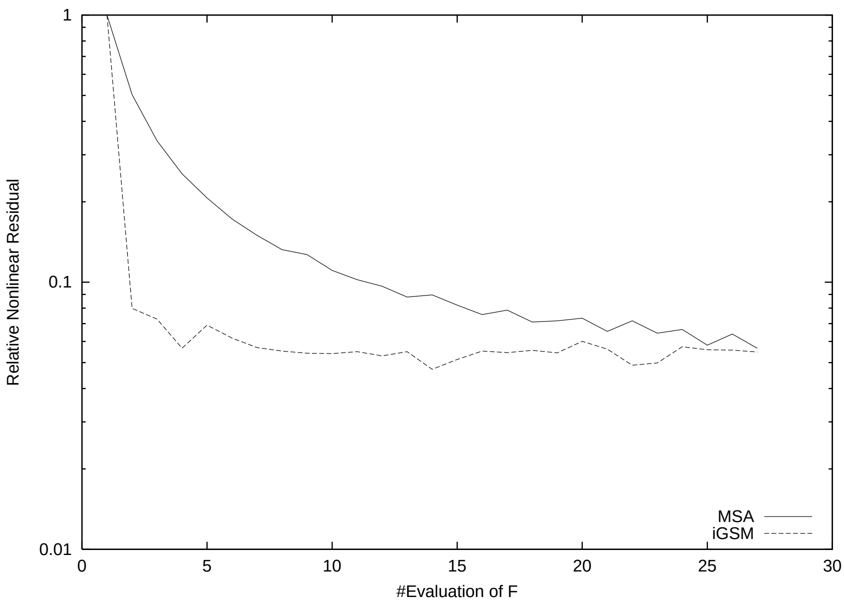


Solving the anticipatory route guidance generation problem using a generalization of secant methods

Frank Crittin, ROSO-IMA-EPFL
Michel Bierlaire, ROSO-IMA-EPFL

Conference paper STRC 2003
Session Model and Generation

STRC | **3rd Swiss Transport Research Conference**

Monte Verità / Ascona, March 19-21, 2003

Solving the anticipatory route guidance generation problem using a generalization of secant methods[1]

Frank Crittin
Institut de Mathématiques
École Polytechnique Fédérale de Lausanne

Phone: +41 21 693 81 00
Fax: +41 21 693 55 70
e-Mail: frank.crittin@epfl.ch

Michel Bierlaire
Institut de Mathématiques
École Polytechnique Fédérale de Lausanne

Phone: +41 21 693 25 37
Fax: +41 21 693 55 70
e-Mail: michel.bierlaire@epfl.ch

[1]This research is supported by SNF grant 2100-63848.00

Abstract

Route guidance refers to information provided to travelers in an attempt to facilitate their decisions relative to departure time, travel mode and route. We are specifically interested in consistent anticipatory route guidance, in which real-time traffic measurements are used to make short-term predictions, involving complex simulation tools, of future traffic conditions. These predictions are the basis of the guidance information that is provided to users. By consistent, we mean that the anticipated traffic conditions used to generate the guidance must be similar to the traffic conditions that the travelers are going to experience on the network. The problem is tricky because, contrarily to weather forecast where the real system under consideration is not affected by information provision, the very fact of providing travel information may modify the future traffic conditions and, therefore, invalidate the prediction that has been used to generate it.

Bottom (2000) has proposed a general fixed point formulation of this problem with the following characteristics. First, as guidance generation involves considerable amounts of computation, this fixed point problem must be solved quickly and accurately enough for the results to be timely delivered to drivers. Secondly the unavailability of analytical forms for the objective function and the presence of noise due to the use of simulation tools prevent from using classical algorithms. We propose in this paper an adaptation of the generalized secant method (cf. the related presentation *"A generalization of secant methods for solving nonlinear systems of equations"*) in order to handle the intrinsic characteristics of the consistent anticipatory route guidance generation, especially the very high dimension associated with real problems.

We present then a number of simulation experiments based on two simulation tools in order to compare the performances of the diverse algorithms. The first is a simple simulator implementing the framework of the route guidance generation on a small network, which is used to illustrate the properties of this problem and the behavior of the algorithms. Then, we present a large-scale case study of size 124575 using DynaMIT, a simulation-based real-time Dynamic Traffic Assignment system designed to compute and disseminate anticipatory route guidance. These results point out the real-time potential of the method as its ability to handle large scale problem.

Keywords

Intelligent transportaion system - consistent anticipatory route guidance - large scale fixed point - 3rd Swiss Transport Research Conference - STRC 2003 - Monte Verità

1 Introduction

Anticipatory guidance informs travelers about the traffic conditions they will experienced during their trip based on prediction of the future condition of the network. But as traffic information affects drivers behavior, it may invalidate predicted traffic conditions that were used to generate it. Therefore the concept of consistency is very important for the generation of reliable information. We declare that the guidance is consistent when the forecasts on which it is based are verified after the reaction of drivers. This problem has been formulated as a fixed point problem. Due to the fact that the computation of anticipatory guidance involves complex simulation tools, the distinctive features of this fixed point problem are stochasticity, large scale and abundant amount of computation. Classical methods to solve the Consistent Anticipatory Route Guidance (CARG) problem are obviously fixed point methods, unfortunately their slow behavior make them inadequate for real-time application.

First we have considered this problem as a large scale optimization problem without derivative (see Bierlaire and Crittin, 2001 for more details). The idea was to consider a population-based procedure to gather variational information about a local model of the objective function. The preliminary numerical results were encouraging, especially for real-time applications, as the method decreases rapidly the objective function in the first iterations. Unluckily the algorithm often get stuck in a subspace where no improvement can be achieved anymore, due certainly to the appearance of local minima of the objective function, which are not fixed point of the initial problem. In that paper (Bierlaire and Crittin, 2001) we had already considered to use a formulation in terms of resolution of a system of nonlinear equations instead of a minimization problem.

This paper will describe a new algorithm considering the CARG problem as a resolution of a system of nonlinear equations. This new method will be a large scale adaptation of a new class of efficient methods proposed by Bierlaire and Crittin, 2003. The concessions made to accommodate the constraints of the CARG problem allow us to design a particularly efficient algorithm for the resolution of large scale systems of equations.

This paper is organized as follow, first we briefly recall the analysis framework which leads to the formulation of the guidance generation problem in terms of fixed point. Section 3 describes different possible formulations of fixed point problems with their associated class of algorithms. Section 3.3.1 identify some existing algorithms especially design to solve large scale systems of equations and we next describe the large scale adaptation of the Generalized Secant Method (GSM) described in (Bierlaire and Crittin, 2003). Numerical results presented in Section 5 are divided into two parts. First we show that this new algorithm is efficient on classical nonlinear systems of equations and secondly we analyze its performance on the CARG problem. Some conclusions and perspectives are outlined in Section 6.

2 Consistent anticipatory route guidance

Route guidance refers to information disseminated to road users with the intent of influencing their route choice decisions. We are interested in here in *anticipatory* route guidance where real-time traffic conditions are used to make predictions of the evolution of the network. Hence information provided to a driver will reflect the conditions that are expected to prevail at network locations at the times when he will actually be there.

A tricky problem in generating anticipatory route guidance is the fact the system under consideration is affected by the dissemination of information. Indeed, contrarily to weather forecast, the reactions of the users receiving the guidance can affect the future conditions of the network and therefore invalidate the predictions on which the guidance was based. The anticipatory guidance is said to be *consistent* if the predictions on which the guidance is based are the same as those that are forecast to result after drivers react to the guidance.

