CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Algorithmic Approaches for Asymmetric Multi-Class Network Equilibrium Problems with Different Class Delay Relationships

Yolanda Noriega
Michael Florian

September 2007

CIRRELT-2007-30

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince, local 2642
Québec (Québec)
Canada G1K 7P4
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

UNIVERSITÉ LAVAL
UQÀM
HEC MONTRĒAL
ÉCOLE POLYTECHNIQUE MONTRÉAL
Université de Montréal

Algorithmic Approaches for Asymmetric Multi-Class Network Equilibrium Problems with Different Class Delay Relationships

Yolanda Noriega[1,*], Michael Florian[1]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7

Abstract. In this paper, we consider a multi-class network equilibrium model where several classes of traffic, with their own travel times, interact on the links of the network. The volume/delay functions depend on each class of vehicles. Hence, the resulting cost functions are nonlinear and asymmetric. The problem is formulated as a nonlinear variational inequality model. The model is solved in three-ways: by using an MSA scheme, a Jacobi approach and a Gauss-Seidel type decomposition. The MSA approach and the Gauss Seidel method exhibit good empirical convergence. Numerical results are provided for each approach.

Keywords. Network equilibrium, multi-class traffic assignment, variational inequality problem.

Acknowledgements. This research was supported in part by an individual operating grant from the Natural Sciences and Engineering Research Council of Canada (NSERC). This support is gratefully acknowledged.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

---

* Corresponding author: Yolanda.Noriega @cirrelt.ca

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2007

© Copyright Noriega, Florian and CIRRELT, 2007

1. Introduction

One of the relevant variants of the multi-class network equilibrium model is the model that considers mixed traffic of cars and trucks. This model has been considered by Mahmassani and Mouskos (1988), Toint and Wynter (1996) and more recently by Marcotte and Wynter (2004). Mahmassani and Mouskos explored the application of the Jacobi method to solve the resulting asymmetric network equilibrium model that results since the time (cost) interaction between cars and trucks is not equal, as the impact of trucks on the travel time of cars is usually higher than the impact of cars on the travel time of trucks. Toint and Wynter further explored the properties of the resulting asymmetric model however did not propose an algorithm that is applicable to large scale problems. Marcotte and Wynter considered a particular nested cost structure that attributes to this model desirable properties, such as the uniqueness of link flows. However, the algorithm that they propose is also difficult to apply to large scale problems since it requires the explicit knowledge of the mapping that gives the dependence of the equilibrium flows of one class on the flows of a secondary class. This mapping is, in general, not analytical nor does it have an explicit expression. Another variant of this model was considered by Wu et al. (2006) where the constant that converts a truck into car equivalents, referred to as PCE (private car equivalent) or PCU (private car units) varies with the composition of the traffic, the slope of the link and its length. The resulting asymmetric, non differentiable model is solved by an adaptation of the method of successive averages (MSA).

The question that is explored in this paper is the solution of the asymmetric multi-class network equilibrium model that results when each class of traffic has its own volume/delay function which depends on the flow of all the vehicle classes present on a link. The model is formulated as variational inequality and its solution is attempted with three algorithmic approaches: the method of successive averages (MSA), the Jacobi method and the Gauss-Seidel method. The computational results indicate that the MSA and Gauss-Seidel methods yield empirically convergent algorithms while the Jacobi method fails when the congestion level in the network is high.

This paper is organized as follows. In Section 2, the mathematical model for the problem is defined. In Section 3, solution methods are developed for the mathematical model. Section 4 describes the implementation of the algorithms. Section 5 describes the data sets used in the tests while Section 6 is devoted to the presentation of the computational results. Section 7 concludes the paper.

2. Model Formulation

In this section, the notation used is introduced in order to state the mathematical formulation for the problem.

In this paper, the following notations are used. The links of the road network are designated by $a \in A$, where A is the set of links. The demand for travel by user class $m \in M$ for the origin-destination pair (p,q) is denoted as g_{pq}^{m} where M is the set of classes. These demands may use paths $K_{pq}^{m} \in K$ where K is the set of all routes $K = \bigcup_{(pq)m} K_{pq}^{m}$ and k is the path index. The set of OD pairs is denoted by I and it is convenient to refer to the OD pair with index $, i = (p,q), i \in I$. The path flow of class m on the route k is denoted h_k^m and give rise to link flows v_a^m of class $m, m = 1,..,|M|$ on link $a, a \in A.$; the total link flow v_a on link a is the weighted sum of the class flows $v_a = \sum_m \gamma^m v_a^m$; for $m = 1,...,|M|$; where γ^m is the conversion factor into private car equivalents, PCE, for the class m.

The total link flow v_a in PCE (passenger car equivalents) includes flows of all classes on link a; $s_a^m(v_a)$ is the travel time on link a for total link flow v_a in PCE for class m and s_k^m is the path travel time of class m on route k;

The feasible region Ω of the problem is defined as follows:

$$\sum_{k \in K_i^m} h_k^m = g_i^m, m \in M, i \in I \tag{1}$$

$$h_k^m \geq 0, \forall m \in M, k \in K_i^m \tag{2}$$

where

$$v_a^m = \sum_{i \in I} \sum_{k \in K_i^m} h_k^m \delta_{ak}, \forall a \in A, m \in M \tag{3}$$

$$s_k^m = \sum_{a \in A} s_a^m(v_a)\delta_{ak}, \forall m \in M, k \in K_i^m \tag{4}$$

where (1) are the equations of conservation of flow, (2) is the non-negativity of the path flows, δ_{ak} is 1 if link a is on route k and is zero otherwise and (4) is the definition of path travel time of class m on route k, s_k^m.

The multi-class network equilibrium problem can be formulated as a variational inequality problem. Find $h^* \in \Omega$ such that

$$\sum_{i \in I} \sum_{m \in M} \sum_{k \in K_i} s_k^m(h^*)(h_k - h_k^*) \geq 0, \forall h \in \Omega \tag{5}$$

It is well known that while the total link flow v in PCE is unique if $s(v) = \left[s_a^m(v), a \in A, m = 1,...,|M| \right]$ is strictly monotone, the class volumes $\left(v^m, m \in M\right)$ may not be unique. However the strict monotonicity conditions are not verified for this model. It is now a classical result that the solution of the problem satisfies the following equilibrium conditions:

$$s_k^m \begin{cases} = u_i^m & \text{if} \quad h_k^m > 0 \\ \geq u_i^m & \text{if} \quad h_k^m = 0 \end{cases}, \ \forall m \in M, k \in K_i, i \in I \tag{6}$$

where $u_i^m = \min_{k \in K_i^m} \{s_k^m\}$ is the minimum travel time for pair i of class m, which are the well known Wardrop (1952) user's optimal conditions. The derivation of the variational inequality formulation (5) from Wardrop's user optimal principle (6) is well known and may be referenced in the seminal work of Smith (1979). One may reference also the survey chapter of Florian and Hearn (1995).

