Using ACCPM in a simplicial decomposition algorithm for the traffic assignment problem

Dulce Rosas, Jordi Castro and Lídia Montero
Statistics and Operations Research Department
Universitat Politècnica de Catalunya
Pau Gargallo 5, 08028 Barcelona (Spain)
{dulce,jcastro,lmontero}@eio.upc.es
DR 2002-04
April 2002

Using ACCPM in a simplicial decomposition algorithm for the traffic assignment problem [1]

Dulce Rosas [2 3] , Jordi Castro [2 4] and Lídia Montero [2 5]

Abstract

The purpose of the traffic assignment problem is to obtain a traffic flow pattern given a set of origin-destination travel demands and flow dependent link performance functions of a road network. In the general case, the traffic assignment problem can be formulated as a variational inequality, and several algorithms have been devised for its efficient solution. In this work we propose a new approach that combines two existing procedures: the master problem of a simplicial decomposition framework is solved through the analytic center cutting plane method. Three variants are considered for solving the master problem. The third one, which heuristically computes an appropriate initial point, has shown to outperform alternative solution methods. Some computational experience is reported in the solution of real large-scale problems, including a subset of the transportation networks of Madrid and Barcelona.

Key words. Traffic assignment problem, variational inequalities, simplicial decomposition, analytic center cutting plane method

1 Introduction

The traffic assignment problem attempts to find the distribution of the traffic flow throughout a network of routes. It is possible to formulate the problem by means of a network model that represents the physical infrastructure and aims to compute the flows of one or more commodities on the links of the network, each commodity being related to the flows for a particular origin-destination node pair.

Whenever congestion phenomena are present, the cost functional associated with the links of the network model are nonlinear and, in most applications convex or monotone. When interactions between network links are present, the problem is known as the asymmetric traffic assignment problem and it can be formulated as a variational inequality problem [5, 29].

The traffic assignment problem has received a lot of attention; partly because of its practical importance, partly because the size of real life problems makes it a challenge for algorithmic development. Many specialized strategies have been developed since [22], where an adaptation of the Frank-Wolfe method [11] was applied to its optimization formulation. Projection algorithms in the space of arc flows [6] and path flows [3] have also been applied. Another projection strategy was developed in [12]. Alternative strategies, i.e., diagonalization and linearization, were, respectively, explored in [10] and [1]. Dual cutting plane methods were proposed in [28], and applied in [24] and [25] using a gap-descent approach. Another gap-descent method was presented in [13].

Some of the most successful approaches were the simplicial and restricted simplicial decomposition algorithms introduced, respectively, in [20] and [21], and implemented for large-scale networks in [26]. Additional strategies have been suggested for solving the traffic assignment problem but only a few of them

[1] The authors are indebted to J. Barceló for his encouragement in the initial stages of this work, and to M. Denault and J.-L. Goffin for providing us with a MATLAB prototype of their ACCPM algorithm for variational inequalities

[2] Statistics and Operations Research Dept., Universitat Politècnica de Catalunya, Pau Gargallo 5, 08028 Barcelona (Spain), e-mails: {dulce,jcastro,lmontero}@eio.upc.es

[3] Author supported by a grant from the Universidad Nacional Autónoma de México

[4] Author supported by CICYT grant TAP 99-1075-C02-02

[5] Author supported by CICYT grant TIC 2000-1750-C06-03

have been based on interior-point methods. For instance, in [7] some limited computational experience was reported using small-scale traffic assignment instances with an analytic center cutting plane method (ACCPM) for variational inequalities.

The algorithm developed in this work combines the above two methods: it is based on a simplicial decomposition algorithm, but solves the resulting master problem through ACCPM. Our main goal is to solve real large-scale traffic assignment instances. For this purpose, three solution variants were considered for the master problem. The third one, which heuristically computes an initial point for ACCPM, has shown to outperform alternative solution methods in some large-scale instances.

The structure of the paper is as follows. Section 2 shows the formulation of the traffic assignment as a variational inequality problem. Section 3 outlines the ACCPM for variational inequalities. In Section 4 we develop an algorithm for the traffic assignment problem based on ACCPM and the simplicial decomposition. Section 5 reports some computational experience with an implementation of this algorithm. Finally Section 6 presents our conclusions.

2 Traffic assignment as a variational inequality problem

The modelling assumption considered in the traffic assignment problem was stated by Wardrop [31]. It postulates that the journey times on all the routes actually used are equal and less than those which would be experienced by a single vehicle on any unused route. The implication of this principle is that the routes actually used are the shortest. The traffic flows that satisfy this principle are usually referred to as *"user optimized flows"*, since each user chooses the route that he perceives the best. By contrast *"system optimized flows"* are characterized by Wardrop's second principle which states that the total travel time is minimum [8,9,28].

Beckmann [2] was the first to consider an optimization formulation of the traffic equilibrium problem and to present necessary conditions for the existence and uniqueness of equilibria. The optimization formulation exists if the partial derivatives of the link cost functions form a symmetric Jacobian. This problem is known as the symmetric traffic assignment. However, these cost functions often become nonseparable and asymmetric and a solution to the Wardrop conditions can then not be formulated as an optimization problem; instead, they are stated as variational inequality or complementarity models. This is the problem considered in this work, usually referred to as the asymmetric traffic assignment problem. We will focus on its variational inequality formulation. An excellent reference on variational inequalities can be found in [18].

We will consider an arc-path formulation on a transportation network $G = (V, A)$, V and A being a set of n nodes and m links, respectively. The nodes represent origins, destinations and intersections of links. The links represent the transportation infrastructure. The set of origin-destination (OD) node pairs will be denoted as P.

For each OD pair $p \in P$ there is a known demand $d_p > 0$ representing traffic entering the network at the origin and exiting at the destination. The demand d_p is to be distributed among a given collection K_p of simple directed paths joining the pair p.

