in Harker and Pang (1990). A similar result is provided by Friesz and Mookherjee (2006) for a DTA model with continuous time formulation.

In practice, the above conditions are not proved to apply in DTA, so that the solution may be not unique and the sequence $\{\mathbf{p}_n\}$ may not converge to it. However, we can consider the probability vector produced by the projection (23) as a good direction for a local search, while the new iterate shall be obtained through a line search or backtracking technique by taking a suitable step size (a more formal discussion on these issues can be found in Facchinei and Pang, 2003).

To this aim, the availability of the gap function $\gamma(\mathbf{p})$ as a metric of equilibrium is only partially helpful, because its calculation may have the same computational cost of the direction itself. Thus we can use a nonsummable diminishing step size.

Similarly, by applying the Method of Successive Averages (MSA) for solving the fixed-point problem CP (15) we obtain the following iteration rule:

$$\mathbf{p}_{n+1} = \alpha_n \cdot P(\mathbf{p}_n - \mathbf{G}^{-1} \cdot c(\mathbf{p}_n)) , S_p , \mathbf{G}) + (1-\alpha_n)\cdot \mathbf{p}_n . \quad (27)$$

Under the assumptions of the Blum theorem (1954), the convergence of this method is guaranteed (almost surely, in presence of random variables) if the sequence of step sizes $\{\alpha_n\}$ satisfies the following conditions:

$$\sum_n \alpha_n = \infty \quad , \quad \lim_{n\to\infty} \alpha_n = 0 \ . \quad (28)$$

The first condition guarantees that the step sizes will not be too large, while the second collectively assures that the step sizes will not be too small. In the conventional MSA it is: $\alpha_n = 1/(n+1)$. However, at the beginning α_n could be too large, and therefore the gap function does not decrese until a number of iterations. In contrast, after a large number of iterations, the step size could become too small, such that the convergence speed becomes extremely slow. Coping with these issues, Liu et al. (2009) provide several methods for choosing a sequence of steps that comply with (28), depending on the required accuracy one wants to reach.

Note that the MSA algorithm is the most widely used approach to solve DTA problems but it is generally applied to the FP (16), as in Mounce R. (2007) and Mounce and Carey (2014), where convergence is proved under strict monotonicity assumptions. However, we shall underline that the proof of convergence does not necessarily imply good performance in practice. This motivated our research for better performing algorithms.

In our DTA model (see Section 3), continuity holds (for a formal proof see Han et al., 2015), but monotonicity doesn't. For this reason, the GP algorithms that will be proposed in the following are only heuristics with no guarantee of convergence. Nevertheless, in our numerical tests the methods always converged to an equilibrium. Further investigation on the Network Congestion Model may allow to identify specific mathematical properties and conditions to ensure convergence and uniqueness, but this is out of the scope of this paper.

2.3 THE PROPOSED METHOD

In this paper we will investigate the performance of a class of GP methods to solve DTA by applying the following generic iterate:

$$\mathbf{p}_{n+1} = P(\mathbf{p}_n - \mathbf{G}_n^{-1} \cdot \alpha_n \cdot c(\mathbf{p}_n)) , S_p , \mathbf{G}_n) . \quad (29)$$

Here, the matrix $\mathbf{G}_n$ can be chosen at each iteration. This allows for a better scaling of the cost/gradient, which due to congestion can assume a large range of values.

Moreover, the nonsummable diminishing step sizes α_n are applied directly to the gradient $\mathbf{c} = c(\mathbf{p}_n)$. This proved to be effective in our numerical tests, because it makes possible to reach a null value of probability for a given alternative (if needed) in a finite number of iterations, which is impossible when adopting the classical MSA approach (27).

We adopted the following formula for the step size of iteration n:

$$\alpha_n = \left(\frac{\eta_1}{\eta_1 + n_{bad}} \right)^{\eta_2} , \tag{30}$$

where n_{bad} is the number of "bad" iterations obtained in the previous n-1 iterations. This events induce to scale down the gradient in order to obtain a smoother, but slower, convergence. An iteration is considered as bad, if the gap function has not decreased sufficiently (as desired), or has increased instead:

$$n_{bad} = \sum_{\iota=2}^{n-1} Bool\left(\gamma_\iota \geq \eta_3 \cdot \gamma_{\iota-1}\right); \tag{31}$$

the Boolean function applies to a Boolean expression x :
$Bool(x) = 1$, if x = TRUE, $Bool(x) = 0$, if x = FALSE.
Note that this check is possible only a posteriori with one iteration of delay, because the gap function γ_n is available after re-computing the minimum costs during iteration n. In our numerical examples we have used a multiplier $\eta_1 = 2$, an exponent $\eta_2 = 0.66$ and a gap reduction factor $\eta_3 = 1$.

We assume that matrix $\mathbf{G}_n / \alpha_n$ is diagonal with known positive entries $g_a > 0$ for $a \in A$, each one suitable to scale the cost c_a into the space of probabilities. A variety of options can be adopted leading to different algorithms.

Two practical ways of defining the scale factors are:

$g_a = c_i^{min} / (\rho \cdot \alpha_n)$, $\forall a \in A_i$; (32)

$g_a = c_a / (\rho \cdot \alpha_n)$, $\forall a \in A_i$. (33)

The additional parameter $\rho > 0$ (whose typical value is 1) is a fixed cost/gradient multiplier and can be used to improve convergence. As mentioned already, values lower than 1 ensure a smoother but slower convergence, while values higher than 1 may accelerate convergence at the risk of instability.

If it is possible to compute the partial derivatives of the cost function, by setting:

$$g_a = \frac{1}{\rho \cdot \alpha_n} \cdot \frac{\partial c_a(\mathbf{p})}{\partial p_a} , \tag{34}$$

we obtain a Linearized Jacobi method, with better convergence properties than the simple Gradient Projection. A similar approach is to consider for $a \in A_i$ the derivative of the non-common cost with the best alternative $a_i^* \in ArgMin(c_a , \forall a \in A_i)$ of the group:

$$g_a = \frac{1}{\rho \cdot \alpha_n} \cdot \left(\frac{\partial c_a(\mathbf{p})}{\partial p_a} - \frac{\partial c_{a_i^*}(\mathbf{p})}{\partial p_a} \right) . \tag{35}$$

Although no proof of convergence is available, we will then consider as a valid iterate for finding the equilibrium the projection $\hat{\mathbf{p}} = \mathbf{p}_{n+1}$ of the anti-gradient $-\mathbf{c}$ in the polytope of feasible probabilities S_p calculated at the current choice patterns $\mathbf{p} = \mathbf{p}_n \in S_p$. Based on (14), this is the result of the following quadratic program:

$$Min\left(0.5 \cdot \sum_{a \in A} \left(p_a - \frac{c_a}{g_a} - \hat{p}_a \right)^2 \cdot g_a , \text{st: } \sum_{a \in A_i} \hat{p}_a = 1, \forall i \in I ; \hat{p}_a \geq 0 , \forall a \in A \right) , \tag{36}$$

where the $\mathbf{G}$-norm distance between point $\mathbf{p} - \alpha_n \cdot \mathbf{G}^{-1} \cdot \mathbf{c}$ and its projection $\hat{\mathbf{p}}$ on the feasible

set S_p is minimized.

Problem (36) can be transposed in the space of *probability shifts* $\Delta p_a = \hat{p}_a - p_a$:

$$Min\left(0.5 \cdot \sum_{a \in A} \left(\Delta p_a + \frac{c_a}{g_a} \right)^2 \cdot g_a \text{ ,st: } \sum_{a \in A_i} \Delta p_a = 0 \,, \forall i \in I; p_a + \Delta p_a \geq 0 \,, \forall a \in A \right) . \tag{37}$$

The solution of (37) is unique (we have a strictly convex objective function and non-empty compact convex domain) and can be applied in the Exact Gradient Projection (EGP) algorithm to find the new iterate of the probabilities:

$$\hat{p}_a = p_a + \Delta p_a \,. \tag{38}$$

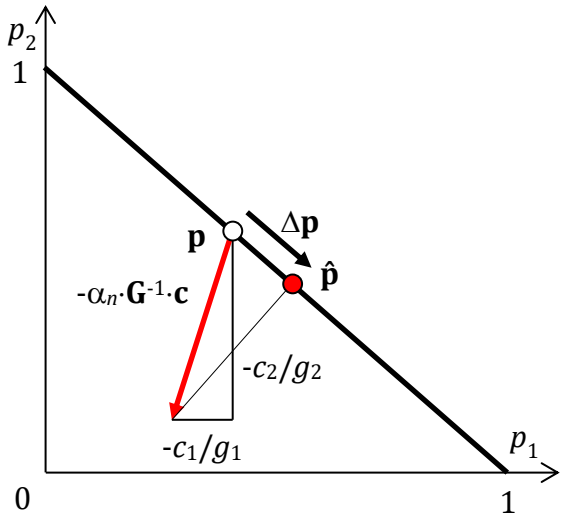

Figure 2. Graphic interpretation of the Gradient Projection iterate for the case of two alternatives. If $c_1/g_1 < c_2/g_2$, like depicted above, then $\Delta p_1 > 0$ and $\Delta p_2 < 0$.

2.4 THE GREEDY SOLUTION APPROACH

The Lagrangian problem of (37) is:

$$Min\left(L = 0.5 \cdot \sum_{a \in A} \left(\Delta p_a + \frac{c_a}{g_a} \right)^2 \cdot g_a - \sum_{i \in I} \lambda_i \cdot \sum_{a \in A_i} \Delta p_a \text{, st: } p_a + \Delta p_a \geq 0 \,, \forall a \in A \right) . \tag{39}$$

The derivative of the Lagrangian for an alternative $a \in A_i$ of group $i \in I$ is:

$$\frac{\partial L}{\partial \Delta p_a} = c_a + \Delta p_a \cdot g_a - \lambda_i \,. \tag{40}$$

The first order conditions of (39) are then:

$$\begin{cases} \dfrac{\partial L}{\partial \Delta p_a} \cdot \left(p_a + \Delta p_a \right) = 0 \quad , \forall a \in A \\ \dfrac{\partial L}{\partial \lambda_i} = 0 \quad , \forall i \in I \\ \dfrac{\partial L}{\partial \Delta p_a} \geq 0 \quad , \forall a \in A \\ p_a + \Delta p_a \geq 0 \quad , \forall a \in A \end{cases} , \quad \begin{cases} \left(\lambda_i - c_a - \Delta p_a \cdot g_a \right) \cdot \left(p_a + \Delta p_a \right) = 0 \quad , \forall a \in A \\ \sum_{a \in A_i} \Delta p_a = 0 \quad , \forall i \in I \\ \lambda_i - c_a \leq \Delta p_a \cdot g_a \quad , \forall a \in A \\ p_a + \Delta p_a \geq 0 \quad , \forall a \in A \end{cases} . \tag{41}$$

Based on the first equation in (41), for each alternative at least one of the two complementarity condition shall be equal to zero. On this base, we can partition the

alternatives of each group $i \in I$ in two sets, namely B_i , including the alternatives that have a positive probability in the projection, and A_i-B_i , including the other alternatives that are set to zero in the projection:

$$\begin{cases} \Delta p_a = \dfrac{\lambda_i - c_a}{g_a} > -p_a \quad ,\forall a \in B_i \\ \Delta p_a = -p_a \quad ,\forall a \in A_i - B_i \end{cases} . \tag{42}$$

Using (42) in the consistency constraint yields:

$$\sum_{a \in A_i} \Delta p_a = 0 \;\rightarrow\; \sum_{a \in B_i} \frac{\lambda_i - c_a}{g_a} - \sum_{a \in A_i - B_i} p_a = 0 \;\rightarrow\; \lambda_i = \frac{\sum_{a \in B_i} \dfrac{c_a}{g_a} + \sum_{a \in A_i - B_i} p_a}{\sum_{a \in B_i} \dfrac{1}{g_a}} , \tag{43}$$

$$\lambda_i = \frac{1 + \sum_{a \in B_i} \left(\dfrac{c_a}{g_a} - p_a \right)}{\sum_{a \in B_i} \dfrac{1}{g_a}} . \tag{44}$$

Let's denote by $c_i^{avg}(B_i)$ the average cost of the alternatives belonging to set B_i weighted by the scale factors reciprocal $1/g_a$:

$$c_i^{avg}(B_i) = \frac{\sum_{a \in B_i} \dfrac{c_a}{g_a}}{\sum_{a \in B_i} \dfrac{1}{g_a}} . \tag{45}$$