In the following sections we present the three algorithms that were tested in this empirical study. Since the sufficient conditions for convergence are not satisfied, as the monotonicity condition of the cost mapping can not be verified, these methods are applied as heuristic or "ad-hoc" algorithms for the solution of this class of asymmetric network equilibrium models.

3. Solution Algorithms

3.1 MSA FOR THE MULTI-CLASS PROBLEM

The method of successive averages (MSA) is one of the most common algorithms used for network equilibrium models that are formulated with cost functions which do not satisfy monotonicity conditions. In some special cases, see Patriksson (1999), the method of successive averages may be shown to converge, such as proving that the cost mapping is non expansive. However, this sufficient condition can not be demonstrated for this class of problems. It is simple and very easy to implement. The statement of the algorithm follows.

MSA ALGORITHM

STEP 0. *Initialization.*

Initialize all class flows : $v_a^{m,l} = 0$ for $m = 1,...,|M|$; set iteration counter $l = 0$

STEP 1. *Compute new extremal flow*.

$l = l + 1$

For each class $m = 1,...,|M|$:

1.1 Update link travel times(costs) based on current link volumes:

$$s_a^{m,l} = s_a^m(v_a^{1,l-1}, v_a^{2,l-1}, \ldots, v_a^{|M|,l-1})$$

1.2 Carry out « all-or-nothing » assignment of the demands g^m on current shortest paths to obtain the extremely $y_a^{m,l}$

STEP 2. *Update link flows*.

For each class $m = 1, \ldots, |M|$ update link flows by using the MSA step size:

$$v_a^{m,l} = v_a^{m,l-1} + \lambda^l (y_a^{m,l} - v_a^{m,l-1}) \quad \text{where} \quad \lambda^l = 1/l$$

STEP 3. *Stopping criterion*.

Compute the relative gap, RG as:

$$RG = \sum_m \frac{\sum v_a^{m,l} s_a^m(v_a^l) - \sum u_i^{m,l} g_i^m}{\sum v_a^{m,l} s_a^m(v_a^l)}$$

where $u_i^{m,l}$ is the current shortest path time (cost).

If $RG \leq \varepsilon$ terminate : otherwise return to step 1.

3.2 THE JACOBI APPROACH

The Jacobi approach, which is sometimes referred to as “diagonalization”, essentially fixes sequentially the asymmetric part of the cost functions, which in this case are the cost interactions among modes. Hence it solves iteratively problems which satisfy the monotonicity condition and hence are equivalent convex cost optimization problems. All the classes are considered sequentially and a step is executed before updating the link costs. A statement of the algorithm follows.

JACOBI ALGORITHM

STEP 0. *Initialization.*

Initialize all class flows : $v_a^{m,l} = 0$ for $m = 1, \ldots, |M|$; set iteration counter $l = 0$;

STEP 1. *Compute a new solution.*

$l = l + 1$

Update link costs $s_a^{m,l} = s_a^m(v_a^{l-1}),\ m = 1, 2, ... |M|,\ a \in A$

For each class *m*, $m = 1, ..., |M|$ solve the single class problem

$$\min\ f(v) = \sum_a \int_0^{v_a^m} s_a^m(v_a^{1,l-1}, v_a^{2,l-1}, ..., v, ..., v_a^{|M-1|,l-1}, v_a^{|M|,l-1})dv$$

Subject to (1) to (3).

For each class $m = 1, ..., |M|$ update link class volumes based on the current solution.

STEP 2. *Stopping Criterion*

Compute the difference between two successive flows:

$$DiffJ = \frac{\sum abs(v_a^l - v_a^{l-1})}{\sum v_a^l} \quad (v_a \text{ in PCE}).$$

If $DiffJ \le \varepsilon$ terminate; otherwise return to step 1.

3.3 THE GAUSS SEIDEL RELAXATION APROACH

The Gauss-Seidel method solves a sequence of symmetric network equilibrium problems for each mode by keeping the flows of the other modes fixed. However, in contrast to the Jacobi method, the link costs are updated as soon as the computations for a mode have terminated. The algorithm is stated below.

GAUSS SEIDEL ALGORITHM

STEP 0. *Initialization.*

Initialize all class flows: $v_a^{m,l} = 0$ for $m = 1, ..., |M|$; set iteration counter $l = 0$

Update costs $s_a^{m,l} = s_a^m(v_a^l),\ m = 1, 2, ... |M|,\ a \in A$

STEP 1. *Compute a new solution.*

$l = l + 1$

For each class *m*, $m = 1, ..., |M|$ solve the single class problem

$$\min\ f(v) = \sum_a \int_0^{v_a^m} s_a^m(v_a^{1,l}, v_a^{2,l}, ..., v, ..., v_a^{|M-1|,l-1}, v_a^{|M|,l-1})dv$$

Subject to (1) to (3).

Update link class volumes based on the current solution.

Update costs $s_a^{m,l} = s_a^m(v_a^{l-1}), a \in A$

STEP 2. *Stopping Criterion*

Compute the difference between two successive flows:

$$DiffGS = \frac{\sum abs(v_a^l - v_a^{l-1})}{\sum v_a^l}$$ (v_a in PCE).

If $DiffGS \le \varepsilon$ terminate; otherwise return to step 1.

4. Implementation

The algorithms have been implemented in the Emme 3 (INRO, 2007) software package by using the macro language provided by this software and the single class equilibrium assignment algorithm. The Emme 3 assignment is solved by the linear approximation method of Frank and Wolfe.

For the three algorithms, an external iteration consists of the solution of a set of single class problems for each mode and in the subsequent update of the link flows and costs. Before the single class problem for a given mode is solved, the set of the corresponding class parameters is read and the flows of the other classes are considered as fixed background flows.

In the case of the Jacobi and Gauss Seidel approaches, once the flows for the classes which are not currently considered are fixed, an internal single class assignment is carried out. This problem can be solved to the optimum or not: a fixed number of internal iterations may be chosen. Sheffi (1985) proposed a streamlined version of the Jacobi algorithm which carries out only one iteration for the single class problem. Mahmassani (1988) found in empirical tests that a 2 iterations version performed best in the majority of the problem instances that he considered.

The main difference between the Jacobi and the Gauss Seidel approaches is the update of flows. In the Jacobi method the update is done after all the sequence of single class assignments is done. In the Gauss Seidel case, the flows update is done after each single class assignment. It is important to mention that the initialization of the volumes is done only once (at step 0) which

means that for the class m, at iteration $l+1$, the class m initial flows for the computations of the single class problem are the solution flows of iteration l.
In the MSA case, once the flows of the classes that are not considered are fixed, an "all-or-nothing" assignment is carried for the single class problem. In the MSA case, the current class link flows are updated by using the MSA step size. As the Jacobi approach, the flows are updated only after all the series of single class assignments are completed.

5. The network test data sets

The computational tests were carried out by using two network data sets originating from the City of Winnipeg and from the City of Montreal. The corresponding networks are displayed in Figures 1 and 2. The MSA algorithm was tested on the Winnipeg network by using up to 12 classes of traffic. All three algorithms were tested on the Montreal network by using 3 classes of traffic: private car and two types of trucks.