This problem was introduced by Ben-Akiva et al. (1996) and developed in the context of DynaMIT by Ben-Akiva et al. (1997), Ben-Akiva et al., 2002, Ben-Akiva et al., 1998).

2.1 Fixed point formulation

First formulations of the CARG generation as a fixed point problem have been proposed by Bottom et al. (1999) and developed by Bottom (2000) in his PhD dissertation. This formulation has been described in (Bierlaire and Crittin, 2001), consequently only a brief overview is given in the following.

In this framework the traffic network is described as a directed graph with an enumeration of feasible paths. The transportation demand is given by origin, destination, departure time, behavorial class and type of access to information. The analysis time horizon is fixed, and divided into a finite number of fixed time steps. In this manner the CARG problem can be formulated as a fixed point problem which may be stated as follows:

$$\text{Find } x \text{ such that } x = T(x) \tag{1}$$

where x belongs to one of the three sets of variables:

- Path flows P, representing the number of trips of a particular user class traveling from a given decision point to a given destination at a given time.
- Network conditions C, that are typically represented by time-dependent link impedances for each time interval in the given time horizon. In most cases, the impedance will be the link traversal time.
- Guidance messages M are quadruple involving message type, location, time and contents. The exact definition of these variables depends on the specific technology under consideration.

Each pair of variables sets are related by causal relationship, expressed by the following maps and diagrammatically represented in Figure 1:

- The Network loading map, denoted by $S : P \to C$, determines the traffic conditions that result from the assignment of a given set of time-dependent path flows P over the network.
- The Guidance map, denoted by $G : C \to M$, represents the generation of actual messages M by the ATIS, based on predicted traffic conditions C. The map captures the technological characteristics of the information system.

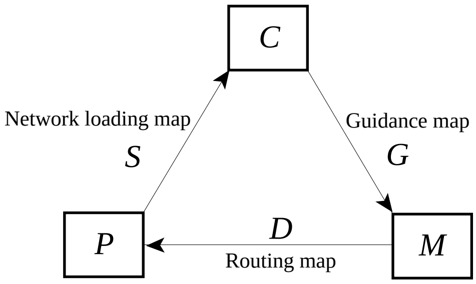


Figure 1: Fixed Point Formulation (CARG)

- The Routing map, $D : M \to P$ relates a given set of guidance messages to the resulting path flows. It captures the drivers response to the information, and is typically based on behavioral models such as route choice, departure time choice and mode choice (see Ben-Akiva and Bierlaire, 1999).

And finally T can be described as one the three following compositions:

$$\begin{cases} \text{If } x \in P \text{ then } T \text{ is defined by } D \circ G \circ S \\ \text{If } x \in C \text{ then } T \text{ is defined by } S \circ D \circ G \\ \text{If } x \in G \text{ then } T \text{ is defined by } G \circ S \circ D \end{cases} \tag{2}$$

The complexity of transportation systems and the necessicity of capturing traveler behavior impose the use of disaggregated models and simulation-based tools to compute one of the three composition map defining T. Consequently these three fixed point problems, defined in (1) and (2), are non-analytical and stochastic, where the level of stochasticity depends on the simulation tools used to compute the composite map. Moreover the three variables x involves a high number of variables producing large scale problems. Indeed if the decision variable associated with the composite map is the link impedance, the size of the fixed point problem is the number of links of the network times the number of time steps. As an example if the network contains 1661 links and we want to produce guidance for the next 75 minutes with time step of one minute the size of the associated fixed point is 124575 as shown for the Swiss network in subsection (5.2.2).

Remark also that these three composite maps are equivalent with respect to the existence of a fixed point: if one has a fixed point then they all do, and if one does not then none does. Unfortunately there is no guarantee that such fixed point exists. Thus in general we will define the *consistency* of a point x, belonging to one of the three sets of variables presented above, as a value $||x - T(x)||$, for some norm $|| \cdot ||$. We will consider that consistency has been achieved, and roughly speaking the fixed point found, if we find an approximate solution to the CARG problem defined by:

$$||x - T(x)|| < \epsilon \tag{3}$$

for some small $\epsilon > 0$.

The non-existence of a fixed point has two concrete implications on the guidance generation problem. First it means that whatever information is sent to travelers, it will never be consistent with the traffic conditions they will experience, *i.e.* the guidance is known to have a poor quality before to be sent to users. Additionally these type of guidance could worsen rather than improve traffic conditions, as shown in (Arnott et al., 1991) and (Hall, 1996). Moreover, from an algorithmic point of view the non-existence of a fixed point invalidate theoretical convergence properties of algorithms used to find it.

3 Existing methods

In this section, we present three different formulations of fixed problems, namely fixed point, optimization and resolution of nonlinear systems of equations, with their classical associated resolution methods. Analysis of advantages and drawbacks of each type of methods in the context of the CARG problem is also given.

3.1 Fixed point methods

The previous section shown route guidance generation problem as a fixed point problem involving one of the three composite maps. Classical algorithms to solve fixed point problems are iterative methods based on the famous Banach Contraction Principle, which can be described as follow: Let $\mathcal{T} : X \to X$ a mapping accepting $x^* \in X$ as a fixed point, *i.e.* $\mathcal{T}(x^*) = x^*$. Choose a starting point $x_0 \in X$ and generate a succession of point of the form:

$$x_{k+1} = x_k + \alpha_k(\mathcal{T}(x_k) - x_k) \tag{4}$$

where $\alpha_k \in [0; 1]$. We will refer to averaging methods for this class of algorithms. We present now some particular implementation of these type of methods. If $\alpha_k = 1, \forall k$ this method is called fixed point iteration, method of successive substitution, or nonlinear Richardson iteration. This method has been proved to be convergent by Banach (1922) if the mapping $\mathcal{T}$ is contractant[2] . If $\alpha_k = \alpha, \forall k$ with $\alpha \in [0, 1]$ this method is referred to as time smoothing algorithm. For the CARG problem Bottom (2000) has tested a number of averaging methods and kept two as best solution algorithms.

1. The method of successive averages (MSA) define by 4 with $\alpha_k = \frac{1}{k}$. Robbins and Monro (1951) and Blum (1954) have shown, under some conditions, the convergence of this method to a fixed point, despite noisy evaluations of $\mathcal{T}$. This method has been successively used for some classical transportation problems, as for example Sheffi and Powell (1982) who used it for stochastic user equilibrium or Cantarella (1997) who applied this algorithm to solve two general fixed point formulation of multimode multi-user equilibrium problem.

2. The Polyak averaging method is a simple off-line running average of points generate by (4). More precisely at each iteration we compute a new iterate, say $\Psi_k = \sum_i^k \frac{x_i}{k}$. Polyak and Juditsky (1992) have shown that the sequence Ψ_k converge to x^* at an optimal rate, if $\alpha_k \to 0$ slower than $o(1/k)$. Remarkably this procedure theoretically equals or surpasses asymptotic performances of any iterative methods define by (4).

Remark

[2] Let $\mathcal{D} \subset \mathbb{R}^n$. $K : \mathcal{D} \to \mathbb{R}^n$ is contractant if $||K(x) - K(y)|| \leq K_{lip}||x - y|| \quad \forall x, y \in \mathcal{D}$ with $K_{lip} < 1$.

- This kind of method are very useful for very large scale problems due to the insignificant amount of linear algebra associated to the generation of each iterate as well as its small need of memory as only the last two iterates have to be conserved.
- In the case of consistent route guidance generation the mapping $\mathcal{T}$ is given by T, one of the three composite maps from (2) and as emphasized before there is no evidence of the existence of a fixed point of T in this case, which invalidates theoretical properties of convergence proof of these algorithms. Furthermore in practice averaging methods exhibit a slow convergence on CARG problem.