Note that, based on (45), if the scale factors of each group are all equal as in (32), then the weighted average cost $c_i^{avg}(B_i)$ becomes the mean cost of used alternatives:

$$c_i^{avg}(B_i) = \frac{\sum_{a \in B_i} c_a}{|B_i|} . \tag{46}$$

Instead, if each scale factor is proportional to the corresponding cost as in (33), then the weighted average cost $c_i^{avg}(B_i)$ becomes the reciprocal of the arithmetic mean of the reciprocal costs (also called the harmonic mean):

$$c_i^{avg}(B_i) = \frac{|B_i|}{\sum_{a \in B_i} \dfrac{1}{c_a}} . \tag{47}$$

Equation (43) can then be rewritten also as follows:

$$\lambda_i = c_i^{avg}(B_i) + \frac{\sum_{a \in A_i - B_i} p_a}{\sum_{a \in B_i} \dfrac{1}{g_a}} . \tag{48}$$

Assume a set B_i is given. We can compute the value of the Lagrangian multiplier λ_i through (48) and then the probability shifts Δp_a through (42). To fully satisfy (41), including the last two conditions, and thus state that B_i is optimal, the following must hold true:

$$\begin{cases} \lambda_i - c_a \le \Delta p_a \cdot g_a \quad ,\forall a \in A_i - B_i \\ p_a + \Delta p_a \ge 0 \quad ,\forall a \in B_i \end{cases} . \tag{49}$$

Using the probability shifts of (42), and denoting:

$x_a = c_a - p_a \cdot g_a$, (50)

the above condition becomes:

$$\begin{cases} \lambda_i \le x_a & ,\forall a \in A_i - B_i \\ \lambda_i \ge x_a & ,\forall a \in B_i \end{cases} . \quad (51)$$

Let's ideally order along a line the alternatives for increasing value of x_a . Based on (51), the optimality of B_i is ensured if λ_i separates the two groups of alternatives. Thus, set B_i is necessarily constituted by the first $|B_i|$ alternatives. The idea is then to eliminate from B_i all the alternatives with $x_a \ge \lambda_i$, because based on (42) the corresponding projection implies a non-positive probability: $p_a + \Delta p_a = (\lambda_i - x_a)/g_a \le 0$, which contradicts the assumption on B_i .

Based on the above considerations, a simple method for finding a set B_i that actually satisfies (51), and thus yields through (44) and (42) the desired probability shifts, is provided by the following "greedy algorithm":

Greedy Algorithm for Exact Gradient Projection
start with $B_i = A_i$,
then iteratively:

- compute λ_i through (44) ,
- compute Δp_a through (42) ,
- eliminate from B_i the alternatives with $p_a + \Delta p_a \le 0$,
- loop until no further alternative has been eliminated .

The algorithm stops in less than $|A_i|$ iterations, because each time some alternative is eliminated from B_i . To prove the validity of the above Greedy Algorithm, let's rewrite (44) using (50) with reference to set B_i as follows:

$$\lambda_i(B_i) = \frac{1 + \sum_{a \in B_i} \frac{x_a}{g_a}}{\sum_{a \in B_i} \frac{1}{g_a}} . \quad (52)$$

The following inductive relation holds:

$$\lambda_i(B_i) = \frac{\lambda_i(B_i - b) \cdot \left(\sum_{a \in B_i - b} \frac{1}{g_a} \right) + x_b \cdot \left(\frac{1}{g_b} \right)}{\sum_{a \in B_i} \frac{1}{g_a}} . \quad (53)$$

which shows that $\lambda_i(B_i)$ is an average with positive weights between $\lambda_i(B_i\text{-}b)$ and x_b . Hence, $\lambda_i(B_i)$ is in the middle. On this base, for each alternative b that we eliminate from B_i it is:

$p_b + \Delta p_b \le 0 \Rightarrow x_b \ge \lambda_i(B_i) \Rightarrow \lambda_i(B_i) \ge \lambda_i(B_i\text{-}b)$. (54)

Therefore during the Greedy Algorithm λ_i decreases while alternatives are removed from B_i ; the algorithm stops at most when B_i is a singleton and in any case when $\lambda_i(B_i)$ separates the two set of alternatives, as required by (51).

In conclusion, the optimal probability shifts are:

$$\begin{cases} \Delta p_a = \dfrac{c_i^{avg}\left(B_i\right) - c_a}{g_a} + \dfrac{\dfrac{1}{g_a}}{\displaystyle\sum_{b\in B_i}\frac{1}{g_b}} \cdot \sum_{b\in A_i - B_i} p_b \quad ,\forall a \in B_i \\ \Delta p_a = -p_a \quad ,\forall a \in A_i - B_i \end{cases} . \tag{55}$$

The probabilities of set A_i-B_i that are put to zero by the shifts are spread among the remaining active alternatives B_i proportionally to the reciprocal of the scale factors.

Because at the solution $\lambda_i(B_i)$ shall separate the two sets, based on (53) it is also minimum. Based on (52) if set B_i is constituted by only one alternative, say b, we have:

$\lambda_i(b) = x_b + g_b$. (56)

The algorithm can then be improved with the following better initialization of B_i :

$$B_i = \left\{a \in A_i : x_a < x_b + g_b\right\} , \tag{57}$$

where b can, for example, be (one of) the alternative with minimum cost c_a or, even better, (one of) the alternative with the lowest $\lambda_i(a)$.

2.5 QUASI GRADIENT PROJECTION

The Quasi Gradient Projection (QGP) is obtained with reference to a specific subset B_i of alternatives without considering the non-negativity constraints, while the probabilities of set A_i-B_i are not modified.

In this case, the first order conditions of (39) yield:

$$\begin{cases} \dfrac{\partial L}{\partial \Delta p_a} = 0 \quad ,\forall a \in B_i \\ \dfrac{\partial L}{\partial \lambda_i} = 0 \quad ,\forall i \in I \end{cases} , \quad \begin{cases} \left(\lambda_i - c_a - \Delta p_a \cdot g_a\right) = 0 \quad ,\forall a \in B_i \\ \displaystyle\sum_{a\in A_i} \Delta p_a = 0 \quad ,\forall i \in I \end{cases} . \tag{58}$$

Substituting the shift given by the first equation of (58) in the second equation we get:

$$\sum_{a\in B_i} \Delta p_a = 0 \;\rightarrow\; \sum_{a\in B_i} \frac{\lambda_i - c_a}{g_a} = 0 \;\rightarrow\; \lambda_i = \frac{\displaystyle\sum_{a\in B_i}\frac{c_a}{g_a}}{\displaystyle\sum_{a\in B_i}\frac{1}{g_a}} = c_i^{avg}\left(B_i\right) . \tag{59}$$

Then probability shifts are:

$$\begin{cases} \Delta p_a = \dfrac{c_i^{avg}\left(B_i\right) - c_a}{g_a} \quad ,\forall a \in B_i \\ \Delta p_a = 0 \quad ,\forall a \in A_i - B_i \end{cases} . \tag{60}$$

On this basis, if the cost is higher than the average, then the probability will decrease; the contrary is true if the cost is lower than the average.

It is immediate to verify that if set A_i-B_i is constituted by alternatives with null probability $p_a = 0$, then the above probability shifts coincide with those of the gradient projection (55).

The shifts provided by (60) can lead to a negative probability for some alternative $a \in B_i$. Moreover, if the current probability p_a of such alternative is null (so that $p_a = 0$ and $\Delta p_a < 0$), then clearly the search direction $\mathbf{p}+\mathbf{\Delta p}$ obtained from set B_i is unfeasible and this alternative should be removed from B_i .

Once B_i has no such alternatives, if $p_a > 0$ and $p_a + \Delta p_a < 0$, then we shall "shorten" the step in the search direction until it satisfies the non-negativity constraint by applying the following correction:

$$\beta_i = Min\left(1, -\frac{p_a}{\Delta p_a}, \forall a \in B_i : \Delta p_a < 0\right), \tag{61}$$

$$\hat{p}_a = p_a + \beta_i \cdot \Delta p_a . \tag{62}$$

The set B_i should be the largest possible. Its determination is then attained through the following algorithm:

Greedy Algorithm for Quasi Gradient Projection
start with $B_i = A_i$,
then iteratively:
- compute Δp_a through (60) ,
- eliminate from B_i the alternatives with $p_a = 0$ and $\Delta p_a < 0$,
- loop until no further alternative has been eliminated .

finally apply (61) and (62).

2.6 REDUCED GRADIENT PROJECTION

The probability of the best alternative $a_i^* \in ArgMin(c_a , \forall a \in A_i)$ of each group $i \in I$ can be eliminated from problem (17) using the consistency constraint: $p_{a_i^*} = 1 - \sum_{a \in A_i - a_i^*} p_a$; thus obtaining a problem with non-negativity constraints only, whose objective function can be written as:

$$\varphi = \int_{\mathbf{0}}^{\bar{\mathbf{p}}} c(\mathbf{x})^T \cdot d\mathbf{x} \quad , \quad \bar{p}_a = p_a \quad \forall a \in A_i - a_i^*, \quad \bar{p}_{a_i^*} = 1 - \sum_{a \in A_i - a_i^*} p_a \ . \tag{63}$$

The Reduced Gradient Projection (RGP) considers in principle as search direction the anti-gradient of the above objective function, for each $a \in A_i - a_i^*$:

$$-\frac{\partial \varphi}{\partial p_a} = c_{a_i^*} - c_a \ . \tag{64}$$

This shall be suitably scaled in order to apply shifts it in the space of probabilities. The scale factor is doubled wrt that used in the EGP and QGP algorithms because the difference cost from the best alternative is (on average) twice as that from the average cost:

$$\Delta p_a = \frac{c_{a_i^*} - c_a}{2 \cdot g_a} \quad , \forall a \in A_i - a_i^* \ . \tag{65}$$

The resulting shifts, which are all non-positive, may however produce some negative probabilities. We have then to limit the projection so that the non-negativity constraints are satisfied. Finally, the new iterate of the best alternative a_i^* is obtained through the consistency constraint:

$$\begin{cases} \hat{p}_a = Max(0, p_a + \Delta p_a) \quad , \forall a \in A_i - a_i^* \\ \hat{p}_{a_i^*} = 1 - \sum_{a \in A_i - a_i^*} \hat{p}_a \end{cases} . \tag{66}$$

Mahut and Florian (2008) propose to consider as reference alternative a_i^* the one with the highest probability, instead of that with the lowest cost. In our numerical test this option did not show evident advantages wrt the classical one. Moreover, they

propose a "quasi" reduced gradient approach by shortening the search direction as in (61). Again, this option did not prove to be clearly convenient wrt the classical one.

The Min-Max Projection (MMP) is a variant of the RGP algorithm (e.g. Dial, 2006). It involves only the best alternative a_i^* and the worst used alternative $a_i^\times \in ArgMax(c_a\ , \forall a \in A_i: p_a > 0)$. The resulting projection is computed as follows:

$$\begin{cases} \Delta p_{a_i^\times} = \dfrac{c_{a_i^*} - c_{a_i^\times}}{2 \cdot g_{a_i^\times}} \\ \hat{p}_{a_i^\times} = Max\left(0, p_{a_i^\times} + \Delta p_{a_i^\times}\right) \\ \hat{p}_a = p_a \quad , \forall a \in A_i - a_i^* - a_i^\times \\ \hat{p}_{a_i^*} = 1 - \sum_{a \in A_i - a_i^*} \hat{p}_a \end{cases} . \tag{67}$$

3. ARC BASED FORMULATION OF DTA

3.1 NETWORK REPRESENTATION

In a dynamic framework each variable is a function of the clock time τ. We assume that the network is empty out of the *simulation period* $T \subseteq \Re$.

The road network constituting the transport *supply* is represented here by means of a directed graph (N, A), where N is the set of *nodes* and $A \subseteq N \times N$ is the set of *arcs*. Each arc $a \in A$ is described through a vector of characteristics $\delta_a(\tau)$ that allow to represent its performance (time and cost) and its dependence on traffic flows (congestion); arc characteristics are usually constant in time, but may presents temporary variations. The initial node of the generic arc $a \in A$ is referred to as *tail* and denoted $a^- \in N$, while the final node is referred to as *head* and denoted $a^+ \in N$. The set of arcs exiting the generic node $i \in N$ is referred to as it *forward star* and denoted $i^+ = \{a \in A: a^- = i\}$. Symmetrically, the set of arcs entering node $i \in N$ is referred to as *backward star* and denoted $i^- = \{a \in A: a^+ = i\}$.

Let $Z \subseteq N$ be the subset of nodes, called *zone centroids*, where trips can start and end. Let G be the set of *user classes*. The travel *demand* is given as a fixed (but time varying) flow $d_{odg}(\tau)$ of class $g \in G$ users departing at time τ from origin $o \in Z$ and directed toward destination $d \in Z$. We assume that on the graph there exists a non-empty set K_{id} of acyclic *paths* connecting each node $i \in N$ to every destination $d \in Z$. Let $K = \cup_{od \in Z \times Z} K_{od}$.

