Solving the traffic assignment problem using ACCPM

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 2002-18
November 2002

Solving the traffic assignment problem using ACCPM

Dulce Rosas, Jordi Castro and Lídia Montero

Statistics and Operations Research Department, Universitat Politècnica de Catalunya,

Pau Gargallo 5, 08028 Barcelona(Spain)

e-mails: {dulce,jcastro,lmontero}@eio.upc.es

Abstract

The purpose of the traffic assignment problem is to obtain a flow pattern, given a set of origin-destination travel demands and flow dependent link performance functions of an urban transportation network. In the general case, the traffic assignment problem can be formulated as a variational inequality problem. Several algorithms have been devised for its efficient solution, but only a few of them have been based on interior-point methods. In this work we propose a new approach that uses the analytic center cutting plane method (ACCPM). Two variants are presented. The first variant directly applies ACCPM to the multicommodity formulation and considers several approaches for computing the Newton direction. In the second variant, at each iteration a different cut is added for each commodity. This disaggregated procedure accurately solved medium sized problems. Some computational experience is also reported.

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

Introduction

The traffic assignment or network equilibrium model is used to predict the steady-state volume of traffic on urban transportation networks. It is possible to formulate the problem as 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 of particular groups of origin-destination pairs.

Whenever congestion phenomena are present, the cost function 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 can be formulated as a variational inequality problem (see Smith, 1979 and Dafermos, 1980).

The traffic assignment problem is of great interest; partly because of its practical importance and partly due to the fact that the size of real life problems makes algorithmic development so challenging. This interest explains the many specialized algorithms that have been developed since LeBlanc et al., 1975 proposed an algorithm to solve the optimization formulation of this problem when no interactions are present.

For the solution of the traffic assignment problem, which do not have equivalent optimization problems, Dafermos, 1982 proposed the projection algorithm in the space of arc flows while Bertsekas and Gafni, 1982 applied it in the space of path flows. Nguyen and Dupuis, 1984 proposed the dual cutting plane method. Marcotte, 1985 also used a cutting plane approach in developing a *gap*-descent method.

Another class of iterative methods which have been successfully applied to the variational inequality problem with compact polyhedral feasible sets are the simplicial decomposition (Lawphongpanich and Hearn, 1984) and restricted simplicial decomposition (Lawphongpanich and Hearn, 1986) algorithms.

However, only a few of the above approaches have been based on interior-point methods. For instance, some limited computational experience were reported by Denault and Goffin, 1999 using small-scale traffic assignment instances with an analytic center cutting plane method (ACCPM) for variational inequalities. Rosas et al., 2002 proposed the use of ACCPM in a simplicial decomposition algorithm for solving the master problem, which is a reduced variational inequality in a derived simplex space. Their computational experience is extended to the solution of real large-scale problems.

The main goal of this work is to solve the traffic assignment problem as a variational inequality using ACCPM. In order to do this, we apply the commodity formulation, obtaining a multicommodity network flow model. We developed two variants. The first of them applies ACCPM to the multicommodity formulation using different approaches for computing Newton's directions. The second variant adds a cut for each commodity at each iteration of the ACCPM. At each iteration is necessary the evaluation of the total vector flow. Observe that the problem is nonseparable. That "disaggregated" procedure turned out to be fairly efficient and permitted the solution of medium sized problems.

The structure of the paper is as follows. Section 1 shows the formulation of the traffic assignment as a variational inequality problem and presents a multicommodity node-arc formulation. Section 2 outlines the ACCPM for variational inequalities taking into account the structure of the problem. In Section 3 we present the structure of the matrix to be factorized. In section 4 the more efficient disaggregated variant of the solution algorithm is introduced. Section 5 shows some computational experience with an implementation of both variants. Finally Section 6 presents our conclusions.

1 The traffic assignment problem as a variational inequality problem

The basis of the model is the concept of *user or Wardrop equilibrium* This concept is a behavioral principle which states that drivers compete noncooperatively for the network resources in order to minimize their travel costs. Thus, the traffic assignment model can be considered a special case of the Nash equilibrium problem (Haurie and Marcotte, 1985).

The user equilibrium principle states that a driver will choose the minimum-cost path between every origin-destination pair and through this process, those utilized paths will have equal costs. Paths with costs higher than the minimum will have no flow.

The user equilibrium principle can be formulated as a nonlinear optimization problem or nonlinear complementarity problem or as a variational inequality over a polyhedral set. The problem considered in this work is usually referred to as the asymmetric traffic assignment problem because cost functions are considered nonseparable and asymmetric. Thus, we will focus on its variational inequality formulation (one of the possibilities, like the nonlinear complementarity problem).

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

$$\text{Find } y^* \in Y \text{ such that } F(y^*)^t(y - y^*) \geq 0, \quad \forall\, y \in Y, \tag{1}$$

y being the vector of link flows over the entire network. The function $F(y)$ models the time delay and is called the volume/delay function. $F(y)$ is monotone as a result of the congestion, i.e. it satisfies

$$(F(y_1) - F(y_2))^t(y_1 - y_2) \geq 0,$$

and it is assumed continuous and differentiable. Y is the nonempty, closed, convex set, which defines the feasible flows to satisfy OD travel demand.

The following *gap* function g associated with *VI(F, Y)* is used to measure the progress and as a stopping criterion:

$$g(y) = \inf_{z \in Y} F(y)^t(z - y). \tag{2}$$

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-approximate solution if $y^* \in Y$ and $g(y^*) \geq -\epsilon$ for a given ϵ tolerance.

1.1 Commodity formulation

In terms of arc flows, the following variational inequality problem, $VI(F, Y)$, solves the traffic assignment problem:

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

where $y_T = \sum_{\rho=1}^{n_c} y^\rho \in I\!R^{na}$ is the total vector flow and y^ρ is the vector flow corresponding to commodity ρ, n_a being the number of arcs and n_c the number of commodities. F is a continuous, monotone mapping in $I\!R^{n_a}$. And Y is a nonempty, closed, convex subset of $I\!R^{n_a}$, defined as

$$Y = \left\{ y_T = \sum_{\rho=1}^{n_c} y^\rho \;\mid y^\rho \in Y^\rho = \{ y^\rho \mid N y^\rho = d^\rho, y^\rho \geq 0 \} \right\} \tag{4}$$

where N is the node-arc incidence matrix and d^ρ the demand vector corresponding to commodity ρ

Problem (3) accepts an alternative formulation, that will be used in the solution procedure. For this purpose let $\mathbf{y} \in I\!R^{n_a \times n_c}$ contain all the vectors for each commodity

$$\mathbf{y} = \begin{pmatrix} y^1 \\ y^2 \\ \vdots \\ y^{n_c} \end{pmatrix},$$

and define the cost function $\mathbf{F} : I\!R^{n_a \times n_c} \longrightarrow I\!R^{n_a \times n_c}$ as follows:

$$\mathbf{F}(\mathbf{y}) = \begin{pmatrix} F(y_T) \\ F(y_T) \\ \vdots \\ F(y_T) \end{pmatrix} = \begin{pmatrix} F\left(\sum_{\rho=1}^{n_c} y^\rho\right) \\ F\left(\sum_{\rho=1}^{n_c} y^\rho\right) \\ \vdots \\ F\left(\sum_{\rho=1}^{n_c} y^\rho\right) \end{pmatrix}.$$