3.2 Derivatives free optimization methods

Bierlaire and Crittin (2001) have considered the fixed point problem (1) as a nonlinear minimization problem:

$$\min_x f(x) \text{ with } f(x) = d(x, T(x)) \tag{5}$$

where d defines a suitable distance. Notice that if x^* is solution of problem (1) then x^* is a global minimum of f with $f(x^*) = 0$. On the contrary if $\bar{x}$ is solution of (5) nothing ensure that $T(\bar{x}) = \bar{x}$. (cf. Dennis and Schnabel, 1996). As described above the evaluation of the function is computationally very expensive, and so global optimization methods are not adapted for this problem, moreover as we can not compute the derivatives of the objective function we have to focus on direct search method. Consult (Bierlaire and Crittin, 2001) for a description and criticisms of existing derivative free algorithms as well as a new method, especially designed for large scale problems, proposed by the authors. The main drawback of this kind of algorithms lie in the fact they can be stuck in local minimizer far from the solution of the original problem. This distinctive feature can explain some mitigated numerical results obtained by these methods.

3.3 Methods solving nonlinear system of equations

Another natural way to express fixed point problems is as resolution of systems of nonlinear equations written:

$$F(x) = 0 \tag{6}$$

with $F : Y \to Y$, $Y \subseteq \mathbb{R}^n$.

The equivalence of this two formulations, *i.e.* if x^* is solution of (1), it is also solution of (6) and reciprocally, is straightforward setting $F(x) = T(x) - x$.

As described, in Bierlaire and Crittin (2003) most of methods used to solve problem (6), referred as quasi-Newton methods, are iterative methods using the following classical framework. At each iteration k solve:

$$\begin{aligned} B_k s_k &= -F(x_k), & (7) \\ x_{k+1} &= x_k + s_k, & (8) \end{aligned}$$

and update B_{k+1}.

In particular, if $B_{k+1} = \nabla F(x_{k+1})$, at each iteration, we obtain the well-known Newton method. Remark that this method is not adapted to CARG problem while it has to compute derivatives of F. One way to avoid computation of derivatives is to carry an approximation of the Jacobian. If these approximations satisfy at each iteration the following secant equation:

$$B_{k+1}(x_{k+1} - x_k) = F(x_{k+1}) - F(x_k) \tag{9}$$

they are called secant methods (cf. (Dennis and Schnabel, 1996), (Kelley, 2002), (Bierlaire and Crittin, 2003)). The most successful secant methods are the Broyden methods, but are not really designed to solve large scale systems of equations. The main drawbacks of quasi-Newton methods when they deal with large scale problems are the storage cost of the matrix B_{k+1} and also the resolution of the associated linear system (7). Therefore very large scale problems, like route guidance generation problem, imply adaptations of classical methods.

3.3.1 Large scale quasi-Newton methods

We have to distinguish here two kinds of methods. The first type of methods approximate the Jacobian in a way that not only avoids computation of the derivatives, but also saves linear algebra work for solving (9), as for example the *"Bad"* Broyden method proposed by Broyden (1965). This method allows to avoid the resolution of the linear system as it updates directly the approximation of the inverse of the Jacobian. Unfortunately, for large scale problems as for example the consistent anticipatory route guidance, this method is unusable. Due to the size of the problem, the construction itself of the entire matrix B_k^{-1} is doomed to failure. In this context limited memory implementation of Broyden's methods have been proposed (Gomes-Ruggiero et al., 1991), which are based on a compact representation of matrices B_k^{-1} (Byrd et al., 1994). Notice that good results have been obtain in optimization but not really for solving nonlinear systems of equations. At this point the most effective method for large scale method has been introduced by Martinez and Zambaldi (1992) and is named Inverse-Column Updating method (ICUM). This is a secant algorithm where B_{k+1}^{-1} is obtain from B_k^{-1} by changing only one of its columns. Following Luksan and Vlcek (1998) this update seems to be the most efficient for solving large systems of nonlinear equations without computing derivatives. Unfortunately this method also needs the explicit construction of the matrix B_{k+1}^{-1}, without, which can be detterent for very large applications.

We have to cite a second type of methods, called Newton-Krylov methods (cf. Kelley, 2002). They use, as their names suggest it, Krylov subspaces based on linear solvers to solve system (7). Despite that Newton-Krylov methods use partial derivatives, and so is helpless in our route guidance generation problem, this family of algorithms, allowing to perform only matrix-vector product is especially well adapted for very large scale problems as underlined by Kelley (2002) and so has to be mentioned.

This short description of the two top rated algorithms to solve large scale nonlinear systems of equations highlights the need of a specific method to solve the consistent anticipatory route guidance, in particular with the constraints of non-availability of the derivatives and the necessity of performing only matrix-vector products during the algorithm.

4 Large scale adaptation of GSM

We propose here an adaptation of the General Secant Method proposed in Bierlaire and Crittin (2003). Motivated first by the performance of this class of algorithms compared with classical secant methods in medium scale and secondly by the fact that a method considering a population of iterates to calibrate the approximation of B_{k+1}^{-1} is likely to be more robust in the presence of noise.

As discussed above, we consider the inverse version of GSM, *i.e.* updating directly the inverse of the approximation of the Jacobian matrix instead of the Jacobian itself. This formulation is necessary in large scale cases as it allows to avoid to solve the system (7). We propose to compute new matrix B_{k+1}^{-1} using a least-square approach in order to calibrate our associated linear model with several previous iterates. Following the approach given in (Bierlaire and Crittin, 2003) we obtain the following least-square problem:

$$B_{k+1}^{-1} = \underset{J}{\operatorname{argmin}} \left\| J \begin{pmatrix} \Omega \cdot Y_{k+1} & \Gamma \cdot I_{n\times n} \end{pmatrix} - \begin{pmatrix} \Omega \cdot S_{k+1} & \Gamma \cdot (B_{k+1}^0)^{-1} \end{pmatrix} \right\|_F^2 \qquad (10)$$

where $\Omega \in \mathbb{R}^{k+1}$ is a diagonal matrix with weights ω_{k+1}^i on the diagonal for $i = 0, \cdots, k$; the matrix Γ contains weights associated with the arbitrary term $(B_{k+1}^0)^{-1}$; $Y_{k+1} = (y_k, y_{k-1}, \ldots, y_0)$; $S_{k+1} = (s_k, s_{k-1}, \ldots, s_0)$ with $y_k = F(x_{k+1}) - F(x_k)$ and $s_k = x_{k+1} - x_k$.