3.2 FIXED-POINT SCHEMA

In this section the DTA model is presented through a conceptual schema, as a Fixed Point problem. A more convenient mathematical formulation as a Variational Inequality problem with full explanation of all variables and functions will be provided later in this paper. Here we concentrate our attention on the essential aspects of the modelling architecture.

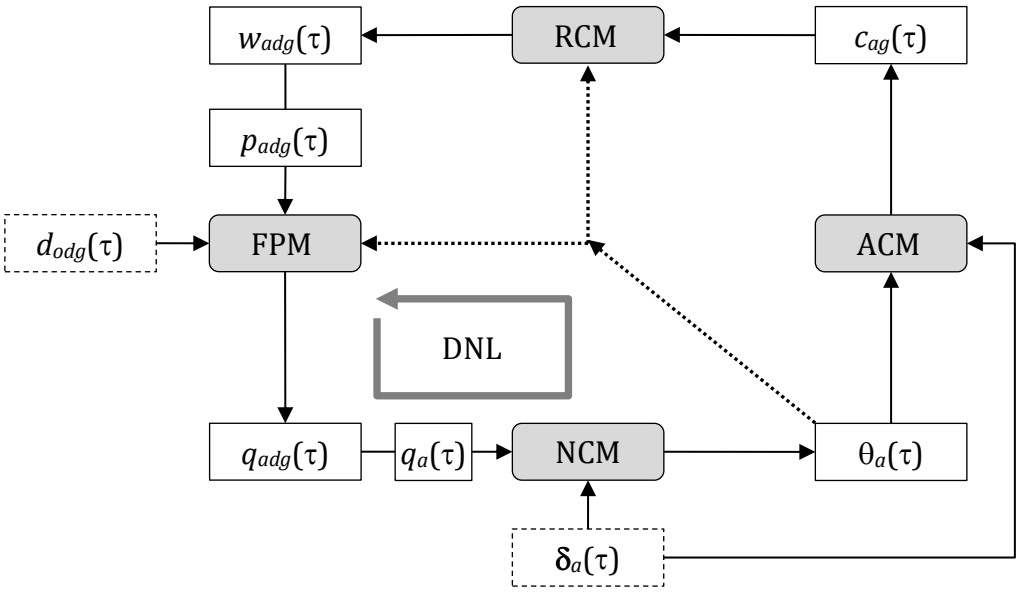

Figure 3. Schema of the DTA model with implicit path enumeration.

Rounded boxes are functionals, while sharp boxes are variables. The dashed boxes indicate external input. The dotted arrows show the crucial role of travel times in DTA, beyond disutility.

The whole outer cycle is the Dynamic User Equilibrium (DUE) problem, while the inner cycle between FPM and NCM is the Dynamic Network Loading (DNL) problem.

NCM – Network Congestion Model. It takes as input the arc volumes, that are typically aggregated from destination specific flows, and the arc characteristic. It yields as output the arc exit times. This sub-model aims at reproducing various traffic phenomena, from hypocritical congestion to queue spillback, which imply a different level of complexity.

ACM – Arc Cost Model. It takes as input the travel times and the arc characteristic. It yields as output the arc costs perceived by each user class, considering their different values of time and tolls.

RCM – Route Choice Model. It takes as input the arc costs, as well as the arc travel times that allow for the dynamic (forward) concatenation of perceived utilities. It yields as output the expected costs (*arc satisfactions*) to reach the destination using each local alternative, that are then used to compute the *arc conditional probabilities*. However, it can be convenient to perform the latter computation directly in the FPM.

FPM – Flow Propagation Model. It takes as input the travel demand and the local choices, as well as the travel times that allow for the dynamic (forward) propagation of flows. It yields as output the arc flows of each class directed towards each destination, that are then aggregated into arc volumes.

Notation of the Dynamic User equilibrium

$p_{agd}(\tau)$	probability that, at time $\tau \in T$, users of class $g \in G$ directed toward destination $d \in Z$ choose to enter arc $a \in A$ conditional on being at its tail node – these are the p_a of the previous section (Splitting Rates)
$d_{odg}(\tau)$	demand flow of class $g \in G$ travelling from origin $o \in Z$ to destination $d \in Z$ and departing at time $\tau \in T$
$\delta_a(\tau)$	characteristic vector of arc $a \in A$ at time $\tau \in T$
$q_{agd}(\tau)$	flow of class $g \in G$ users entering arc $a \in A$ at time $\tau \in T$ directed to $d \in Z$

$q_a(\tau)$	volume entering arc $a \in A$ at time $\tau \in \mathrm{T}$
$\theta_a(\tau)$	exit time of arc $a \in A$ for users entering at time $\tau \in \mathrm{T}$
$c_{ag}(\tau)$	cost of arc $a \in A$ perceived by users of class $g \in G$ entering it at time $\tau \in \mathrm{T}$
$w_{agd}(\tau)$	expected cost perceived by users of class $g \in G$ entering arc $a \in A$ at time $\tau \in \mathrm{T}$ and directed toward destination $d \in Z$ – these are the c_a of the previous section

Arc conditional probabilities reflect the route choice taken at each tail node by users of a certain class directed toward a given destination. For the deterministic case, the global path choices form origins to destinations can be consistently derived as a sequence of such local choices; note that the same is true for the Logit case, while it is false for the Probit case (Gentile and Papola, 2006). Arc probabilities play a crucial role in the proposed model formulation, since they can be exploited in the definition of a sound variational inequality, because their feasibility set is simple and well defined (non-negativity and tail sum equal to one).

From a model perspective, arc probabilities $p_{adg}(\tau)$ are the result of the route choice model; they are derived by the arc satisfactions $w_{adg}(\tau)$ applying a discrete choice model (in our case the deterministic Wardrop model) locally at the tail node. But from an algorithm perspective, they can be conveniently computed together with the flow propagation model; actually, here the new arc probabilities at each iteration are also the result of the Gradient Projection algorithm which modifies the old (current) variables to determine a new search direction. All this can be technically done at once in the FPM.

3.3 MODEL EQUATIONS

The performance pattern of the network is given by the travel time $t_a(\tau)$ of each arc $a \in A$ for users entering it at time τ and by the corresponding travel cost $c_{ag}(\tau)$ for each class $g \in G$. The travel time is obtained from the exit time as follows:

$$t_a(\tau) = \theta_a(\tau) - \tau \,, \tag{68}$$

while the cost is a linear function of the travel time; for example we can assume:

$$c_{ag}(\tau) = \hat{c}_{ag}(\tau) + \beta_g^{vot} \cdot t_a(\tau) \,, \tag{69}$$

where $\hat{c}_{ag}(\tau)$ is the monetary cost of arc $a \in A$ and class $g \in G$ while $\beta_g{}^{vot} \geq 0$ is the class value of time. Note that the proposed supply model is multiclass but not multimodal, because the travel time of an arc is the same for all users and only the cost may differ among classes. The extension to users classes with different vehicle speeds is not trivial as in the static case, as it requires a more complex traffic model.

Due to congestion phenomena, the exit time of the generic arc $a \in A$ at time τ depends in general on the entire flow pattern $\mathbf{q}_A = \{q_a(\tau), \forall a \in A\}$:

$$\theta_a(\tau) = \theta_a(\mathbf{q}_A, \tau) \,. \tag{70}$$

The above Network Congestion Model is crucial for DTA, but is not the focus of the present paper. In particular, we will consider the Network Performance Function (NPF) proposed in Gentile (2015), which is based on the GLTM (Gentile, 2010). However, the proposed framework is valid also for other supply models.

We first introduce the model with explicit path enumeration. This will allow to better highlight the difference with the proposed model with implicit path enumeration.

The generalized cost $c_{kg}(\tau)$ of path $k \in K$ for class $g \in G$ is assumed to be additive, i.e. it is given by the sum of the costs associated with the $A_k \subseteq A$ arcs constituting the path,

each taken at the time $\theta_{ka[-]}(\tau)$ when a user that entered k at time τ reaches the initial node a^- of arc $a \in A_k$ by following k :

$$c_{kg}(\tau) = \sum_{a \in A_k} c_{ag}\left(\theta_{ka[-]}(\tau)\right) . \quad (71)$$

The exit time $\theta_{ki}(\tau)$ of the sub-path of $k \in K$ from its initial node until node $i \in N_k$ (whose arcs are $A_{ki} \subseteq A_k$) for users that entered k at time τ is given by the following concatenation formula:

$$\theta_{ki}(\tau) = \tau + \sum_{a \in A_{ki}} t_a\left(\theta_{ka[-]}(\tau)\right) . \quad (72)$$

The route choice of users is reproduced through a deterministic model where only (dynamic) shortest paths (i.e. with minimum cost) are utilized. Then, the probability $p_{kg}(\tau)$ of choosing the generic path $k \in K_{od}$ at time τ for users of class $g \in G$ travelling between $o \in Z$ and $d \in Z$ shall satisfy, beyond non-negativity $p_{kg}(\tau) \geq 0$ and consistency $\sum_{k \in K_{od}} p_{kg}(\tau) = 1$, the following complementarity condition:

$$\left(c_{kg}(\tau) - w_{odg}(\tau)\right) \cdot p_{kg}(\tau) = 0 , \quad (73)$$

where $w_{idg}(\tau)$ is in general the cost of the shortest path from node $i \in N$ to destination $d \in Z$ for users of class $g \in G$ leaving i at time τ :

$$w_{idg}(\tau) = Min\left(c_{kg}(\tau), \forall k \in K_{id}\right) . \quad (74)$$

Based on (73) if k is used, i.e. $p_{kg}(\tau) > 0$, then it is shortest, i.e. $c_{kg}(\tau) = w_{odg}(\tau)$.

The flow $q_{kg}(\tau)$ of class $g \in G$ users entering the generic path $k \in K_{od}$ at time τ can be simply obtained by multiplying its choice probability $p_{kg}(\tau)$ by the demand flow of class g between $o \in Z$ and $d \in Z$:

$$q_{kg}(\tau) = p_{kg}(\tau) \cdot d_{odg}(\tau) . \quad (75)$$

The flow $q_{adg}(\tau)$ of class $g \in G$ users travelling toward destination $d \in Z$ that enter arc $a \in A$ at time τ is given by:

$$q_{adg}(\tau) = \sum_{o \in Z} \sum_{k \in K_{od} : a \in A_k} q_{kg}\left(\theta^{-1}_{ka[-]}(\tau)\right) \cdot \frac{\partial \theta^{-1}_{ka[-]}(\tau)}{\partial \tau} . \quad (76)$$

Equation (76) comes from the FIFO rule, that applied to a generic network element k states:

$$n^{in}_k\left(\theta^{-1}_k(\tau)\right) = n^{out}_k(\tau) , \quad (77)$$

where $n_k^{in}(\tau)$ and $n_k^{out}(\tau)$ are, respectively, the cumulative inflow and outflow of k at time τ, while $\theta_k^{-1}(\tau)$ yields the entrance time for given exit time τ, that is the inverse of $\theta_k(\tau)$ yielding the exit time for given entrance time τ. Indeed, taking the derivatives of (77) we have:

$$\frac{\partial n^{in}_k\left(\theta^{-1}_k(\tau)\right)}{\partial \theta^{-1}_k(\tau)} \cdot \frac{\partial \theta^{-1}_k(\tau)}{\partial \tau} = \frac{\partial n^{out}_k(\tau)}{\partial \tau} \rightarrow q^{in}_k\left(\theta^{-1}_k(\tau)\right) \cdot \frac{\partial \theta^{-1}_k(\tau)}{\partial \tau} = q^{out}_k(\tau) , \quad (78)$$

where $q_k^{in}(\tau)$ and $q_k^{out}(\tau)$ are, respectively, the inflow and outflow of k at time τ.

The volume $q_a(\tau)$ entering arc $a \in A$ at time τ is given by the sum of all the relative destination and class arc flows multiplied by the class vehicle equivalents $\beta_g^{eqv} \geq 0$:

$$q_a(\tau) = \sum_{d \in Z} \sum_{g \in G} \beta^{eqv}_g \cdot q_{adg}(\tau) . \quad (79)$$

We now introduce the model with implicit path enumeration.

We can define the cost of the shortest path from arc $a \in A$ to destination $d \in Z$ for users of class $g \in G$ leaving its initial node $a^- \in N$ at time τ through the following equation:

$$w_{adg}(\tau) = c_{ag}(\tau) + w_{a^+dg}\left(\theta_a(\tau)\right). \quad (80)$$

On this base we can express the cost of the shortest path from node $i \in N$ as the result of the following local choice among the arcs i^+ of the forward star:

$$w_{idg}(\tau) = Min\left(w_{adg}(\tau), \forall a \in i^+\right). \quad (81)$$

The combination of (80) and (81) yields a recursive problem (a square system of non-linear equations) that can be solved by processing nodes in reversed topological order starting from the destination, as shown later on in this paper.