Consider the alternative variational inequality $VI(\mathbf{F}, \mathbf{Y})$ problem

$$\text{Find } \mathbf{y}^* \in \mathbf{Y} \text{ such that } \mathbf{F}(\mathbf{y}^*)^t(\mathbf{y} - \mathbf{y}^*) \geq 0, \quad \forall\, \mathbf{y} \in \mathbf{Y}, \tag{5}$$

where the set $\mathbf{Y}$ of feasible flows contains upper bounds $\bar{c}$ in order to avoid problems in finding the optimal solution. Note that the original problem (4) is uncapacitated. The choice of the particular values of $\bar{c}$ is related to the magnitudes of the solution components. Hence $\mathbf{Y}$ can be represented as follows

$$\begin{aligned} \mathbf{Y} &= \{\, \mathbf{y} \mid A^t\mathbf{y} \leq c \,\} \\ &= \left\{ \mathbf{y} \mid \begin{pmatrix} I \\ -I \\ B \\ -B \end{pmatrix} \mathbf{y} \leq \begin{pmatrix} \bar{c} \\ \mathbf{0} \\ \mathbf{d} \\ -\mathbf{d} \end{pmatrix} \right\} \end{aligned} \tag{6}$$

with

$$B = \begin{pmatrix} N & & & \\ & N & & \\ & & \ddots & \\ & & & N \end{pmatrix} \quad \text{and} \quad \mathbf{d} = \begin{pmatrix} d^1 \\ d^2 \\ \vdots \\ d^{n_c} \end{pmatrix},$$

where I is the identity matrix of dimension m, with $m = n_c \times n_a$, $\mathbf{0}$ is a zero vector of dimension m, $\bar{c}$ is an arbitrarily large upper bound vector that does not restrict (4) and B is a $q \times m$ matrix with $q = n_c \times n_n$, where n_n is the number of nodes. We do not work with equality constraints in (6) to avoid computing an additional factorization of size $q \times q$ at each step.

Expanding the variational inequality $VI(\mathbf{F}, \mathbf{Y})$ formulation (5) we obtain

$$\mathbf{F}(\mathbf{y}^*)^t(\mathbf{y} - \mathbf{y}^*) \quad \geq \quad 0 \quad \forall\, \mathbf{y} \in \mathbf{Y}$$

$$(F(y_T^*)^t, F(y_T^*)^t, \dots, F(y_T^*)^t) \left(\begin{pmatrix} y^1 \\ y^2 \\ \vdots \\ y^{n_c} \end{pmatrix} - \begin{pmatrix} y^{1*} \\ y^{2*} \\ \vdots \\ y^{n_c*} \end{pmatrix} \right) \quad \geq \quad 0 \quad \forall\, \mathbf{y} \in \mathbf{Y}$$

$$F(y_T^*)^t \left(\sum_{\rho=1}^{n_c} y^\rho \; - \; \sum_{\rho=1}^{n_c} y^{\rho*} \right) \quad \geq \quad 0 \quad \forall\, \mathbf{y} \in \mathbf{Y},$$

which coincides with the variational inequality *VI(F, Y)* formulation (3) of the asymmetric traffic assignment problem, with

$$\mathbf{y}* = \begin{pmatrix} y^{1*} \\ y^{2*} \\ \vdots \\ y^{n_c*} \end{pmatrix} \quad \text{and} \quad y_T^* = \sum_{\rho=1}^{n_c} y^{\rho*}.$$

2 ACCPM for variational inequalities

ACCPM, initially developed as a nondifferentiable optimization algorithm (Goffin et al., 1992), permits solving generalized monotone variational inequalities (Goffin et al., 1997b). 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 Lemaréchal et al., 1991.

Definition 2.1 *Let F be a mapping. Let Y be a nonempty convex subset of $\mathbb{R}^m$. Then a weak solution to the VI(F, Y) problem, is a point y^* such that*

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

Theorem 2.1 *Let Y be a nonempty, closed, convex subset of $\mathbb{R}^m$, 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:

$$Y^* = \{y^* \in Y \mid F(y)^t(y - y^*) \geq 0, \quad \forall\, y \in Y\} \tag{8}$$

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 (7). $Y^* \subset Y$ ensures that Y^* is bounded, while (7) ensures both the convexity and closedness of Y^*.

2.1 Analytic centers

Analytic centers, formally introduced by Sonnevend, 1988, are defined as centers of polyhedrons. Given a set

$$Y = \{y \mid A^t y \leq c, \} \tag{9}$$

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

$$y^c = \arg \max_{y \in int(Y)} \varphi_D(y). \tag{10}$$

Note that the feasible set for the traffic assignment problem as defined in (6) matches (9).

Problem (10) can be solved through the equivalent mathematical program

$$\begin{array}{ll} \max\limits_{y,s} & \sum\limits_i \ln s_i \\ \text{subject to} & A^t y + s = c \\ & s > 0. \end{array} \tag{11}$$

The first-order KKT optimality conditions of (11) are

$$\begin{aligned} Ax &= 0 && (12)\\ A^t y + s &= c && (13)\\ Xs &= e && (14)\\ x, s &> 0 && (15) \end{aligned}$$

where x are the Lagrange multipliers associated with constraints $A^t y + s = c$, (12) impose primal feasibility, (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 (12–15) can be solved using a damped Newton method. In practice, the nonlinear complementarity conditions (14) are usually relaxed, obtaining an approximate analytic center that satisfies $\|e - Xs\| \leq \eta < 1$ for a given η tolerance. More details about the solution of (12–15) can be found in Denault and Goffin, 1999.

2.2 An ACCPM algorithm for variational inequalities

The algorithm outlined in this subsection was fully described by Goffin et al., 1997b and Denault and Goffin, 1999. The method generates a sequence of shrinking sets Y_k that converge to the solution set (8) of *VI(F, Y)*:

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

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 1 shows the main steps of this procedure.

Algorithm 1 ACCPM for *VI(F, Y)*.

step 0: **Initialization**
Find a initial interior point and set $k = 0$, $Y_0 = Y$
step 1: **Analytic center**
Find an approximate analytic center y_k of Y_k
step 2: **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)*
step 3: **New cut**
$Y_{k+1} := Y_k \bigcap \{y \mid F(y_k)^t y \leq F(y_k)^t y_k\}$
$k := k + 1$
Return to step 1

3 The structure of the matrix. First variant

During the process of ACCPM we find an approximate analytic center at each iteration by solving the optimality conditions (12–15) of problem (11) using Newton's method. Each Newton iteration involves linear systems with

$$\Delta_k = A_k S_k^{-1} X_k A_k^t,$$

where k refers to the ACCPM iteration, A_k is the constraint matrix of the current localization set Y_k in Algorithm 1, and S_k and X_k are diagonal positive definite matrices derived from s and x. Δ_k has dimension $m \times m$, regardless of the large number of possible cuts generated. Due to the density of the k new cuts added to A_k^t the matrix Δ_k becomes dense.

In our first implementation of ACCPM for the variational inequality $VI(\mathbf{F}, \mathbf{Y})$ formulation, we solved the linear systems by dense Cholesky's factorization of Δ_k. However, we were unable to solve medium size problems, due to the high cost of these factorizations. We improved that solution procedure by exploiting the structure of Δ_k and using the implicit inverse of Sherman-Morrison-Woodbury formula (SMW) (Gondzio and Sarkissian, 2000).

Let A_S be the initial sparse inequality constraints and let A_D be the dense generated inequalities. Thus,

$$A_k \quad = \quad \begin{pmatrix} A_S & A_D \end{pmatrix}$$

where

$$A_S = \begin{pmatrix} I_m & -I_m & B^t_{m\times q} & -B^t_{m\times q} \end{pmatrix}$$

with

$$B^T \quad = \quad \begin{pmatrix} N^t & & & \\ & N^t & & \\ & & \ddots & \\ & & & N^t \end{pmatrix} \qquad \text{and} \qquad A_D \quad = \quad \begin{pmatrix} A_{D_1} \\ A_{D_2} \\ \vdots \\ A_{D_{n_c}} \end{pmatrix},$$

where $N^t \in I\!R^{n_a \times n_n}$, and $A_{D_i} \in I\!R^{n_a \times k}$, $i = 1, ..., n_c$. Matrix A_k has then $m = n_c \times n_a$ rows and $n_k = 2m + 2(n_c \times n_n) + k$ columns.

