A Computational Comparison of two Simplicial Decomposition Approaches for the Separable Traffic Assignment Problem: RSDTA and RSDVI

Lídia Montero, Esteve Codina and Jaume Barceló

Draft pel 4th Meeting del Euro Working Group
on Transportation (Newcastle 9-11 Setembre de 1.996)
Report de Recerca: Projecte Europeu PETRI E0261.
Juliol de 1.996

A Computational Comparison of two Simplicial Decomposition Approaches for the Separable Traffic Assignment Problem: RSDTA and RSDVI.

Lídia Montero, Esteve Codina and Jaume Barceló

Department of Statistics and Operations Research, Technic University of Catalonia. FME. C/Pau Gargallo 5, 08028 Barcelona. Tlno.(93) 4.01.58.68. Fax (93) 4.01.58.55. E-mails: lmontero@eio.upc.es, esteve@eio.upc.es and barcelo@eio.upc.es. Spain

Abstract

The class of simplicial decomposition methods has shown to constitute efficient tools for the solution of the variational inequality formulation of the general traffic assignment problem. The paper presents a particular implementation of such an algorithm, called RSDVI, and a restricted simplicial decomposition algorithm, developed adhoc for diagonal, separable, problems named RSDTA. Both computer codes are compared for large scale separable traffic assignment problems. Some meaningful figures are shown for general problems with several levels of asymmetry.

Keywords: Traffic Equilibria, Restricted Simplicial Decomposition, Variational Inequalities, Projection Methods, Quadratic Programming.

1 Introduction

The main objective of this work relies on the determination of the average extra-cost due to the resolution of non interaction models by a general purpose asymmetric assignment code (RSDVI) based on a simplicial decomposition scheme such as described by Hearn and Lawphongpanich (1984), *versus* an specific restricted simplicial decomposition code developed for diagonal traffic assignment models based on the proposal of Hearn et al. (1987) and called RSDTA. Another objective that the authors wish to evaluate is the performance of RSDVI algorithm in solving real large separable models and the effect, in the computational sense, of including some levels of asymmetry in test probelms.

RSDVI algorithm is a generic code that might require some adaptations to improve its efficiency when dealing with separable models. The analysis of a first group of executions will define the enhancements and modifications requested to improve performance in these particular problems.

This work is included in a research line which deals with the integration of a real time traffic assignment tool in the GETRAM/AIMSUN environment as carried out in the PETRI Project (PETRI, 1995).

Both algorithms are programmed in SUN/SOLARIS F77. Computational tests will run in a Sparc Station 2 under SUN/SOLARIS Operating System. The structure of this abstract is: Section 2 states the traffic assignment formulations (TAP) that are the base for computer programs development (RSDVI and RSDTA), Sections 3 and 4 describe

RSDVI and RSDTA algorithms respectively. Section 5 deals with test design and Section 6 shows computational result tables and graphics. Conclusions and Bibliography sections are also included at the end.

2. TAP Formulations

Assuming no interactions between the links in the network, the resulting separable model for the traffic assignment problem has been shown to be equivalent to a convex mathematical programming problem (Beckmann *et al*. 1956). The separable model becomes a nonlinear optimization problem with linear constraints. Variables *h* refer to path flows, variables *v* to link flows and functions *c* to link travel-costs.

$$\underset{h_{ijr}}{Min}\; F(v(h)) \quad = \quad \sum_{a \in \mathrm{A}} \int_o^{v_a(h)} c_a(s)\, ds$$

$$s.t. \quad \sum_{r \in R_{ij}} h_{ijr} = d_{ij} \qquad \forall (i,j) \in \mathrm{K}$$

$$h_{ijr} \quad \geq \quad 0 \qquad \forall\, r \in R_{ij},\ \forall (i,j) \in \mathrm{K}$$

where $v_a(h)$ are defined as $v_a(h) \;\equiv\; \sum_{(i,j)\in K} \sum_{r \in R_{ij}} \delta_{ar}\, h_{ijr} \quad \forall\, a \in \mathrm{A}$

In practical applications such an assumption may be too restrictive, e.g. when studying congested junctions. The asymmetric model cannot be reformulated as an optimization program by the use of the same arguments as in the separable case.

The Wardrop conditions of user equilibrium for the general traffic assignment model can be reformulated into three main types of equivalent problems: variational inequality formulations, nonlinear complementary formulations and fixed point formulations (see Montero (1992) for a survey). The variational inequality formulations is chosen in this work. The foundation of simplicial decomposition schemes for the separable traffic assignment problem resides on the fact that a polyhedron not only can be represented as the intersection of a finite number of subspaces, but as a convex combination of its extreme points (vertexes) and a positive linear combination of its rays. In our case, the feasible region of the problem is a polytope and therefore, it can be represented as the convex hull of a finite set of vertexes.

The arc flow variational inequality formulation of the traffic assignment problem will be referred to subsequently as TAP-VI. It serves as the basis for developing the RSDVI algorithm:

Find $v^* \in V$ such that $c(v^*)^T.(v-v^*) \geq 0 \quad \forall v \in V$ and

$$\boldsymbol{V} = \left\{ v \;\; s.t. \;\; v_a = \sum_{(i,j)\in K} \; \sum_{r \in R_{ij}} \delta_{ar} h_{ijr}\,;\;\; \forall a \in A,\;\; h \in H \right\}$$

The additivity property is used in the present work, because dealing with path variables causes an exponential growing of the dimensionality of subproblems and data structures. Instead of this, working with an aggregate space of link variables (so-called arc flow formulations) drastically reduces dimensionality and makes possible the development of efficient algorithmic approaches for large scale problems.