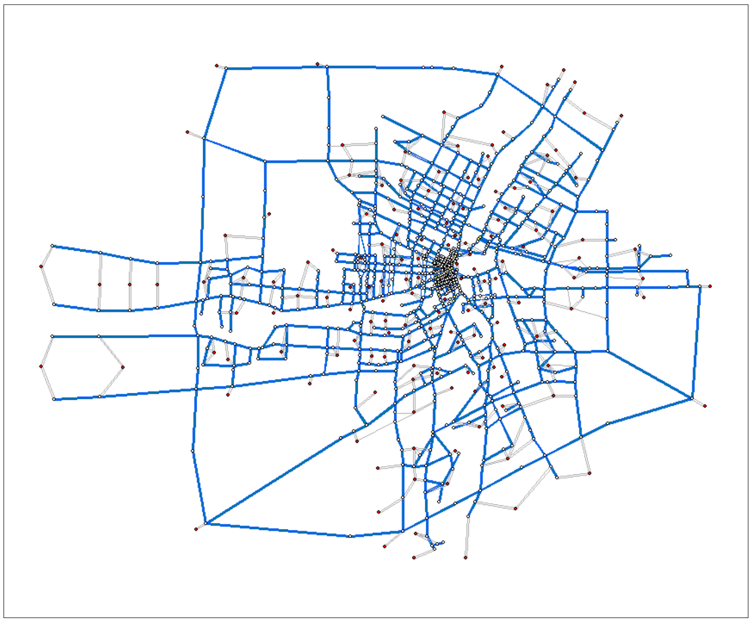

Figure 1. The Winnipeg network

The network characteristics are given in tables 1 to 3. Only data for the AM peak was available for the Winnipeg network. The data for the Montreal network includes five periods of the day.

The total assigned demand for the Winnipeg network consists of 56,219 vehicles. The demand for the Montreal network is given in Table 4.

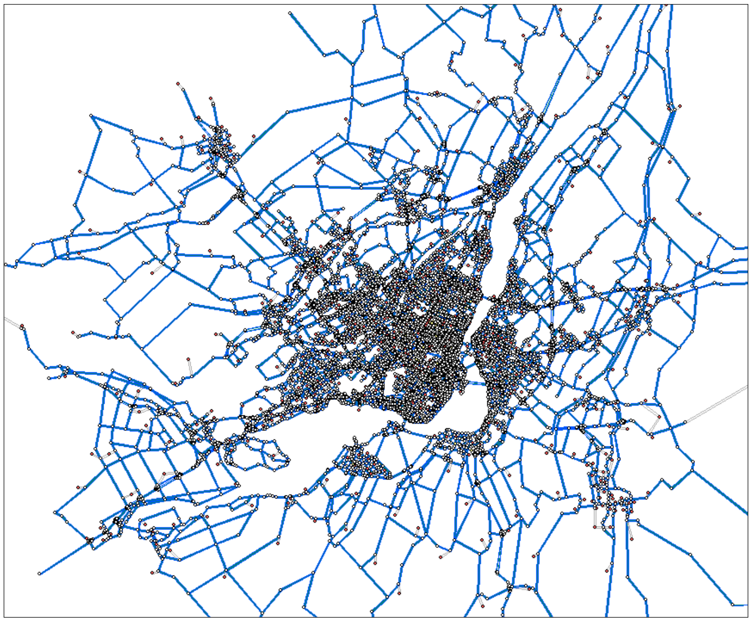

Figure 2. The Montreal Network

	Winnipeg	Montréal		
	AM peak	AM peak	PM peak	Off peak
Zones (centroids)	154	1 425	1 425	1 425
Regular nodes	903	12 771	12 756	13 019
Links	2975	30 681	30 663	32 284
Classes	3,6 or 12	4	4	4

Table 1. The network parameters

Class	Characteristics
Auto	
Regular truck	One unit, 2or 3 axles
Heavy truck	One unit, 4 axles, or more than one unit

Table 2. The Montréal Network : Classes

Code	Period	Hours
NI	Night	0 :00 to 6 :00
AM	AM Peak	6 :00 to 9 :00
OD	Off Peak Day	9 :00 to 15 :30
PM	PM Peak	15 :30 to 18 :30
ON	Off Peak Night	18 :30 to 24 :00

Table 3. The Montréal Time Periods

Period	Auto	Regular truck	Heavy truck
AM	976,715	26,631	15,367
OD	1'905,037	84,091	42,157
PM	1'259,606	24,305	14,804
ON	1'007,921	47,703	27,111
NI	246,212	7,048	7,542

Table 4. The Montreal demand

The multi-class cost functions used are of the BPR type. The specific form of the BPR functions, which was used to differentiate among the classes, is the following:

$$s_a^m(v_a) = t_a^0 * \varphi^m * (1 + \alpha_a * (v_a / c_a) \wedge \beta_a)$$

where $s_a^m(v_a)$ is the travel time on link a for class m; t_a^0 is the travel time in the link at free flow speed; φ^m is a class time factor (for the slower classes); α_a and β_a are the BPR parameters for the link a; c_a is the capacity of the link (in PCE) and $v_a = \sum_m \gamma^m v_a^m$ as previously mentioned (in PCE too).

In the case of Winnipeg the delay functions had been calibrated for this network and demand for a single class of traffic. In the case of Montreal the standard values of α and β were used: 0.15 and 4, respectively, for all the links $a \in A$. For Montreal, a second set of delay functions was used: a set of logistic functions which have an upper bound on the travel time, even in presence of very high flows. The form of these logistic functions is as follows:

$$s_a^m(v_a) = t_a^0 * \varphi^m * (1 + \frac{\eta_a}{1 + \alpha_a / ((v_a + \theta_a) / c_a) \wedge \beta_a})$$

where $s_a^m(v_a)$, t_a^0, φ^m, c_a and v_a are the same as for the BPR functions. The parameters: η_a, α_a, θ_a and β_a were previously calibrated for all the links of the network.

The BPR delay functions are increasing monotonically as a function of the flows. The logistic functions are continuous positive non decreasing functions of the flow. The behaviour of the two kinds of functions is shown in Figure 3.

These functions represent satisfactorily the behaviour of the different classes of vehicles at free flow conditions and before congestion. However, they do not represent perfectly the travel times when the flows are approaching the link capacities. This is especially true for the upper bounded logistic functions. At “practical” capacity the travel times ratios of the functions for the different classes must approach 1 that is all the modes considered have equal travel times. However, these functions differentiate sufficiently the classes of traffic considered in these computational experiments.