Let $A = \begin{pmatrix} \Omega \cdot Y_{k+1} & \Gamma \cdot I_{n\times n} \end{pmatrix}$ and $C = \begin{pmatrix} \Omega \cdot S_{k+1} & \Gamma \cdot (B_{k+1}^0)^{-1} \end{pmatrix}$, using these notations, (10) can be written as $B_{k+1}^{-1} = \underset{J}{\operatorname{argmin}} \|A - C\|_F^2$. Solving the normal equations we can directly compute the associated quasi-Newton step given in (7):

$$s_k = -(CA^T)(AA^T)^{-1}F(x_k) \qquad (11)$$

With a small amount of linear algebra[3] we can show that s_k defined by (11) is equivalent to the following:

$$\begin{cases} \text{1. Solve} & x = \underset{y}{\operatorname{argmin}} \|Ay - F(x_k)\|_2^2 \\ \text{2. Compute} & s_k = -Cx \end{cases} \qquad (12)$$

Remark that the least-square associated with (12) is now a vector least-square, contrarily to (10) which is a generalized matrix least-square. Moreover with this formulation there is no need to store or even construct the matrix B_{k+1}^{-1}, and consequently the method can be implemented as a matrix-free algorithm, *i.e.* only matrix-vector products have to be computed, which is decisive for large scale problems. The only matrices that we need to store are Y_k, S_k, $(B_{k+1}^0)^{-1}$, Ω and Γ. More precisely, matrices Y_k and S_k have size $n \times (\kappa - 1)$ where n is the size of the problem and κ the number of iterates kept in the population. The matrix $(B_{k+1}^0)^{-1}$ is an a priori matrix whose role is to overcome the possible underdetermination of the problem (10) which can also be written as following:

$$B_{k+1}^{-1} = B_k^{-1} + \left(\Gamma^2 + Y_{k+1}\Omega^2 Y_{k+1}^T\right)^{-1} Y_{k+1}^T \Omega^2 \left(S_{k+1} - B_k^{-1} Y_{k+1}\right) \qquad (13)$$

A classical choice for these matrices is to choose, at each iteration, $(B_{k+1}^0)^{-1} = B_k^{-1}$. In that case (13) becomes an update formula and local convergence can be proved using the same approach as Bierlaire and Crittin (2003). Nevertheless this approach can be difficult to apply

[3] Remarking that $A^T(AA^T)^{-1} = (A^TA)^{-1}A^T$

on large scale cases due to intensive linear algebra computations and high storage costs. In addition the choice of this a priori matrix is not fundamental for the behavior of the method. Therefore we propose to use the identity matrix because of storage cost and numerical stability considerations. Additionally, considering the CARG problem, the choice of the identity matrix stands out because it can be shown that if the weights associated to under-determination are sizeable then the sequence of iterates generated will be close to those generated by averaging methods.

The matrix Ω, as described before, is a diagonal matrix stocked as a vector. It captures the relative importance of each iterate in the population assigning more weight to points close to x_{k+1}, and less weight to points faraway. The matrix Γ apprehends the importance of the arbitrary term defined by $(B^0_{k+1})^{-1}$. Bierlaire and Crittin (2003) propose to use modified Cholesky factorization of the matrix $S_{k+1}\Omega^2 S^T_{k+1}$ to compute Γ, addressing both the problem of overcoming the under-determination and guaranteeing the numerical stability of the associated least-square (10). Naturally in large scale case this approach is very costly in term of computation time, moreover the information contained in matrix $(B^0_{k+1})^{-1}$ is low, as we have chosen to set it equal to identity. As a consequence the only role of Γ is to ensure the positive definiteness of matrix:

$$(\Gamma^2 + Y_{k+1}\Omega^2 Y^T_{k+1}) \tag{14}$$

coming from (13), to guarantee the numerical stability of the associated least-square. Therefore to assure the safely positiveness we first remark that $Y_{k+1}\Omega^2 Y^T_{k+1}$ is positive semidefinite. Indeed for $\forall x \in \mathbb{R}^n$ we have:

$$\begin{aligned} x^T Y_{k+1}\Omega^2 Y_{k+1} x &= (x^T Y^T_{k+1}\Omega^T)(\Omega Y_{k+1} x) && (15)\\ &= \|\Omega Y_{k+1} x\|^2_2 && (16)\\ &\geq 0 && (17) \end{aligned}$$

Consequently the only dubious case is when eigenvalues of $Y_{k+1}\Omega$ are close to zero. To elude this to happen it is sufficient to choose $\Gamma = \tau I_{n\times n}$ where $\tau \in \mathbb{R}$, with first τ as small as possible in order not to perturb too much our model and secondly significantly[4] different from zero in order to make matrix (14) invertible.

We propose to use the LSQR algorithm to solve the least-square defined in (12). This algorithm proposed by Paige and Saunders (1982), analytically equivalent to a conjugate gradient method, requires only matrix-vector products, allowing to keep this feature for the whole algorithm and explicitly account for the problem's sparsity, since A and C are very sparse by construction.

4.1 Description of the algorithm

Following the observations of the previous section we described the adaptation of GSM methods for large scale problems. It is an iterative method, called iGSM (inverse General Secant Method), defined by:

$$x_{k+1} = x_k + s_k \tag{18}$$

where s_k is computed using (12), more precisely:

Algorithm iGSM(x_0, F, κ, τ)

[4] In terms of machine precision.

1. Compute $s_1 = -F(x_0)$, $k = 1$
2. For $k = 1, \ldots$ do
   (a) Compute $x_{k+1} = x_k + s_k$ and evaluate $F(x_{k+1})$,
   (b) Compute $s_i = \frac{x_{k+1}-x_i}{\|x_{k+1}-x_i\|_2^2}$ and $y_i = \frac{F(x_{k+1})-F(x_i)}{\|F(x_{k+1})-F(x_i)\|_2^2}$ for $i = \max(0, k-\kappa), \ldots, k$,
   (c) Construct $S = (s_k\ s_{k-1}\ \ldots)$ and $Y = (y_k\ y_{k-1}\ \ldots)$,
   (d) Solve $x = \text{argmin}_y \|Ay - F(x_{k+1})\|_2^2$ with $A = (Y\ \ \tau I_{n\times n})$ using LSQR algorithm,
   (e) Compute $s_{k+1} = -Cx$ with $C = (S\ \ \tau I_{n\times n})$.

Remarks

- Parameter τ has to be small in order to perturb the model as little as possible and big enough to ensure numerical stability. Accordingly we suggest to choose practically $\tau = \sqrt{\epsilon}$ with ϵ equal to the machine epsilon. It is defined as the smallest number ϵ^* such that $1 + \epsilon^* > 1$ and is system dependent.
- Parameter κ is the maximum number of previous iterates that we keep in our population to calibrate B_{k+1}^{-1}. The size of the parameter is determined by the size of the problem and the memory availability of the system. We can also consider to keep trace of previous iterates using a limited memory approach to improve the storage cost of the method.
- The matrices A and C do not need to be explicitly constructed. All we need is to be able to compute their matrix-vector product to build next iteration. This property is a major advantage for methods planned to solve very large scale systems of nonlinear equations, as the CARG problem. To our knowledge it is the only method solving nonlinear systems of equations without derivatives which has this feature.

5 Numerical results

In this section we present the numerical results of the algorithm described in 4.1. We briefly show that this algorithm is efficient on classical nonlinear systems of equations and then compare its numerical performances with averaging methods to solve the consistent anticipatory route guidance problem.

5.1 iGSM solving classical problems

We expose here a preliminary performance analysis of iGSM method, in comparison with classical algorithms to solve medium scale nonlinear systems of equations. These results tend to validate the ideas developed in previous sections. We also give a comparison on a classical large scale problem with a Newton-Krylov algorithm to emphasize the potential of iGSM method. All algorithms and test functions have been implemented with the package Octave (see `http://www.octave.org/`) and computations have been done on a laptop equipped with 1066MHz CPU in double precision. The machine epsilon is about 2.2204e-16.