3 RSDVI algorithm

The RSDVI algorithm is a restricted simplicial decomposition framework for solving general traffic assignment problems, it is based on the variational inequality formulation and uses a linear variable projection scheme for master problems resolution. Several variants of the generic algorithm can be applied to problem resolution, all of them under user request in a user-friendly execution environment. This feature makes RSDVI a valid and useful tool for testing the efficiency of some techniques for general models. Some elements that might be tuned according to test purposes are: the quality of the initial starting point, the size of the initial active set of extreme points, the column dropping criteria to be applied, the linear approximation to be applied in master problem resolution, accuracy of master problem solution and the stopping criteria type.

The primal gap function $G(v)$ might be used to monitor the generic TAP-VI algorithmic approach. It is defined as follows,

$$G(v) = \max_{w \in V} \{c(v)^T . (v-w)\}$$

$G(v)=0$ precisely when no traveller has an incentive to change route, that is, when the flow satisfies Wardrop equilibria. Algorithms based on the primal gap function are given by Hearn (1982); methods based on simplicial decomposition are given by Lawphongpanich and Hearn (1984).

The approach is based on that by Lawphongpanich and Hearn (1984). In this section, a high-level version of the RSDVI restricted simplicial decomposition algorithm is presented.

RSDVI algorithm:

Step 0: Initialization.

Let $v^0 \in V$ be a demand feasible link flow for TAP-VI and set iteration counter to 1, $t=1$.
Set the best upper bound variable to infinity, i.e., $BUB^0 = \infty$.
Set the active working set, W^0 , to v^0.

Step 1: Initial simplex computation.

Obtain η>2 extreme link flow vectors $\{e_k\}_{k=1}^{k=\eta}$ of the demand feasible set V and update $W^0 = W^0 + \{e_k\}_{k=1}^{k=\eta}$.

Step 2: Solve **VIS=VI(c(v), W^t)** .

Find $v^t \in H(W^t)$ such that $c(v^t).(v- v^t) \geq -\varepsilon \quad \forall v \in H(W^t)$ where $H(W^t)$ is the convex hull of columns of W^t.

Step 3: Extreme flow generation.

Solve $f^t = \min_{f \in W^t} \{c(v^t).f\}$

Step 4: Convergence test for diagonal tests

Evaluate gap function $G(v^t)$ and $F(v^t)$, the objective function value. Compute relative gap $G'(v^t)$ as

$$G'(v^t) = \frac{F(v^t) - BLB}{F(v^t)}.$$ where BLB is the Best Bound to gap function

If $G'(v^t) < \varepsilon'$ then STOP.

Step 5: Active extreme flow vectors updating W^{t+1}.

Add f^t to W^{t+1} and replace the current point v^t in W^{t+1}.

Apply extreme dropping criteria to W^{t+1} to preserve master size

Set $BUB^{t+1}=min\ (BUB^t, G(v^t))$ and increment t.

GOTO Step 2.

□

where

ε'	RSDVI stopping criteria tolerance
ε	VIS stopping criteria tolerance
W^t	Set of active extreme points at iteration t
$H(W^t)$	Convex hull of set W^t

A few comments to RSDVI algorithmic proposal are:

- An incremental loading technique will be used for calculating a demand feasible starting point in computational tests.

- The master problem or VIS subproblem resolution, as we will refer to it, requires at least two extreme flows in W^t, i.e., an initial simplex dimension of $\eta = 2$. A linear projection method with variable metric on the simplicial space has been implemented to solve the master problem (Bertsekas and Gafni, 1982).

- A new extreme flow vector generation is involved in one step of the algorithm, this subproblem is the familiar linear problem in the Frank and Wolfe type algorithm, and it is decomposable into a fixed number of shortest path tree computations plus a demand loading process for building the extreme flow vector. The algorithm selected for computational tests is L-Deque (Gallo and Pallotino, 1984).

- The stopping criteria possibilities for global convergence includes a stopping criteria based on the *relative gap* as used in the commercial software EMME/2.

- Column dropping criteria has to preserve master problem maximum size, that is the maximum number of active extreme flows(W^t).

4 RSDTA Algorithm

As in the classical Frank and Wolfe method for TAP, the simplicial decomposition approach consists of two steps, generally named the *subproblem* (new extreme point computation) and the *master problem*. At the master problem step, instead of carrying out a line search of the optimal value of the objective function along the descent direction, the minimization of the objective function is performed over the convex hull of all the extreme points generated so far or a subset of them.

In the first version of the algorithm, the number of vertexes used in the master problem might increase infinitely and therefore all of the generated vertexes were used. Hohenbalken introduced a modification in the scheme consisting of the removal of those vertexes with null baricentric coordinates in the previous master problem iteration. Hearn *et al*. (1987) developed a simplicial decomposition algorithm where the maximum number of vertexes to be used in the master problem was limited *a priori*. This version is called *Restricted Simplicial Decomposition*. It follows below, a version of the restricted simplicial decomposition for the case in which the set of feasible points is a polytope. Let's assume a polytope X and F a convex function on X.

<u>RSDTA algorithm:</u>

<u>Step 0</u>: *Initialization.*

Let $v^0 \in V$ be a demand feasible link flow and set iteration counter to 1, $t=1$.

Set the best upper bound variable to infinity, i.e., $BLB^0 = -\infty$.