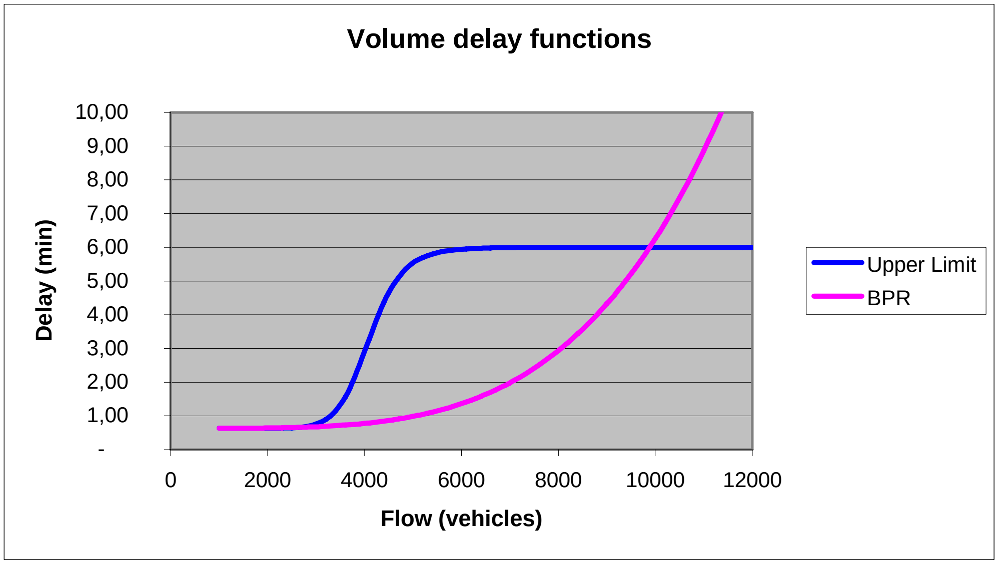

Figure 3. The volume delay functions

6. The Computational Results

The MSA algorithm was tested on the Winnipeg network which is of modest size and relatively un-congested. In order to evaluate the performance of the algorithm the demand for one class was subdivided into several classes. Five experiments were carried out: the first one has three classes, the last one has 12, and the other 3 cases consider 6 classes. The subdivision of the original demand is given in table 5. The values of the parameters φ^m and γ^m are showed in table 6.

	3 classes	**6 classes**	**12 classes**
Class 1	75	60	20
Class 2	15	10	15
Class 3	10	10	15
Class 4		10	10
Class 5		5	5
Class 6		5	5
Class 7			5
Class 8			5
Class 9			5
Class 10			5
Class 11			5
Class 12			5

.

Table 5. Percentage of the original demand of Winnipeg

	φ^m					γ^m				
	3c	6c-s1	6c-s2	6c-s3	12c	3c	6c-s1	6c-s2	6c-s3	12c
Class 1	1.00	1.00	1.00	1.00	1.00	1.0	1.0	1.0	1.0	1.0
Class 2	1.10	1.05	1.05	1.10	1.00	2.0	1.5	1.5	2.0	1.0
Class 3	1.15	1.10	1.10	1.10	1.00	3.0	2.5	1.5	2.0	1.5
Class 4		1.10	1.10	1.20	1.10		2.5	1.5	3.0	1.5
Class 5		1.15	1.15	1.20	1.10		3.5	2.5	3.0	2.0
Class 6		1.15	1.15	1.20	1.10		3.5	2.5	3.0	2.0
Class 7					1.10					2.5
Class 8					1.15					2.5
Class 9					1.15					3.0
Class 10					1.15					3.0
Class 11					1.20					3.5
Class 12					1.20					3.5

Table 6. Delay functions parameter values: Winnipeg

The convergence of the algorithm is shown in Figures 4 and 5. The link flows convergence is computed as the sum of absolute link flow differences divided by the sum of the current flows:

$$\frac{\sum abs(v_a^l - v_a^{l-1})}{\sum v_a^l}, \quad \text{(where } v_a \text{ is expressed in PCE).}$$

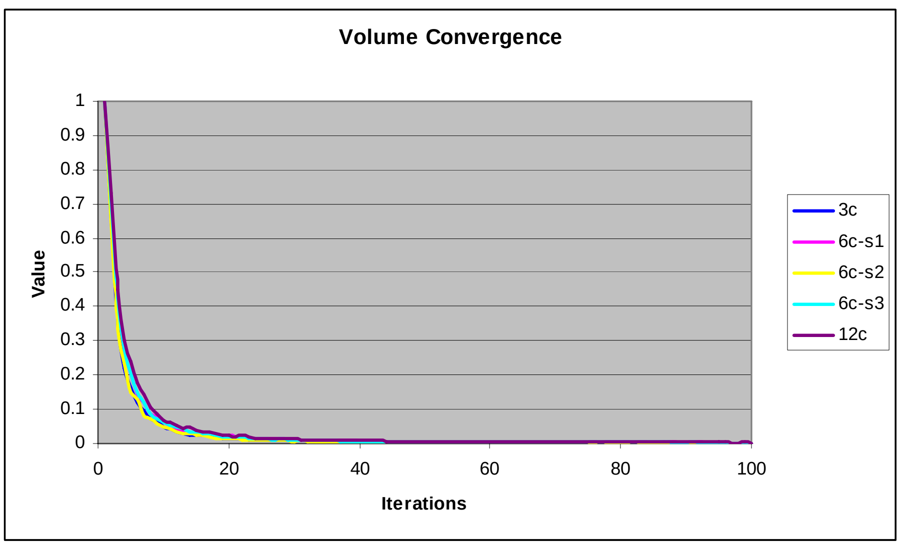

Figure 4. Volume Convergence of the MSA algorithm: Winnipeg

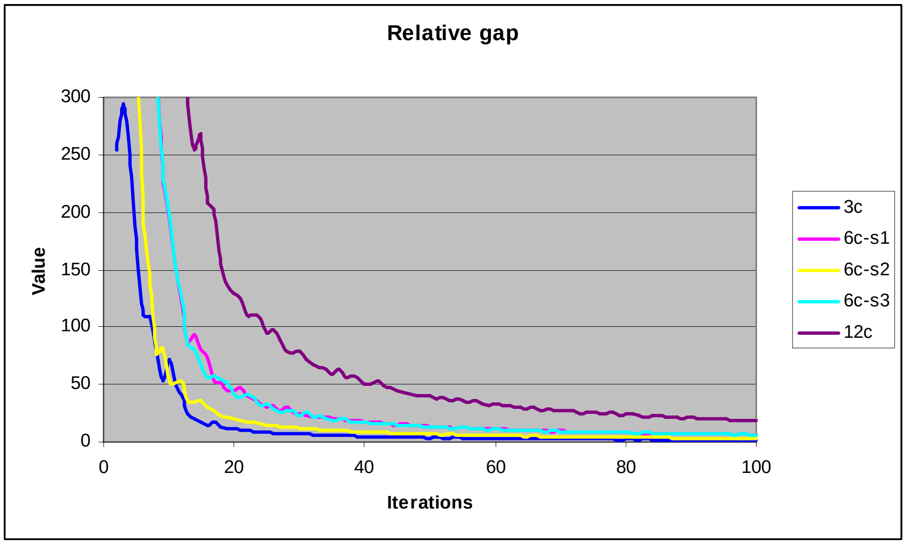

Figure 5. MSA algorithm. Relative gap : Winnipeg

The MSA algorithm converges in all these cases. As one can see, the total relative gap depends on the distribution of the demand in different number of classes as well as on the value of the parameters.