The probability $p_{adg}(\tau)$ that users of class $g \in G$ travelling toward destination $d \in Z$ enter arc $a \in i^+$ conditional on being in its initial node $i \in N$ at time τ shall satisfy, beyond non-negativity $p_{adg}(\tau) \geq 0$ and consistency $\sum_{a \in i^+} p_{adg}(\tau) = 1$ conditions, the following complementarity condition, that is the local version of (73):

$$\left(w_{adg}(\tau) - w_{idg}(\tau)\right) \cdot p_{adg}(\tau) = 0 \ . \quad (82)$$

Based on (82), if a is used, i.e. $p_{adg}(\tau) > 0$, then it is shortest, i.e. $w_{adg}(\tau) = w_{idg}(\tau)$.

Then, the arc flows are the result of local choices:

$$q_{adg}(\tau) = p_{adg}(\tau) \cdot q_{idg}(\tau) \ . \quad (83)$$

The flow $q_{idg}(\tau)$ of class $g \in G$ users travelling toward destination $d \in Z$ that exit node $i \in N$ at time τ, based on (77), is given by:

$$q_{idg}(\tau) = d_{idg}(\tau) + \sum_{a \in i^-} q_{adg}\left(\theta_a^{-1}(\tau)\right) \cdot \frac{\partial \theta_a^{-1}(\tau)}{\partial \tau} \ . \quad (84)$$

The combination of (83) and (84) yields a recursive problem (a square system of linear equations) that can be solved by processing nodes in topological order starting from origins, as shown later on in this paper.

The probability of the generic path $k \in K_{od}$ resulting from the sequential model is:

$$p_{kg}(\tau) = \prod_{a \in A_k} p_{adg}\left(\theta_{ka[-]}(\tau)\right) \ . \quad (85)$$

3.4 VARIATIONAL INEQUALITY FORMULATIONS OF DUE

The DUE problem can be formulated through the following Variational Inequality in terms of path probabilities:

$$\sum_{k \in K} \sum_{g \in G} \int_{\tau \in T} c_{kg}\left(\mathbf{p}^*_{KGT}, \tau\right) \cdot \left(p^*_{kg}(\tau) - p_{kg}(\tau)\right) \cdot d\tau \leq 0 \quad , \forall \mathbf{p}_{KGT} \in S_p^{KGT}$$

$$S_p^{KGT} = \left\{ \begin{array}{l} \mathbf{p}_{KGT} \in \Re^{KGT} : p_{kg}(\tau) \geq 0 \ , \forall k \in K, \forall g \in G \ ; \\ \sum_{k \in K_{od}} p_{kg}(\tau) = 1 \ , \forall od \in Z \times Z, \forall g \in G \end{array} \right\} \quad , \quad (86)$$

where, based on (75),(76),(79),(70),(68),(69), from (71)-(72) we have that each path cost $c_{kg}(\tau)$ is a function of the path choice pattern $\mathbf{p}_{KGT}$:

$$c_{kg}(\tau) = c_{kg}\left(\mathbf{p}_{KGT}, \tau\right) \ . \quad (87)$$

Note that (87) includes the solution of DNL, because (76) requires the exit times which are produced by (70). This is essentially the VI formulation proposed by Friesz et al. (1993), but casted in the space of path probabilities instead of path flows. Although here the departure time choice is not included, it is possible to do so as in Bellei et al. (2006).

As an alternative, we propose here a formulation of DUE based on local choices at each node $i \in N$ among its forward star made by users of class $g \in G$ directed toward each

destination $d \in Z$; the resulting Variational Inequality is in terms of arc conditional probabilities:

$$\sum_{d\in Z}\sum_{g\in G}\sum_{i\in N-d}\sum_{a\in i^+}\int_{\tau\in \mathrm{T}} w_{adg}\left(\mathbf{p}^*_{ADG\mathrm{T}},\tau\right)\cdot\left(p^*_{adg}(\tau)-p_{adg}(\tau)\right)\cdot d\tau \le 0 \quad ,\forall \mathbf{p}_{ADG\mathrm{T}} \in S_p^{ADG\mathrm{T}}$$

$$S_p^{ADG\mathrm{T}} = \left\{ \begin{array}{l} \mathbf{p}_{ADG\mathrm{T}} \in \Re^{ADGT} : p_{adg}(\tau) \ge 0 \,, \forall a \in A \,, \forall d \in Z, \forall g \in G \,; \\ \sum_{a\in i^+} p_{adg}(\tau) = 1 \,, \forall i \in N-d \,, \forall d \in Z, \forall g \in G \end{array} \right\} \quad , \tag{88}$$

where, based on (83)-(84), (79),(70),(68),(69), from (81)-(80) we have that each arc satisfaction $w_{adg}(\tau)$ is a function of the arc probability pattern $\mathbf{p}_{ADG\mathrm{T}}$:

$$w_{adg}(\tau) = w_{adg}\left(\mathbf{p}_{AGD\mathrm{T}},\tau\right) . \tag{89}$$

Note that also (89) includes the solution of DNL, because (84) requires the exit times which are produced by (70). The exit times could be, for example, obtained from the conditional probabilities through the Cell Transmission Model or Link Transmission Model (although here we did a different choice considering the NPF).

The gap function of the VI (88) can be conveniently written, like in (9), as:

$$\gamma\left(\mathbf{p}^*_{ADG\mathrm{T}}\right) = 1 - \frac{\sum_{d\in Z}\sum_{g\in G}\sum_{i\in N-\{d\}}\int_{\tau\in \mathrm{T}} w_{idg}\left(\mathbf{p}^*_{ADG\mathrm{T}},\tau\right)\cdot d\tau}{\sum_{d\in Z}\sum_{g\in G}\sum_{i\in N-\{d\}}\sum_{a\in i^+}\int_{\tau\in \mathrm{T}} w_{adg}\left(\mathbf{p}^*_{ADG\mathrm{T}},\tau\right)\cdot p^*_{adg}(\tau)\cdot d\tau} \; ; \tag{90}$$

if the average cost of local choices at the current solution (denominator) is equal to the minimum cost of the node, then the gap function is null, no local choice can be improved, and we hence have equilibrium.

In the following we prove that if $\mathbf{p}^*_{ADG\mathrm{T}}$ is a solution of VI (88) for implicit path enumeration, i.e. an equilibrium at the node local level, then the $\mathbf{p}^*_{KG\mathrm{T}}$ obtained through (85) is a solution of VI (86) for explicit path enumeration, i.e. an equilibrium at the path global level. Therefore, if the proposed heuristic has success and finds a solution of the arc-based equilibrium, then also a solution of the path-based equilibrium is found.

Theorem 1. Equivalence of local and global equilibrium for DUE.

Based on (6) we can equivalently prove the following for each $d \in Z$:

$$\begin{array}{l} \left(w_{adg}(\tau) - w_{a^-dg}(\tau)\right)\cdot p^*_{adg}(\tau) = 0, \quad \forall a \in A \Rightarrow \\ \left(c_{kg}(\tau) - w_{odg}(\tau)\right)\cdot p^*_{kg}(\tau) = 0 \quad \forall k \in K_{od}, \forall o \in Z \end{array} \tag{91}$$

Proof.

Consider the generic path $k \in K_{od}$ and let:

$$p^*_{kg}(\tau) = \prod_{a\in A_k} p^*_{adg}\left(\theta_{ka[-]}(\tau)\right) . \tag{92}$$

If $p_{kg}{}^*(\tau) = 0$, then the right side of (91) is satisfied. Let's then consider the case where $p_{kg}{}^*(\tau) > 0$. Based on (92), it is:

$$p^*_{adg}\left(\theta_{ka[-]}(\tau)\right) > 0, \quad \forall a \in A_k \; . \tag{93}$$

Then, from (80), based on the left side of (91) we have:

$$c_{ag}(\tau) = w_{a^-dg}(\tau) - w_{a^+dg}\left(\theta_a(\tau)\right), \quad \forall a \in A_k . \tag{94}$$

Finally, (71) based on (94), becomes:

$$c_{kg}(\tau) = \sum_{a\in A_k} w_{a^-dg}\left(\theta_{ka[-]}(\tau)\right) - w_{a^+dg}\left(\theta_{ka[+]}(\tau)\right) = w_{odg}(\tau) . \tag{95}$$

Thus, also in this case, the right side of (91) is satisfied. ■

4. APPLYING GP ALGORITHMS TO LOCAL CHOICES

In the following we present a class of Gradient Projection methods for solving the formulation of Dynamic Traffic Assignment with implicit path enumeration.

The simulation period T is discretized into η subsequent time intervals separated by an ordered list T of instants with indices from 0 to η, whose generic clock time is $\tau_t \in \Re$, $t \in T$.

Besides the transposition into vectors of the temporal profiles introduced in section 3, which is self-explanatory (just note that flows and probabilities are referred to intervals while times and costs to instants), some additional notation is required:

- $h_t = (\tau_{t+1} - \tau_t)$ is the duration of interval $t \in T$; $\tau_{\eta+1} = \infty$;
- $e_{at} \in T \geq t$ is the interval which includes the exit time $\tau_t + t_{at}$ of $a \in A$ for users who enter at τ_t ;
- m_{ate} is the share of inflow during interval $t \in T$ that exits $a \in A$ during interval $e \in T \geq t$ – clearly, it is: $m_{ate} = 0$ for $e < e_{at}$ or $e > e_{a\,t+1}$;
- $\omega_{it} \in \Im$ is the reverse topological order of node $i \in N$ at instant $t \in T$;
- $i_{\omega t} \in N$ is the ω-th node in reverse topological order.

We now present the iterative equilibrium procedure which implements the schema of Figure 3 and solves the Variational Inequality formulation (88). The stop criterion is based on the gap function (90) and the search direction is based on GP.

```
* Dynamic User Equilibrium
function DUE
   q_ADGT ← 0_ADGT                                  * reset destination flows
   n ← 0 , n_bad ← 0; γ_num ← 1 , γ_den ← 1         * initialization
   do                                               * main assignment cycle

      call NCM                                      * Network Congestion Model
      c_agt ← ĉ_ag + β_g^vot · t_at , ∀a∈A , ∀g∈G , ∀t∈T   * Arc Cost Model

      γ_n = γ_num / γ_den                           * compute gap function
      if n ≥ n_max or γ_n < γ_min then exit loop    * stop criterion
      n ← n + 1 ; γ_num ← 0 , γ_den ← 0             * start new iteration
      if n > 2 and γ_n-1 ≥ η3 · γ_n-2 then n_bad ← n_bad + 1   * update bad iterations
      α_n ← (η1/(η1+n_bad))^η2                      * update the gradient step

      for each d∈Z                                  * for each destination
         for each g∈G                               * for each class
            call DSP(d∈Z, g∈G)                      * Dynamic Shortest Paths
            call FPM(d∈Z, g∈G)                      * Flow Propagation Model
         next g
      next d

   loop
end function
```

The great advantage of the VI formulation (88) based on arc conditional probabilities $p_{adg}(\tau)$ is that the feasible set is a simple polytope, thus leading to the possibility of applying the gradient projection approach as a solution algorithm.

The disadvantage is that its gradient function requires the solution of a DNL. In the proposed solution algorithm in each iteration of the DUE we perform just one iteration of fixed point problem that solves the DNL, starting from the current exit times of the previous iteration.

If we look at the fixed point schema of Figure 3 we can see how the arc flows for each destination $q_{adg}(\tau)$ would represent a more convenient iterate, because this variable is internal to both DUE and DNL . However, the feasible set of arc flows is very complex as it includes the result of the DNL for each feasible arc probability pattern.

On these basis, the proposed algorithm can be considered at the same time a method for the solution of: a fixed point problem based on destination arc flows, or a VI problem based on arc conditional probabilities, where the DNL is solved approximatively with one iteration.

The Network Congestion Model is one of the main components of DTA, but it is not the focus of this paper. For completeness we present here the simple example of a space-separable whole link model with final bottleneck (no interaction at nodes nor spillback) based on Average Kinematic Waves (Gentile et al, 2005). We used here the following notation for the characteristics of the generic arc $a \in A$: $s_a > 0$ is the free flow speed; l_a is the length; κ_a is the capacity; g_a is the green share.