Expanding the matrix Δ_k and using an appropriate partitioning for S_k and X_k, we obtain

$$\begin{aligned}
\Delta_k \quad &= \quad A_k S_k^{-1} X_k A_k^t \\
&= \quad \begin{pmatrix} A_S & A_D \end{pmatrix} \begin{pmatrix} S_S^{-1} X_S & \\ & S_D^{-1} X_D \end{pmatrix} \begin{pmatrix} A_S^t \\ A_D^t \end{pmatrix} \\
&= \quad A_S S_S^{-1} X_S A_S^t \; + \; A_D S_D^{-1} X_D^2 A_D^t \\
&= \quad \begin{pmatrix} I & -I & B^t & -B^t \end{pmatrix} \begin{pmatrix} S_a^{-1} X_a & & & \\ & S_b^{-1} X_b & & \\ & & S_c^{-1} X_c & \\ & & & S_d^{-1} X_d \end{pmatrix} \begin{pmatrix} I \\ -I \\ B \\ -B \end{pmatrix} \\
& \qquad + A_D S_D^{-1} X_D A_D^t \\
&= \quad S_a^1 X_a + S_b^{-1} X_b \; + \; B^t (S_c^{-1} X_c + S_d^{-1} X_d) B \; + \; A_D S_D^{-1} X_D A_D^t.
\end{aligned}$$

In general, notice that Δ, for any iteration k, has the following dual block-angular structure

$$
\begin{aligned}
\Delta &= AS^{-1}XA^t \\
&= \begin{pmatrix} A_1 S_1^{-1} X_1 A_1^t & & & \\ & A_2 S_2^{-1} X_2 A_2^t & & \\ & & \ddots & \\ & & & A_{n_c} S_{n_c}^{-1} X_{nc} A_{n_c}^t \end{pmatrix} + A_D S_D^{-1} X_D A_D^t \\
&= \begin{pmatrix} \Delta_{S_1} & & & \\ & \Delta_{S_2} & & \\ & & \ddots & \\ & & & \Delta_{S_{n_c}} \end{pmatrix} + A_D S_D^{-1} X_D A_D^t
\end{aligned}
$$

where

$$
\begin{aligned}
\Delta_{S_\rho} &= A_\rho S_\rho^{-1} X_\rho A_\rho^t \\
&= S_{a_\rho}^{-1} X_{a_\rho} \;+\; S_{b_\rho}^{-1} X_{b_\rho} \;+\; N^t (S_{c_\rho}^{-1} X_{c_\rho} \;+\; S_{d_\rho}^{-1} X_{d_\rho}) N \quad \forall\ \rho = 1, ..., n_c.
\end{aligned}
$$

Defining

$$
\Delta_{D_\rho} = A_{D_\rho} \left(S_{D_\rho}^{-1} X_{D_\rho} \right)^{1/2} \qquad \forall\ \rho = 1, ..., n_c,
$$

the SMW formula requires the computation of the following factorizations:

$$
\begin{aligned}
\Delta_{S_\rho} &= L_\rho L_\rho^t \qquad \rho = 1, 2, ..., nc \\
\tilde{\Delta}_{D_\rho} &= L_\rho^{-1} \Delta_{D_\rho} \quad \rho = 1, 2, ..., nc \\
M = I_k + \sum_{\rho=1}^{n_c} \tilde{\Delta}_{D_\rho}^t \tilde{\Delta}_{D_\rho} &= L_M L_M^t,
\end{aligned} \tag{16}
$$

where I_k is the identity matrix of dimension k.

The inverse of Δ can be expressed as follows:

$$
\begin{aligned}
\Delta^{-1} &= (LL^t \;+\; \Delta_D \Delta_D^t)^{-1} \\
&= (LL^t)^{-1} \;-\; (LL^t)^{-1} \Delta_D M^{-1} \Delta_D^t (LL^t)^{-1}
\end{aligned} \tag{17}
$$

with all inversions easy to compute, where L and Δ_D are matrices defined from respectively L_ρ and Δ_{D_ρ}. It is even possible to exploit the symmetry and simplify (17), thus obtaining

$$
\begin{aligned}
\Delta^{-1} &= L^{-t}(I \;-\; L^{-1} \Delta_D M^{-1} \Delta_D^t L^{-t}) L^{-1} \\
&= L^{-t}(I \;-\; \tilde{\Delta}_D M^{-1} \tilde{\Delta}_D^t) L^{-1}.
\end{aligned} \tag{18}
$$

To solve

$$
\Delta z = g, \tag{19}
$$

where $z, g \in I\!R^m$, we partitioned both vectors accordingly to the row partition of matrix Δ into blocks $z = (z_1, z_2, ..., z_{n_c})$ and $g = (g_1, g_2, ..., g_{n_c})$. The solution of (19) is thus obtained by the following sequence of multiplications and triangular systems:

$$
\begin{array}{rcll}
L_\rho v_\rho & = & g_\rho, & \rho = 1,2,...,n_c \\
u & = & \sum_{\rho=1}^{n_c} \Delta^t_{D_\rho} L_\rho^{-t} v_\rho & \\
L_M w & = & u & \\
L_M^t v & = & w & \\
L_\rho^t z_\rho & = & v_\rho - L_\rho^{-1}\Delta_{D_\rho} v, & \rho = 1,2,...,n_c.
\end{array}
\tag{20}
$$

It is possible to compute M in (16) as a sum of outer products of columns of the matrices $\tilde{\Delta}^t_{D_\rho}$. Similarly, for the evaluation of $\tilde{\Delta}_{D_\rho} v = L_\rho^{-1}\Delta_{D_\rho} v$ in (20) we compute the multiplication of $\Delta_{D_\rho} v$ first and finally a triangular system with L_ρ is solved.

We solved the linear systems using the SMW formula in two different ways. In the first case we used dense Cholesky factorizations. And in the second case we applied specialized techniques for sparse systems using the SPARSPAK package (George and Lui, 1981).

4 n_c-cuts. Second disaggregated variant

As shown in Section 1.1 the variational inequality $VI(\mathbf{F},\mathbf{Y})$ problem (5) can be written as follows

$$
\text{Find } \mathbf{y}^* \in \mathbf{Y} \text{ such that } \sum_{\rho=1}^{n_c} F(y_T^*)^t (y^\rho - y^{\rho *}) \geq 0, \quad \forall\, \mathbf{y} \in \mathbf{Y} \subset I\!R^{n_a \times n_c} \tag{21}
$$

where

$$
y_T^* = \sum_{\rho=1}^{n_c} y^{\rho *}.
$$

Defining the cost function $\mathbf{F}^\rho : I\!R^{n_a\times n_c} \rightarrow I\!R^{n_a \times n_c}$ as follows

$$
\mathbf{F}^\rho(\mathbf{y}) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ F\left(\sum_{\rho=1}^{n_c} y^\rho\right) \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{array}{l} \\ \\ \\ \leftarrow \text{commodity } \rho \\ \\ \\ \\ \end{array} \qquad \text{where} \quad \mathbf{y} = \begin{pmatrix} y^1 \\ y^2 \\ \vdots \\ y^{n_c} \end{pmatrix},
$$

we can formulate the following set of n_c variational inequalities $VI(\mathbf{F}^\rho, \mathbf{Y})$, $\rho = 1,...,n_c$

$$
\text{Find } \mathbf{y}^* \in \mathbf{Y} \text{ such that } \mathbf{F}^\rho(\mathbf{y}^*)^t(\mathbf{y} - \mathbf{y}^*) \geq 0, \quad \forall\, \mathbf{y} \in \mathbf{Y},\ \rho = 1,...,n_c. \tag{22}
$$

