Multi-Class Equilibrium Assignment & O-D Matrix Adjustment

Michael Florian, Yolanda Noriega
CRT, University of Montreal
CIRRELT

21st Annual International EMME Users Conference. Toronto, October 2007

Multi-Class Equilibrium Assignment & O-D Matrix Adjustment

Yolanda Noriega, Michael Florian,
CRT, University of Montreal
CIRRELT

21st Annual International EMME Users Conference. Toronto, October 2007

Contents

Motivation

- In the context of multi-class assignments that involve private cars and several types of trucks
- the Department of Transport, UK asked if each class of traffic can have its own volume/delay (or speed/flow) function.
- Our answer was : yes! And here is how it may be done.
- (Actually, this version of the multi-class equilibrium assignment attracted attention among academic researchers; many references are available)

The Model and Solution Algorithms

- Several classes of traffic;
- Each class has its own volume (cost)/delay function;
- The cost functions are nonlinear and asymmetric
- Solved by several methods:
- By using an MSA scheme, a Jacobi approach and a Gauss-Seidel decomposition

The MSA 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 new extremal flow.*

$l = l + 1$

For each class $m = 1,\ldots,|M|$:

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.

The MSA Algorithm

STEP 0. Initialization.

Initialize all class flows; set the iteration counter

STEP 1. Compute new « all-or-nothing » assignment

For each class :

* Update link travel costs based on current link volumes

* Carry out « all-or-nothing » assignment of the demand on the current shortest paths to obtain the new solution

STEP 2. Update link flows.

For each class update the link flows by using the MSA step size

STEP 3. Stopping criterion.

Compute the relative gap (*RG*)

If $RG \leq \varepsilon$ STOP; otherwise return to step 1.

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

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

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

Subject to (1) to (3).

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

STEP 2. *Stopping Criterion*

Compute the difference between two successive flows:

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

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

The Jacobi Algorithm

STEP 0. Initialization.

Initialize all class flows; set iteration counter.

STEP 1. Compute a new solution.

Update link costs

For each class solve the single class minimization problem.

For each class update the link class volumes based on the current solution.

STEP 2. Stopping Criterion.

Compute the difference between two successive flows (*DJ*).

If $DJ \leq \varepsilon$ STOP; otherwise return to step 1.

The Gauss-Seidel Algorithm

STEP 0. *Initialization.*

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

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

STEP 1. *Compute a new solution.*

$l = l + 1$

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

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

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:

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

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

The Gauss-Seidel Algorithm

STEP 0. Initialization.

Initialize all class flows; set iteration counter.

Update link costs.

STEP 1. Compute a new solution.

For each class

Solve the single class minimization problem.

Update the link class volumes based on the current solution.

Update link costs

STEP 2. Stopping Criterion.

Compute the difference between two successive flows (*DGS*).

If $DGS \leq \varepsilon$ STOP; otherwise return to step 1.

The Montreal data set

	AM Peak	PM Peak	Off Peak
Zones	1 425	1 425	1 425
Regular nodes	12 771	12 756	13 019
Links	30 681	30 663	32 284
Classes	3	3	3

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

Period	Auto demand	Regular truck demand	Heavy truck demand
NI	246,212	7,048	7,542
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

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

The Montreal data set

The volume/delay functions

- BPR multi-class cost functions:

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

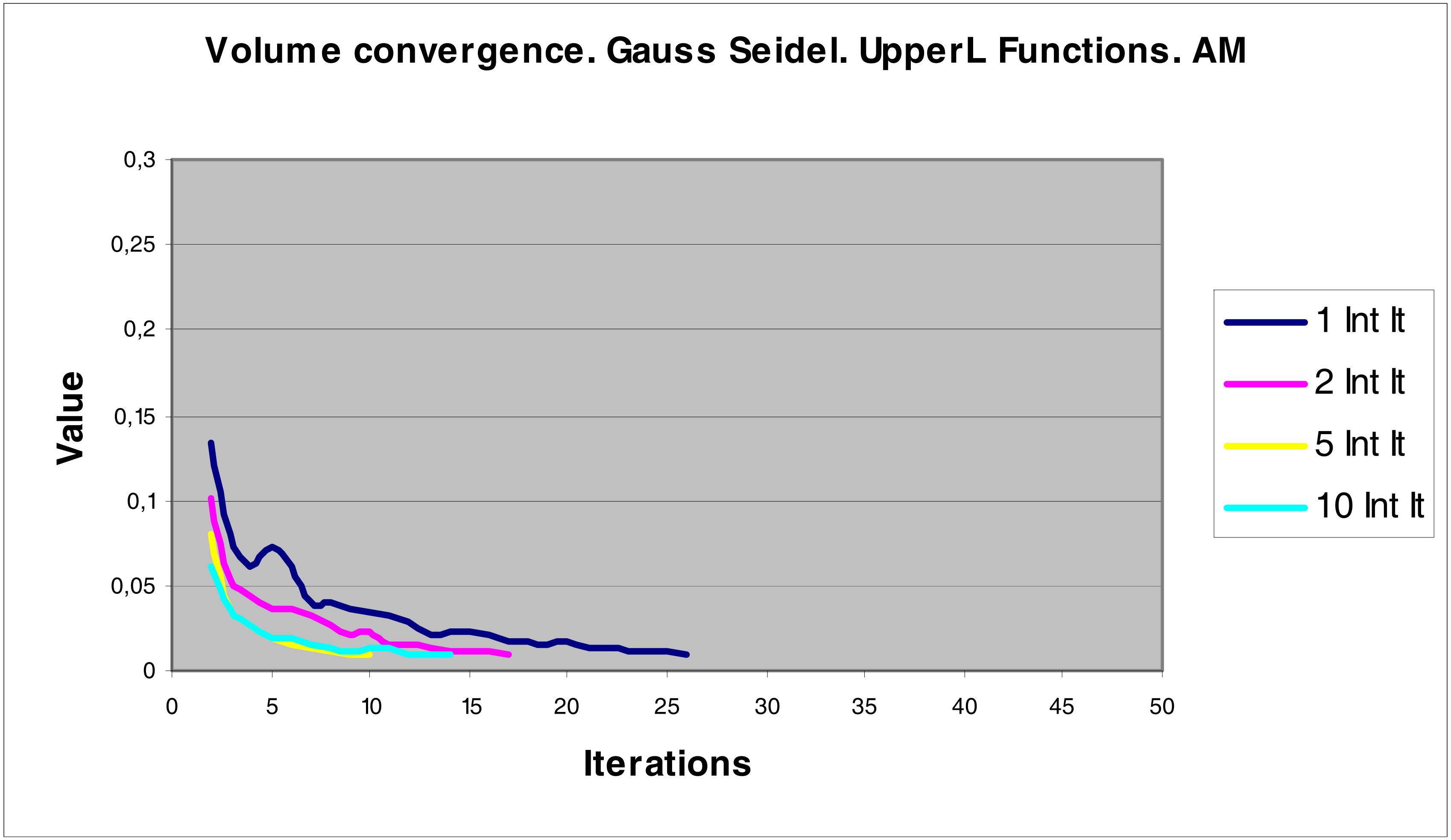
- where $s_a^m(v_a)$ is the travel time on link α for class m ; t_a^0 is the travel time in the link at free flow speed; φ^m is a class time factor (for the slower classes); α_a and β_a are the BPR parameters for the link ; c_a is the capacity of the link (in PCE) and $v_a = \sum_m \gamma^m v_a^m$ (in PCE). Standard values of α and β were used: 0.15 and 4, respectively, for all the links.

- Logistic upper bounded functions:

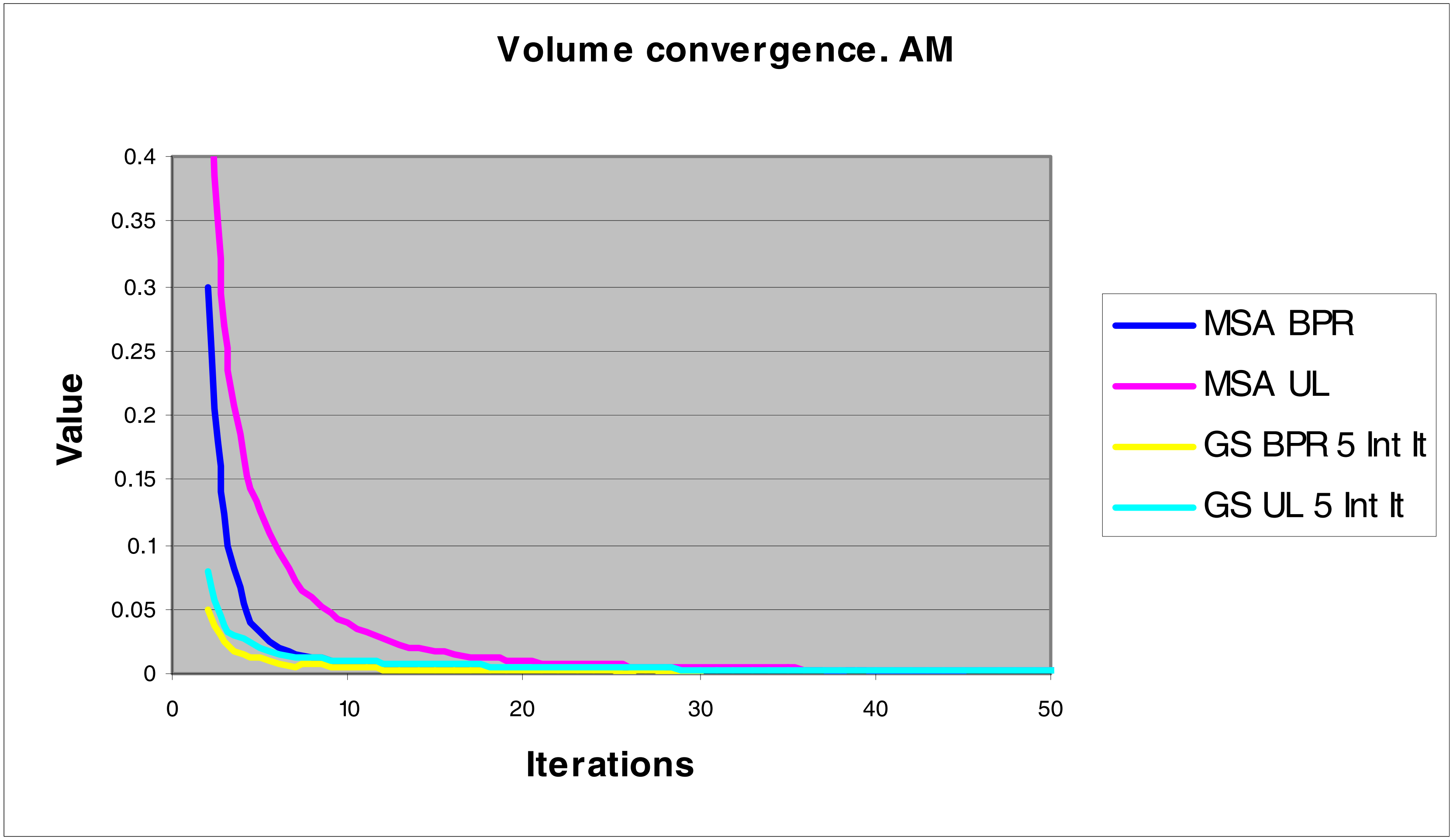
$$s_a^m(v_a) = t_a^0 * \varphi^m * (1 + \frac{\eta_a}{1 + \alpha_a / ((v_a + \theta_a) / c_a) \wedge \beta_a})$$