** Network Congestion Model – an example based on Average Kinematic Waves*
function *NCM*
for each $a \in A$

$q_{at} \leftarrow \sum_{d \in Z} \sum_{g \in G} \beta_g^{eqve} \cdot q_{adgt}$, $\forall t \in T$ ** arc volumes*

** Average Kinematic Waves*
$t_{a0} \leftarrow l_a / s_a$
$\rho \leftarrow s_a$
for $t = 1$ **to** η
$s \leftarrow s(q_{at}) = 0.5 \cdot s_a \cdot (1 + (1 - q_{at} / \kappa_a)^{0.5})$ ** speed from the fundamental diagram*
$t_{at} \leftarrow l_a / s$
$\mu \leftarrow \rho + s - s_a$ ** shockwave speed* (valid for parabolic diagrams)
if $(h_{t-1} + t_{at}) \cdot \mu < l_a$ **then** $t_{at} \leftarrow l_a / \rho + h_{t-1} \cdot \mu / (s - \mu) \cdot (1 - s / \rho)$
$\rho \leftarrow l_a / t_{at}$
$t_{at} \leftarrow Max(t_{a\,t-1} + h_{t-1} \cdot (q_{at} / (\kappa_a \cdot g_a) - 1) , t_{at})$ ** bottleneck model*
next t

** Arc Propagation Map* (valid for constant exit capacity. i.e. no spillback)
$e = 0$
do until $\tau_{e+1} > \tau_0 + t_{a0}$: $e = e + 1$ **loop**
$e_{a0} = e$
$\mathbf{m}_{aTT} \leftarrow \mathbf{0}_{aTT}$
for $t = 0$ **to** η-1
do until $\tau_{e+1} > \tau_t + t_{at}$: $e = e + 1$ **loop**
$e_{a\,t+1} = e$

```
        h ← h_t + t_{a t+1} - t_at
        if h > 0 then
            for e = e_at to e_{a t+1}
                if e = e_at then t_0 ← τ_t + t_at else t_0 = τ_e
                if e = e_{a t+1} then t_1 ← τ_{t+1} + t_{a t+1} else t_1 = τ_{e+1}
                m_ate ← (t_1 − t_0) / h
            next e
        else
            m_ate ← 1
        end if
    next t
    m_{aηη} ← 1

  next a
end function
```

The introduction of the arc propagation map (Gentile, 2015) highly improves the quality of the solution, especially in case of time varying exit capacity (not considered above).

The computation of Dynamic Shortest Trees for each destination (or origin) is the heart of any model for deterministic route choices. Here we adopt the Temporal Layer approach proposed by Chabini (1998) adapted to continuous cost functions and long time intervals as in Gentile (2016), which requires to handle a list L of nodes, to be visited in reverse topological order, inside each temporal layer, visited in reverse chronological order.

```
* Dynamic Shortest Paths
function DSP(d∈Z, g∈G)
  for t = η to 0 step -1                          * in reverse chronological order
    w_it ← ∞ , ∀i∈N-d ; w_dt ← 0                  * initialize node labels
    L ← d                                         * initialization of nodes list
    ω ← 0 ; ω_it ← ∞ , i_ωt ← 0 , ∀i∈N            * init of topological order
    do until L = ∅                                * until the node list is empty
      j ← ArgMin(w_it , ∀i∈L) ; L ← L − j          * extract the node with least label
      ω ← ω + 1 ; ω_jt ← ω ; i_ωt ← j              * increase topological order
      for each a∈j⁻                               * for each arc of the backward star
        i ← a⁻ ; e ← e_at                         * set tail and exit interval
        if ω_it = ∞ then                          * to avoid absorbing cycles
          w_at ← c_agt + w_je                     * interpolation of minimum cost
          if e < η then w_at ← w_at + (τ_t + t_at − τ_e) · (w_{j e+1} − w_je) / h_e   * ... follows
          if w_it > w_at then                     * Bellman check
            w_it ← w_at                           * improve the label
            if i∉L then L ← L + i                 * insert the node in the list
          end if
        end if
      next a
    loop
    i_0t ← ω                                      * rem the first index in the top ord
```

next t
end function

The Flow Propagation Model proposed in this paper introduces a relevant novelty. Usually, the network loading is performed on an acyclic sub-graph, for example that of efficient arcs (i.e. arcs that bring the user closer to the destination with respect to some topological order). Here, we allow for (only temporary) solutions that may include cycles, solving the Flow Propagation Model as a sequence of Square Linear Systems (SLS), one for each temporal layer.

Each equation represents the flow conservation at a node during the current interval and the unknowns are the flows exiting from each node during the same interval. The matrix of each system depends on the search direction in terms of arc conditional probabilities and on the arc propagation map (i.e. on the travel times). It is in general rather sparse and almost triangular (if nodes are in topological order). It is the identity matrix if all the flow entering one arc in a given interval exits in later intervals.

Moreover, the matrix is diagonally dominant (i.e., for every column, the magnitude of the diagonal entry is larger than the sum of the magnitudes of all the other entries) and then non-singular, because the sum of arc conditional probabilities of each node forward star in the search direction is one, while the elements of the arc propagation map are smaller than one. The Jacobi and Gauss–Seidel methods for solving the linear system converge. The latter can then be usually solved (depending on the congestion level) to nearly double (10^{-16}) precision (the highest possible on standard computers) through a few (say, less than 10) iterations of an iterative method, such as the BICGSTAB (Van der Vorst, 1992).

Preconditioning through the solution of a simplified problem, as suggested in Saad (2003), provides a remarkable speed-up. For instance, we can triangularize the matrix. In our case, we can load on the network the vector to be preconditioned as if it was a demand flow, by following the topological order resulting from dynamic shortest paths, as in the classical case of efficient arcs.

The iterative solution of the SLS is initialized by loading the demand on efficient arcs.

** Flow Propagation Model*
function *FPM*($d \in Z, g \in G$)
$q_{it} \leftarrow 0$, $\forall i \in N$, $\forall t \in T$ ** reset node flows*
for $t = 0$ **to** η ** in direct chronological order*
$q_{ot} \leftarrow q_{ot} + d_{odgt}$, $\forall o \in Z$ ** load travel demand on nodes*
$q_i^{lin} \leftarrow q_{it}$, $\forall i \in N$ ** set the SLS constants*
$p_a^{dir} \leftarrow 0$, $p_a^{lin} \leftarrow 0$, $\forall a \in A$ ** initialize the search direction*
for $\omega = i_{0t}$ **to** 2 **step** -1 ** in topological order*
$i \leftarrow i_{\omega t}$ ** for each node*
call *GPA*($d \in Z, g \in G, t \in T, i \in N$) ** find the search direction p_a^{dir}*
$p_a^{lin} = m_{att} \cdot p_a^{dir}$, $\forall a \in i^+$ ** compute the SLS coefficients*
$q_{a[+]\,t} \leftarrow q_{a[+]\,t} + q_{it} \cdot p_a^{lin}$, $\forall a \in i^+$ ** propagate node flows in this interval*
next ω
$q_{it} - \sum_{a \in i[-]} q_{a[-]\,t} \cdot p_a^{lin} = q_i^{lin}$, $\forall i \in N$ ** solve for $q_{\cdot t}$ this Square Linear System*
$q_{a[+]\,e} \leftarrow q_{a[+]\,e} + q_{it} \cdot p_a^{dir} \cdot m_{ate} \cdot h_t / h_e$, $\forall e > t: m_{ate} > 0$, $\forall a \in i^+$ ** prop. to future int.*
$q_{adgt} \leftarrow q_{a[-]\,t} \cdot p_a^{dir}$, $\forall a \in A$ ** update arc inflows by class and dest.*
next t
end function

The following procedure implements the proposed Gradient Projection algorithms for the local route choice of class $g \in G$ users directed toward destination $d \in Z$ that during interval $t \in T$ pass through node $i \in N$.

The procedure computes the gap function numerator and denominator as well.

** search direction with Gradient Projection Algorithms*
function *GPA*($d \in Z, g \in G, t \in T, i \in N$)
  if $t = \eta$ **then** $r \leftarrow \eta$ **else** $r \leftarrow t+1$ ** r is the cost reference instant for current interval*
  $p_a^{dir} \leftarrow 0$, $\forall a \in i^+$
  $B \leftarrow \{a \in i^+ : \omega_{a[+]t} < \infty\}$
  $q_{tot} \leftarrow \sum_{a \in B} q_{adgt}$
  $a^* \leftarrow ArgMin(w_{ar}$, $\forall a \in B)$
  $a^\times \leftarrow ArgMax(w_{ar}$, $\forall a \in B: q_{adgt} > 0)$
  $w_{min} \leftarrow w_{a^* r}$
  $w_{max} \leftarrow w_{a^\times r}$
  $w_{med} \leftarrow \sum_{a \in B} q_{adgt} \cdot w_{ar}$
  if $q_{tot} = 0$ **then**
    $B \leftarrow a^*$
  else
    $\gamma_{num} \leftarrow \gamma_{num} + \sum_{a \in B} (w_{ar} - w_{min}) \cdot q_{adgt} \cdot h_t$
    $\gamma_{den} \leftarrow \gamma_{den} + w_{med} \cdot h_t$
  end if
  if $|B| = 1$ **then**
    $a \leftarrow B$
    $p_a^{dir} \leftarrow 1$
  else if $w_{max} = w_{min}$ **then**
    $p_a^{dir} \leftarrow q_{adgt} / q_{tot}$, $\forall a \in B$
  else if *GP* = *MSA* **then**
    $p_a^{dir} \leftarrow \alpha_n \cdot Bool(a = a^*) + (1\text{-}\alpha_n) \cdot q_{adgt} / q_{tot}$, $\forall a \in B$
  else

    ** different gradient scaling*
    if *GS* = *wnod* **then**
      $g_a \leftarrow w_{min}$
    else if *GS* = *warc* **then**
      $g_a \leftarrow w_{ar}$
    else if *GS* = *wavg* **then**
      $g_a \leftarrow w_{med} / q_{tot}$
    end if
    $g_a \leftarrow g_a / (\rho \cdot \alpha_n)$
    $\Delta p_a \leftarrow 0$, $\forall a \in i^+$

    ** different gradient projection*
    if *GP* = *EGP* **then**
      $w_{sup} \leftarrow w_{min} + g_{a^*} \cdot (1 - p_{a^*})$
      $w_{sum} \leftarrow 0$
      $p_{sum} \leftarrow 0$

$g_{den} \leftarrow 0$
for each $a \in B$
if $w_{ar} - g_a \cdot p_a < w_{sup}$ **then**
$w_{sum} \leftarrow w_{sum} + w_{ar} / g_a$
$g_{den} \leftarrow g_{den} + 1 / g_a$
else
$B \leftarrow B - a$
$\Delta p_a \leftarrow - q_{adgt} / q_{tot}$
$p_{sum} \leftarrow p_{sum} + q_{adgt} / q_{tot}$
end if
next a
do
$EndLoop \leftarrow TRUE$
$w_{avg} \leftarrow (w_{sum} + p_{sum}) / g_{den}$
for each $a \in B$
$\Delta p_a \leftarrow (w_{avg} - w_{ar}) / g_a$
if $\Delta p_a < 0$ **and** $\Delta p_a + q_{adgt} / q_{tot} < 0$ **then**
$EndLoop \leftarrow FALSE$
$B \leftarrow B - a$
$\Delta p_a \leftarrow - q_{adgt} / q_{tot}$
$w_{sum} \leftarrow w_{sum} - w_{ar} / g_a$
$p_{sum} \leftarrow p_{sum} + q_{adgt} / q_{tot}$
$g_{den} \leftarrow g_{den} - 1 / g_a$
end if
next a
loop until $EndLoop$

else if $GP = QGP$ **then**
$w_{sum} \leftarrow \sum_{a \in B} w_{ar} / g_a$
$g_{den} \leftarrow \sum_{a \in B} 1 / g_a$
do
$EndLoop \leftarrow TRUE$
$w_{avg} \leftarrow w_{sum} / g_{den}$
for each $a \in B$
$\Delta p_a \leftarrow (w_{avg} - w_{ar}) / g_a$
if $\Delta p_a < 0$ **and** $q_{adgt} / q_{tot} < 0$ **then**
$EndLoop \leftarrow FALSE$
$B \leftarrow B - a$
$\Delta p_a \leftarrow 0$
$w_{sum} \leftarrow w_{sum} - w_{ar} / g_a$
$g_{den} \leftarrow g_{den} - 1 / g_a$
end if
next a
loop until $EndLoop$
$\beta \leftarrow Min(1, - q_{adgt} / q_{tot} / \Delta p_a , \forall a \in B: \Delta p_a < 0)$
$\Delta p_a \leftarrow \Delta p_a \cdot \beta, \forall a \in B$

else if $GP = RGP$ **then**

```
        for each a∈B
            if w_ar > w_min and q_adgt / q_tot > 0 then
                Δp_a ← 0.5 · (w_min − w_ar) / g_a
                if Δp_a < − q_adgt / q_tot then Δp_a ← − q_adgt / q_tot
                Δp_a* ← Δp_a* − Δp_a
            end if
        next a

    else if GP = MMP then
        a ← a^×
        if q_adgt / q_tot > 0 then
            Δp_a ← 0.5 · (w_min − w_max) / g_a
            if Δp_a < − q_adgt / q_tot then Δp_a ← − q_adgt / q_tot
            Δp_a* ← Δp_a
        end if

    end if
    p_a^dir ← q_adgt / q_tot + Δp_a , ∀a∈i+
  end if
end function
```

5. NUMERICAL EXPERIMENTS

We present here some experiments of the proposed Gradient Projection algorithms on test networks, comparing them also with MSA. The convergence pattern of the gap function is analysed for different time discretizations, with time intervals h_t of 6, 60 and 600 sec. Different levels of demand wrt (storage and bottleneck) capacity are also considered, so that three different kinds of congestion are experimented, namely: hypocritical, queuing and spillback.