5.1.1 Performance profile

We have decided to compare the proposed algorithm with three methods on classical problems. The first one is an hybrid method proposed by Martinez (1982). It is based on conjecture allowing to choose at each iteration of the algorithm between the Broyden Good or the Broyden Bad method. This method clearly outperforms both Broyden methods. The second algorithm is named ICUM (Inverse Column-Updating Method). It has been introduced by Martinez and Zambaldi (1992) to reduce the computational cost of the Broyden Bad method for large scale problems. Following Spedicato and Huang (1997) and Ruggiero et al. (1996) this method is currently considered as the best secant method for large scale problems without derivative. We also consider the GSM method proposed by Bierlaire and Crittin (2003) to visualize the impact of the adaptation for large scale cases. Note here that algorithms designed to solve fixed point problems based on the Banach contraction principle have not been considered as they are generally not convergent for solving systems of nonlinear equations, because the associated fixed point equivalent formulation is usually not contractant.

The numerical experiment has been carried out on the same set of test functions as in (Bierlaire and Crittin, 2003) with dimension 10 and 50. We also present the results in the form of performance profiles analysis proposed by Dolan and More (2002). First it appears surprisingly from

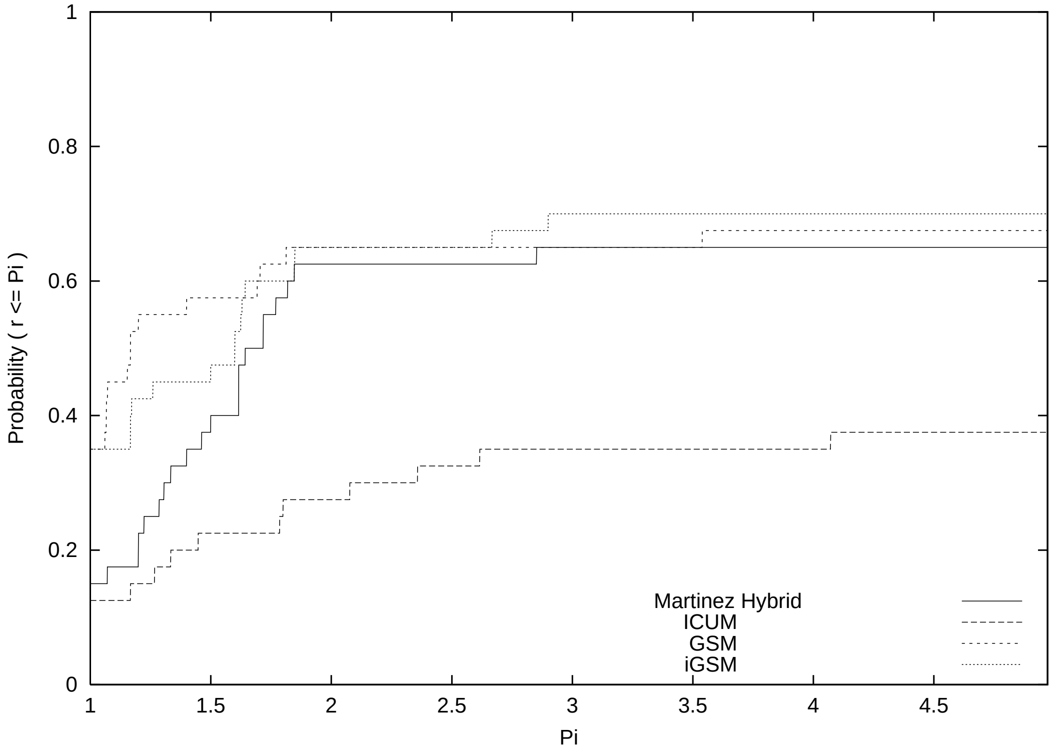


Figure 2: Performance Profile

Figure 2 that the adaptations of GSM algorithm to large scale problems do not deteriorate the performance of the method as the profile of GSM is very similar to the one of iGSM in terms of speed of convergence and robustness. The difference of profiles on small values of π between this two methods can be explained by the fact that using the modified cholesky factorization to calibrate Γ guarantees that the role of Γ is minimal, but most of the time this diagonal matrix is very small (near or equal to zero). So choosing $\Gamma = \tau I_{n\times n}$, as proposed in section 4, do not change drastically the behavior of the algorithm, it only slows down a little bit the method. Remark also the difference of speed of convergence, when they converge, between iGSM and ICUM, two algorithms design for large scale problems. The reason is mainly that ICUM, at each iteration, change only one column of approximate Jacobian at each iteration, while iGSM re-calibrate an entire matrix. In this way iGSM adapts quicker to the function shape, as algorithm progresses, and so converge faster. Same remark can be made about their robustness as ICUM solves only 30% of proposed problems against 70% for iGSM as shown by Figures 2. It also shows that the hybrid method converges within a factor 2 of the best algorithm for nearly all the problems it has been able to solve. This behavior is certainly due to the fact that this method uses only the last two iterates to update the approximation of the Jacobian, even though GSM and iGSM use a population of iterates.

5.1.2 Convection-Diffusion example

We will present here results comparing iGSM, Broyden, ICUM and Newton-Krylov method on a classical large scale problem.

This example, described in details by Kelley (2002), is a semi-linear convection-diffusion equation of size 961. The objective function is right preconditioned using a fast Poisson solver. We have examined the performance of this four algorithms in terms of number of evaluations of F. Note that we have not plotted the result for ICUM as it diverges.

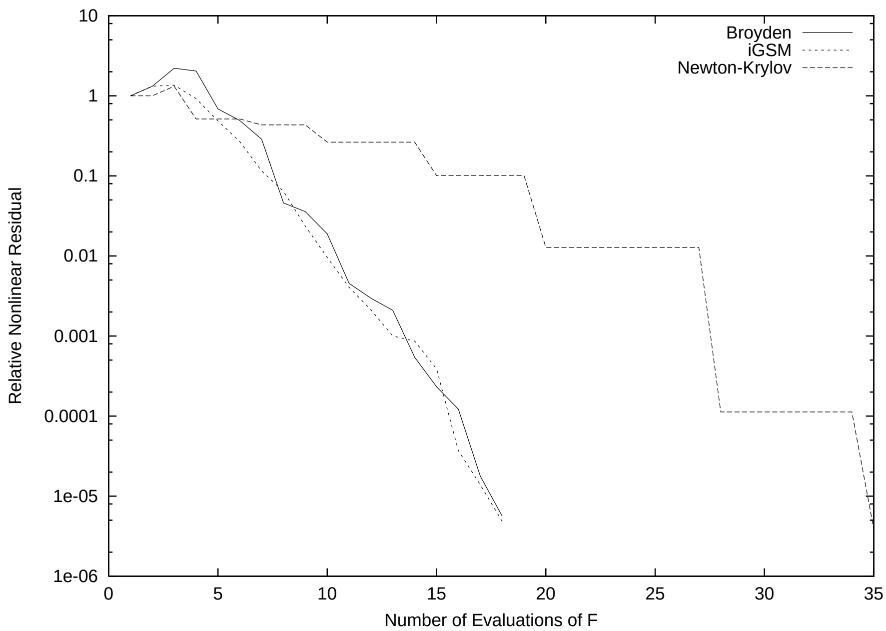


Figure 3: Convection-Diffusion equation

This result shows that iGSM algorithm is very efficient as shown by Figure 3 even compared with methods using derivatives information to solve nonlinear systems of equations, like Newton-Krylov methods. Remark the horizontal "step" of the Newton-Krylov method which represent the computation of the partial derivatives of F. Moreover the computational time to reach convergence on this problem is 14.9 [sec] for iGSM and 26.4 [sec] for the Newton-Krylov method, illustrating the potential of such a method on large scale problems. For comparison, contrarily to ICUM, the classical Broyden method has converged on this example but in 759.8 [sec], this result is logical while Broyden method is not design for large scale problems. However notice that in term of behavior the Broyden method is very similar to iGSM. This can be explain by the fact that when iGSM converges rapidly iterates are relatively distant each other. Due to the choice of s_i and y_i given in 4.1 it is reasonable that the sequence of iterates is similar to the one produced by the Broyden method as shown by Figure 3.