- 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:
- α_a, θ_a and φ^m were previously calibrated for all the links of the network.

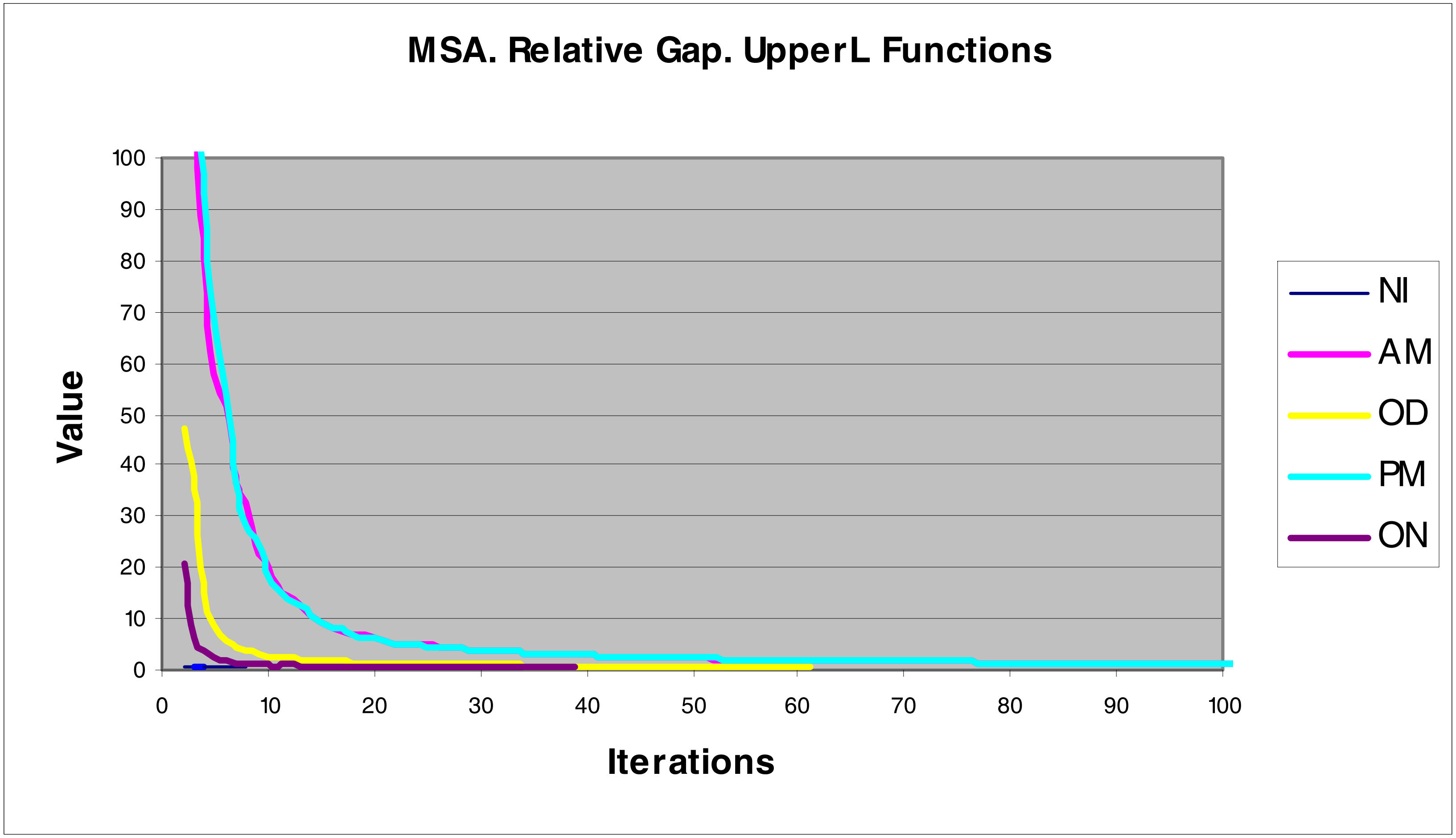
Computational results: GS



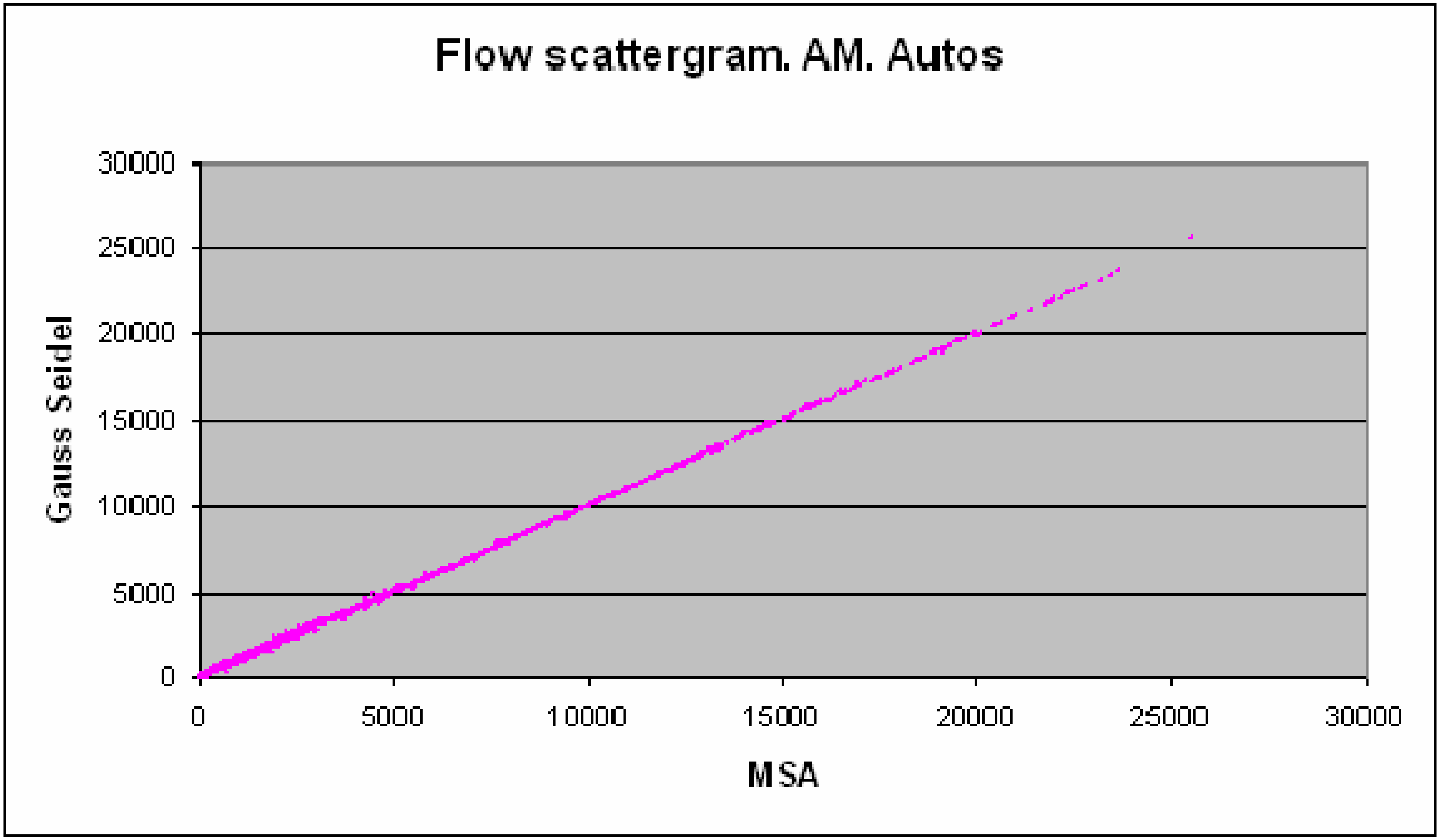
Computational results: MSA-GS



Computational results: MSA



Comparison: GS vs. MSA

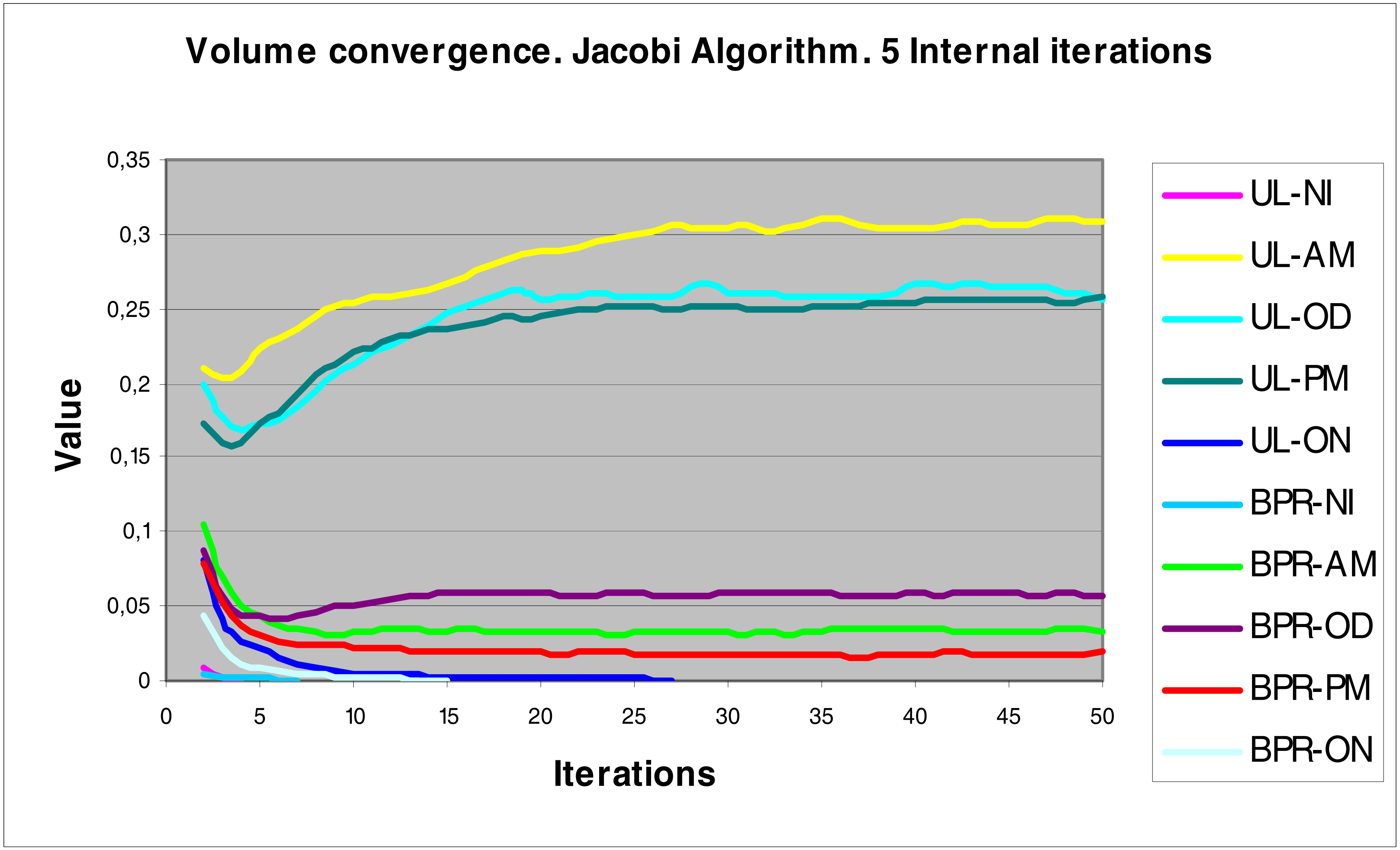


Computation times (sec.)

Period	MSA UL	Gauss Seidel 1 Int. It. UL	Gauss Seidel 5 Int. It. UL	MSA BPR	Gauss Seidel 5 Int. It. BPR
NIGHT	253	153	263	129	146
AM PEAK	3226	4276	4178	1352	1778
OFF PEAK DAY	2018	2312	2254	921	1449
PM PEAK	2965	3808	3982	1429	1701
OFF PEAK NIGHT	1231	1146	1444	535	575