Each directed link $a \in A$ is associated with a positive travel time, or transportation cost $F_a(y) : \mathbb{R}^m \to \mathbb{R}$, where $y \in \mathbb{R}^m$ is the vector of link flows over the entire network. The function $F(y) = (F_a(y))_{a\in A} : \mathbb{R}^m \to \mathbb{R}^m$ models the time delay for the journey on each arc a and is called the volume-delay function. $F(y)$ is monotone as a result of the congestion, i.e., it satisfies

$$[F(y') - F(y'')]^T (y' - y'') \geq 0,$$

and is continuous and differentiable.

We denote by y_a the flow of trips on a link a. Clearly $y_a = \sum_{p\in P} \sum_{k\in K_p} \delta_{ak} h_k$ for all $a \in A$, where h_k is the flow carried by the path k and

$$\delta_{ak} = \begin{cases} 1 & \text{if link } a \text{ belongs to path } k \\ 0 & \text{otherwise.} \end{cases}$$

The set of feasible flows can thus be written as

$$Y = \left\{ \begin{array}{ll} y = (y_a) \mid \exists\, h = (h_k) \geq 0 & \text{with} \quad y_a = \sum_{p \in P} \sum_{k \in K_p} \delta_{ak} h_k, \forall a \in \mathcal{A} \\ & \text{and} \quad \sum_{k \in K_p} h_k = d_p, \forall p \in P \end{array} \right\}. \tag{1}$$

The set Y accepts the following alternative node-arc formulation

$$Y = \{ \ y = \textstyle\sum_{p \in P} (y^p) \ |(y^p) = (y^p_a)_{a \in A} \in I\!R^m, N y^p = b^p, y^p \geq 0 \ \}. \tag{2}$$

(2) are the equations of a multicommodity network flow model, where $N \in I\!R^{n \times m}$ denotes the node-arc network matrix, $y^p \in I\!R^m$ the flows for commodity p, and $b^p \in I\!R^n$ the injection vector for OD pair p (i.e., $b^p_O = d_p$, $b^p_D = -d_p$ and $b^p_v = 0$ for the remaining nodes).

The traffic assignment problem can be formulated as the following variational inequality *VI(F, Y)*:

$$\text{Find } y^* \in Y \text{ such that } F(y^*)^T (y - y^*) \geq 0, \quad \forall y \in Y, \tag{3}$$

F being a continuous, monotone cost function and Y the nonempty, closed, convex subset of $I\!R^m$ defined in (1) or (2). The following gap function g associated with *VI(F, Y)* is used to measure the progress and as stopping criterion:

$$g(y) = \inf_{z \in Y} F(y)^T (z - y). \tag{4}$$

Since Y is compact and polyhedral, the "inf"can be replaced by a "min", and $g(y)$ can be evaluated by solving a linear optimization problem. In general $g(y) \leq 0$ and in particular y^* is a solution of *VI(F, Y)* if and only if $g(y^*) = 0$. In practice the point y^* is considered an ϵ_g-approximate solution if $y^* \in Y$ and $g(y^*) \geq -\epsilon_g$ for a given ϵ_g tolerance. It must be noted that (4) is equivalent to the solution of p shortest-path problems

$$\min_{z^p} F(y)^T z^p \quad \text{subject to} \quad N z^p = b^p, z^p \geq 0. \tag{5}$$

The optimal point of (4) can be computed as $z = \sum_{p \in P} z^p$.

3 ACCPM for variational inequalities

ACCPM, initially developed as a nondifferentiable optimization algorithm [15], permits solving generalized monotone variational inequalities [16]. The key idea is that under the assumptions that F is a monotone and continuous mapping and that Y is a closed, convex and nonempty set, *VI(F, Y)* can be formulated as a convex feasibility problem:

$$\text{Find a point } y^* \in Y^*,$$

where Y^* is a closed, convex and bounded set. The above result comes from the following definition and theorem, both from [23].

Definition 3.1 *Let F be a mapping. Let Y be a nonempty convex subset of $I\!R^m$. Then a weak solution to the VI(F, Y) problem, is a point y^* such that*

$$F(y)^T (y - y^*) \geq 0 \quad \forall y \in Y \tag{6}$$

Theorem 3.1 *Let Y be a nonempty, closed, convex subset of $I\!R^m$, with nonempty interior and let F be a monotone mapping with domain $dom(F)$. If $int(F) \subset dom(F) \subset Y$ then, for the variational inequality problem VI(F, Y), any weak solution is a solution and any solution is a weak solution.*

The theorem above justifies the formulation of the solution set Y^* as the intersection of an infinite number of half-spaces:

$$Y^* = \{ y^* \in Y \mid F(y)^T (y - y^*) \geq 0, \quad \forall y \in Y \} \tag{7}$$

which eventually might consist of a unique point. In other words, there is a convex feasibility formulation of *VI(F, Y)*, with the feasible set Y^* implicitly defined by the infinite family of cutting planes (6). $Y^* \subset Y$ ensures that Y^* is bounded, while (6) ensures both the convexity and closedness of Y^*.

3.1 Analytic centers

Analytic centers, formally introduced by Sonnevend [30], are defined as centers of polyhedrons. Given a set

$$Y = \{y \mid A^T y \leq c, By = d\} \tag{8}$$

and the associated dual potential function

$$\varphi_D(y) = \sum_i \ln(c_i - A_i^T y),$$

where the index i refers to the components of c and the rows of A^T, the analytic center y^c of Y is defined as the point maximizing the dual potential function over the interior of Y

$$y^c = \arg \max_{y \in int(Y)} \varphi_D(y). \tag{9}$$

Note that the feasible set for the traffic assignment problem as defined in (1) or (2) matches (8) using appropriate A and B matrices.