Set the active working set, W_x^0 , to v^0 and $W_s^0 = \varnothing$

<u>Step 1</u>: *Linearization of the objective function in v^t and computation of a new vertex (subproblem).*

Solve $f^t = \min_{f \in X^t} \left\{ c(v^t) \,.\, f \right\}$

<u>Step 2</u>: *Extremal points updating.*

If $\left| W_s^t \right| < \rho$ the new vertex is stored: $W_s^{t+1} = W_s^t \cup \left\{ f^t \right\}$, $W_x^{t+1} = W_x^t$.

If $\left| W_s^t \right| = \rho$ then replace the extreme with the smallest baricentric coordinate from W_s^t, $\bar{f}$, with the new vertex f^t: $W_s^{t+1} = W_s^t \cup \left\{ f^t \right\} / \left\{ f^t \right\}$, $W_x^{t+1} = W_x^t$

The new set of extreme flows for master problem is: $W^{t+1} = W_s^{t+1} \bigcup W_x^{t+1}$.

Step 3: *Best Lower Bound Updating (stopping criteria).*

Evaluate gap function $G(v^t)$ and $F(v^t)$, the objective function value. Compute relative gap $G'(v^t)$ as

$$G'(v^t) = \frac{F(v^t) - BLB^t}{F(v^t).}$$ where BLB is the Best Bound to gap function

If $G'(v^t) < \varepsilon$ then STOP otherwise ***goto*** Step 4.

Step 4: *Master Problem Resolution*.

$$\begin{aligned} &Min_v \quad C(v) \\ &s.t. \quad v \in H(W^t) \end{aligned}$$

Set v^t to the solution of master problem and remove extreme points resulting with baricentric coordinate equal to 0 from W_s^t.

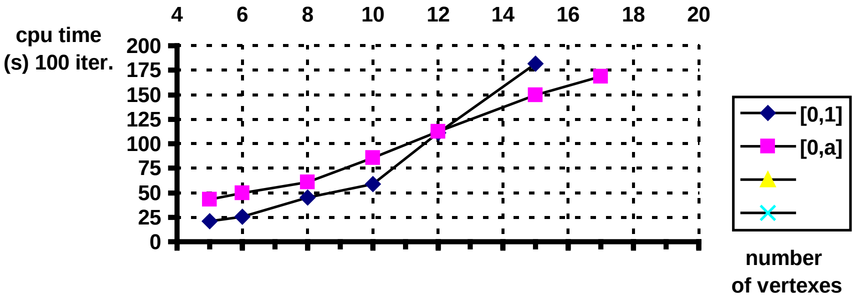
Increment iteration counter t and ***goto*** Step 1.

□

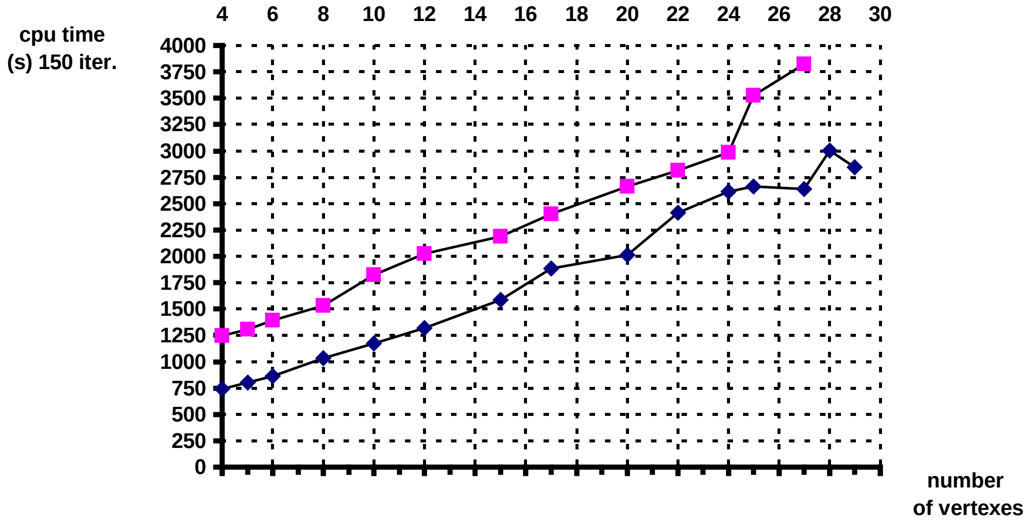
where

ε	RSDTA stopping criteria tolerance
W^t	Set of active extreme points at iteration t
$H(W^t)$	Convex hull of set W^t
ρ	Maximum number of vertexes allowed