5.2 iGSM on the CARG problem

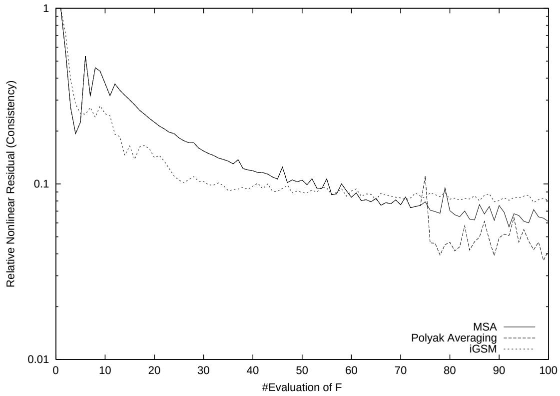
We provide here numerical comparison between averaging methods and iGSM to solve the consistent anticipatory route guidance problem. All algorithms have been implement in C++ and computations have been done with a laptop equipped with 1066MHz CPU in double precision. First we use a simple simulator to illustrate the behavior of these algorithms on a small synthetic network. Then we present two case studies to compare the numerical performances of these algorithms on real networks using DynaMIT system.

5.2.1 Simple Simulator

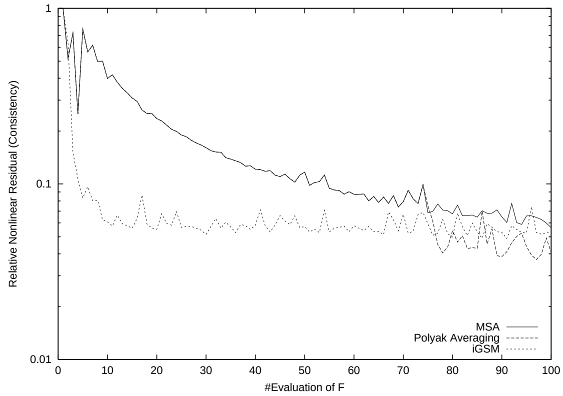
We present here the first results analysis concerning the anticipatory route guidance problem using a simple simulator implemented by Bottom (2000). This code implements in C++ a simple version of the network loading, routing and guidance components maps described in Section 2.1. This software is intended as a test bed for investigating different problems formulation and solutions methods for the consistent anticipatory route guidance. Runs presented here were made using a simple 14-link network with a single OD pair and eleven OD paths. We simulate the evolution of the network for 40 minutes with time intervals of 1 seconds. We also assume that 50 % of drivers have access to information, with a demand rate of 10800 trips per hour from origin to destination over 20 minutes. All guidance generation are computed using the composite link condition map expressed as link impedance, as described in section 2.1. With this specifications the size of the problem is 33600 (14×2400). We produce here four replications of the problems, all taking the free flow traversal times as initial point. Note that for the Polyak method the rolling averaging has been performed from iteration 75 based on MSA iterates.

We see in Figure 4 that iGSM perform very well in the first iterates of every replication, which is a really good feature for real-time consideration, but struggles afterwards. For MSA or Polyak methods this struggle, named tail effect by Bottom (2000), arises later and so allows the algorithm to decrease a little bit more the consistency, especially on Figure (4(c)). This can be explained by the fact that these averaging methods are really designed to handle stochastic problems, allowing to perform very small steps between each iterate and so accomplishing a sort of smoothing of the iterates. This feature seems to be effective, mainly with small values of the consistency. We also see from iteration 75 that Polyak is really powerful and decreases even more the value of the objective functions, as already pointed out by Bottom (2000).

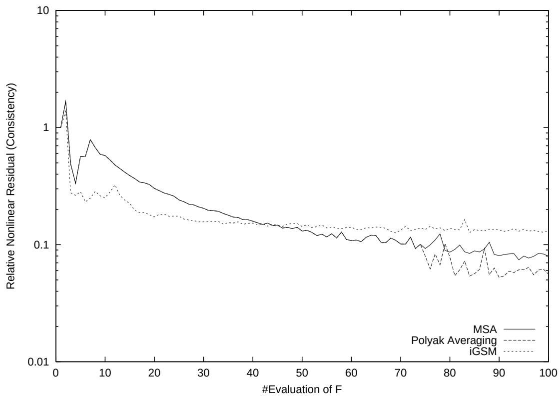
In Figure (5) we compare the solution obtain by both algorithm for Replication 1, in terms of



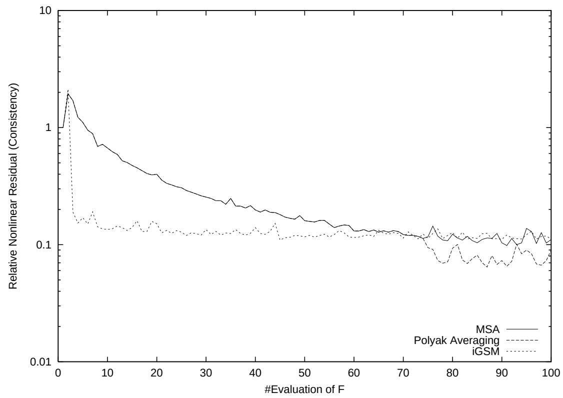
(a) Replication 1



(b) Replication 2



(c) Replication 3



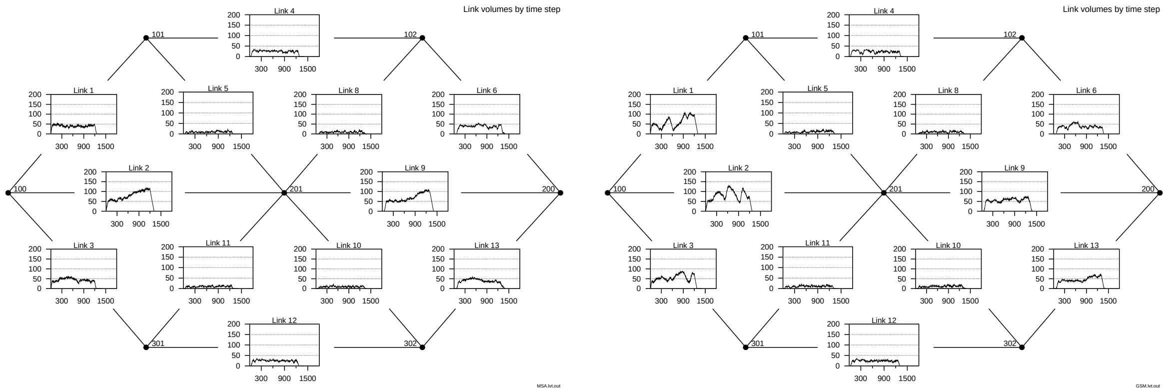
(d) Replication 4

Figure 4: Simple Simulator

density of traffic on each link during the simulation period. It is interesting to notice that the guidance generated by MSA, which has quasi the same consistency as the one generated by iGSM, seems to produce more regular flows on links 1 and 2 of the network, while at the end of the time period iGSM seems to produce smaller flow on link 9. These considerations attempt to point out that same level of consistency can be produced by different anticipatory guidances, influencing differently the future state of the network. This statement underlines the necessity of considering others criteria, in addition to the consistency, to generate consistent anticipatory route guidance.