Problem (9) can be solved through the equivalent mathematical program

$$\begin{array}{ll} \max\limits_{y,s} & \sum\limits_i \ln s_i \\ \text{subject to} & A^T y + s = c \\ & By = d \\ & s > 0. \end{array} \tag{10}$$

The first-order KKT optimality conditions of (10) are

$$Ax + B^T \mu = 0 \tag{11}$$
$$A^T y + s = c \tag{12}$$
$$By = d \tag{13}$$
$$Xs = e \tag{14}$$
$$x, s > 0 \tag{15}$$

where x and μ are, respectively, the Lagrange multipliers associated with constraints $A^T y + s = c$ and $By = d$, (11) impose primal feasibility, (12) and (13) impose dual feasibility, (14) are the centrality conditions, (15) are the bounds of the variables, and e denotes a vector of ones of appropriate dimension. According to this notation, the analytic center lies in the dual space. As usual in interior-point methods, system (11–15) can be solved using a damped Newton method. In practice, the nonlinear complementarity conditions (14) are usually relaxed, obtaining and approximate analytic center that satisfies $\|e - Xs\| \leq \eta < 1$ for a given η tolerance. More details about the solution of (11–15) can be found in [7].

3.2 An ACCPM algorithm for variational inequalities

The algorithm outlined in this subsection was fully described in [16] and [7]. The method generates a sequence of shrinking sets Y_k that converge to the solution set (7) of *VI(F,Y)*:

$$Y_0 \supset Y_1 \supset \ldots \supset Y_k \supset Y_{k+1} \supset Y^*.$$

Each new set is obtained by adding a cutting plane to the current set. This cutting plane is computed from the analytic center of the current set, and it is used to remove a region that does not contain any solution. Algorithm 3.1 shows the main steps of this procedure.

Algorithm 3.1 *ACCPM for VI(F,Y).*

step 0: **Initialization**
$k = 0$, $Y_0 = Y$
step 1: **Analytic center**
Find an approximate analytic center y_k *of* Y_k
step 2: **New cut**
$Y_{k+1} := Y_k \bigcap \{y \mid F(y_k)^T y \leq F(y_k)^T y_k\}$
step 3: **Termination Criterion**
Compute gap $g(y_k)$
if $g(y_k) \geq -\epsilon_g$ *then*
stop: y_k *is a solution of VI(F,Y)*
else
$k := k + 1$ *and return to step 1*

A comprehensive explanation of the above procedure and its convergence properties can be found in [7,16,27].

4 ACCPM in a simplicial decomposition framework for variational inequalities

There are two possible approaches for solving (3) using ACCPM. The first one is to apply Algorithm 3.1 to (3), considering the node-arc formulation (2) of the feasible set. As already done in specialized interior-point methods [4], the multicommodity structure of the problem should be exploited. This procedure is under development by the authors for large-scale problems. The second approach is to use ACCPM within an alternative solution method for (3). This was the choice adopted in this work. We solved (3) through a simplicial decomposition for variational inequalities (SDVI) algorithm, using ACCPM in the solution of the master problem that appears at each SDVI iteration. This master problem is itself a reduced variational inequality. For optimization problems, ACCPM has already been successfully applied in the master of alternative decomposition approaches [14,17].

4.1 Simplicial decomposition algorithm

The SDVI algorithm, originally applied to the traffic assignment problem in [20], is a column generation method where feasible flows are written as convex combinations of the extreme points of Y. Let $E \in \mathbb{R}^{m \times t}$ be a matrix with all the t extreme flows of Y. Feasible flows can be written as a convex combination of the columns of E, i.e.,

$$y = E\lambda, \qquad \lambda \in \Lambda$$

where

$$\Lambda = \{\lambda \mid \sum_{i=1}^{t} \lambda_i = 1,\ \lambda_i \geq 0\}. \tag{16}$$

The traffic assignment problem (3) can thus be rewritten as

$$\text{Find } \lambda^* \in \Lambda \text{ such that } (F(E\lambda^*)^T E)(\lambda - \lambda^*) \geq 0, \quad \forall \lambda \in \Lambda. \tag{17}$$

Since enumerating all the t extreme flows is impractical, the SDVI algorithm considers an initial set of them and generates new ones as needed at each iteration. Algorithm 4.1 outlines the main steps of this procedure.

Algorithm 4.1 *A generic SDVI algorithm for* (17).

step 0: **Initialization**
$k = 0$, E_0 *matrix with initial set of* t_0 *extreme points*
step 1: **Find y_k, solution of the master problem** $VI(F, H(E_k))$
where $H(E_k) = \{y | y = E_k\lambda, \lambda \in \Lambda_k\}$, Λ_k *defined as in* (16) *for* $t = t_k$
step 2: **Find the new extreme point** z_k
where z_k *is the solution of the gap* $g(y_k)$ *defined in* (4)
step 3: **Stopping criteria**
If $g(y_k) \geq -\epsilon_g$ *then stop:* y_k *is a solution of VI(F,Y)*
step 4: **Add the new extreme point**
$E_{k+1} := E_k \cup \{z_k\}$
$k := k+1$ *and return to step 1*

Comprehensive descriptions of the simplicial decomposition method can be found in [20] and [19] for variational inequalities—asymmetric traffic assignment—and nonlinear optimization—symmetric traffic assignment—problems, respectively.

4.2 Master problem through ACCPM

The main computational burden of Algorithm 4.1 is the solution of the master problem—a reduced variational inequality—at step 1, which can be solved by any effective method. In the past, projection methods [3] were considered as an efficient choice [20, 26]. In this work we applied ACCPM, which means adapting Algorithm 3.1—originally formulated in the space of flows—to work in the space of λ's. The master problem to be solved through ACCPM at iteration k of Algorithm 4.1, denoted as $VI(\widetilde{F}, \Lambda_k)$, can thus be stated as

$$\text{find } \lambda^* \text{ such that } \widetilde{F}(\lambda^*)(\lambda - \lambda^*) \geq 0 \quad \forall\, \lambda \in \Lambda_k, \quad \text{where } \widetilde{F}(\lambda) = (F(E_k\lambda)^T E_k), \tag{18}$$

where $y_k = E_k\lambda^*$. Algorithm 4.2 details the main steps to be performed:

Algorithm 4.2 *Detail of step 1 of Algorithm* 4.1 *through ACCPM*

step 1: **Find y_k, solution of the master problem** $VI(F, H(E_k))$ **through** (18)
(i) **Initialization**
$j = 0$, $\Lambda_0 = \Lambda_k$
(ii) **Analytic center**
Find an approximate analytic center λ_j *of* Λ_j
(iii) **New cut**
$\Lambda_{j+1} := \Lambda_j \bigcap \{\lambda \mid \widetilde{F}(\lambda_j)^T\lambda \leq \widetilde{F}(\lambda_j)^T\lambda_j\}$
(iv) **Termination criterion**
Compute gap $g(\lambda_j) = \min_{z \in \Lambda_k} \widetilde{F}(\lambda_j)^T(z - \lambda_j)$
if $g(\lambda_j) \geq -\epsilon_g$ *then*
$y_k = E_k\lambda_j$ *and go to step 2 of Algorithm* 4.1
else
$j := j+1$ *and return to step (ii)*

Iterations of Algorithm 4.1 are usually named "major iterations", whereas those of Algorithm 4.2 are referred to as "minor iterations".

Problem (18) was solved considering three variants of Algorithm 4.2. The first two obtain an initial feasible point at step *(ii)* for $j = 0$ by performing primal-dual Newton iterations. They differ in the representation of the feasible set Λ_k. The third variant considers the same representation of the feasible set than second variant, but heuristically computes an initial feasible point at step *(ii)* for $j = 0$. For all three variants and $j > 0$, the last center λ_{j-1} was used as a warm start at step *(ii)*, performing additional primal-dual Newton steps to recover both feasibility and centrality [7].

4.2.1 First variant

In the first variant the initial feasible set Λ_k, defined as in (16) considering $t = t_k$, is represented as

$$\Lambda_k = \left\{ \lambda \mid A^T \lambda \leq c,\ B\lambda = 1 \right\}, \tag{19}$$

where $A^T = -I_{t_k} \in I\!R^{t_k \times t_k}$ is the minus identity matrix, $c \in I\!R^{t_k}$ is a zero vector, and $B \in I\!R^{1 \times t_k}$ is a vector of ones. The new inequalities computed at step *(iii)* of Algorithm 4.2 will be successively added to matrix A_j^T (initially $A_0^T = A^T$). At iteration j the dimension of A_j^T is $(t_k + j) \times t_k$.

This representation of the feasible set clearly matches (8), replacing y by λ. It can be shown (see [7] for details) that if we solve the optimality conditions (11–15) of (10), each Newton iteration involves linear systems with

$$\Delta = A_j S^{-1} X A_j^T \qquad \text{and} \qquad H = B \Delta^{-1} B^T, \tag{20}$$

where S and X are diagonal positive definite matrices derived from s and x. Δ has dimension $t_k \times t_k$, independent of the possible large number of cuts generated. The solution of the linear systems is performed by dense Cholesky factorization of Δ because of the density of the j new cuts added to A_j^T. To compute the scalar H we need to perform an additional backward and forward substitution with the factorization of Δ.

4.2.2 Second variant

In the second variant the equality constraints of Λ_k are duplicated into two inequalities as follows

$$\begin{aligned} \Lambda_k &= \left\{ \lambda \mid A^T \lambda \leq c \right\} \\ &= \left\{ \lambda \mid \begin{pmatrix} -I_{t_k} \\ B \\ -B \end{pmatrix} \lambda \leq \begin{pmatrix} \mathbf{0} \\ 1 \\ -1 \end{pmatrix} \right\}, \end{aligned} \tag{21}$$

where I_{t_k} is the identity matrix of dimension t_k, $\mathbf{0}$ is a zero vector of dimension t_k, and $B \in I\!R^{1 \times t_k}$ is a vector of ones. At iteration j the dimension of matrix A_j^T (initially $A_0^T = A^T$) is $(t_k + 2 + j) \times t_k$. This representation of the feasible set again matches (8), removing constraints $By = d$ and considering variables λ. To compute the analytic center of Λ_j we have to solve problem (10) without constraints $By = d$. The optimality conditions of this problem are a subset of (11–15), i.e.,

$$Ax = 0 \tag{22}$$
$$A^T \lambda + s = c \tag{23}$$
$$Xs = e \tag{24}$$
$$x, s, > 0. \tag{25}$$

The solution of (22–25) through Newton iterations involves systems of equations with matrix

$$\Delta = A_j S^{-1} X A_j^T.$$

This is the same matrix that for the first variant. This second variant saves the computation of H in (20).

It is important to note that this second variant can not provide a strictly feasible analytic center. Indeed, the interior of the feasible set $\Lambda_k = \left\{ \lambda, s \geq 0 \mid A^T \lambda + s = c \right\}$ is empty, and system (22–25) is infeasible. To overcome this inconvenient we used a feasibility tolerance ϵ in the range $[10^{-6}, 10^{-5}]$ when performing the primal-dual Newton iterations. This can be seen as finding a center λ such that $1 - \epsilon < \sum_{i=1}^{t_k} \lambda_i < 1 + \epsilon$. As it will be discussed later, the use of this feasibility tolerance did not have a great repercussion in the quality of the solution found, for the traffic assignment problem.

4.2.3 Third variant

The third variant also represents the feasible set by (21) and the optimality conditions of its analytic center are (22–25). However, unlike the first two variants, that rely on primal-dual Newton iterations, the starting point is heuristically obtained as

$$\begin{array}{rcll} \lambda_i & = & 1/t_k & i = 1,\ldots,t_k \\ s_i & = & 1/t_k & i = 1,\ldots,t_k \\ x_i & = & t_k & i = 1,\ldots,t_k \\ s_i & = & \epsilon & i = t_k+1, t_k+2 \\ x_i & = & 1/\epsilon & i = t_k+1, t_k+2, \end{array} \tag{26}$$

for a fixed $\epsilon > 0$ tolerance. The above point satisfies

$$\begin{array}{rcl} Ax & = & t_k e \\ A^T\lambda + s & = & c + \epsilon(\mathbf{0}^T, 1, 1)^T \\ Xs & = & e \\ x, s, & > & 0, \end{array} \tag{27}$$

where $\mathbf{0}$ and e are vectors of dimension t_k of, respectively, zeros and ones. Considering a feasibility tolerance of ϵ, this point satisfies the dual feasibility optimality condition (23), and it can be considered a dual ϵ-feasible starting point.