Expanding the set of n_c variational inequalities (22) we obtain

$$(0, ..., 0, F(y_T^*)^t, 0, ..., 0) \left(\begin{pmatrix} y^1 \\ y^2 \\ \vdots \\ y^{n_c} \end{pmatrix} - \begin{pmatrix} y^{1*} \\ y^{2*} \\ \vdots \\ y^{n_c*} \end{pmatrix} \right) \geq 0 \quad \forall\, \mathbf{y} \in \mathbf{Y},\ \rho = 1, ..., n_c$$

$$F(y_T^*)^t \, (y^\rho \;-\; y^{\rho*}) \geq 0 \quad \forall\, \mathbf{y} \in \mathbf{Y},\ \rho = 1, ..., n_c,$$

where

$$y_T^* = \sum_{\rho=1}^{n_c} y^{\rho*}.$$

Proposition 4.1 *Any solution of the variational inequality $VI(\mathbf{F}, \mathbf{Y})$ problem (5) written as (21) is a solution of the set of n_c variational inequalities $VI(\mathbf{F}^\rho, \mathbf{Y})$ (22) and conversely.*

Proof: Firstly, we suppose that $y^{\rho*}$, $\rho = 1, ..., n_c$ solves the set of n_c variational inequalities $VI(\mathbf{F}^\rho, \mathbf{Y})$ (22). If we add them together, we observe that we also find a solution of the variational inequality $VI(\mathbf{F}, \mathbf{Y})$ problem (5) written as (21).

Next, we suppose that $\mathbf{y}^*$ is a solution of $VI(\mathbf{F}, \mathbf{Y})$ problem (21). That means

$$\begin{aligned} 0 = gap &= \min_{\mathbf{z} \in \mathbf{Y}} \mathbf{F}(\mathbf{y}^*)^t (\mathbf{z} - \mathbf{y}^*) \\ &= \min_{\mathbf{z} \in \mathbf{Y}} \left(F(y_T^*)^t, F(y_T^*)^t, \ldots\ , F(y_T^*)^t \right) \left(\begin{pmatrix} z^1 \\ z^2 \\ \vdots \\ z^{n_c} \end{pmatrix} - \begin{pmatrix} y^{1*} \\ y^{2*} \\ \vdots \\ y^{n_c*} \end{pmatrix} \right) \\ &= \min_{\mathbf{z} \in \mathbf{Y}} \sum_{\rho=1}^{n_c} F(y_T^*)^t (z^\rho \;-\; y^{\rho*}), \end{aligned}$$

and by the separability structure of the commodity characterization of Y

$$\sum_{\rho=1}^{n_c} \left(\min_{z^\rho \in Y^\rho} F(y_T^*)^t (z^\rho - y^{\rho*}) \right) = 0.$$

It must be shown that each term of the above sum is equal to zero. If a term were positive then

$$\min_{z^\rho \in Y^\rho} F(y_T^*)^t (z^\rho - y^{\rho*}) > 0,$$

which is not possible since for $z^\rho = y^{\rho*}$, thus, the objective function value is zero. The nonexistence of positive terms means that negative terms are neither possible, since the overall sum must be zero.

Therefore

$$\min_{z^\rho \in Y^\rho} F(y_T^*)^t (z^\rho - y^{\rho*}) = 0 \qquad \forall\, \rho = 1, ..., n_c$$

which means $F(y_T^*)(y^\rho - y^{\rho*}) \geq 0$ for every $\rho = 1, ..., n_c$, and $\mathbf{y}^* = (y^{1*t}, ..., y^{n_c*t})^t$ is also a solution of (22).

4.1 Solving $VI(\mathbf{F}^\rho, \mathbf{Y})$ by ACCPM

Consider the following set of n_c variational inequalities $VI(\mathbf{F}^\rho, \mathbf{Y})$

$$\text{Find } \mathbf{y}^* \in \mathbf{Y} \quad \text{such that} \quad F(y_T^*)^t(y^\rho - y^{\rho *}) \geq 0, \quad \forall\, \mathbf{y} \in \mathbf{Y} \subset I\!R^{n_a \times n_c}, \quad \rho = 1, ..., n_c,$$

where

$$y_T \in Y \subset I\!R^{n_a} \quad \text{and} \quad Y = \left\{ y_T = \sum_{\rho=1}^{n_c} y^\rho \;\mid\; y^\rho \in Y^\rho = \{ y^\rho \mid N y^\rho = d^\rho, y^\rho \geq 0 \} \right\}.$$

The cost function $F : I\!R^{n_a} \to I\!R^{n_a}$ is continuous and monotone. The set Y^ρ for each commodity can be written as

$$\begin{aligned} Y^\rho &= \{ y^\rho \mid A^t y^\rho \leq c^\rho \} \\ &= \left\{ y^\rho \mid \begin{pmatrix} I \\ -I \\ N \\ -N \end{pmatrix} y^\rho \leq \begin{pmatrix} \bar{c}^\rho \\ \mathbf{0} \\ d^\rho \\ -d^\rho \end{pmatrix} \right\}, \end{aligned} \tag{23}$$

where A is a $n_a \times 2(n_a + n_n)$ matrix, I is the identity matrix of dimension n_a, $\mathbf{0}$ is a zero vector of dimension n_a and N is the node-arc network matrix of size $n_n \times n_a$. $\bar{c}^\rho \in I\!R^{n_a}$ is an arbitrarily large upper bound vector, necessary to avoid problems in finding the optimal solution (note that the original problem is uncapacitated), and d^ρ is the corresponding demand for commodity ρ.

This representation of the $\rho = 1, ..., n_c$ feasible sets also matches (9) considering variables y^ρ. Thus, it is possible to follow the same computations of Algorithm 1 to find an approximate analytic center of each localization set in a given iteration.

Observe that the cost function depends on the total flow, that is

$$F(y_T) = F\left(\sum_{\rho=1}^{n_c} y^\rho \right),$$

hence, Algorithm 1 can not be applied independently to each commodity, due to the need of computing the total flow before computing the new inequalities. However, with this evaluation of the cost function, it is possible to compute n_c new cuts, as follows

$$F(y_{T_k})^t y^\rho \leq F(y_{T_k})^t y_k^\rho, \quad \rho = 1, ..., n_c.$$

Algorithm 2 shows the main steps followed in order to find a solution to the set of n_c variational inequalities $VI(\mathbf{F}^\rho, \mathbf{Y})$ (problem (22)).