The various versions of GP (EGP, QGP, RGP, MMP) have been tested in the above different situations. However, it was not possible to clearly state the superiority of one method over the others: performance patterns are in most cases very similar and the differences seem to be related with the specific instance of the problem. We reached a similar conclusion when testing several definitions of the scaling factor, specifically that of Equation (33) and (34) wrt to current and initial conditions.

The first battery of tests is performed on the simple dipole network of Figure 4, where it is possible to have expectations on the solution. All links share the following characteristics: free flow speed of 90 km/h, link capacity of 1800 veh/h, jam density of 150 veh/km, jam wave speed of 30 km/h, parabolic hypocritical branch of the fundamental diagram, linear hypercritical branch. All links have a base length equal length of 1 km. Travel demand is constant for 40 min with entry: d_{14} = 1500 veh/h.

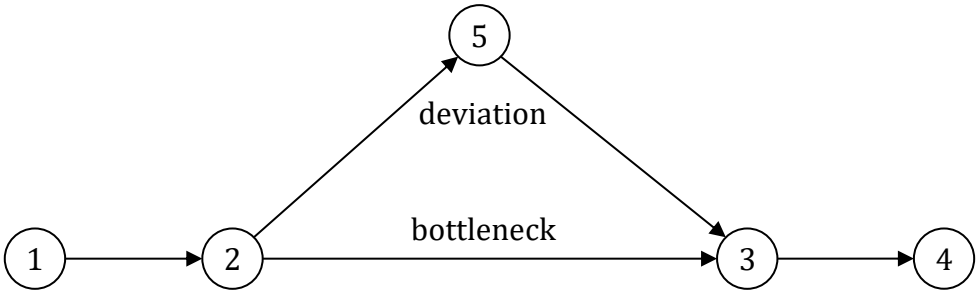

Figure 4. Topology of the dipole network.

In the dipole network there is one diversion with only two alternatives. In this case the 4 GP algorithms essentially coincide.

A glance to the temporal profiles of diversion flows and travel times is presented in the following, for the different levels of congestion. These results, obtained through EGP with the gradient scaling of Equation (34) for the case of one minute intervals, show how the proposed method is capable of obtaining the expected solution.

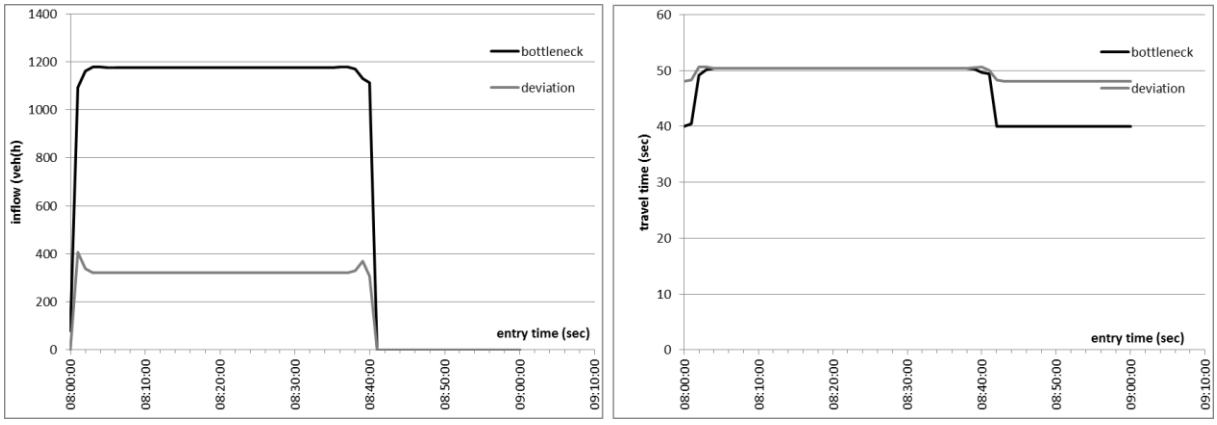

Figure 5. Dipole network with hypocritical congestion. The links of the deviation 2-5 and 5-3 have length of 600 m each. The bottleneck 2-3 has exit capacity of 1200 veh/h.

Figure 5 shows how in the hypocritical case the shorter bottleneck receives more flow than the (slightly longer) deviation so that the two travel times are equal. This is very similar to what happens in the static case.

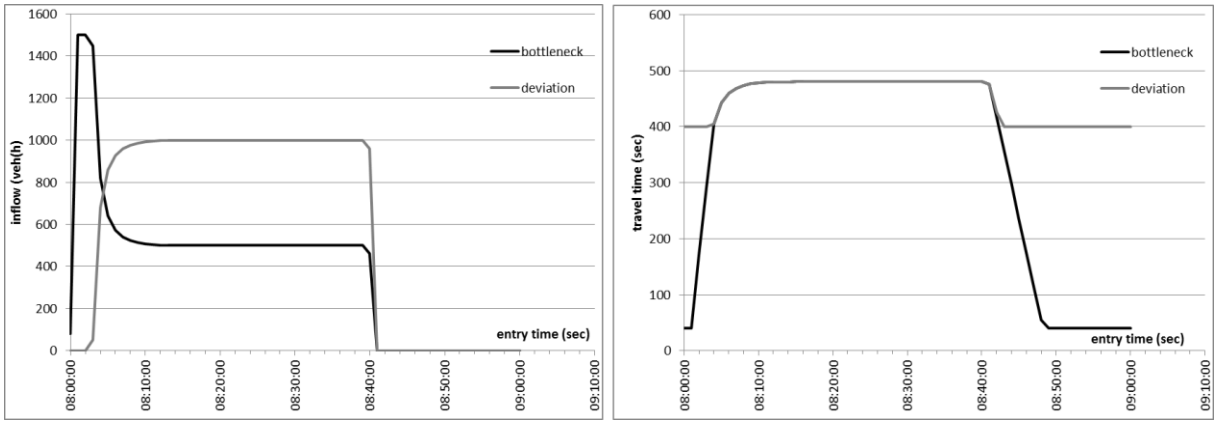

Figure 6. Dipole network with queue. The links of the deviation 2-5 and 5-3 have length of 5 km each. The bottleneck 2-3 has an exit capacity of 500 veh/h.

Figure 6 shows how the shorter bottleneck receives initially the whole demand flow, which is higher than the exit capacity, until the queue gets so big that the (much longer) deviation becomes convenient. From that point on, the share of demand entering into the bottleneck decreases until it receives exactly a flow equal to its exit capacity, so as to maintain a stable queue with a constant travel time.

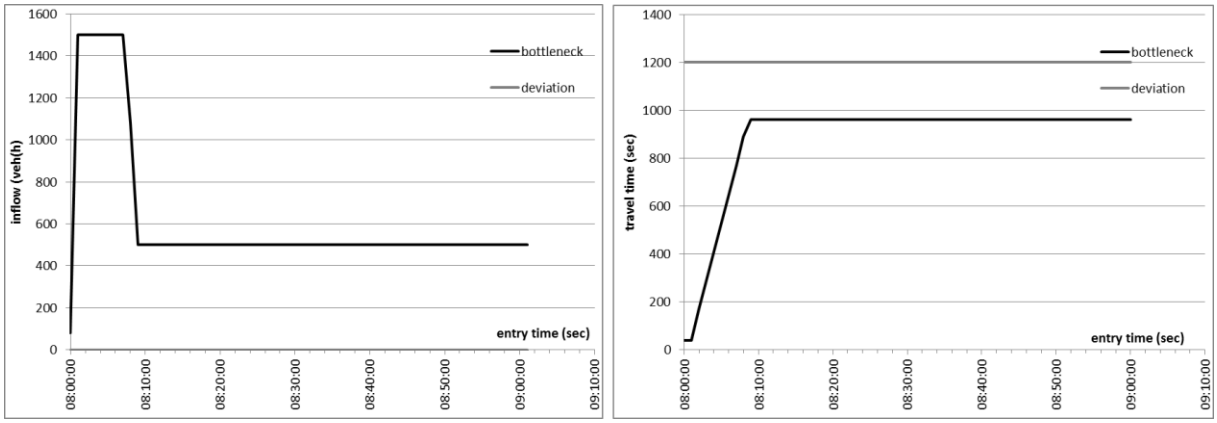

Figure 7. Dipole network in spillback. The links of the deviation 2-5 and 5-3 have length of 15 km each. The bottleneck 2-3 has an exit capacity of 500 veh/h.

Figure 7 shows how the shorter bottleneck receives the whole demand flow during the entire simulation, because the queue spills back to the incoming link before its delay makes the (very long) deviation convenient. At that point the travel time along the bottleneck becomes constant while the inflow drops to the level of the exit capacity, although the link capacity is higher. The queue develops further on the first link. Note that it would be convenient in terms of total network cost if some users is directed to the deviation as this allows a better usage of the existing capacity with a global saving of time. Moreover, in a more complex network this would avoid the impediment of other turns at the node 2 for users who are not directed to node 3. But clearly in a descriptive equilibrium user are selfish consumers, and once arrived at node 2 they chose for the shorter time of the bottleneck. Thus, when spillback occurs, the DTA becomes a DNL.

Interestingly, the diminishing step size is not required by this simple network, as we always found a proper value of the base gradient multiple (ρ = 5, 2, 2, respectively, in the hypocritical, queue and spillback cases) for which the best convergence is achieved assuming $\alpha_n = 1$ in all iterations.

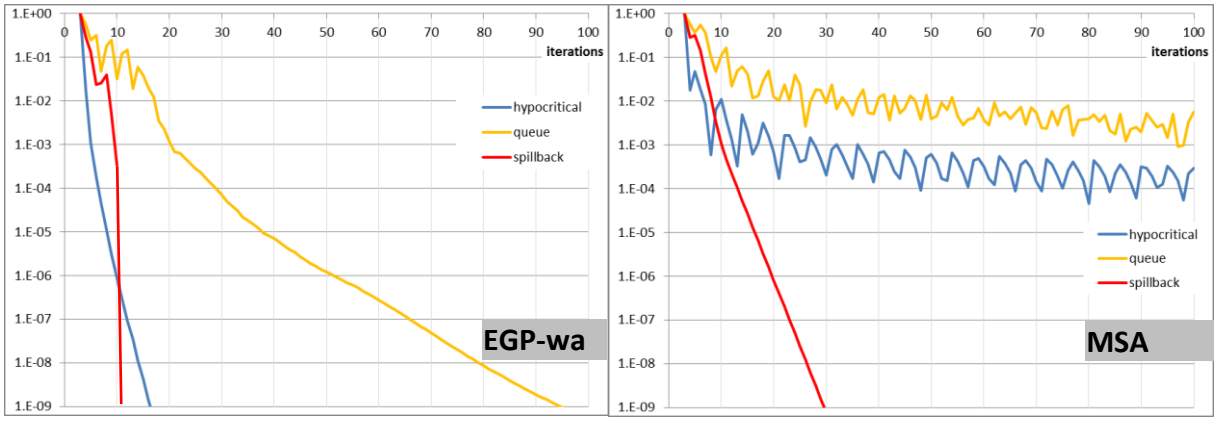

Figure 8. Convergence pattern for the dipole network – EGP outperforms MSA.

Figure 8 show that, as expected, the gradient projection algorithm outperforms MSA for all the possible kinds of congestion: hypocritical, queue and spillback.

The Dial network (Dial, 2006) has been considered for testing how the proposed algorithms scale in the case where several O-D couples (from 4 origins to 4 destinations) interact on the same links (100 arcs). This highly congested network was specifically conceived to involve relevant changes of the shortest trees from the free flow condition to the equilibrium condition, with links that are used in the opposite direction in the two cases; it can thus be considered a difficult problem, despite the small size.