Equations (27) are approximately the optimality conditions (22–25), but for the primal feasibility. Indeed, it can be easily proved that (27)—setting $\epsilon = 0$—are the optimality conditions of the perturbed analytic center problem

$$\begin{array}{ll} \max\limits_{\lambda, s} & \sum\limits_{i=1}^{t_k+2} \ln s_i + t_k e^T \lambda \\ \text{subject to} & A^T\lambda + s = c \\ & s > 0. \end{array} \tag{28}$$

(26) can thus be considered a fairly good approximation to the analytic center. In practice this variant provided by far the best computational results.

Through the ϵ feasibility parameter we can perform a trade-off between the quality of the solution and the computation time. Small values (e.g., $\epsilon = 10^{-7}$) provide (almost) the exact solution of (18) but large execution times. Values about 10^{-2} (which is the default of the implementation developed) have shown to provide good enough approximate solutions very efficiently. Such large feasibility tolerances were not appropriate for the previous second variant: execution times did not reduce, even some numerical instabilities were found. However, in combination with the heuristically computed initial point, they provided the fastest execution times.

The use of this ϵ feasibility parameter means that we can provide solutions where $\sum_{i=1}^{t_k} \lambda_i \neq 1$ (indeed, the constraints impose $1 - \epsilon < \sum_{i=1}^{t_k} \lambda_i < 1 + \epsilon$). Therefore, the point y_k computed in the master problem of Algorithm 4.1—which eventually will be reported as the solution of the traffic assignment problem—only satisfies approximately the demands for the different OD pairs. In this sense, we can state that we are solving a traffic assignment problem with slightly perturbed demands at the OD pairs. However, since in practice OD demands are approximations of the real unknown values, the solution of a perturbed-demands problem can be acceptable. Moreover, as shown in next Section, we empirically observed that the patterns of flows of the approximate solution are similar to those of the exact solution. This third variant can thus be seen as a fast method for computing approximations of the main patterns of flows in the traffic assignment problem, which is of great interest in practice. Moreover, the balance between the quality of the solution and the performance through the ϵ feasibility parameter makes the method a versatile tool.

5 Computational results

The three ACCPM variants of the previous Section have been implemented in C and included in the Fortran code for large-scale traffic assignment problems described in [26]. That code implements a restricted version of the simplicial decomposition algorithm. It solves the master problem through the linear projection method, which can be considered the standard solution procedure in the traffic assignment context [3, 20].

We considered the transportation networks of Sioux Falls and Winnipeg, and a subset of the transportation networks of Barcelona and Madrid. Table 1 reports the dimensions of these networks, e.g., number of nodes, links and OD node pairs. Column "centroids" gives the number of nodes with nonzero demands/supplies. Columns "Variables" and "Constraints" show the overall number of variables and constraints of the associated multicommodity network model (note that "variables"= "links" $\times$ "OD pairs", and "constraints"= "nodes" $\times$ "OD pairs").

Problem	Nodes	Centroids	Links	OD pairs	Constraints	Variables
Sioux Falls	48	24	124	528	25344	65472
Barcelona	930	110	2522	7922	7367460	19979284
Winnipeg	1017	154	2976	4345	4418865	12930720
Madrid	2776	490	6871	26037	72278712	178900227

Table 1: *Test networks dimensions*

For each network of Table 1 we developed three different categories of traffic assignment instances: a diagonal, a symmetric and an asymmetric problem. Diagonal problems involve separable cost functions (e.g., the Jacobian of the travel cost function $F(y)$ is diagonal). Symmetric problems consider symmetric link interactions (if link a interacts with link b, then b is also assumed to interact with a). The Jacobian of the cost function $F(y)$ is thus symmetric but not diagonal. The asymmetric problems (i.e., Jacobian of $F(y)$ asymmetric) were artificially built by including additional link interactions.

We used a general form of the standard BPR (Bureau of Public Roads) cost function. It provides the journey time for each link of the network. For a diagonal problem, it can be written as

$$F_a(y_a) = t_0 \left(1 + \alpha \left(\frac{y_a}{c_a}\right)^{\beta}\right),$$

where c_a is the capacity of link a, and t_0 is the travel time through this link when it is empty (zero flow). Parameters α and β were set to the standard values of, respectively, 0.15 and 4.

For real world instances, the estimation of exact asymmetric cost functions is a difficult task. We then generated asymmetric problems by adding interactions between incoming links at junctions. For each link a we considered the following asymmetric cost function:

$$F_a(y) = t_0 \left(1 + \alpha \left(\frac{\sum_{b \in I_a} w_{ab} y_b}{c_a}\right)^{\beta}\right),$$

where $t_0 = 60 \cdot l_a$, l_a being the link length, I_a is the set of links interacting with link a, and w_{ab} are the weight interaction factors between links a and b ($w_{aa} = 1$).

Tables 2–4 report the results obtained for, respectively, diagonal, symmetric and asymmetric instances. Columns "SIO", "BCN", "WIN" and "MAD" show the results for the transportation networks of Sioux Falls, Barcelona, Winnipeg and Madrid, respectively. Column "Master" gives the method used for the solution of the master problem: the standard linear projection method ("LPM"), and the three variants based on ACCPM described in previous Section ("ACCPM-V1", "ACCPM-V2" and "ACCPM-V3"). The ϵ feasibility parameter of the third ACCPM variant was set to 10^{-2} for all the instances. For each transportation network and solution method the following information is provided. Rows "initial gap" and "final gap" show the relative gap for, respectively, the first and last major iterations (thus, "final gap" is the gap of the solution provided). Row "major it." gives the number of major iterations performed. Row "minor it." provides the average number of minor iterations required for the master problems. Row "$\max\{t_k\}$" is the maximum number of extreme points considered in the simplicial decomposition procedure (i.e., maximum dimension of the master problems). "Global-CPU" gives the total execution time in seconds. "M.P.-CPU" is the execution time spent in the solution of the master problems, in seconds. All the runs were carried on a Sparc Sun-4 workstation with a 198 MHz CPU.