As in Algorithm 1, after k iterations the sets of localization are:

$$Y_k^\rho = \left\{ y_k^\rho \mid A_{k_\rho}^t y^\rho \leq c_k^\rho \right\}, \qquad \rho = 1, ..., n_c$$

Algorithm 2 ACCPM for $VI(\mathbf{F}^\rho, \mathbf{Y})$.

step 0: **Initialization**
Find a initial interior point and set $k = 0$, $Y_0^\rho = Y^\rho$, $\rho = 1, ..., n_c$
step 1: **Analytic center**
Find n_c approximate analytic centers
y_k^ρ of Y_k^ρ, $\rho = 1, ..., n_c$
step 2: **Termination Criterion**
Compute the primal *gap* $g(\mathbf{y}_k, \mathbf{Y})$
if $g(\mathbf{y}_k, \mathbf{Y}) \geq -\epsilon$ then stop
step 3: **New cuts**
Compute n_c-cuts $\quad F(y_{T_k})^t y^\rho \leq F(y_{T_k})^t y_k^\rho, \quad \rho = 1, ..., n_c$
where $\quad y_{T_k} = \sum_{\rho=1}^{n_c} y_k^\rho$
Set $Y_{k+1}^\rho := Y_k^\rho \bigcap \{y^\rho \mid F(y_{T_k})^t y^\rho \leq F(y_{T_k})^t y_k^\rho\}, \quad \rho = 1, ..., n_c$
$k := k + 1$
Return to step 1

where each matrix A_{k_ρ} is of dimension $na \times n_k$, and $n_k = n + k$ includes both the $n = 2(n_a + n_n)$ initial inequality constraints of (23) represented by A_S and for each $\rho = 1, ..., n_c$ the k generated inequalities represented by A_{D_ρ}, i.e.

$$A_{k_\rho} = \begin{pmatrix} A_S & A_{D_\rho} \end{pmatrix}, \quad \text{with} \quad A_S = \begin{pmatrix} I_{n_a} & -I_{n_a} & N^t_{n_a \times n_n} & -N^t_{n_a \times n_n} \end{pmatrix}, \quad \rho = 1, ..., n_c$$

The main difference between Algorithm 1 and 2 is that the later adds n_c cuts at step 3 and disaggregates the analytic center computation for each commodity. As already shown in other multicommodity problems (e.g. Goffin et al., 1997a) disaggregation can significantly improve the computational performance.

Again, we solve the optimality conditions (12–15) by Newton method, each Newton's iteration involves linear systems for each $\rho = 1, ..., n_c$ matrix

$$\Delta_{k_\rho} = A_{k_\rho} S_{k_\rho}^{-1} X_{k_\rho} A_{k_\rho}^t,$$

where Δ_{k_ρ} has dimension $n_a \times n_a$, regardless of the large number of possible cuts generated.

Due to the density of the k new cuts added to $A_{k_\rho}^t$ the matrix Δ_{k_ρ} becomes dense. As in the first variant we solve the systems by dense Cholesky factorization of Δ. We also exploit the structure of the matrix Δ_{k_ρ} by applying the implicit inverse of Sherman-Morrison-Woodbury formula (Gondzio and Sarkissian, 2000) and solving the sparse systems with the SPARSPAK package (George and Lui, 1981).

5 Computational experience

The two variants described in previous Sections were implemented in C. We also implemented different approaches for computing Newton's directions in ACCPM: dense Cholesky factorization and the Sherman-Morrison-Woodbury formula (Gondzio and Sarkissian, 2000) either by dense Cholesky factorization or using the SPARPAK package (George and Lui, 1981) for the sparse systems.

5.1 Test problems

The transportation networks used for testing purposes are: P (Montero and Barceló, 1996), ND (Nguyen and Dupuis, 1984), Sioux Falls (LeBlanc et al., 1975), and Atlanta and Norway (Gunluk, 1999); the last two instances were adapted to the traffic assignment problem. Table 1 reports the dimension of these networks. In the first two columns we give the number of "nodes" and "links". Column "centroids" gives the number of nodes with nonzero demands/supplies. Column "Variables" shows the overall number of variables of the associated multicommodity network model (note that "variables"= "links" × "Commodities").

Problem	Nodes	Links	Centroids	Commodities	Variables
P	8	16	4	2	32
ND	13	19	4	2	38
Sioux Falls	24	76	24	24	1824
Atlanta	15	44	15	15	660
Norway	27	102	27	27	2754

Table 1: *Test networks description*

We have developed two different categories of traffic assignment instances:

1. **Diagonal problems** involve a separable user equilibrium assignment. All the drivers are identical, they do not differ from one to another in either their travel cost definitions or their vehicle size or vehicle performance and there are no interaction between network links.

   The function that we used to relate journey speed or its reciprocal, journey time, per unit distance and flow of traffic on a network, is one of the best known and most widely used. We refer to the BPR function (Bureau of Public Roads). In this work we use the general form of the BPR function

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

   where c_a is the capacity in the current link a and t_0 is the travel time necessary to traverse the link at free flow speed in minutes. Free flow speed is assumed to hold at zero link flow. Usual values for α and β are $\alpha = 0.15$ and $\beta = 4$.

2. **Asymmetric problems** were artificially built by adding interactions between incoming links at intersections. For link a we have

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

where I_a is the set of link interacting with the current link a, $w_{ab} < 1$ is the weight factor for link interactions randomly generated to guarantee diagonal dominance of the Jacobian matrix, and $w_{aa} = 1$.

5.2 Computational results

For the first variant, we solved the traffic assignment problem formulation $VI(\mathbf{F}, \mathbf{Y})$ presented in Section 2. We also implemented three different ways for solving the linear equation systems (LS in the following tables):

1. By using dense Cholesky factorizations.

2. By applying the Sherman-Morrison-Woodbury formula and solving the linear systems by only dense Cholesky factorizations.

3. By applying the SMW formula, solving the dense systems by dense Cholesky factorizations and solving the sparse systems with the SPARSPAK package.

In the second variant, we solved the disaggregated formulation $VI(\mathbf{F}^{\rho}, \mathbf{Y})$ presented in Section 4. We implemented two different ways for solving the linear equations systems

4. By using dense Cholesky factorizations.

5. By applying the SMW formula, solving the dense systems by dense Cholesky factorizations and solving the sparse systems with the SPARSPAK package.

Tables 2 and 3 report the results obtained respectively, for diagonal and asymmetric traffic assignment instances. Column "Problem" refers to the test transportation network P, ND and SIO (Sioux Falls). Column "LS", which takes values from 1 to 5, gives the solution method for the linear systems, as presented above.

For each transportation network, the following information is provided. Column "Initial *gap*" refers to the relative *gap* of the first iteration. Column "Final *gap*" is the desired approximation of the solution in *gap* terms. Column "No. Iter." is the number of iterations proposed to reach the criteria of convergence. Column "CPU/(No. Iter.)" gives the average time per iteration. Column "CPU" gives the total execution time in seconds. All the runs were carried out on a Sun-4, SPARC-based with a 198.3 MHz CPU.

For the smallest (P) network and the slightly larger (ND) network it was possible to find the solution with both variants and any LS value. As we observe, the variant with the lowest execution time was the second with LS=4.

Problem	LS	Initial *gap*	Final *gap*	No. Iter.	CPU/(No. Iter.)	CPU
P	1	.600e-2	.416e-4	100	.019	1.9
	2	.600e-2	.416e-4	100	.037	3.7
	3	.600e-2	.416e-4	100	.038	3.8
	4	.605e-2	.962e-4	100	.015	1.5
	5	.603e-2	.704e-4	100	.042	4.2
ND	1	.142e+2	.993e-5	100	.024	2.4
	2	.142e+2	.924e-5	100	.044	4.4
	3	.142e+2	.924e-5	102	.046	4.7
	4	.142e+2	.322e-5	61	.014	0.9
	5	.142e+2	.924e-5	66	.025	1.7
SIO	1	.132e+4	.113e+4	10	113.77	1137.7
	2	.132e+4	.427e+3	200	14.06	2813.6
	3	.132e+4	.427e+3	200	12.10	2421.5
	4	.132e+4	.940	133	3.37	448.9
	5	.132e+4	.971	133	3.53	470.6

Table 2: *Results for diagonal traffic assignment problems using ACCPM-VI*

Problem	LS	Initial *gap*	Final *gap*	No. Iter.	CPU/(No. Iter.)	CPU
P	1	.917e-1	.465e-4	100	.019	1.9
	2	.917e-1	.466e-4	100	.038	3.8
	3	.917e-1	.467e-4	100	.038	3.8
	4	.917e-1	.101e-3	100	.016	1.6
	5	.917e-1	.737e-4	100	.046	4.6
ND	1	.141e+2	.968e-5	103	.024	2.5
	2	.141e+2	.932e-5	103	.046	4.8
	3	.141e+2	.932e-5	105	.046	4.9
	4	.141e+2	.720e-5	60	.015	0.9
	5	.141e+2	.682e-5	69	.027	1.9
SIO	1	.191e+4	.147e+4	10	111.01	1110.1
	2	.191e+4	.470e+3	200	14.02	2805.1
	3	.191e+4	.470e+3	200	12.10	2420.2
	4	.191e+4	.956	132	3.30	435.9
	5	.191e+4	.999	131	3.61	473.6

Table 3: *Results for asymmetric traffic assignment problems using ACCPM-VI*

For the larger (SIO) network and the first variant, we see that none of the solution methods (LS=1,2 or 3) provided an acceptable solution (the *gap* was not sufficiently reduced) after several thousands of execution seconds. With the second variant, we were able to reduce the *gap* enough to obtain a good solution. However, observe that solving the linear systems by dense Cholesky factorization (LS=4) requires less CPU time than when using the SMW formula (LS=5).