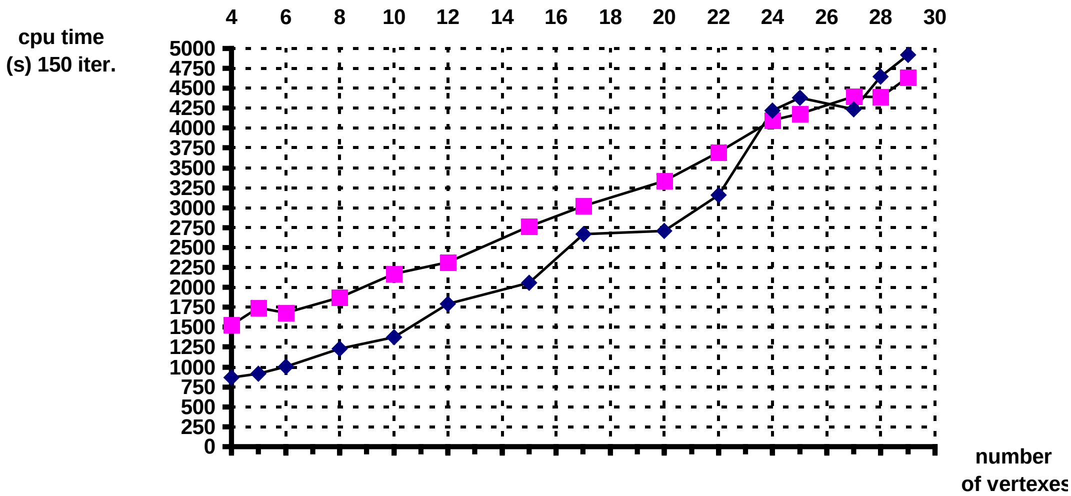
In the implemented version of RSDTA algorithm, the *relative gap* stopping criteria is used. Hearn *et al*. (1987) proved that the algorithm is convergent even when no exact solutions to master problems are computed. In this work, two ways of solving the master problem have been implemented previously to the comparison with the asymmetric algorithm. The first method implemented has been the quasi-Newton method due to Bertsekas (1982). In this method we have observed that the Armijo-like line search method requires a high percentage of the CPU time consumed in the master problem step of the RSDTA algorithm. As shown in Figure 1 for some of the test networks defined in Section 5, by restricting the line search step to the [0,1) interval a great amount



(a)



(b)



(c)

Figure 1. CPU times in a Sun Sparc 10/30 (2.3 times faster than Sun Station 2). The master problem is solved using the quasi-Newton method of Bertsekas restricting or not the line search step length for three of the test networks defined in section 5: Sioux Falls (a), Barcelona (b) and Winnipeg (c).

of CPU time is saved and this trend seems to switch when the number of vertexes used in the RSDTA algorithm increases. Computer times shown in Figure 1 correspond with a medium to high accuracy level of resolution of the master problem being allowed a maximum of 15 iterations in the master problem step.

A more efficient method than the quasi-Newton of Bertsekas for the master problem step is a quadratic approximation of the objective function as suggested in Hearn *et al.* (1987). The quadratic approximation implemented here for the hessian of the objective function consists of the complement of the Davidon Fletcher Powell formula. Additionally it is known that the master problem step can be solved with a low level of accuracy and the benefits of using this method to solve the master problem together with the limitation in the maximum accuracy attainable are obvious when observing results shown in Table 2 for the RSDTA algorithm. For this table a maximum of 3 iterations in the master problem step are allowed.

5 Test Design

The test networks used to define the diagonal traffic assignment tests are presented in this section. Networks are listed above, in an increasing order depending on their size, with a brief description:

- SIO Network. Sioux Falls test network due to LeBlanc *et al.* (1975). It consists on 24 nodes (centroids all of them), 76 links and a demand matrix defined by 538 OD pairs leading to 3616 trips. One standard volume-delay function is considered: the classical BPR formulation with α=0.25 and β=4.
- BCN Network. Barcelona's network, with a detailed representation of the CMB area (Eixample) and without Olympic Games Infrastructures. Provided by the municipality of Barcelona, it consists on 930 nodes (110 centroids), 2522 links and a demand matrix with 7922 OD pairs, representing peak morning period. BPR type delay functions.
- WIN Network. Winnipeg's network, developed by the University of Montreal, consisting on 1017 nodes (154 centroids), 2976 links and a total demand of 54,459 trips in 4345 OD pairs. BPR type delay functions.
- CMB Network. Barcelona's extended network. It includes Olympic Games corridors and rings, and a more detailed representation of the surrounding CMB area. Demand is defined by 8004 OD pairs with 373,113 trips. It contains 2253nodes (90 centroids) and 5171 links. BPR type delay functions.
- MAD Network. Madrid's network, delivered by USM Consultancy for research purposes. It consists on 2776 nodes (490 centroids) and 6871 links. Demand pattern is defined by almost 185,000 OD pairs and a basic matrix of 374,633 trips. BPR type delay functions.

For each network, several tests are designed according to the definition of:

- Two levels of congestion. Level A (medium) and level B (high). Congestion level A is defined by an average link ratio of congestion into capacity of 0.35 and level B increases the average ratio until 0.7.

- Accuracy of the solution. Three levels of error in the relative gap function are used as in the stopping criteria: Low (15%), Medium (5%) and High (1.0% and 0.1% in SIO Network).
- Maximum size of simplicial space (master problem cardinality). Several options are selected: 4, 8, 12, 15, 20, 25 and 30 vertexes. And extra option of non restricted dimensionality is also included in some tests (referred to it as *maxver=100*).

A L-Deque algorithm for shortest path computations has been selected in the new extreme point computation. A moderate precision in the master problem resolution is required and a robust set of tolerances has been defined *a priori* for each computer code. Round-off errors are prevented by requesting a double precision arithmetic in test executions.

A set of tests including three levels of asymmetry (low, medium and high) in some of the former networks (BCN and WIN) have been considered in order to evaluate the effect of asymmetry in computer cost and the selection of the best execution options in RSDVI program. Asymmetric patterns have been randomly generated among incoming/outcoming links to intersections; the level of asymmetry refers to the proportion of intersections (nodes) affected by asymmetric relations:

- Considered levels are 25%, 50% and 75% of total network nodes.

- Cost functional has been assummed to be a modified BPR type function where link costs are calculated from linear combinations of intersection link flows, where the coefficients of the combination are randomly generated (with some diagonal dominance effect restrictions).

- Neither realistic asymmetric cost functions are available to the authors, nor urban data for the calibration of asymmetric functions in local networks (BCN, CMB or MAD).