		Transportation network			
	Master	SIO	BCN	WIN	MAD
initial gap	LPM	.182e+03	.669e+04	.369e+02	.126e+04
	ACCPM-V1	.182e+03	.669e+04	.369e+02	.126e+04
	ACCPM-V2	.182e+03	.669e+04	.369e+02	.126e+04
	ACCPM-V3	.180e+03	.636e+04	.359e+02	.122e+04
final gap	LPM	.875e+00	.989e+00	.832e+00	.897e+00
	ACCPM-V1	.875e+00	.989e+00	.832e+00	.897e+00
	ACCPM-V2	.706e+00	.990e+00	.938e+00	.936e+00
	ACCPM-V3	.925e+00	.999e+00	.880e+00	.956e+00
major it.	LPM	29	81	16	41
	ACCPM-V1	29	81	16	41
	ACCPM-V2	30	85	15	41
	ACCPM-V3	20	54	10	28
minor it.	LPM	4.24	3.09	3.18	3.34
	ACCPM-V1	180.17	462.76	111.5	247.12
	ACCPM-V2	99.53	216.41	58.53	112.71
	ACCPM-V3	18.35	21.72	7.8	14.33
$\max\{t_k\}$	LPM	31	83	18	43
	ACCPM-V1	31	83	18	43
	ACCPM-V2	32	87	17	43
	ACCPM-V3	22	56	12	30
Global-CPU	LPM	0.5	201.9	7.4	342.9
	ACCPM-V1	484.1	361764.5	79.1	3940.5
	ACCPM-V2	105.0	27740.8	29.0	1231.8
	ACCPM-V3	1.3	60.7	4.7	202.2
M.P.-CPU	LPM	0.5	186.6	3.3	111.2
	ACCPM-V1	484.0	361748.9	75.1	3713.8
	ACCPM-V2	104.9	27724.2	25.1	996.4
	ACCPM-V3	1.3	49.8	1.7	33.7

Table 2: *Results for the diagonal traffic assignment problems*

From Tables 2–4 it can be concluded that the linear projection method keeps on being competitive compared to the first two ACCPM variants. However, the third ACCPM variant provides significant better execution times for the largest and most difficult instances. Although it performed, on average, more minor iterations than the linear projection method, it required much less major iterations to reach a solution. This good behaviour was observed for the three categories of instances (diagonal, symmetric and asymmetric).

As stated before, the ϵ feasibility parameter of the third ACCPM variant can be used to balance efficiency and accuracy. To show this fact, we solved the diagonal Winnipeg problem for several values of ϵ. Table 5 reports the results obtained, for the third ACCPM variant with different ϵ values (columns"ACCPM-V3") and for the linear projection method (column "LPM"). Row "$\mathrm{Obj}(y^*)$" provides the objective function value of the equivalent optimization problem formulation. The objective value of column "LPM" is assumed to be that of the optimal solution. Rows "major it." and "minor it." show the major and average minor iterations, respectively. Row "Global-CPU" gives the overall execution time. Clearly, the smaller the ϵ, the better the objective cost of the solutions provided by ACCPM. On the other hand, execution times

	Master	SIO	BCN	WIN	MAD
		Transportation network			
initial gap	LPM	.339e+03	.505e+04	.201e+0	.482e+03
	ACCPM-V1	.339e+03	.505e+04	.201e+02	.482e+03
	ACCPM-V2	.339e+03	.505e+04	.201e+02	.482e+03
	ACCPM-V3	.334e+03	.483e+04	.192e+02	.472e+03
final gap	LPM	.914e+00	.997e+00	.914e+00	.984e+00
	ACCPM-V1	.914e+00	.984e+00	.914e+00	.983e+00
	ACCPM-V2	.958e+00	.945e+00	.846e+00	.929e+00
	ACCPM-V3	.873e+00	.972e+00	.862e+00	.901e+00
major it.	LPM	25	84	18	41
	ACCPM-V1	25	87	18	41
	ACCPM-V2	25	91	16	42
	ACCPM-V3	19	54	10	26
minor it.	LPM	3.36	3.36	3	3.29
	ACCPM-V1	158.28	490.19	122.0	242.19
	ACCPM-V2	88.48	226.67	61.0	123.69
	ACCPM-V3	18.52	23.12	7.5	12.11
$\max\{t_k\}$	LPM	27	86	20	43
	ACCPM-V1	27	89	20	43
	ACCPM-V2	27	93	18	44
	ACCPM-V3	21	56	12	28
Global-CPU	LPM	0.9	741.8	23.0	852.3
	ACCPM-V1	251.4	511172.5	178.8	5436.5
	ACCPM-V2	55.0	27852.8	63.6	2058.1
	ACCPM-V3	1.6	118.8	9.5	376.5
M.P.-CPU	LPM	0.8	710.1	14.1	369.0
	ACCPM-V1	251.3	511139.4	169.9	4973.1
	ACCPM-V2	54.8	27817.2	55.3	1563.1
	ACCPM-V3	1.5	96.5	3.5	57.2