Stopping criterion $\varepsilon = 0.001$

Computational results: Jacobi



Multi-Class OD Matrix adjustment

- Over the past 15 years, ever since the gradient method for O-D matrix adjustment had been made available, we had questions from Emme users about extending the method for multi-class OD matrix adjustment;

- Until the release of Emme 3 one could do sequential adjustments by class;

- Now, one can use the simultaneous multi-class path analysis for implementing multi-class OD matrix adjustments;

- The multi-class method was implemented and results on the Montreal region network will be shown.

The Mathematical Model

$$Min\ Z(g) = \frac{1}{2} \sum_{m \in M} \sum_{a \in \hat{A}} (v_a^m - \hat{v}_a^m)^2$$

Subject to

v=assign(g)

Minimize the sum of squared differences between observed flows and assigned flow.

The Multi-Class Algorithm

Step 0. Initialization. Iteration $l = 0$

Step 1. Multi-class assignment of demand $g^{m\,l}$ ($\forall m \in M$) to obtain link volumes $v_a^{m\,l}$ for $a \in A, \quad m \in M$

Step 2. Computation of the link derivatives $(v_a^{m\,l} - \hat{v}_a^{m\,l})$ for $a \in \hat{A}, \quad m \in M$

and the objective function $\frac{1}{2}\sum_{m \in M}\sum_{a \in \hat{A}}(v_a^{m\,l} - \hat{v}_a^{m\,l})^2$

If the maximum number of iterations L is reached go to Step 7.