From Tables 2 and 3 it can be concluded that disaggregation is instrumental for the solution of the (SIO) network. This is due to the dimension of the dense systems (LS=1). Even when we use the SMW formula (LS=2,3), note that the dense systems increase at each iteration.

We now present the results obtained with different origin-destination (OD) matrices. That is, for the same transportation network, we multiply the OD matrix by different constants, in order to increase the difficulty of the problem.

Due to our previous computational experience. We only considered the second variant of Section 4. We also compared the two different techniques implemented for the solution of the linear systems, (LS=4,5) with the implementation of the simplicial decomposition method for variational inequalities (called RSDVI) described by Montero and Barceló, 1996. For that purpose we considered the transportation networks of Sioux Falls, Atlanta and Norway. Their dimensions are presented in Table 1.

We report in Tables 4, 5 and 6 the results obtained respectively for, Sioux Falls, Atlanta and Norway using the diagonal traffic assignment problem with four different OD matrices. Column "Method" provides the method used to solve the traffic assignment problem, such as, ACCPM with LS=4,5 and RSDVI. Column "No. Iter." is the number of iterations performed to reach the criteria of convergence. Column "*gap*" is the desired approximation of the solution in *gap* terms. Column "Obj(y_k)" shows the objective function value of the equivalent mathematical programming formulation. Column "CPU/(No. Iter.)" gives the average time for iteration. Column "CPU" gives the total execution time in seconds. For each method the three different lines in the tables present the results for the first iteration computed, when the *gap* is less than one and for the last iteration computed respectively. All the runs were carried out on a Sun-4, SPARC-based with a 198.3 MHz CPU.

We also report in Tables 7, 8 and 9 the computational results obtained respectively for, Sioux Falls, Atlanta and Norway networks for the asymmetric traffic assignment problem using four different OD matrices. For comparison purposes and because asymmetric problems do not have an equivalent mathematical programming, we show the value of the first volume variable in these columns.

From the results reported in Tables 4–9 it can be concluded that for medium sized networks, the solutions of the linear systems with dense Cholesky's factorization (LS=4) is more efficient than with the SMW formula (LS=5). In general both methods require a similar number of iterations to reach the solution. However, for the first iterations the SMW formula is more efficient, due to the reduced sized of the dense linear systems. This size successively increases at each iteration, significantly reducing its initial efficiency.

For all the different OD matrices, we observe that ACCPM takes in general the same amount of time to solve the different problems. Using the RSDVI the number of itera-

OD matrix	Method	No. Iter.	*gap*	Obj(y_k)	CPU/(No. Iter.)	CPU
1×OD	4.	1	.132e+4	.9219053e+13	.70	.7
		133	.940	.3862184e+08	3.37	448.9
		231	.778e-3	.3698446e+08	9.14	2112.1
	5.	1	.132e+4	.9219053e+13	.40	.4
		133	.971	.3867407e+08	3.53	470.6
		231	.409e-3	.3698389e+08	17.02	3933.5
	RSDVI	1	.182e+3	.1591003e+09	.00	.0
		29	.875	.3742640e+08	.02	.6
		117	.332e-3	.3699634e+08	.15	17.5
2×OD	4.	1	.132e+4	.2949585e+15	.70	.7
		133	.976	.1235962e+10	3.41	454.7
		236	.566e-2	.1181840e+10	9.43	2225.9
	5.	1	.132e+4	.2949585e+15	.3	.3
		133	.956	.1236042e+10	3.56	473.7
		238	.196e-2	.1181680E+10	18.58	4423.6
	RSDVI	1	.182e+3	.5089085e+10	.00	.0
		28	.863	.1198149e+10	.01	.5
		131	.149e-3	.1182115e+10	.22	29.0
5×OD	4.	1	.132e+4	.2880169e+17	.80	.8
		134	.977	.1207323e+12	3.44	462.0
		262	.354e-3	.1154145e+12	10.87	2849.3
	5.	1	.132e+4	.2880169e+17	.40	.4
		134	.939	.1205071e+12	3.64	488.5
		262	.259e-3	.1154139e+12	23.89	6261.4
	RSDVI	1	.182e+3	.4969675e+12	.00	.0
		28	.863	.1169954e+12	.02	.6
		129	.104e-3	.1154297e+12	.23	29.1
10×OD	4.	1	.132e+4	.9216465e+18	.80	.8
		134	.950	.3859816e+13	3.48	466.4
		274	.661e-3	.3693499e+13	11.72	3213.3
	5.	1	.132e+4	.9216465e+18	.40	.4
		133	.951	.3860152e+13	3.57	475.1
		274	.386e-3	.3693472e+13	27.05	7413.7
	RSDVI	1	.182e+3	.1590295e+14	.00	.0
		28	.863	.3743845e+13	.02	.6
		129	.275e-3	.3693743e+13	.24	30.7

Table 4: *Results for diagonal traffic assignment Sioux Falls instance, with different OD matrices*

OD matrix	Method	No. Iter.	*gap*	Obj(y_k)	CPU/(No. Iter.)	CPU
1×OD	4.	1	.249e+5	.1138975e+12	.200	.2
		105	.954	.2344665e+03	.526	55.3
		168	.457e-3	.2316421e+03	.798	134.1
	5.	1	.249e+5	.1138975e+12	.100	.1
		105	.956	.2344768e+03	.762	80.1
		168	.760e-3	.2316421e+03	2.710	455.3
	RSDVI	1	.381e+1	.2390976e+03	.000	.0
		5	.466	.2319535e+03	.000	.0
		17	.511e-2	.2316425e+03	.005	.1
2×OD	4.	1	.249e+5	.3644733e+13	.200	.2
		109	.979	.9925889e+03	.535	58.4
		179	.351e-3	.9633613e+03	.842	150.8
	5.	1	.249e+5	.3644733e+13	.100	.1
		109	.915	.9912072e+03	.818	89.2
		180	.663e-3	.9633381e+03	3.304	594.8
	RSDVI	1	.317e+2	.1348940e+04	.000	.0
		15	.850	.9772048e+03	.006	.1
		40	.449e-3	.9633841e+03	.015	.6
5×OD	4.	1	.249e+5	.3559231e+15	.200	.2
		109	.998	.5549700e+05	.534	58.3
		180	.352e-2	.5298702e+05	.841	151.5
	5.	1	.249e+5	.3559231e+15	.100	.1
		110	.922	.5534325e+05	.835	91.9
		179	.557e-3	.5299080e+05	3.231	578.4
	RSDVI	1	.334e+2	.8800627e+05	.00	.0
		11	.988	.5359837e+05	.01	.1
		36	.897e-3	.5299858e+05	.02	.6
10×OD	4.	1	.249e+5	.1138892e+17	.200	.2
		110	.923	.1738394e+07	.539	59.3
		174	.247e-3	.1663349e+07	.824	143.5
	5.	1	.249e+5	.1138892e+17	.100	.1
		109	.969	.1742876e+07	.819	89.3
		173	.971e-3	.1663392e+07	2.942	509.0
	RSDVI	1	.337e+2	.2794643e+07	.000	.0
		19	.670	.1678854e+07	.010	.2
		49	.371e-2	.1663557E+07	.026	1.3