Table 3: *Results for the symmetric traffic assignment problems*

	Master	SIO	BCN	WIN	MAD
		Transportation network			
initial gap	LPM	.427e+02	.156e+04	.249e+03	.154e+05
	ACCPM-V1	.427e+02	.156e+04	.249e+03	.154e+05
	ACCPM-V2	.427e+02	.156e+04	.249e+03	.154e+05
	ACCPM-V3	.412e+02	.148e+04	.240e+03	.147e+05
final gap	LPM	.887e+00	.984e+00	.769e+00	.922e+00
	ACCPM-V1	.887e+00	.984e+00	.769e+00	.922e+00
	ACCPM-V2	.837e+00	.882e+00	.768e+00	.876e+00
	ACCPM-V3	.502e+00	.910e+00	.757e+00	.775e+00
major it.	LPM	12	69	11	39
	ACCPM-V1	12	69	11	39
	ACCPM-V2	12	66	11	41
	ACCPM-V3	11	45	8	31
minor it.	LPM	6.5	8.35	3.9	7.25
	ACCPM-V1	89.33	379.21	86.0	238.07
	ACCPM-V2	45.16	172.85	48.45	131.42
	ACCPM-V3	16.1	34.87	11.25	34.19
$\max\{t_k\}$	LPM	14	71	13	41
	ACCPM-V1	14	71	13	41
	ACCPM-V2	14	68	13	43
	ACCPM-V3	13	47	10	33
Global-CPU	LPM	0.2	533.1	6.6	610.8
	ACCPM-V1	11.3	127616.9	33.8	3185.1
	ACCPM-V2	3.5	6065.4	17.6	1280.5
	ACCPM-V3	0.3	77.0	5.1	270.0
M.P.-CPU	LPM	0.2	518.5	3.0	403.0
	ACCPM-V1	11.0	127602.5	30.4	2977.2
	ACCPM-V2	3.4	6050.5	14.1	1058.7
	ACCPM-V3	0.3	66.7	2.1	104.7

Table 4: *Results for the asymmetric traffic assignment problem*

tend to considerably increase for small values. However, for $\epsilon = 10^{-2}$ we already obtained a solution with a good enough objective value—the relative error is $1.5 \cdot 10^{-3}$—in a fraction of the time required by the linear projection method.

	ACCPM-V3				LPM
	$\epsilon = 10^{-2}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-7}$	
major it.	10	16	16	16	16
minor it.	7.8	42.87	79.18	96.5	3.18
GLOBAL-CPU	4.7	22.2	46.1	61.8	7.4
Obj(y^*)	704207.8	705252.4	705277.1	705277.2	705277.2

Table 5: *accuracy vs. efficiency for the diagonal Winnipeg instance*

It could be argued that the good behaviour of the third ACCPM variant shown in Table 5 is merely due to the use of a greater feasibility and optimality tolerances than the linear projection method. However, the linear projection method did not perform better when such tolerances were relaxed. Similar execution times were obtained when its optimality tolerance was increased by four orders of magnitude. Imposing $\sum_{i=1}^{t_k} \lambda_i = 0.99$ (which is the relaxed value used by ACCPM for $\epsilon = 10^{-2}$), we got some convergence problems. Only the diagonal Winnipeg problem could be solved in 9.5 seconds, against the 7.4 seconds required when $\sum_{i=1}^{t_k} \lambda_i = 1$. Moreover the objective function value obtained for the equivalent optimization problem was $\text{Obj}(y^*) = 699502.8$, much farther from the optimal cost of 705277.2 than the value of 704207.8 reported by the third ACCPM variant for $\epsilon = 10^{-2}$.

We also compared the link flows of the optimal solutions provided by the third variant of ACCPM and the linear projection method. For the diagonal Winnipeg problem, Figure 1 shows the linear regression between the optimal flows provided by both methods. y_L are the optimal flows of the linear projection method, while y_A are those of the third ACCPM variant using $\epsilon = 10^{-2}$. Zero flows were removed and are not showed. From Figure 1 it is clear that both optimal flows have a large positive correlation. Moreover, the slope of the regression equation was 1.007, which means that the flow values are very similar.

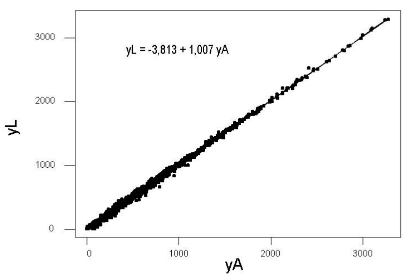

Figure 1: *Comparison of LPM and ACCPM-V3 optimal flows by regression.*

As an additional test of the quality of the solution reported by the third ACCPM variant, the difference of the above two optimal flows (i.e., diagonal Winnipeg problem and $\epsilon = 10^{-2}$) was compared to their average value. Figure 2 shows the results obtained. Clearly, discrepancies in link flows tend to decrease as the link flows increase. This means that, for the main links of the transportation network (those with large flows), differences in the solutions reported by both methods are expected to be small. To refine the analysis, the average of the link flows were divided into groups and the relative error $|y_{L_a} - y_{A_a}|/((y_{L_a} + y_{D_a})/2) \cdot 100$ for each link a was computed. Table 6 shows, for each group, the number of flows (column "No."), the mean relative error (column "mean"), and the 75% percentile (column "Q3"). For the largest flows, the relative errors were not greater than 1%. Again this means that, as for the main links of the transportation network, both the linear projection method and the third ACCPM variant provide similar flow patterns.

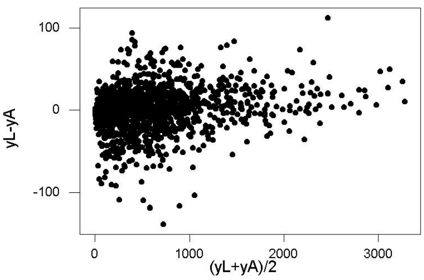

Figure 2: *Comparison of flow differences and average flows.*

Flows group	No.	Mean	Q3
0 – 500	1330	10%	12%
500 – 1000	514	3%	4%
1000 – 1500	163	2%	2%
1500 – 2000	61	1%	2%
2000 – 3200	49	1%	1%

Table 6: *Relative errors for link flows groups*

6 Conclusions

From the computational experience reported, it can be stated that the solution of the master problem of a simplicial decomposition algorithm through the third ACCPM variant is an efficient method for computing very good approximate solutions to large scale traffic assignment problems. Moreover, through the ϵ feasibility parameter we can control the trade-off between quality of the solution and computation time, which makes the method a versatile tool.