One hour of simulation has been executed, where for the first half an hour a travel demand given by the original O-D matrix (with all 500 veh/h entries) is loaded on the network. Note that the static outlet of this network produces huge queues in the dynamic simulation. Moreover, the capacity of the links is here halved at the final section, to reproduce bottlenecks (e.g. traffic signals).

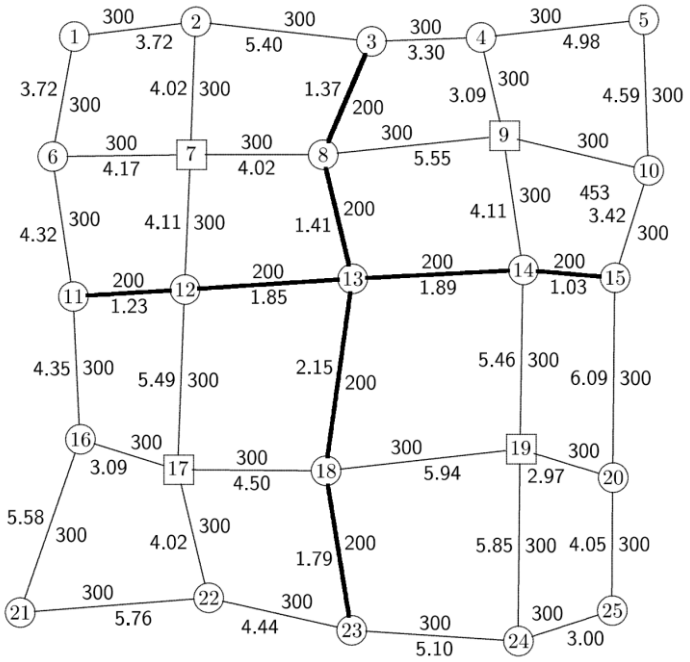

Figure 9. Topology of the Dial network and symmetric link characteristics – Capacities are 200 veh/h along the faster central corridors (thick black line), and 300 veh/h on the other slower links; the other numbers are free flow travel times.

The jam density of links is assumed equal to 1/10 of the capacity, the free flow speed is assumed equal to 60 km/h and the jam wave speed to 25 km/h.

Three types of congestion are considered, corresponding to different demand levels: moderate queues (mded = 0.2 → cong = 0.2); heavy queues (mdem = 0.5 → cong = 1.5); heavy spillback (mdem = 1.0 → cong = 4.5). Here, *mdem* is the demand multiplier, and *cong* is the relative cost of congestion wrt free flow travel times for equilibrium flows (for real cities *cong* is typically is the range 0.2-1.0, see for example the Traffic Index data at www.tomtom.com).

Below we present the sensitivity of the algorithms wrt time discretization, which shows that convergence is slower if we have to adjust decisions (probabilities) at many intervals.

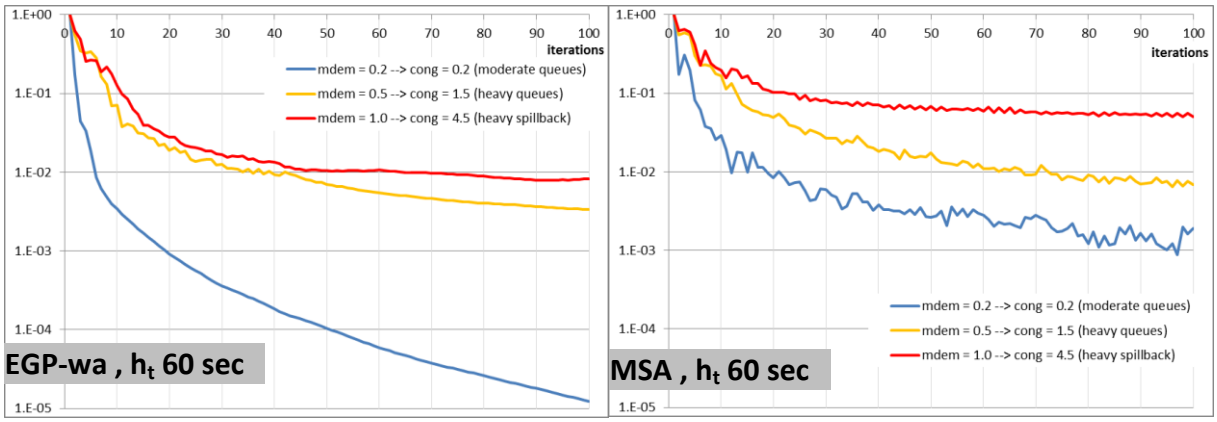

Figure 10. Convergence pattern for Dial network – EGP outperforms MSA.

Figure 10 shows how gradient projection outperforms MSA for all demand levels, both in the long term (after 100 iterations) and in the short term (after 20 iterations). Moreover convergence appears to be smoother for EPG than for MSA. Clearly, the higher the demand, the higher is the congestion and its non-separability (in time and space), the more difficult is the problem, and thus the convergence is slower.

However, as already experimented in Yang and Jayakrishnan (2012), the relative improvement of gradient projection to MSA can be modest in heavy congestion and on larger networks. To improve the efficiency of the method they used a Gauss-Seidel approach with early recalculation of costs. This approach can be possibly applied also to our method and will be the object of future research.

There are not many papers aimed at finding methods with a better convergence than MSA in the DTA literature. We provide in the follwing a qualitative comparison with the method proposed by Yang and Jayakrishnan (2012). In the case of moderate congestion, in 100 iterations our methods converged to a relative gap of 10^{-4}, while their method reached $0.5 \cdot 10^{-2}$. For heavier congestion our method reaches the above minimal target (10^{-2}), while their method could not really converge (as reported by the authors).

Clearly, convergence with heavy congestion is an issue; but this is true also for many static assignment algorithms. In DTA we face the additional complexity of a highly non-separable supply model.

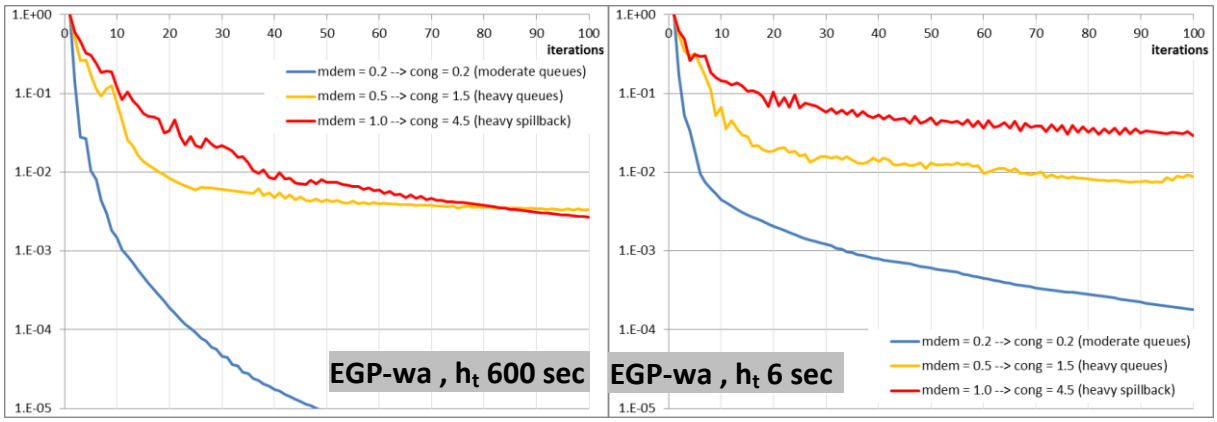

Figure 11. Convergence pattern for Dial network – EPG works well with different h_t .

Figure 11 shows how EGP supports well different time discretization, with intervals from 6 to 600 seconds. Clearly, the more intervals the more difficult is the problem, and thus the convergence is slower.

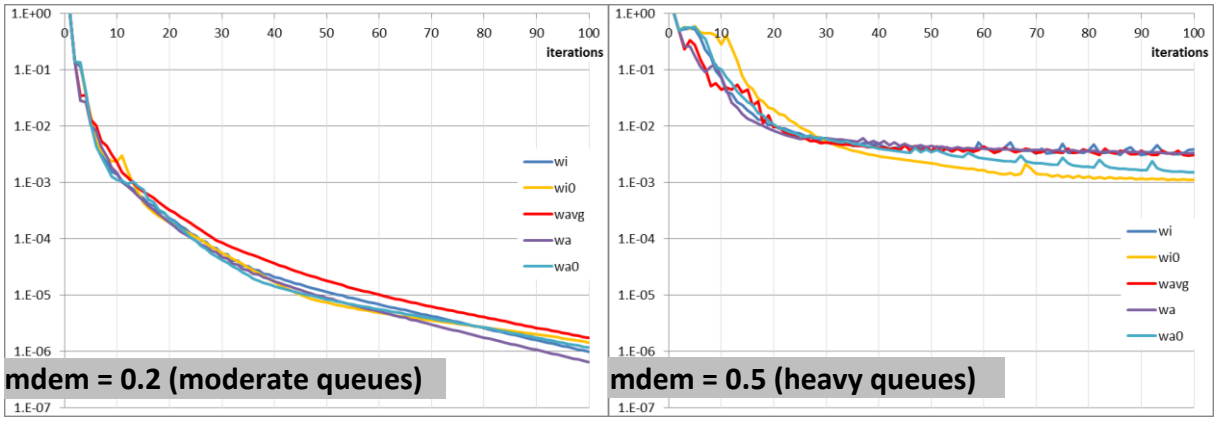

Figure 12. Convergence pattern for Dial network – EPG performing similarly with different scaling factors.

Figure 12 shows how EGP performs similarly with any of the scaling factors tested: wi (current node min cost), wi0 (initial node min cost), wavg (current average node cost), wa (current arc satisfaction), wa0 (initial arc satisfaction). Tests are executed for h_t of 600 sec, and for two congestion levels: moderate (mded = 0.2) and heavy (mdem = 0.5).

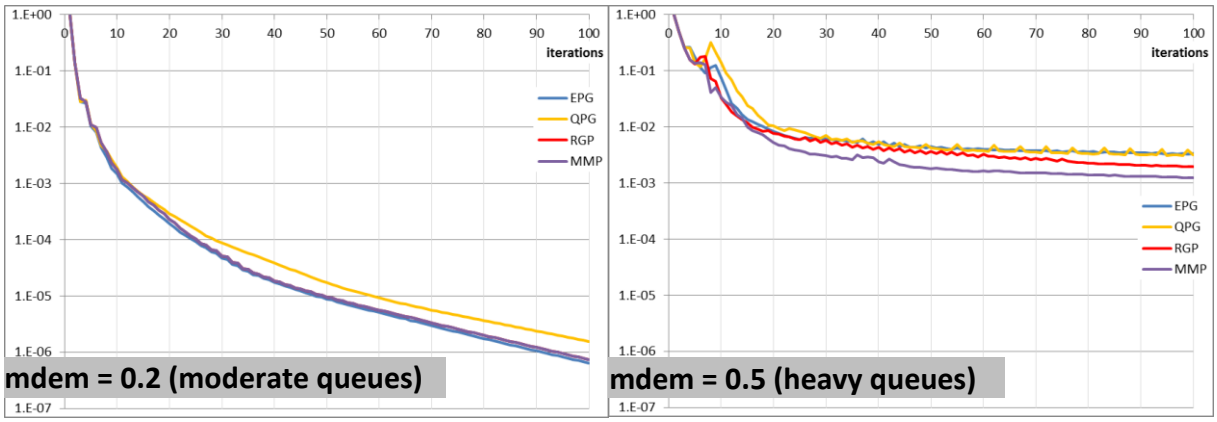

Figure 13. Convergence pattern for Dial network – All gradient projection methods performing similarly.

Figure 13 shows how all the proposed gradient projection methods (EPG, QGP, RGP, MMP) perform similarly. Tests are executed for h_t of 600 sec, and for two congestion levels: moderate (mded = 0.2) and heavy (mdem = 0.5).

These results merit some comments, because the higher (analytical more than computational) effort required to solve the projection problem exactly with EPG, instead that approximatively with RGP or MMP, seems not to be justified. This algorithm behaviour can be explained with the fact that the implicit path enumeration method operates with a limited number of used local alternatives (e.g. a few links of the node forward star). When this number reduces to two all the proposed algorithm basically coincide. We have experimented that in some cases in presence of more alternatives, MMP converges less smoothly than the other three methods, because it involves only two alternatives. Instead the difference between EPG, QGP and RGP is in practice minor, despite the theoretical superiority of EPG.