The total computation times range from 387 sec (for the three classes' case) to 1475 sec (for the 12 classes' case). The tests were carried out on an Intel® Pentium® 4, 2.40GHz, 512 Mb of RAM.

In order to evaluate the performance of the three proposed approaches under a realistic multi-class case and set of parameters, the Montreal network was considered next.

The values of the parameters φ^m (time factor for the slower vehicles) and γ^m (PCE factors) are listed in Table 7. The same values are used for the evaluation of the three algorithms over the 5 periods of the day, and for all the network links. The value of φ^m was chosen as a compromise between the reality and the fact that the travel times ratios of the different classes of vehicles should approach 1 when approaching to congestion.

Class	φ^m	γ^m
Auto	1.0	1.0
Regular truck	1.1	2.5
Heavy truck	1.15	3.5

Table 7. The delay functions parameter values: Montreal

The link flows convergence measure was used to evaluate and compare all three algorithms. It is computed as described above. The convergence of the Jacobi and the Gauss Seidel (GS) algorithms was evaluated for four different maximal numbers of internal iterations: 1, 2, 5 and 10 (it is clear that sometimes the optimal solution is obtained before the maximal number of iterations is reached). From the results obtained with both algorithms, it can be concluded that carrying out 5 internal iterations is sufficient to obtain good convergence. Figure 6 shows the convergence of the Gauss Seidel algorithm when the number of internal iterations is varied. The numerical results are listed in table 8. As one can observe, five internal iterations are enough to find the solution for the less congested periods (NI, OD and ON).

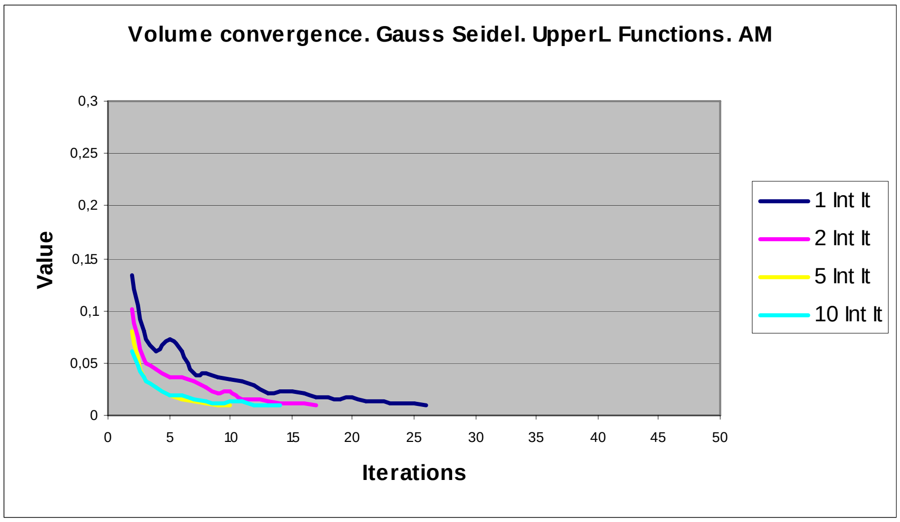

Figure 6. Comparison of the convergence according to the number of internal iterations

Period	**1 internal iteration**		**2 internal iterations**		**5 internal iterations**		**10 internal iterations**	
	Computation time	External Iterations	Computation time	External Iterations	Computation time	External Iterations	Computation time	External Iterations
NI	93	2	148	2	147	2	147	2
AM	824	26	1023	17	735	10	982	14
OD	469	13	549	8	579	8	579	8
PM	797	26	1044	18	884	13	1130	17
ON	286	8	336	5	346	5	346	5

Table 8. Computation times (in sec.) and number of external iterations with the Gauss Seidel Algorithm. Logistic Functions. Stopping criterion $\varepsilon = 0.01$

As previously mentioned, two sets of volume delay functions were used: the classical BPR and a set of logistic functions which have an upper bound on the travel time. They are referred to as UpperL or UL. The computation times and the convergence are strongly dependent on the kind of delay functions used. As it can be seen in figure 7, the Jacobi algorithm does not converge for the most congested periods: AM, PM and OD, neither with the BRP functions nor with the UL ones.

Figures 8 to 10 display the link flow convergence for the other two algorithms and sets of functions. Convergence for the PM period is very similar to the AM period even if it is more congested. In the case of the NI period, convergence is almost immediate. The MSA and Gauss Seidel algorithms always converge, even with the logistic functions.