5.2.2 DynaMIT

DynaMIT is a state-of-the-art, real-time computer system for traffic estimation prediction and generation of traveler information and route guidance. DynaMIT is the result of about 10 years of intense research and development at the Intelligent Transportation Systems Program of the Massachusetts Institute of Technology (for description and details, see (Ben-Akiva et al., 2002), (Bottom et al., 1999) and (Ben-Akiva et al., 1998)). DynaMIT is designed to operate in real time, using traffic volume and control system state data to estimate and predict time-dependent



(a) MSA (b) iGSM

Figure 5: Link Volume Trajectories

origin-destination flows and network conditions, and generating descriptive and prescriptive information that should be consistent with the predicted traffic conditions. DynaMIT's consistent guidance generation algorithm is currently the time smoothing algorithm described in section 3.1. We have implemented iGSM algorithm in order to compare results between this two methods.

Small Networks

The first network, called Florian, is a really small synthetic network composed of 10 links with only one OD pair. We simulate from 8h00 to 9h00 with interval time of 1 minute and analyze the guidance generation algorithm for the interval 8h00 to 8h30. The size of the CARG problem for this tiny network is 300 (30×10). As shown in Figure (6) the consistency is nearly reached in 3 iterations using iGSM but it needs more than 10 iterations using time smoothing algorithm to obtain the same consistency. This can be explained by the fact the initial point is the free flow traversal time table, which is a very bad starting point in this case. As TS algorithm performs an averaging between iterates the negative influence of the initial point spreads longer than with iGSM. The second network is the Central Artery/Third Harbor Tunnel network, currently under construction at Boston. It is a real medium scale network, with 211 links. The scenario contains 10 OD pairs. We simulate from 7h00 to 8h00 with interval time of 1 minute and analyze the guidance generation algorithm for the period 7h00 to 7h30 with a time interval of 1 minute. The size of the fixed point problem associated with the CARG problem is 6330 (211×30). For this network the iGSM algorithm reaches the stopping criteria, set at $10e - 5$ for this simulation, at the second iteration. The TS algorithm returns the following sequence of consistency:

Iterations	1	2	3	4	5	6	7	8	9	10	11
Consistency	318.04	7.00	3.50	1.75	0.87	0.43	0.21	0.10	0.05	0.02	0.01

Here again this huge difference of performance between these two algorithms can be explained by the fact that free flow link traversal is notably a poor starting point, that the time smoothing algorithm trail behind it all along the iterations. This real example illustrates the impact of the formulation in term of numerical efficiency. In the first iterates considering the CARG

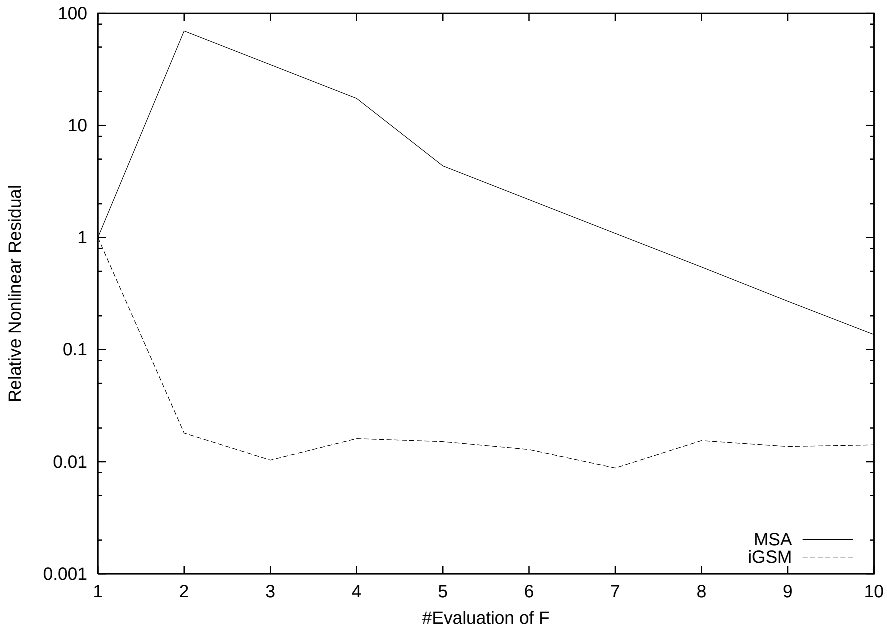


Figure 6: Florian Network

problem as solving a nonlinear system of equations clearly outperforms its classical fixed point formulation.

We already see in this two examples that the behavior of iGSM, compared to the TS algorithm, with DynaMIT is similar to the one experienced on the simple simulator compared to MSA. Indeed, iGSM algorithm behaves definitely better than algorithms of type (4) on the first iterates.

Swiss Network

As part of the project "Plan de gestion du trafic: Etude pilote pour la Suisse Occidentale" in collaboration with the engineer office RGR SA, represented by Mr Robert Grandpierre, and the Swiss Federal Roads Authority (OFROU) a calibration of DynaMIT has been achieved. This large-scale network represents the swiss highway system from Geneva to Schaffausen and is composed of 1661 links. We simulate from 7h00 to 8h15 in the morning with time interval of one minute and analyze the guidance generation for 75 minutes. The size of the fixed point problem associated with the CARG problem is 124575 (1661×75). This big network application attests of the applicability of iGSM algorithm on very large problems. One more time, as it appears in Figure 7 our methods decrease the consistency very fast during the first iterates, after which it seems to struggle. The TS algorithm reaches the same consistency about 28 iterations later. In terms of real-time applications, the fast decreasing consistency, at the beginning, associated with iGSM algorithm seems a very good alternative to averaging methods. Those preliminary results on the consistent anticipatory route guidance problem are very encouraging, principally for real-time applications, even if a deeper analysis is required to better understand the algorithm behavior as well as the problem formulation.

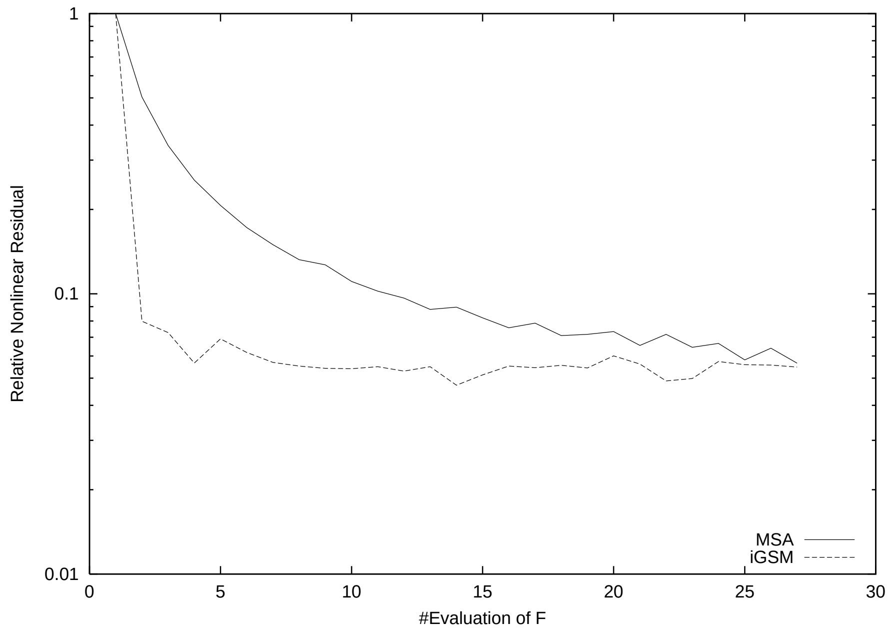


Figure 7: Swiss Network

6 Conclusion and perspectives

In this paper we have proposed an adaptation of the generalized secant method described by Bierlaire and Crittin (2003) to solve the anticipatory route guidance problem. Because GSM algorithm is not designed for large scale applications we have apply some simplifications to the method leading to an algorithm performing only matrix-vector products, a very crucial property for very large scale problems. In order to validate the algorithm's transformations we have shown its performance on standard nonlinear system of equations and compared the obtained results to classical algorithms. These results are very impressive on medium scale problems, as the efficiency of iGSM is very similar to GSM algorithms. Moreover, it clearly outperforms ICUM currently considered as one of the best method solving large scale nonlinear system of equations without derivatives and it is even able to compete with methods using derivative information as Newton-Krylov methods.