6 Computational Results

This section presents computational results obtained by executing the algorithms described in Sections 3 and 4 on the test networks described in Section 5 and only results corresponding to the lowest level of gap, i.e. 1% are considered. We shall proceed as follows: first we consider the computational comparison of RSDVI and RSDTA algorithms. Data for the comparison is contained in Tables 1 and 2 and is graphically summarized in Figure 2. Next the performance of RSDVI code depending on the level of asymmetries is presented having into account results reported in Table 3. Computational tests considered in this section have been carried out on a Sun Sparc Station 2 and all cpu times are referred to this machine.

Description of Tables 1 and 2. These tables contain computer times for RSDTA and RSDVI codes on diagonal problems. Basically they show the increase of cpu times *(CPUTotal)* and number of iterations *(Niter)* for an increasing number of vertexes used by the corresponding algorithm (*Maxver*) and the percent required by the master problem

RSDVI Computational Results (STOPPING GAP 1%) NO INTERACTION TESTS							
NET WORK	**Maxver**	**Low Congestion**			**High Congestion**		
		CPUTotal	*CPUMp(%)*	*Niter*	*CPUTotal*	*CPUMp(%)*	*Niter*
SIO	**4**	1.8	55.55	8	9.9	63.63	55
SIO	**8**	1.8	55.55	8	10.2	78.43	33
SIO	**12**	1.8	55.55	8	12.8	82.03	34
SIO	**15**	1.8	55.55	8	11.0	79.04	34
SIO	**20**	1.8	55.55	8	11.0	79.04	34
SIO	**30**	1.8	55.55	8	11.0	79.04	34
SIO	**Unlim.**	1.9	57.89	8	18.8	87.76	33
BCN	**4**	253.9	22.76	31	902.3	28.78	112
BCN	**8**	272.7	35.60	28	961.9	36.07	106
BCN	**12**	284.4	42.40	26	882.6	46.43	81
BCN	**15**	268.5	39.03	26	975.8	53.36	78
BCN	**20**	268.6	39.03	26	1113.7	61.08	74
BCN	**30**	268.7	39.03	26	1054.5	58.43	75
BCN	**Unlim.**	364.9	65.13	26	2409.0	83.13	69
WIN	**4**	171.7	18.52	18	1254.4	25.04	161
WIN	**8**	178.8	27.79	16	1192.4	34.39	132
WIN	**12**	175.9	26.83	16	1455.3	45.39	132
WIN	**15**	175.2	26.88	16	1558.3	52.57	121
WIN	**20**	177.6	26.80	16	1789.2	62.57	111
WIN	**30**	174.4	26.77	16	1848.7	65.33	107
WIN	**Unlim.**	197.9	35.47	16	7271.3	91.65	103
CMB	**4**	1005.6	26.38	79	4733.9	30.22	360
CMB	**8**	1104.0	38.47	73	4467.0	38.00	299
CMB	**12**	1434.0	49.61	78	5083.9	49.17	276
CMB	**15**	1665.9	56.05	79	5602.7	56.46	263
CMB	**20**	1790.3	64.02	69	8249.0	66.39	299
CMB	**30**	1884.5	65.86	69	11590.8	78.46	268
CMB	**Unlim.**	4657.1	86.18	69	-	-	-
MAD	**4**	1885.9	3.77	14	9771.0	4.85	95
MAD	**8**	1820.1	6.27	13	9937.4	7.64	94
MAD	**12**	1813.3	5.96	13	10478.3	11.39	95
MAD	**15**	1817.2	5.96	13	9182.0	14.26	80
MAD	**20**	1823.6	5.96	13	9228.2	18.76	76
MAD	**30**	1831.2	5.95	13	9352.0	19.87	76
MAD	**Unlim.**	1851.3	7.46	13	13801.5	47.54	73

Table 1. RSDVI Computational Results on Diagonal Tests

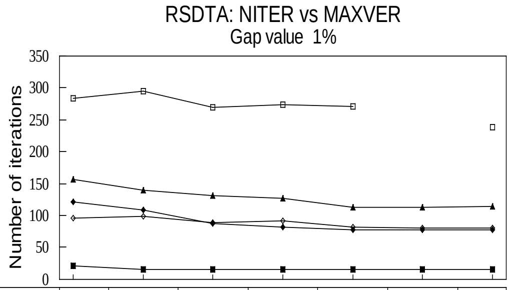
step *(CPUMaster(%))*. The row labeled as **Unlim**. corresponds to no restriction in the storage requirements on the number of extreme points. It must be noted that the RSDTA algorithm discards an extreme point always that its baricentric coordinate is zero after a master problem step even if there is no limitation in the storage of extreme points whereas the RSDVI does not and the vertex is kept for subsequent iterations. For this reason in Table 2 the maximum number of vertexes reached during the execution is