Table 5: *Results for diagonal traffic assignment Atlanta instance, with different OD matrices*

OD matrix	Method	No. Iter.	*gap*	Obj(y_k)	CPU/(No. Iter.)	CPU
1×OD	4.	1	.175e+4	.2688172e+8	1.90	1.9
		184	.985	.1740918e+4	14.17	2608.8
		339	.350e-3	.1722349e+4	31.99	10845.0
	5.	1	.175e+4	.2688172e+8	.70	0.7
		184	.980	.1740779e+4	13.66	2515.5
		338	.509e-3	.1722353e+4	67.46	22804.8
	RSDVI	1	.438e+1	.1743579e+4	.00	.0
		3	.837	.1732292e+4	.00	.0
		64	.124e-3	.1722358e+4	.07	4.4
2×OD	4.	1	.175e+4	.8592915e+09	1.70	1.7
		205	.980	.4842330E+04	16.40	3363.4
		363	.451e-3	.4748415e+04	39.91	12674.4
	5.	1	.175e+4	.8592915e+09	.60	.6
		204	.981	.4842126e+04	17.98	3668.7
		364	.690e-3	.4748356e+04	80.04	29137.4
	RSDVI	1	.557e+2	.5759233e+04	.00	.0
		24	.915	.4775728e+04	.02	.4
		131	.215e-3	.4748705e+04	.17	21.9
5×OD	4.	1	.175e+4	.8390840e+11	1.40	1.4
		193	.964	.1388900e+06	15.15	2924.0
		359	.250e-3	.1334976e+06	34.49	12383.7
	5.	1	.175e+4	.8390840e+11	.50	.5
		192	.977	.1390205e+06	15.40	2958.6
		354	.651e-3	.1335211e+06	75.27	26646.8
	RSDVI	1	.241e+3	.2657788E+06	.00	.0
		36	.986	.1342624E+06	.03	1.1
		150	.235e-3	.1335691E+06	.33	49.7
10×OD	4.	1	.175e+4	.2684619e+13	1.40	1.4
		191	.984	.4173724e+07	14.87	2841.2
		360	.194e-2	.4001678e+07	34.64	12472.4
	5.	1	.175e+4	.2684619e+13	.50	.5
		191	.985	.4177534e+07	15.07	2880.1
		364	.191e-2	.4001254e+07	79.89	29080.1
	RSDVI	1	.242e+3	.8392274e+07	.00	.0
		43	.844	.4017448e+07	.04	1.9
		141	.615e-3	.4002987e+07	.37	51.6

Table 6: *Results for diagonal traffic assignment Norway instance, with different OD matrices*

OD matrix	Method	No. Iter.	*gap*	Obj(y_k)	CPU/(No. Iter.)	CPU
1×OD	4.	1	.191e+4	1068.02517	.70	.7
		132	.956	58.11928	3.30	435.9
		211	.826e-3	57.93587	8.03	1695.2
	5.	1	.191e+4	1068.02562	.40	.4
		131	.999	58.19730	3.43	450.3
		211	.117e-3	57.94149	13.38	2824.2
	RSDVI	1	.427e+2	41.542936	.10	.1
		12	.887	54.934993	.10	1.2
		23	.654e-1	58.221992	.18	4.2
2×OD	4.	1	.192e+4	2133.94572	.70	.7
		132	.954	116.37132	3.37	445.9
		214	.895e-3	115.93951	8.25	1775.5
	5.	1	.192e+4	2133.94523	.30	.3
		131	.996	116.27136	3.42	448.7
		214	.140e-3	115.97691	13.90	2976.1
	RSDVI	1	.427e+2	83.08635	.00	.0
		12	.887	102.62620	.02	.2
		41	.300e-2	114.49680	.33	13.6
5×OD	4.	1	.192e+4	5339.77793	.70	.7
		132	.959	290.95380	3.33	439.8
		232	.285e-3	289.98963	9.17	2128.0
	5.	1	.192e+4	5339.64268	.40	.4
		132	.955	290.62945	3.45	455.5
		231	.518e-3	289.91724	17.09	3948.7
	RSDVI	1	.427e+2	207.71596	.00	.0
		12	.887	274.67786	.04	.5
		38	.411e-2	287.33569	.12	4.7
10×OD	4.	1	.192e+4	10679.29541	.80	.8
		132	.964	581.95184	3.75	445.6
		242	.445e-3	579.98093	9.81	2374.3
	5.	1	.192e+4	10678.99369	.40	.4
		132	.962	581.32612	3.53	465.3
		240	.984e-3	579.93029	19.10	4585.5
	RSDVI	1	.427e+2	415.43192	.10	.1
		12	.887	549.35574	.10	1.2
		38	.411e-2	574.67046	.32	12.0

Table 7: *Results for asymmetric traffic assignment Sioux Falls instance, with different OD matrices*

OD matrix	Method	No. Iter.	*gap*	Obj(y_k)	CPU/(No. Iter.)	CPU
1×OD	4.	1	.234e+5	187.57412	.200	.2
		106	.939	13.66117	.525	55.7
		174	.920e-3	13.60099	.821	143.0
	5.	1	.234e+5	187.57246	.200	.2
		106	.939	13.66117	.772	81.9
		173	.961e-3	13.60099	2.918	504.9
	RSDVI	1	.405e+1	13.40050	.000	.0
		4	.914	13.72748	.025	.1
		19	.912e-3	13.60392	.026	.5
2×OD	4.	1	.234e+5	937.86698	.200	.2
		108	.992	27.20369	.533	57.6
		173	.280e-2	26.90308	.815	141.0
	5.	1	.234e+5	937.86546	.100	.1
		108	.992	27.20369	.797	86.1
		173	.917e-3	26.90529	2.916	504.5
	RSDVI	1	.327e+2	30.14088	.000	.0
		16	.652	27.24317	.031	.5
		34	.100e-2	26.90760	.068	2.3
5×OD	4.	1	.234e+5	937.86698	.200	.2
		109	.951	68.02605	.538	58.7
		171	.467e-2	67.28044	.816	139.7
	5.	1	.234e+5	937.86546	.100	.1
		109	.951	68.02603	.812	88.6
		171	.466e-2	67.28042	2.799	478.7
	RSDVI	1	.370e+2	75.47850	.000	.0
		14	.900	69.32901	.014	.2
		41	.684e-2	67.28119	.027	1.1
10×OD	4.	1	.234e+5	1875.71089	.200	.2
		109	.951	136.03977	.540	58.9
		167	.950e-3	134.56454	.792	132.3
	5.	1	.234e+5	1875.71350	.100	.1
		109	.951	136.03981	.811	88.4
		167	.271e-2	134.55764	2.624	438.3
	RSDVI	1	.371e+2	150.96596	.000	.0
		14	.901	138.65834	.036	.5
		39	.232e-2	134.56420	.100	3.9

Table 8: *Results for asymmetric traffic assignment Atlanta instance, with different OD matrices*