Step 3. Multi-class assignment with path analysis to compute the gradient matrix:

$$\nabla Z(g)^{m\,l} = \frac{\partial Z(g^m)^l}{\partial g_i^{m\,l}} = \sum_{k \in K_i^{m\,l}} p_k^{m\,l} \sum_{a \in \hat{A}} \delta_{ak}^{m\,l}(v_a^{m\,l} - \hat{v}_a^{m\,l})$$

Step 4. Multi-class assignment with path analysis to obtain the derivatives:

$$v_a^{m\,l\,\prime} = -\sum_{i \in I} g_i^{m\,l} \nabla Z(g)^{m\,l} \sum_{k \in K_i^{m\,l}} \delta_{ak}^{m\,l} p_k^{m\,l}$$

Step 5. For each class $m \in M$

Computation of the maximal gradient as $\max g^{m\,l} = \max(\nabla Z(g)^{m\,l} / g_i^{m\,l})$

Computation of the optimal step length as $\lambda^{m\,l\,*} = \dfrac{\sum_{a \in \hat{A}} v_a^{m\,l\,\prime}\,(\hat{v}_a^{m\,l} - v_a^{m\,l})}{\sum_{a \in \hat{A}} v_a^{m\,l\,\prime\,2}}$

Update of the demand matrix:

$$g_i^{m\,l+1} = g_i^{m\,l} + \min(\lambda^{m\,l\,*}, 1) * \nabla Z(g)^{m\,l} / \max g^{m\,l}$$

Step 6. Update the step counter $l = l + 1$ and go to Step 1.

Step 7. End

The Multi-Class Algorithm

Step 0. Initialization.

Step 1. Multi-class assignment to obtain the link volumes.

Step 2. Computation of the link derivatives and the objective function.

If the maximum number of iterations is reached STOP.

Step 3. Multi-class assignment with path analysis to compute the gradient matrix.

Step 4. Multi-class assignment with path analysis to obtain the derivatives.

Step 5. For each class

Computation of the maximal gradient.

Computation of the optimal step length.

Update of the demand matrix.

Step 6. Update the step counter and go to Step 1.

Three Approaches for Multi-Class Adjustments

Three approaches :

•The first approach is the multi-class adjustment where the demand for every class is adjusted iteration by iteration (MC Adjustment).

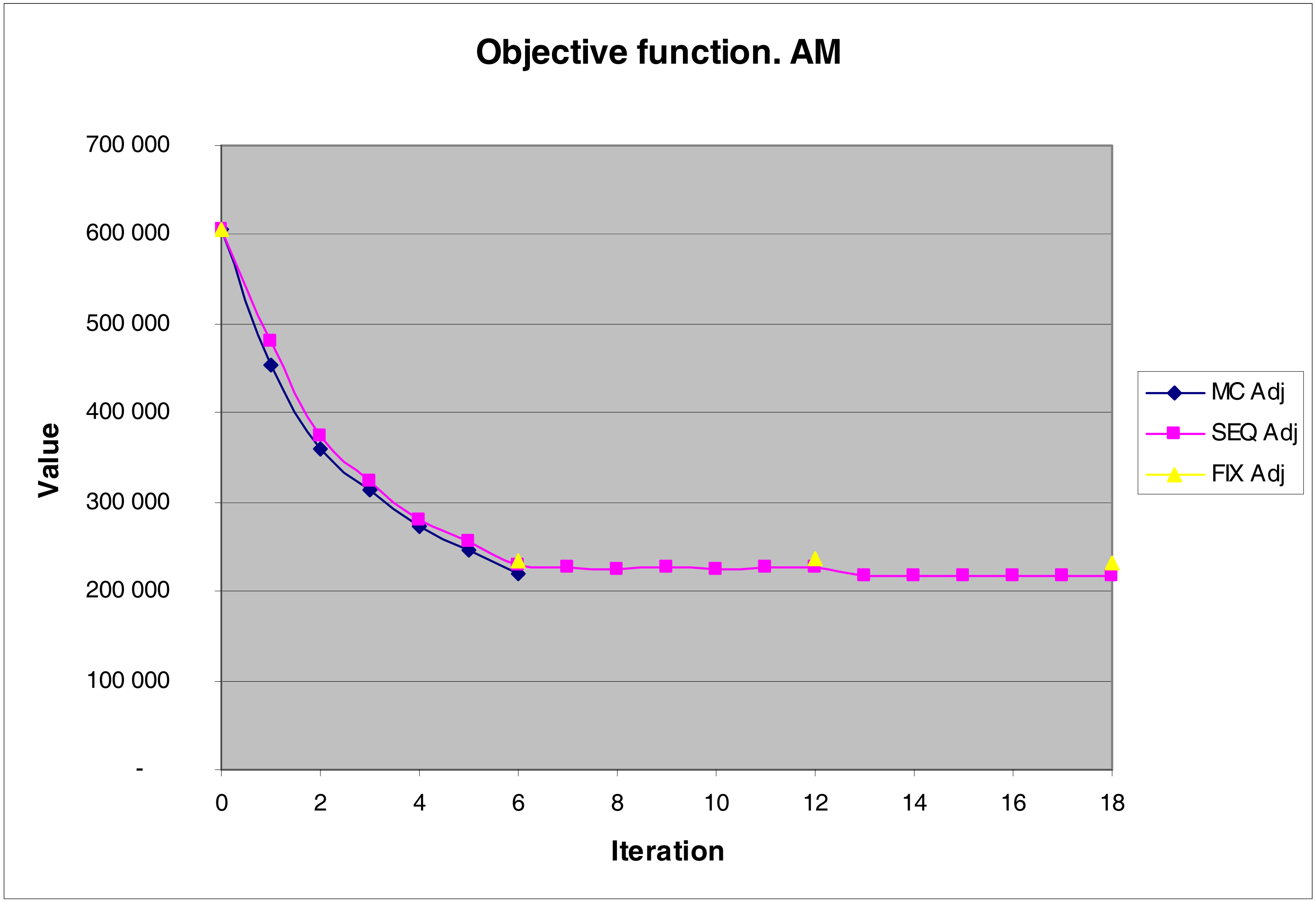
•The second approach consists on adjusting the demand for one class at the time, leaving however the flows of all classes variable during the assignments (SEQ Adjustment).