Among the related tasks that are being performed, we mention the solution of the variational inequality formulation of the traffic assignment problem through ACCPM. Obtaining the real perturbed OD demands considered by the third ACCPM variant is also one of the additional tasks to be developed.

References

[1] H. Z. Aashtiani and T. L. Magnanti, "A linearization and decomposition algorithm for computing urban traffic equilibria", in Proceedings of the IEEE Large Scale Systems Symposium, 1982, pp. 8–19.

[2] M. J. Beckmann, C. McGuire and C. Wisten, "Studies in the economics of transportation", 1956.

[3] D. P. Bertsekas and E. M. Gafni, "Projection methods for variational inequalities with application to the traffic assignment problem", Mathematical Programming Study, vol. 17, pp. 139–159, 1982.

[4] J. Castro, "A specialized interior-point algorithm for multicommodity network flows", SIAM Journal on Optimization, vol. 10, no. 3, pp. 852–877, 2000.

[5] S. Dafermos, “Traffic equilibrium and variational inequalities”, Transportation Science, vol. 14, pp. 42–54, 1980.

[6] S. Dafermos, “Relaxation algorithms for the general asymmetric traffic equilibrium problem”, Transportation Science, vol. 16, pp. 231–240, 1982.

[7] M. Denault and J. L. Goffin, “On a primal-dual analytic center cutting plan method for variational inequalities”, Computational Optimization and Applications, vol. 12, pp. 127–156, 1999.

[8] M. Florian, “An introduction to network models used in transportation planning”, In M. Florian, ed. Transportation Planning Models. North-Holland, pp. 137–152, 1984.

[9] M. Florian, “Nonlinear cost network models in transportation analysis”, Mathematical Programming Study, vol. 26, pp. 167–196, 1986.

[10] M. Florian and H. Spiess, “The convergence of diagonalization algorithms for asymmetric network equilibrium problems”, Transportation Research, vol. 16B, pp. 477–483, 1982.

[11] M. Frank and P. Wolfe, “An algorithm for quadratic programming”, Naval Research Logistics Quarterly, vol. 3, pp. 95–110, 1956.

[12] M. Fukushima, “A relaxed projection method for variational inequalities”, Mathematical Programming, vol. 35, pp. 58–70, 1986.

[13] M. Fukushima, “Equivalent differentiable optimization problems and descent methods for asymmetric variational inequalities”, Mathematical Programming, vol. 53, pp. 99–110, 1992.

[14] J.-L. Goffin, J. Gondzio, R. Sarkissian and J.-P. Vial, “Solving nonlinear multicommodity flow problems by the analytic center cutting plane method”, Math. Programming, 76, pp. 131–154, 1996.

[15] J.-L. Goffin, A. Haurie and J.-P. Vial, “Decomposition and nondifferentiable optimization with the projective algorithm”, Management Science, vol. 38-2, pp. 284–302, 1992.

[16] J.-L. Goffin, P. Marcotte and D. Zhu, “An analytic cutting plane method for pseudo-monotone variational inequalities”, Operations Research Letters, vol. 20, pp. 1–6, 1997.

[17] J. Gondzio, R. Sarkissian and J.-P. Vial, “Using an interior point method for the master problem in a decomposition approach”, European Journal of Operational Research,vol. 101, pp. 577–587, 1997.

[18] P. Harker and J.-S. Pang, “Finite dimensional variational inequality and nonlinear complementarity problems: a survey of theory, algorithms and applications”, Mathematical Programming, vol. 48, pp. 161–220, 1990.

[19] D. W. Hearn, S. Lawphongpanich and J. A. Ventura, “Restricted simplicial decomposition: computation and extensions”, Mathematical Programming Study, vol. 31, pp. 99–118, 1987.

[20] S. Lawphongpanich and D. W. Hearn, “Simplicial decomposition of the asymmetric traffic assignment problem”, Transportation Research, vol. 18B, pp. 123–133, 1984.

[21] S. Lawphongpanich and D. W. Hearn, “Restricted simplicial decomposition with application to the traffic assignment problem”, Ricera Operativa, vol. 38, pp. 97–120, 1986.

[22] L. J. LeBlanc, E. K. Morlok and W. Pierskalla, “An efficient approach to solving the road network equilibrium traffic assignment problem”, Transportation Research, vol. 9, pp. 309–318, 1974.

[23] C. Lemaréchal, A. Nemiroskii and Y. Nesterov, “New variants of bundle methods”, Tech. Rep. No. 1508, INRIA-Rocquencourt, 1991.

[24] P. Marcotte, “A new algorithm for solving the variational inequalities with application to the traffic assignment problem”, Mathematical Programming Study, vol. 33, pp. 339–351, 1985.

[25] P. Marcotte and J.-P. Dussault, “A modified newton method for solving variational inequalities”, in Proceedings of the 24th IEEE conference on Decision and Control, 1985, vol. 33, pp. 1433–1436.

[26] L. Montero, A Simplicial Decomposition Approach for Solving the Variational Inequality Formulation of the General Traffic Assignment Problem for Large Scale Networks. PhD thesis, Universitat Politècnica de Catalunya, Barcelona-Spain, 1992.

[27] Y. Nesterov and J.-P. Vial, “Homogeneous analytic center cutting plane methods for convex problems and variational inequalities”, Logilab Technical Report, vol. 4, 1997.

[28] S. Nguyen and C. Dupuis, “An efficient method for computing traffic equilibria in networks with asymmetric transportation costs”, Transportation Science, vol. 18, pp. 185–202, 1984.

[29] M. J. Smith, “Existence, uniqueness and stability of traffic equilibria”, Transpn Res., vol. 13B, pp. 295–304, 1979.

[30] G. Sonnevend, “New algorithms in convex programming based on a notion of “centre”(for systems of analytic inequalities) and on rational extrapolation”, Trends in Mathematical Optimization, pp. 311–326, 1988.

[31] J. G. Wardrop, “Some theoretical aspects of road traffic research”, in Proceedings of the Institute of Civil Engineers, Part II, 1952, vol. 1, pp. 325–378.