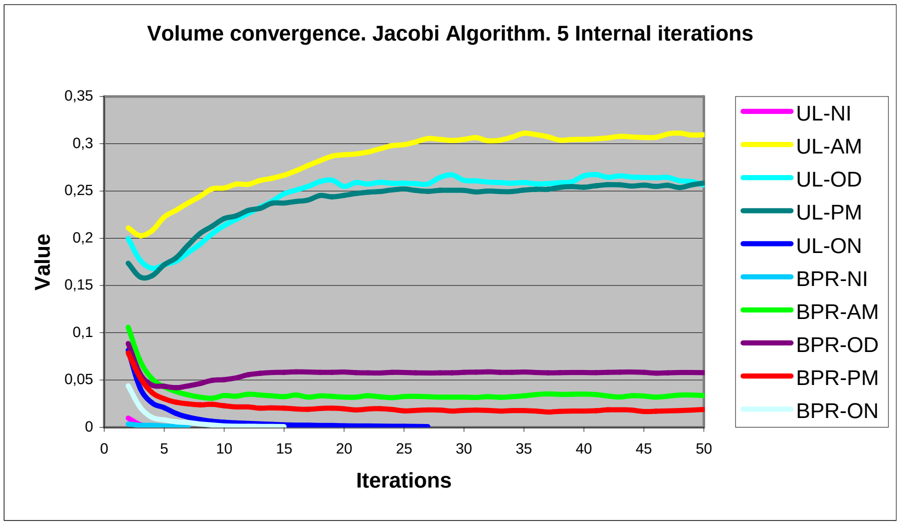

Figure 7. The Jacobi algorithm convergence

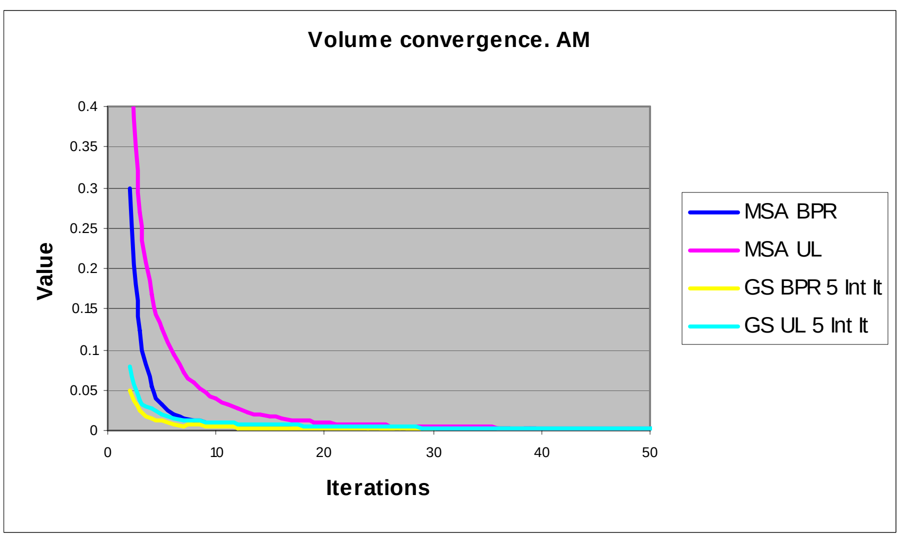

Figure 8. MSA and Gauss Seidel: Convergence of the algorithms. AM Peak

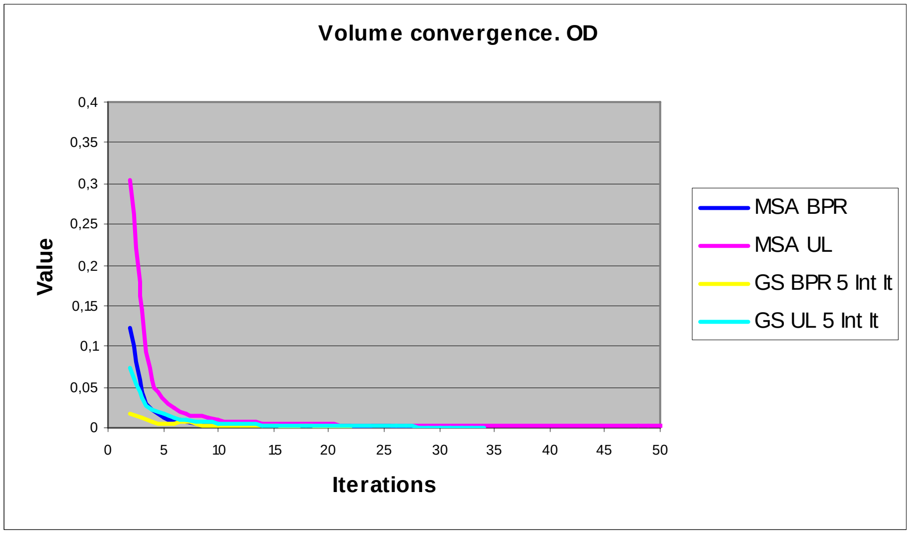

Figure 9. MSA and Gauss Seidel: Convergence of the algorithms. Off Peak Day

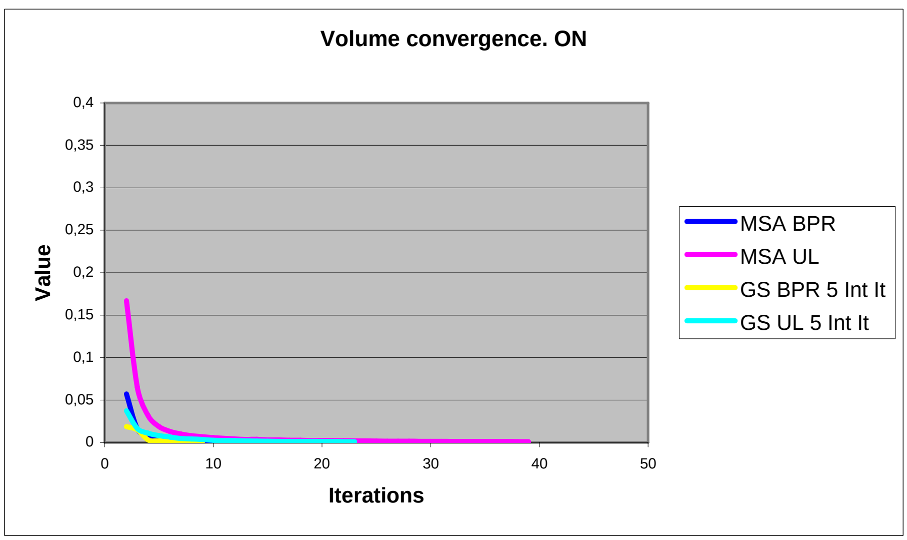

Figure 10. MSA and Gauss Seidel: Convergence of the algorithms. Off Peak Night

A summary of the computation times and number of iterations is listed in tables 9 and 10. The experiments were carried out on the same Intel® Pentium® 4, 2.40GHz, 512 Mo de RAM. The MSA algorithm is faster than the Gauss Seidel method in almost all the cases. The choice of the number of external and internal iterations for a given precision indicates that more internal iterations result in less external iterations.

Period	MSA UpperL Functions	Gauss Seidel 1 Int. It UpperL Functions	Gauss Seidel 5 Int. It UpperL Functions	MSA BPR Functions	Gauss Seidel 5 Int. It BPR Functions
NI	253	153	263	129	146
AM	3226	4276	4178	1352	1778
OD	2018	2312	2254	921	1449
PM	2965	3808	3982	1429	1701
ON	1231	1146	1444	535	575

Table 9. Computation times (in sec.). Stopping criterion $\varepsilon = 0.001$

Period	MSA UpperL Functions	Gauss Seidel 1 Int. It UpperL Functions	Gauss Seidel 5 Int. It UpperL Functions	MSA BPR Functions	Gauss Seidel 5 Int. It BPR Functions
NI	8	4	4	4	2
AM	106	139	69	45	30
OD	61	68	34	28	22
PM	102	128	68	48	29
ON	39	35	23	17	9

Table 10. Number of external iterations. Stopping criterion $\varepsilon = 0.001$

Figures 11 and 12 present the Relative Gap values for the MSA algorithm.