Although in theory the solution of the dynamic shortest paths and (more important) the solution of the linear system for the flow propagation can imply each time a different

computation cost, we have found that in practice the running times of each iteration is fairly stable on a given network. To see this we can consider a larger test network (representing the city of Cosenza, Italy) with 524 arcs and 40 zones; 120 intervals of 1 minute each were used to simulate a time varying demand with a multiplier *mded* of the original matrix from 1.0 to 1.4.

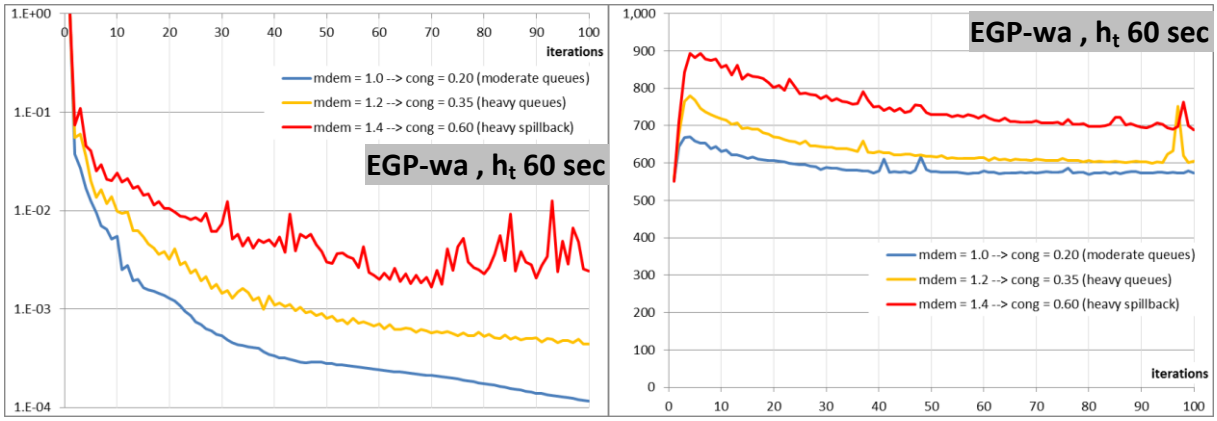

Figure 14. Convergence pattern for the Cosenza network – The computation time (in milliseconds) per iteration is fairly stable.

Figure 14 shows the convergence pattern experimented in the case of moderate queues (mded = 1.0 → cong = 0.20), heavy queues (mded = 1.2 → cong = 0.35), and heavy spillback (mded = 1.4 → cong = 0.60). The computing time per iteration ranges from 550 to 900 milliseconds, depending on the level of congestion, which is higher in the early iterations where the arcs used to reach each destination do not form yet a bush, like they shall at equilibrium. A mini-desktop computer with Intel Core i5-4210U 1.70 GHz was used to run the tests.

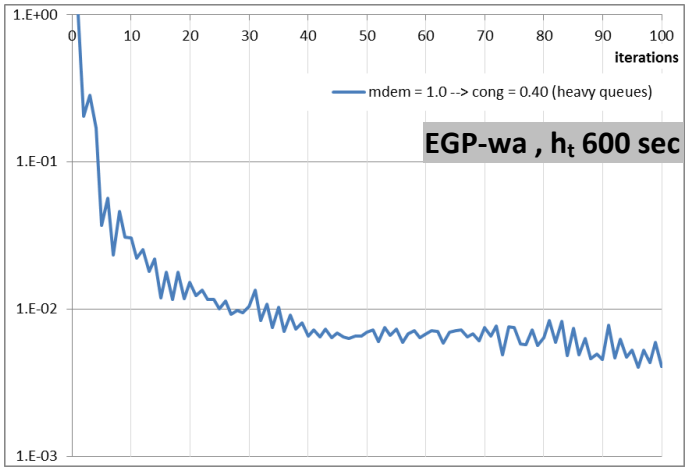

Figure 15. Convergence pattern for the Rome network.

Figure 15 shows the convergence pattern experimented for the larger network of Rome (with 9365 links and 453 zones) in the case of heavy queues (cong = 0.40), which is the daily condition in the eternal city, as reported by the TomTom Traffic Index (Cohn, 2014). Simulating the whole day with 10 minutes intervals required almost 3 hours of computation, for 100 iterations.

The larger dimension of the network and especially of the number of zones produces a harder problem to solve. After all the proposed algorithm partitions the assignment

model by destinations and the search directions refer to each sub-problem separately. Nonetheless, an acceptable convergence with a relative gap of 10^{-2} is reached after 30 iterations and one of 4×10^{-3} after 100 iterations. This is suitable for many applications and certainly a relevant step forward in the direction of a more precise computation of Dynamic User Equilibrium, wrt other assignment methods.

ACKNOWLEDGMENTS

The author wishes to thank two anonymous referees and Prof. Francisco Facchinei who helped in better casting this article in the consolidated field of Variational Inequalities.

REFERENCES

Adamo V., Astarita V., Florian M., Mahut M., Wu J. (1999) Modeling the spillback of congestion in link based dynamic network loading models: A simulation model with application. In Transportation and Traffic Theory, ed. A. Ceder, Pergamon-Elsevier, New York, USA, pp. 555-573.

Barcelo J. (2010) Fundamentals of Traffic Simulation, Springer, New York, USA.

Beckmann M.J., McGuire C.B., Winsten C.B. (1956) Studies in the Economics of Transportation. Yale University Press, Connecticut, USA.

Bellei G., Gentile G., Papola N. (2005) A within-day dynamic traffic assignment model for urban road networks. Transportation Research B 39, pp. 1-29.

Bellei G., Gentile G., Meschini L., Papola N. (2006) A demand model with departure time choice for within-day dynamic traffic assignment. European Journal of Operational Research 175, 1557-1576.

Ben-Akiva M., Bierlaire M., Bottom J., Koutsopoulos N., Mishalani R. (1997) Development of a route guidance generation system for real-time application. Proceedings of the 8th IFAC symposium on transportation systems, Chania, Greece.

Ben-Akiva M, Bierlaire M, Koutsopoulos H., Mishalani R. (2002) Real-time simulation of traffic demand–supply interactions within DynaMIT. In Transportation and network analysis: current trends. Miscellanea in honour of Michael Florian, ed.s M. Gendreau and P. Marcotte, Kluwer, Boston, USA, pp. 19-36.

Bertsekas D.P. (1976) On the Goldstein-Levitin-Polyak gradient projection method. IEEE Transactions on Automatic Control 21, 174–184.

Blum J.R. (1954) Multidimensional stochastic approximation methods. The Annals of Mathematical Statistics 25, 737-744.

Cantarella G.E. (1997) A general Fixed-Point approach to multi-mode multi-user equilibrium assignment with elastic demand. Transportation Science 31, 107-128.

Chabini I. (1998) Discrete dynamic shortest path problems in transportation applications: complexity and algorithms with optimal run time. Transportation Research Record 1645, pp. 170-175.

Cohn N. (2014) TomTom Traffic Index: Toward a global measure. Presented at ITS France, Paris.

Dafermos S. (1980) Traffic equilibrium and variational inequalities. Transportation Science 14, 42-54.

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

Facchinei F., Pang J.-S. (2003) Finite-dimensional variational inequalities and complementarity problems. Springer, New York, USA.

Friesz T., Bernstein D., Smith T., Tobin R., Wie B. (1993) A variational inequality formulation of the dynamic network user equilibrium problem. Operations Research 41, pp. 179-191.

Friesz T., Mookherjee R. (2006) Solving the dynamic network user equilibrium problem with state-dependent time shifts. Transportation Research B 40, 207-229.

Gentile G. (2010) The General Link Transmission Model for Dynamic Network Loading and a comparison with the DUE algorithm. In New developments in transport planning: advances in Dynamic Traffic Assignment (selected papers from the DTA 2008 Conference, Leuven), ed.s L.G.H. Immers, C.M.J. Tampere, F. Viti, Transport Economics, Management and Policy Series, Edward Elgar Publishing, MA, USA, 153–178.

Gentile G. (2014) Local User Cost Equilibrium: a bush-based algorithm for traffic assignment. Transportmetrica A: Transport Science 10, 15-54.

Gentile G. (2015) Using the General Link Transmission Model in a Dynamic Traffic Assignment to simulate congestion on urban networks. Transportation Research Procedia 5, 66-81

Gentile G. (2016) Dynamic routing on transit networks. Multi-Modal transit systems modelling in the context of ITS/ICT, ed.s. A. Nuzzolo and W. Lam., Taylor and Francis (to appear).

Gentile G., Meschini L. (2011) Using dynamic assignment models for real-time traffic forecast on large urban networks. In Proceedings of the 2nd International Conference on Models and Technologies for Intelligent Transportation Systems, Leuven, Belgium.

Gentile G., Papola A. (2006) An alternative approach to route choice simulation: the sequential models. Proceedings of the European Transport Conference 2006 – ETC 2006, Strasbourg, France.

Gentile G., Meschini L., Noekel K. (2006) Dynamic User Equilibrium – DUE, in VISUM 10 Manual, Karlsruhe, Germany.

Gentile G., Meschini L., Papola N. (2005) Macroscopic arc performance models with capacity constraints for within-day dynamic traffic assignment. Transportation Research B 39, 319–338.

Gentile G., Meschini L., Papola N. (2007) Spillback congestion in dynamic traffic assignment: a macroscopic flow model with time-varying bottlenecks. Transportation Research B 41, 1114–1138.

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

Jayakrisham R., Mahmassani H., Hu T. (1994) An evaluation tool for advanced traffic information and management systems in urban networks. Transportation Research C 2, pp. 129-147.

Han K., Piccoli B., Friesz T. (2015) Continuity of the path delay operator for LWR-based network loading with spillback. Submitted to Transportation Research B.

Harker T., Pang J.-S. (1990) Finite-dimensional variational inequality and nonlinear complementarity problem: A survey of theory, algorithms and applications. Mathematical Programming 48, 161-220.

Heydecker B., Addison J. (1998) Traffic models for dynamic traffic assignment. In Transport networks: recent methodological advances, ed M.G.H. Bell, Pergamon-Elsevier, Oxford, UK, pp. 35-49.

Himpe W., Corthout R., Tampère C (2013) An Implicit Solution Scheme for the Link Transmission Model. Proceeding of the 16th IEEE ITSC, The Hague, The Netherlands.

Liu H., He X., He B. (2009) Method of Successive Weighted Averages (MSWA) and self-regulated averaging Schemes for solving Stochastic User Equilibrium Problem. Networks and Spatial Economics 9, 485-503.

Lo H., Szeto W., 2002. A cell-based variational inequality formulation of the dynamic user optimal assignment problem. Transportation Research B 36, pp. 421-443.

Mahmassani H. (2001) Dynamic network traffic assignment and simulation methodology for advanced system management applications. Network and Spatial Economics 1, pp. 267.292.

Mahut M., Florian M., Tremblay N. (2008) Comparison of assignment methods for simulation based dynamic-equilibrium traffic assignment. Presented at the 87th Annual Meeting of the Transportation Research Board, Washington DC, USA.

Mounce R. (2007) Convergence to equilibrium in dynamic traffic networks when route cost is decay monotone. Transportation Science 41, 409-414.

Mounce R., Carey M. (2014) On the convergence of the Method of Successive Averages for calculating equilibrium in traffic networks. Transportation Science 49, 535-542.

Papageourgiou M. (1990) Dynamic modelling, assignment and route guidance in traffic networks. Transportation Research B 24, pp. 471-95.

Ramadurai G., Ukkusuri S. (2011) B-Dynamic: An Efficient Algorithm for Dynamic User Equilibrium Assignment in Activity-Travel Networks. Computer-Aided Civil and Infrastructure Engineering 26, 254–269.

Ran B., Boyce D. (1994) Dynamic urban transportation network models: theory and implications for intelligent vehicle-highway systems. Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, New York, USA.

Rosen J.B. (1960) The gradient projected method for nonlinear programming, part I: linear constraints. J Soc Indus Appl Math 8,181–217.

Saad Y. (2003) Iterative Methods for Sparse Linear Systems (2nd ed.). SIAM.

Smith M. (1979) The existence, uniqueness and stability of traffic equilibria. Transportation Research B, 295-304.

Smith M., Mounce R. (2011) A splitting rate model of traffic re-routeing and traffic control. Transportation Research B 45, 1389–1409.

Van der Vorst H.A. (1992) Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. SIAM J. Sci. and Stat. Comput. 13, 631–644.

Yang I., Jayakrishnan R. (2012) Gradient projection method for simulation-based dynamic traffic assignment. Transportation Research Record 2284, 70-80.

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