Results concerning the CARG problem have been provided using a simple traffic simulation software system that has allowed us to emphasize some characteristics of the problem revealed by the use of different algorithms. It also underlines the good behavior of iGSM in the first iterates compared to averaging methods presently used to solve the CARG problem. Finally, we presented results from reals networks using DynaMIT to demonstrate the applicability of our algorithm to very large problems. But it is clear that further numerical experiences are needed to really understand the behavior of iGSM algorithm for stochastic problems, and the CARG problem in particular.

Finally the presented numerical results also open new fields of investigations for the generation of consistent anticipatory route guidance generation itself, in particular:

- Additional characterizations of CARG problem should be investigated, like the question of the nature and interpretation of solutions supplied from different algorithms, as illustrated by the differences in terms of traffic pattern for two different guidance nevertheless with the same consistency. Moreover we can suppose that sometimes the CARG problem might have multiple approximate fixed point solutions. In that case also we need to be able to select from among various solutions using other criteria in addition to consistency.

- In this paper we have investigated only one of the three proposed composite maps of the problem, namely the composite link condition map. Other formulations need to be investigated, as for example the composite path time map, which appeared to involve the least amount of stochasticity and hopefully allows iGSM to perform better.

The results obtained with iGSM indicate that this method will be one of the first choice to solve practical large scale nonlinear system of equation in absence of derivatives information, especially due to the feature that allows to perform only matrix-vector product, that make it very cheap in terms of computation time and memory size. We have strong feelings that solving the consistent anticipatory route guidance problem with a nonlinear system of equations approach rather than a fixed point approach will permit to really speed up the process of generation of consistent guidance particularly in the context of real-time applications.

References

Arnott, R., de Palma, A. and Lindsey, R. (1991). Does providing information to drivers reduce traffic congestion?, *Transportation Research A* **25**(5): 309–318.

Banach, S. (1922). Sur les operations dans les ensembles abstraits et leur application aux equations integrales, *Fund. Math.* **3**: 133–181.

Ben-Akiva, M. and Bierlaire, M. (1999). Discrete choice methods and their applications to short-term travel decisions, *in* R. Hall (ed.), *Handbook of Transportation Science*, Kluwer, pp. 5–34.

Ben-Akiva, M., Bierlaire, M., Bottom, J., Koutsopoulos, H. N. and Mishalani, R. G. (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. N. and Mishalani, R. (2002). Real-time simulation of traffic demand-supply interactions within DynaMIT, *in* M. Gendreau and P. Marcotte (eds), *Transportation and network analysis: current trends. Miscellenea in honor of Michael Florian*, Kluwer Academic Publishers, Boston/Dordrecht/London.

Ben-Akiva, M. E., Bierlaire, M., Koutsopoulos, H. N. and Mishalani, R. G. (1998). DynaMIT: a simulation-based system for traffic prediction, *Proceedings of the DACCORD Short-Term forecasting workshop*.

Ben-Akiva, M. E., de Palma, A. and Kaysi, I. (1996). The impact of predictive information on guidance efficiency: an analytical approach, *in* L. Bianco and P. Toth (eds), *Advanced Methods in Transportation Analysis*, Springer-Verlag, pp. 413–432.

Bierlaire, M. and Crittin, F. (2001). New algorithmic approaches for the anticipatory route guidance generation problem, *Proceedings of the 1st Swiss Transportation Research Conference,* Ascona, Switzerland.

Bierlaire, M. and Crittin, F. (2003). Generalization of secant methods for solving nonlinear systems of equations, *Proceedings of the 3rd Swiss Transportation Research Conference*, Ascona, Switzerland.

Blum, J. R. (1954). Multidimensional stochastic approximation methods, *Annals of mathematical statistics* **25**(4): 737–744.

Bottom, J. (2000). *Consistent Anticipatory Route Guidance*, PhD thesis, Massachusetts Institute of Technology.

Bottom, J., Ben-Akiva, M., Bierlaire, M., Chabini, I., Koutsopoulos, H. and Yang, Q. (1999). Investigation of route guidance generation issues by simulation with DynaMIT, *in* A. Ceder (ed.), *Transportation and Traffic Theory. Proceedings of the 14th ISTTT*, Pergamon, pp. 577–600.

Broyden, C. G. (1965). A class of methods for solving nonlinear simultaneous equations, *Mathematics of Computation* **19**: 577–593.

Byrd, R., Nocedal, J. and Schnabel, R. B. (1994). Representation of quasi-Newton matrices and their use in limited memory methods, *Mathematical Programming* **63**: 129–136.

Cantarella, G. E. (1997). A general fixed-point approach to multimode multi-user equilibrium assignment with elastic demand, *Transportation Science* **31**(2): 107–128.

Dennis, J. E. and Schnabel, R. B. (1996). *Numerical methods for unconstrained optimization and nonlinear equations*, Society for Industrial and Applied Mathematics.

Dolan, E. and More, J. (2002). Benchmarking optimization software with performance profiles, *Mathematical Programming, Serie A* **91**: 2.

Gomes-Ruggiero, M. A., Martinez, J. M. and Moretti, A. C. (1991). Comparing algorithms for solving sparse nonlinear systems of equations, *Siam J. Sci. Stat Comp.* .

Hall, R. W. (1996). Route choice and advanced traveler information systems on a capacited and dynamic network, *Transporation research* **4**(5): 289–306.

Kelley, C. T. (2002). *Solving nonlinear equations with Newton's method*, Draft stage. http://www4.ncsu.edu/ ctk/newton.html.

Luksan, L. and Vlcek, J. (1998). Computational experience with globally convergent descent methods for large sparse systems of nonlinear equations, *Optimization Methods and Software* **8**: 185–199.

Martinez, J. M. (1982). Sobre dois metodos de broyden, *Matematica Aplicada e Computational* **1**.

Martinez, J. M. and Zambaldi, M. C. (1992). An inverse column-updating method for solving large-scale nonlinear systems of equations, *Optimization method and software* **1**.

Paige, C. C. and Saunders, M. A. (1982). LSQR: an algorithm for sparse linear equations and sparse least squares, *ACM Transactions on Mathematical Software* **8**: 43–71.

Polyak, B. T. and Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging, *SICON* **30**(4): 838–855.

Robbins, H. and Monro, S. (1951). A stochastic approximation method, *Annals of mathematical statistics* **22**(3): 400–407.

Ruggiero, M. G., D.N.Kozakevich and J.M.Martínez (1996). A numerical study on large-scale nonlinear solvers, *Computers and Mathematics with Applications: An International Journal* **32**: 1–13.

Sheffi, Y. and Powell, W. (1982). An algorithm for the equilibrium assignment problem with random link times, *Networks* **12**: 191–207.

Spedicato, E. and Huang, Z. (1997). Numerical experience with Newton-like methods for nonlinear algebraic systems, *Computing* **58**: 69–99.