RSDTA Computational Results (STOPPING GAP 1%) NO INTERACTION TESTS							
NET WORK	**Maxver**	**Low Congestion**			**High Congestion**		
		CPUTotal	***CPUMp(%)***	***Niter***	***CPUTotal***	***CPUMp(%)***	***Niter***
SIO	**4**	0.969	50.65	6	3.212	63.57	21
SIO	**8**	0.911	54.43	(4) 6	2.747	67.64	15
SIO	**12**	0.913	54.77	(4) 6	2.731	67.68	(8) 15
SIO	**15**	0.906	54.20	(4) 6	2.735	67.65	(8) 15
SIO	**20**	0.908	54.36	(4) 6	2.729	67.69	(8) 15
SIO	**30**	0.908	54.36	(4) 6	2.729	67.69	(8) 15
SIO	**Unlim.**	0.909	54.31	(4) 6	2.732	67.63	(8) 15
BCN	**4**	366.6	15.99	34	1182.9	17.98	121
BCN	**8**	277.2	22.89	25	1201.9	26.83	109
BCN	**12**	274.6	25.12	(10) 24	1087.6	34.69	87
BCN	**15**	275.0	24.98	(10) 24	1105.4	39.27	82
BCN	**20**	274.8	25.10	(10) 24	1252.7	49.23	78
BCN	**30**	275.6	25.16	(10) 24	1156.7	45.29	(23) 78
BCN	**Unlim.**	275.5	25.15	(10) 24	1156.6	45.29	(23) 78
WIN	**4**	197.6	19.92	22	1199.9	24.70	156
WIN	**8**	207.8	28.08	21	1270.5	36.02	140
WIN	**12**	178.1	26.79	(9) 18	1403.0	45.86	131
WIN	**15**	178.5	26.85	(9) 18	1556.3	52.51	127
WIN	**20**	178.2	26.91	(9) 18	1688.1	60.83	113
WIN	**30**	178.2	26.91	(9) 18	1688.1	60.83	113
WIN	**Unlim.**	178.2	26.93	(9) 18	1805.5	63.09	(28)114
CMB	**4**	1033.8	28.35	82	3644.1	29.24	284
CMB	**8**	1188.7	40.02	80	4570.8	40.64	295
CMB	**12**	1421.7	49.62	80	4999.9	51.43	269
CMB	**15**	1609.4	55.63	79	5809.8	58.13	273
CMB	**20**	1849.5	62.80	76	7279.7	66.75	271
CMB	**30**						
CMB	**Unlim.**	2063.9	67.47	(30) 74	15377.0	86.07	(49)239
MAD	**4**	1492.5	3.22	12	9843.9	4.52	96
MAD	**8**	1609.4	4.28	(9) 13	10492.8	7.49	98
MAD	**12**	1608.9	4.32	(9) 13	9924.7	10.68	89
MAD	**15**	1607.9	4.30	(9) 13	10681.6	13.48	92
MAD	**20**	1611.4	4.31	(9) 13	10021.5	17.01	82
MAD	**30**	1609.5	4.32	(9) 13	9794.8	17.86	(25) 81
MAD	**Unlim.**	1609.5	4.32	(9) 13	9794.8	17.86	(25) 81

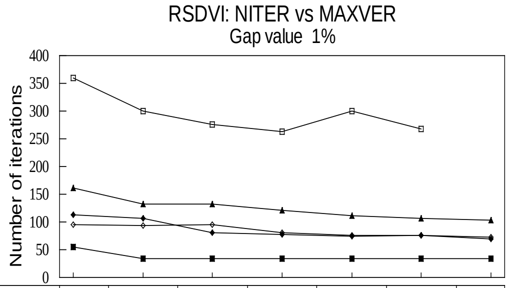
Table 2. RSDTA Computational Results on Diagonal Tests

specified in parenthesis when it is less than the limit *Maxver*.

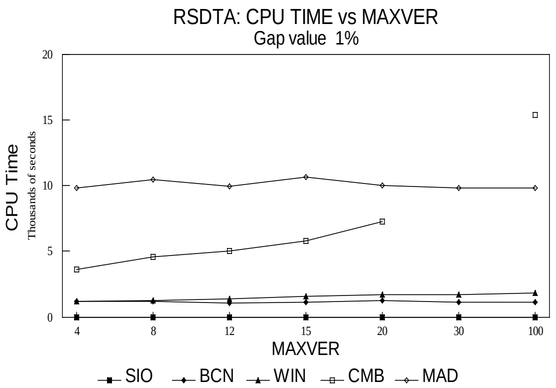
Figures 2. Comparison RSDTA vs RSDVI on NITER (nb. iterations), CPU TIME and Master Problem Percentual Time.

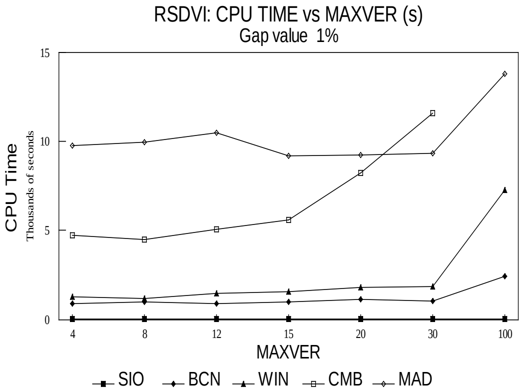


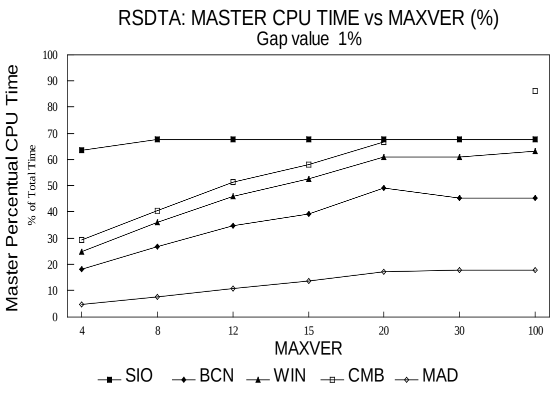
MAXVER	4	8	12	15	20	30	100
SIO	21	15	15	15	15	15	15
BCN	121	109	87	82	78	78	78
WIN	156	140	131	127	113	113	114
CMB	284	295	269	273	271		239
MAD	96	98	89	92	82	81	81