•Similarly, in the third approach the demand of one class is adjusted at the time, but here the volumes of the other classes are considered as fixed (FIX Adjustment).

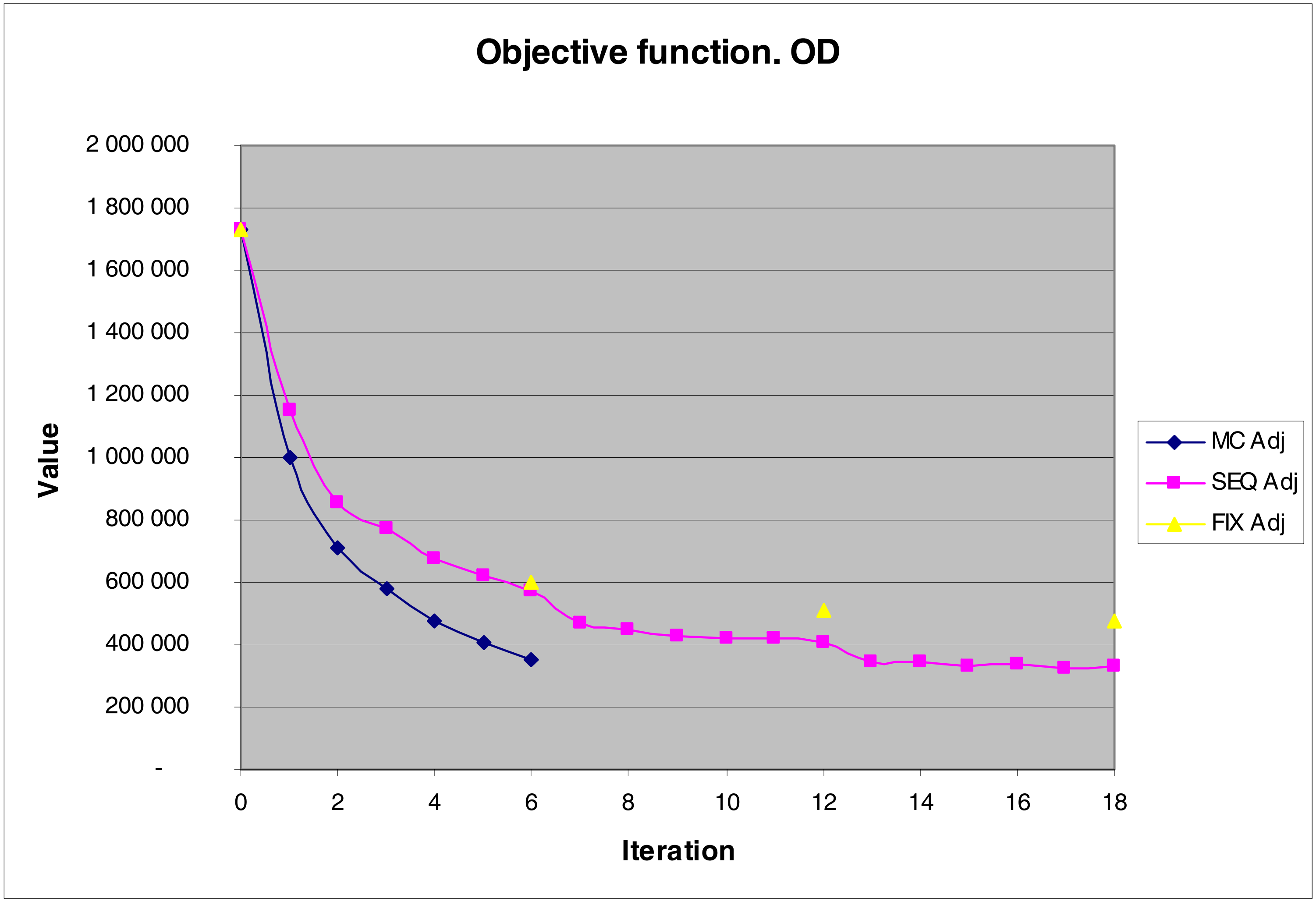
Computational results: R^2

		To be adjusted	MC Adjustment
NI	Auto	0.96	0.99
	Regular truck	0.93	0.96
	Heavy truck	0.91	0.98
AM	Auto	0.93	0.97
	Regular truck	0.87	0.93