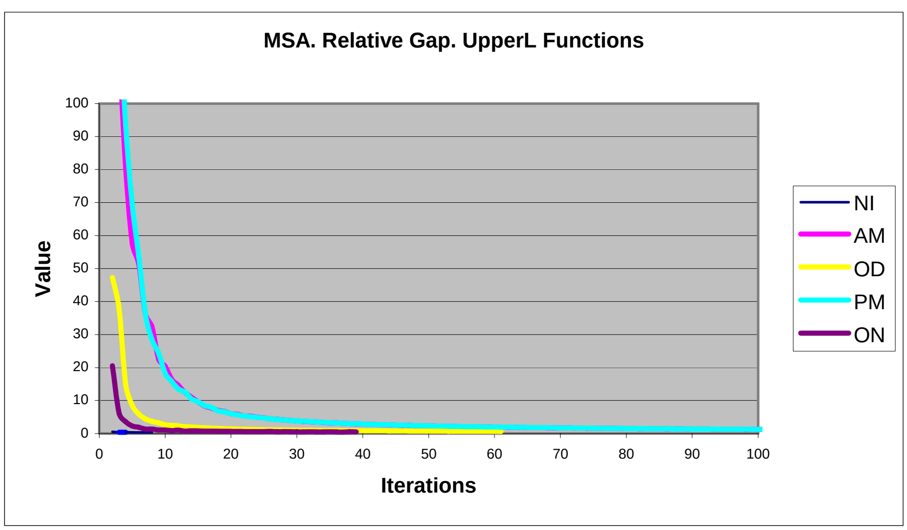

Figure 11. MSA Convergence: Relative Gap for Logistic functions

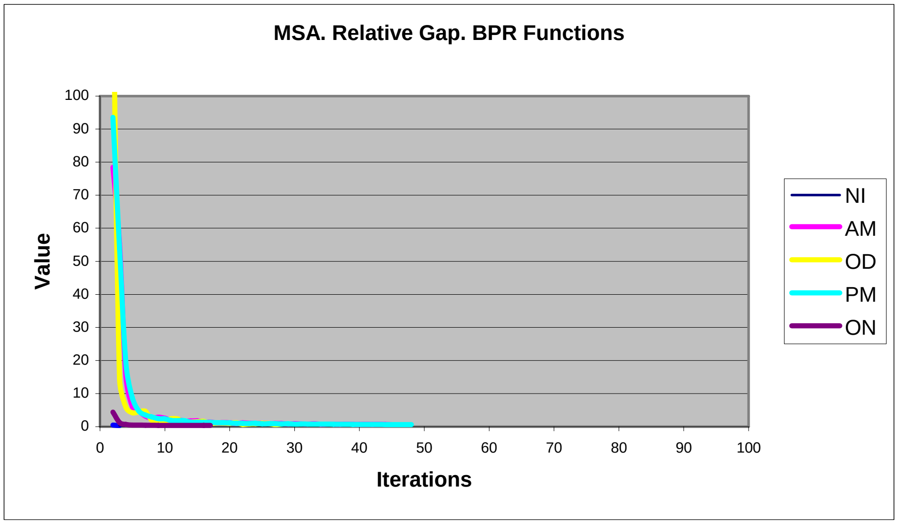

Figure 12. MSA Convergence : Relative Gap for BPR functions

The next step was to analyse the flows. Figures 13 to 15 show a comparison between the link flows obtained with both algorithms for the AM period. The solutions can not be proven to be unique, however the MSA and Gauss Seidel algorithms yield very close solutions, in particular for the autos and the heavy trucks. The same result is found for the other periods of the day. For the case of the Montreal network observed flows were available for almost all the classes and all the periods of the day. Table 11 lists the regression parameters obtained from the comparison between the observed and the simulated flows. The fit is excellent.