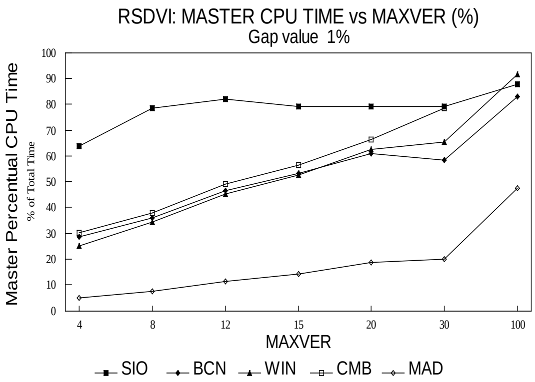


MAXVER	4	8	12	15	20	30	100
SIO	55	33	34	34	34	34	33
BCN	112	106	81	78	74	75	69
WIN	161	132	132	121	111	107	103
CMB	360	299	276	263	299	268	
MAD	95	94	95	80	76	76	73

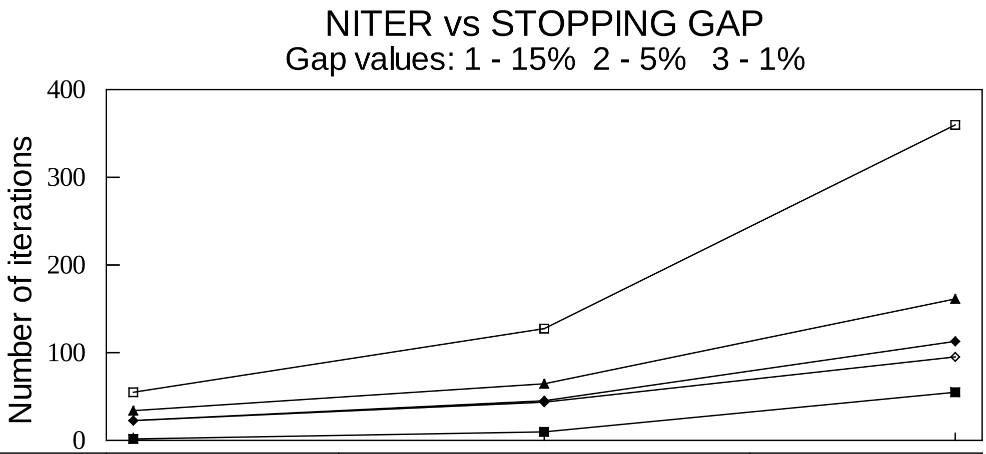








For the SIO network the specialized code RSDTA shows better computer times and number of iterations to reach the 1% gap level. When the network size increases then both algorithms present a very similar performance as shown for BCN, WIN, CMB and MAD networks provided that the number of vertexes is limited to Maxver. When the number of vertexes is not limited then the RSDTA algorithm presents a clear advantage because the master problem size is smaller as the extreme points with null baricentric coordinate are dropped from the working set. Thus keeping all generated vertexes up to the current iteration in the master problem step adds an unnecessary computational burden to the algorithm.



Stopping Gap	1	2	3
SIO	2	9	55
BCN	23	45	112
WIN	34	65	161
CMB	55	127	360
MAD	22	44	95

Figure 3. *Effect of the precision (Stopping gap) on the number of iterations. RSDVI code.*

The effect of the requested precision in the completion of executions is quite similar in both computer codes (RSDTA and RSDVI): accuracy increments the number of iterations until convergence in a nonlinear way that mainly depends on the level of congestion (the more congested the more iterations required for convergence) and the severity of BPR link cost functions (the more severity the more iterations). A detailed study of the large computer time for convergence in CMB tests shows that the severity in the definition of delay functions is the cause of the behaviour. Extra computer tests have been performed with relaxed delay functions and reveal a computer time reduction for convergence.

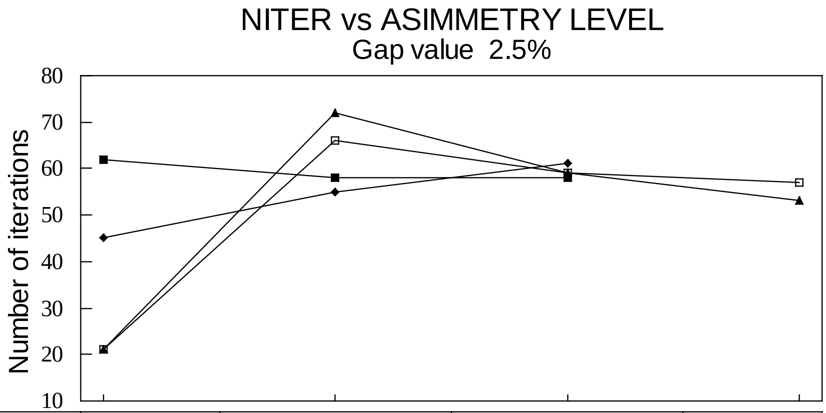
The effect of congestion on convergence is well known and our purposes were acomplished when computer results lead to an almost identical behaviour of both algorithms.

Table 3. This table is graphically depicted in Figure 4 and represents the effect of asymmetry in computer times. In general, master problem resolution phase increases its

portion of the global CPU time consumption. The number of iterations for convergence in the unlimited master resolution strongly increases under asymmetric presence; in the limited master size version is not clear, because the effect of the limitation and the asymmetry are combined and more detailed computer tests should be performed. Tunning master problem parameters, for each network, is critical.