OD matrix	Method	No. Iter.	*gap*	Obj(y_k)	CPU/(No. Iter.)	CPU
1×OD	4.	1	.205e+4	242.75270	1.90	1.9
		178	.982	3.03302	13.44	2393.3
		316	.262e-3	2.90253	29.16	9214.8
	5.	1	.205e+4	242.75410	.70	0.7
		179	.968	3.03071	13.69	2451.8
		316	.688e-3	2.90261	56.27	17782.5
	RSDVI	1	.369e+1	3.99751	.10	.1
		3	.881	2.90000	.07	.2
		30	.478e-2	2.90000	.25	7.5
2×OD	4.	1	.205e+4	1213.74678	1.70	1.7
		186	.971	6.59905	14.28	2656.9
		329	.321e-3	5.82283	30.60	10069.8
	5.	1	.205e+4	1213.74677	.60	.6
		186	.972	6.60431	14.08	2619.6
		329	.398e-3	5.82315	62.99	20726.1
	RSDVI	1	.123e+2	11.17582	.10	.1
		11	.842	5.800000	.10	1.1
		46	.812e-3	5.800000	.42	19.5
5×OD	4.	1	.205e+4	1213.74678	1.40	1.4
		185	.994	17.29706	14.15	2618.9
		308	.961e-3	14.64116	28.14	8667.8
	5.	1	.205e+4	1213.74677	.60	.6
		185	.992	17.27361	13.70	2534.6
		307	.148e-2	14.64249	52.65	16166.0
	RSDVI	1	.645e+2	41.73746	.10	.1
		13	.949	13.35894	.08	1.1
		54	.895e-3	14.50000	.32	17.1
10×OD	4.	1	.205e+4	2428.85386	1.30	1.3
		185	.992	34.56697	14.11	2611.9
		312	.427e-3	29.24969	28.71	8959.1
	5.	1	.205e+4	2428.85253	.60	.6
		185	.995	34.58760	13.84	2560.9
		312	.881e-3	29.25538	54.75	17085.1
	RSDVI	1	.648e+2	90.09222	.00	.1
		12	.817	29.00000	.10	1.2
		62	.454e-3	29.00000	.64	39.5

Table 9: *Results for asymmetric traffic assignment Norway instance, with different OD matrices*

tions slightly increases as the OD matrix is multiplied by 2, 5 or 10. For the asymmetric Sioux Falls instance we encountered some convergence problems using RSDVI. However, for all instances the simplicial decomposition method was significantly more efficient than ACCPM.

6 Conclusions

The performance of the direct application of ACCPM to the solution of the variational inequality formulation of the traffic assignment problem has been evaluated using two different proposed variants. For both of them different techniques were considered to solve the necessary linear systems. According to the obtained results, it can be concluded that for the first iterations of ACCPM, the SMW formula (LS=5) is more efficient than dense Cholesky's factorization (LS=4). Nevertheless, during the process of ACCPM, when using the SMW formula, there is an increase in the dimension of the dense system. This increase means a rise in the cost per iteration and therefore, dense Cholesky's factorization becomes more competitive than the SMW formula.

Using the first variant we were only able to solve small sized instances. This is not surprising since the multicommodity formulation increases the dimension of the problem, which is the "number of arcs" multiplied by the "number of commodities". In general, ACCPM requires the generation of many cuts in order to find a solution. Therefore, the direct application of ACCPM is not as competitive as other specialized methods for the traffic assignment problem as simplicial decomposition.

However, we believe that the second disaggregated variant, which adds at each iteration a different cut for each commodity is of considerable importance. According to the computational results, it achieves substancial improvements in the solution of medium sized instances. In our opinion, this is an interesting issue in the direct application of ACCPM in solving the traffic assignment problem. Furthermore, these results could be significantly improved by developing a parallel algorithm for this second disaggregated variant. Another possible extension could be the use of quadratic cuts in ACCPM to improve the speed of convergence (Denault and Goffin, 1998).

7 Acknowledgments

The authors are indebted to Prof. J. Barceló for his encouragement in the initial stages of this work. The first author was supported by a grant from the Universidad Nacional Autónoma de México (UNAM). The second author was supported by CICYT grant TAP 99-1075-C02-02. The third author was supported by CICYT grant TIC 2000-1750-C06-03.

References

Bertsekas, D. P. and Gafni, E. M. "Projection methods for variational inequalities with application to the traffic assignment problem". *Mathematical Programming Study*, 17, 139–159 (1982).

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

Dafermos, S. “Relaxation algorithms for the general asymmetric traffic equilibrium problem”. *Transportation Science*, 16, 231–240 (1982).

Denault, M. and Goffin, J. L. “Variational inequalities with quadratic cuts”. Techical report, GERAD and McGill University, Montréal, CN (1998).

Denault, M. and Goffin, J. L. “On a primal-dual analytic center cutting plan method for variational inequalities”. *Computational Optimization and Applications*, 12, 127–156 (1999).

George, J. and Lui, J. “Computer solution of large sparse positive definite systems”. New Jersey, USA. Prentice-Hall, Englewood Cliffs (1981).

Goffin, J.-L., Gondzio, J., Sarkissian, R., and Vial, J.-P. “Solving nonlinear multicommodity problems by the analytic center cutting plane method”. *Mathematical Programming*, 76, 131–154 (1997a).

Goffin, J.-L., Haurie, A., and Vial, J.-P. “Decomposition and nondifferentiable optimization with the projective algorithm”. *Management Science*, 38-2, 284–302 (1992).

Goffin, J.-L., Marcotte, P., and Zhu, D. “An analytic cutting plane method for pseudomonotone variational inequalities”. *Operations Research Letters*, 20, 1–6 (1997b).

Gondzio, J. and Sarkissian, R. “Parallel interior point solver for structured linear programs”. Techical report, Department of Mathematics and Statistics. The University of Edinburgh, United Kingdom (2000).

Gunluk, O. “A branch-and-cut algorithm for capacitated network design problems”. *Mathematical Programming*, 86, 17–39 (1999).

Haurie, A. and Marcotte, P. “On the relationship between Nash-Cournot and Wardrop equilibria”. *Networks*, 15, 295–308 (1985).

Lawphongpanich, S. and Hearn, D. W. “Simplicial decomposition of the asymmetric traffic assignment problem”. *Transportation Research*, 18B, 123–133 (1984).

Lawphongpanich, S. and Hearn, D. W. “Restricted simplicial decomposition with application to the traffic assignment problem”. *Ricera Operativa*, 38, 97–120 (1986).

LeBlanc, L. J., Morlok, E. K., and Pierskalla, W. P. “An efficient approach to solving the road network equilibrium traffic assignment problem”. *Transportation Research*, 9, 309–318 (1975).

Lemaréchal, C., Nemiroskii, A., and Nesterov, Y. “New variants of bundle methods”. Techical report No. 1508, INRIA-Rocquencourt (1991).

Marcotte, P. "A new algorithm for solving the variational inequalities with application to the traffic assignment problem". *Mathematical Programming Study*, 33, 339–351 (1985).

Montero, L. and Barceló, J. "A simplicial decomposition algorithm for solving the variational inequality formulation of the general traffic assignment problem for large scale networks". *TOP*, 4(2), 225–256 (1996).

Nguyen, S. and Dupuis, C. "An efficient method for computing traffic equilibria in networks with asymmetric transportation costs". *Transportation Science*, 18, 185–202 (1984).

Rosas, D., Castro, J., and Montero, L. "Using accpm in a simplicial decomposition algorithm for the traffic assignment problem". Techical report DR 2002-04, Statistics and Operation Research Department: Universitat Politècnica de Catalunya, Barcelona, Spain (2002).

Smith, M. J. "Existence, uniqueness and stability of traffic equilibria". *Transportation Research*, 13B, 295–304 (1979).

Sonnevend, G. "New algorithms in convex programming based on a notion of "centre"(for systems of analytic inequalities) and on rational extrapolation". *Trends in Mathematical Optimization*, pages 311–326 (1988).