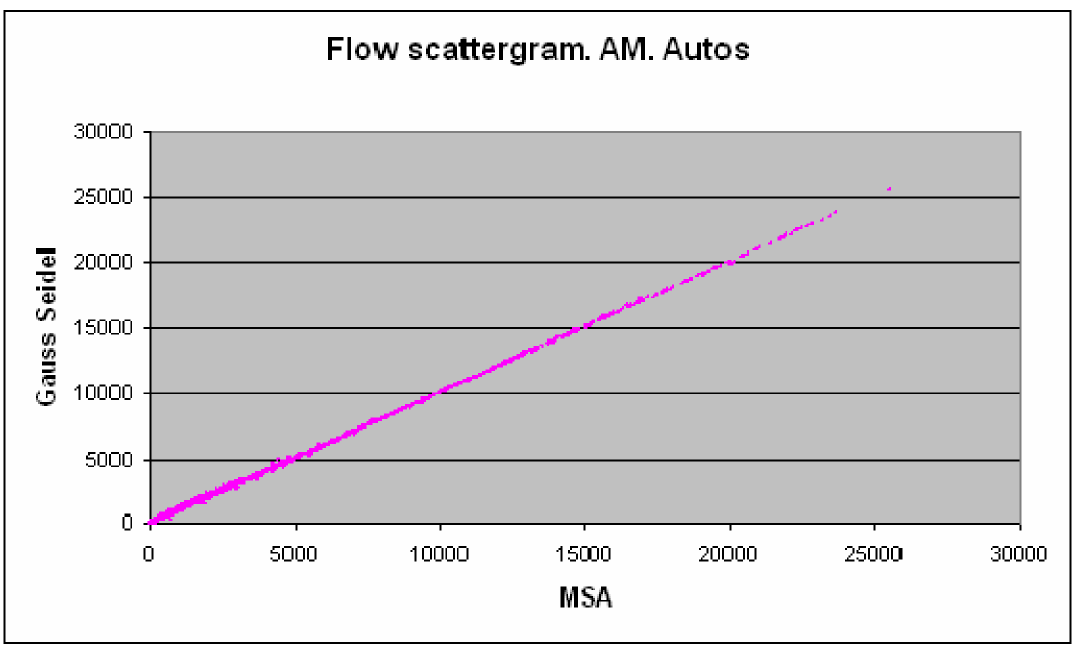

Figure 13. Link flow comparison: Autos, BPR Functions, 5 internal iterations

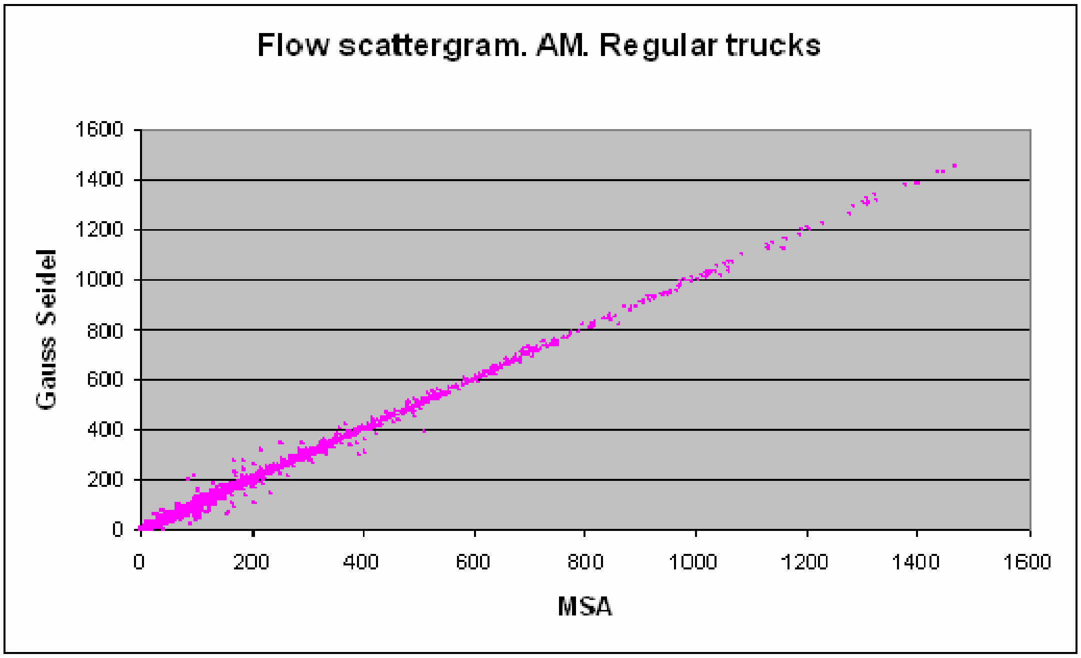

Figure 14. Flow comparison: Regular trucks, BPR Functions, 5 internal iterations

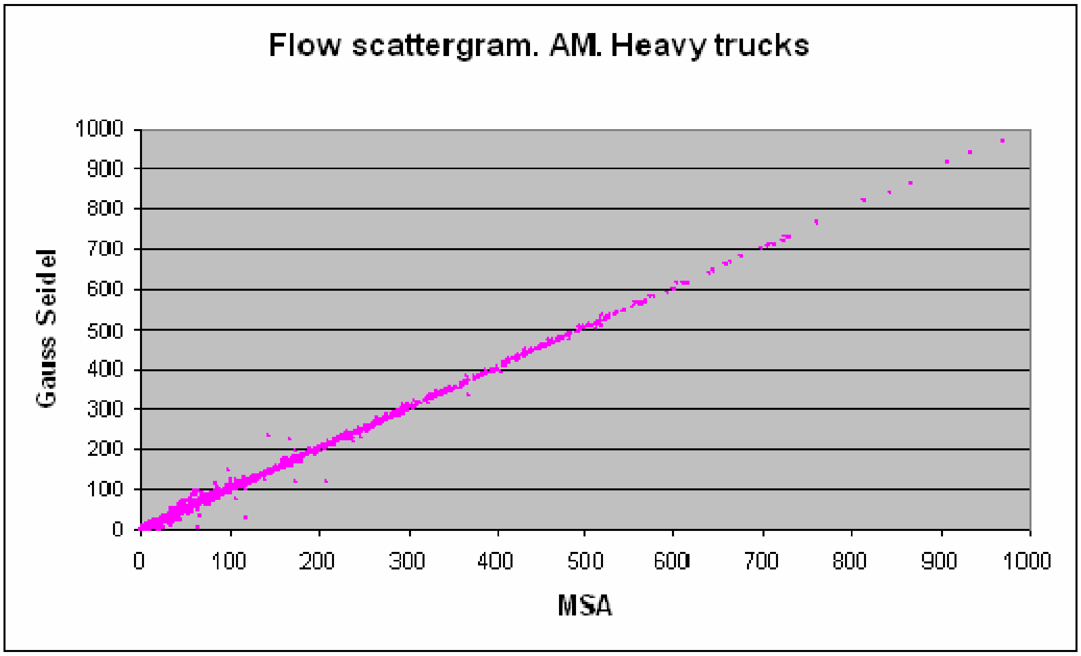

Figure 15. Flow comparison: Heavy trucks, BPR Functions, 5 Internal iterations

	Class	MSA				Gauss Seidel – 5 Internal iterations			
		A	B	R^2	RSTD	A	B	R^2	RSTD
NI	Auto	14.22	1.00	0.99	101.16	13.67	1.00	0.99	100.87
	Regular truck	-1.26	0.95	0.97	11.15	-0.81	0.94	0.93	16.10
	Heavy Truck	-0.60	1.00	0.98	11.54	-0.63	1.00	0.98	11.76
AM	Auto	-248.07	1.07	0.90	1,253.84	-248.14	1.07	0.90	1,255.17
	Regular truck	-20.42	1.02	0.91	61.89	-20.40	1.02	0.91	62.96
	Heavy Truck	3.64	0.91	0.93	34.57	3.28	0.91	0.93	34.24
OD	Auto	17.82	0.94	0.92	1,370.18	8.65	0.94	0.92	1,373.65
	Regular truck	-77.80	0.96	0.94	153.27	-77.23	0.95	0.94	151.15
	Heavy Truck	-8.51	0.97	0.97	63.89	-8.97	0.97	0.97	65.96
PM	Auto	-516.03	1.14	0.88	1,535.21	-518.49	1.14	0.88	1,542.22
	Regular truck	-26.41	1.00	0.91	55.80	-26.57	1.00	0.90	57.28
	Heavy Truck	-4.25	0.96	0.95	35.54	-4.49	0.96	0.95	35.66
ON	Auto	56.97	1.06	0.94	823.35	57.12	1.06	0.94	818.98
	Regular truck	-28.40	1.00	0.97	55.41	-27.71	1.00	0.96	63.63
	Heavy Truck	-0.93	1.02	0.95	52.17	-1.06	1.02	0.95	52.71

Table 11. Observed vs. Computed flows Regression parameters.
BPR Functions. Stopping criterion $\varepsilon = 0.001$

7. Conclusion

The algorithms tested empirically for this multi-class network equilibrium problem indicate that either the MSA method or the Gauss-Seidel approach yield convergent algorithms. The failure of the Jacobi method to converge for the problem instances that exhibited high congestion levels is probably due to the fact that the Jacobian of the cost functions are not diagonally dominant, which is a sufficient condition for the convergence of this method. However, the problem instances were of much too large scale to attempt a verification of this hypothesis.

It is concluded that this asymmetric network equilibrium problem instance may be solved satisfactorily.

References

Florian M., Hearn D., (1995), *Network Equilibrium Models and Algorithms*, Handbooks in OR & MS Vol. 8, 485-550.

Florian M., Spiess H., (1982), *The Convergence of Diagonalization Algorithms for Asymmetric Network Equilibrium Problems,* Transpn. Res. B Vol. 16B, No. 6, 477-483.

INRO Consultants Inc., (2007), *Emme On-line Documentation*.

Magnanti T. L., Perakis G., (1994), *Averaging schemes for variational inequalities and systems of equations*, Mathematics of Operations Research, 22, 568-587.

Magnanti T. L., Perakis G., (1997), *Solving variational inequality and fixed point problems by averaging and optimizing potentials*, Technical Report OR 324-97, MIT.

Mahmassani H. S., Mouskos K., (1988), *Some Numerical Results on the Diagonalization Algorithm for Networks Assignment with Asymmetric Interactions between cars and trucks*, Transpn. Res. B Vol. 22B, No. 4, 275-290.

Marcotte P., Wynter L., (2004), *A New Look at the Multiclass Network Equilibrium Problem*, Transpn. Science Vol. 38, No. 3, 282-292.

Patriksson, M., (1999), *Nonlinear Programming and Variational Inequality Problems—A Unified Approach*, Applied Optimization Series, vol. 23, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Sheffi Y., (1985), *Urban Transportation Networks*, Prentice-Hall.

Smith M. J., (1979), *The existence, uniqueness and stability of traffic equilibria*, Transportation Research, 13B, 295-304.

Toint Ph., Wynter L., (1996), *Asymmetric multiclass traffic assignment: A coherent formulation*, J.-B. Lesort, ed. Proc. 13th Internat. Sympos. Transportation Traffic Theory, 237-260.

Wardrop J. G., (1952), Some theoretical aspects of road traffic research, Inst. Civil Eng., Part II, 325-378.

Wu J. H., Florian M., He S., (2006), *An algorithm for Multi-class Network Equilibrium Problem in PCE of Trucks: Application to the SCAG Travel Demand Model*, Transpormetrica Vol. 2, No. 1, pp.1-9.