			RSDVI Computational Results (STOPPING GAP 2.5%) ASYMMETRIC TESTS					
LEVEL ASIMM.	**NET WORK**	**Mxver**	**Low Congestion**			**High Congestion**		
			CPUTotal	***CPUMp***	***Niter***	***CPUTotal***	***CPUMp***	***Niter***
NONE	**BCN**	**8**	180.7	65.8	17	590.7	216.6	62
NONE	**BCN**	**Unlim.**	187.7	78.3	16	1047.4	772.0	45
NONE	**WIN**	**8**	130.2	33.8	11	240.2	78.4	21
NONE	**WIN**	**Unlim.**	133.4	37.5	11	317.5	159.5	21
25%	**BCN**	**8**	393.4	255.3	21	1654.1	1306.0	58
25%	**BCN**	**Unlim.**	556.3	416.9	21	7627.1	7294.0	55
25%	**WIN**	**8**	256.7	148.0	13	1715.7	1282.7	71
25%	**WIN**	**Unlim.**	264.0	155.6	13	10679.7	10275.2	66
50%	**BCN**	**8**	460.6	326.8	20	2357.0	1996.8	58
50%	**BCN**	**Unlim.**	692.8	556.0	20	17129.4	16746.8	61
50%	**WIN**	**8**	320.2	196.6	15	1793.4	1406.7	59
50%	**WIN**	**Unlim.**	413.7	290.5	15	10879.6	10491.9	59
75%	**BCN**	**8**	515.3	380.5	20	NO CONV.		
75%	**BCN**	**Unlim.**	771.9	634.8	20	NO CONV.		
75%	**WIN**	**8**	243.6	135.3	12	1508.0	1137.0	53
75%	**WIN**	**Unlim.**	290.6	182.9	12	11798.2	11403.6	57

Table 3RSDVI Computational Results on Asymmetric Tests



ASIMM.	0	1	2	3
BCN-8	62	58	58	
BCN-100	45	55	61	
WIN-8	21	72	59	53
WIN-100	21	66	59	57

Figure 4. *RSDVI general tests. Comparison of the number of iterations until convergence (2.5%) for two levels of maxver: 8 and unlimited.*

7 Conclusions

While the RSDVI algorithm solves the variational inequality formulation of a fixed demand traffic assignment with link dependencies, the RSDTA algorithm solves just the diagonal case. The computational performance of both algorithms RSDVI and RSDTA have been compared on a set of test networks and special care has been devoted in the adaptation of the codes so that both run in “equity of conditions”: a) the subproblem step is solved using the same shortest path algorithm and no post ordering is used to evaluate the solution flow of this step and b) the master problem step is solved by means of an approximation to the VI problem (or optimization problem for the RSDTA algorithm) on a convex hull of the set of extreme points and it must be noted that this process has contributed to the enhancement of the computational codes. After examining the results it can be concluded that both algorithms have approximately the same performance on diagonal problems for medium to large size networks and that therefore the advantages of using a code to solve a more general model are highlighted. For non diagonal problems the effects of asymmetries on the performance of the RSDVI algorithm results on the increase of the CPU time required, but the extension of the RSDTA algorithm into a diagonalization scheme to solve general problems may not be computationally competitive.

8 Bibliography

Beckmann, M., Mc Guire C.B., Winsten C.B. (1956), *Studies in the Economics of Transportation*. Yale University Press, New Haven CT.

Bertsekas, D.P. (1982), “Projected Newton Methods for optimization problems with simple constraints”. *Siam J. Control and Optimization 20 (2),* pp. 221-246.

Bertsekas, D.P., Gafni E.M. (1982), “Projection methods for variational inequalities with applications to the traffic assignment problem”. *Mathematical Programming Study 17*, pp. 139-159.

Bureau of Public Roads (1964), *Traffic Assignment Manual*. U.S. Department of Commerce, Urban Planning Division, Washington DC.

Gallo, G., Pallotino S. (1984), “Shortest Paths Methods in Transportation Models”. In *Transportation Planning Models* (Edited by M.Florian), pp. 227-256 Elsevier, NY.

LIOS (1995), *GETRAM Environment. TEDI: The Getram Editor User’s Guide*. Dep. Statistics and Operational Research. UPC.

Hearn, D.W. (1982), “The gap function of a convex program”. *Operations Research Letters 1*, pp. 67-71.

Hearn, D.W., Lawpongpanich S., Ventura J. (1987) “Restricted Simplicial Decomposition: Computation and Extensions”. *Mathematical Programming Study Vol 31* , pp 99-118.

Hohenbalken, B. (1977), “Simplicial Decomposition in Nonlinear Programming Algorithms”. *Mathematical Programming 13,* pp 49-68.

Lawpongpanich, S., Hearn, D.W. (1984), “Simplicial Decomposition of the asymmetric traffic assignment problem”. *Transportation Research 18B*, pp. 123-133.

LeBlanc, L.J., Morlok, E.K., Pierskalla, W. (1975), “An efficient approach to solving the road network equilibrium traffic assignment problem”. *Transp.Res. 9(5*), pp. 309-318.

Montero, L. (1992), *A Simplicial Decomposition Approach for Solving the Variational Inequality Formulation off the General Traffic Assignment Problem for Large Scale Networks.* Ph.D. Thesis supervised by Prof. Jaume Barceló, Politechnical University of Catalonia in Barcelona (Spain).

PETRI Project (1995), Parallel Computing for Spain (PACOS). PCI Project EP906, Deliverable 1 *Detailed System Design.* UPC, Municipality of Barcelona and UITESA.

Smith, M.J. (1979), “Existence, uniqueness and stability of traffic equilibria”. *Transportation Resarch 13B (4),* pp. 295-304.