	Heavy truck	0.88	0.95
OD	Auto	0.92	0.98
	Regular truck	0.92	0.96
	Heavy truck	0.94	0.97
PM	Auto	0.92	0.97
	Regular truck	0.87	0.92
	Heavy truck	0.83	0.94
ON	Auto	0.94	0.98
	Regular truck	0.93	0.96
	Heavy truck	0.92	0.97

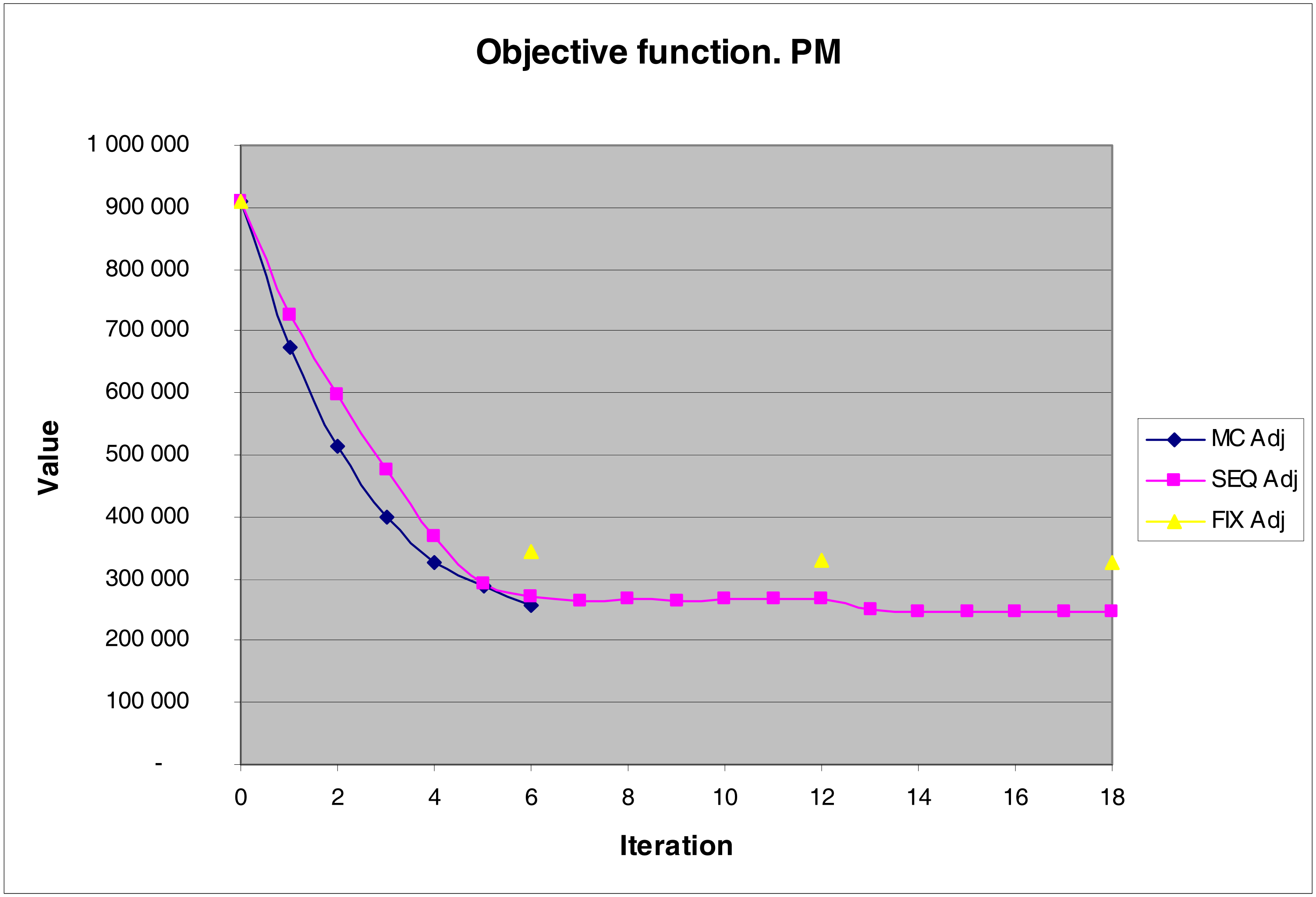
Computational results



Computational results



Computational results



Computational results: the demand

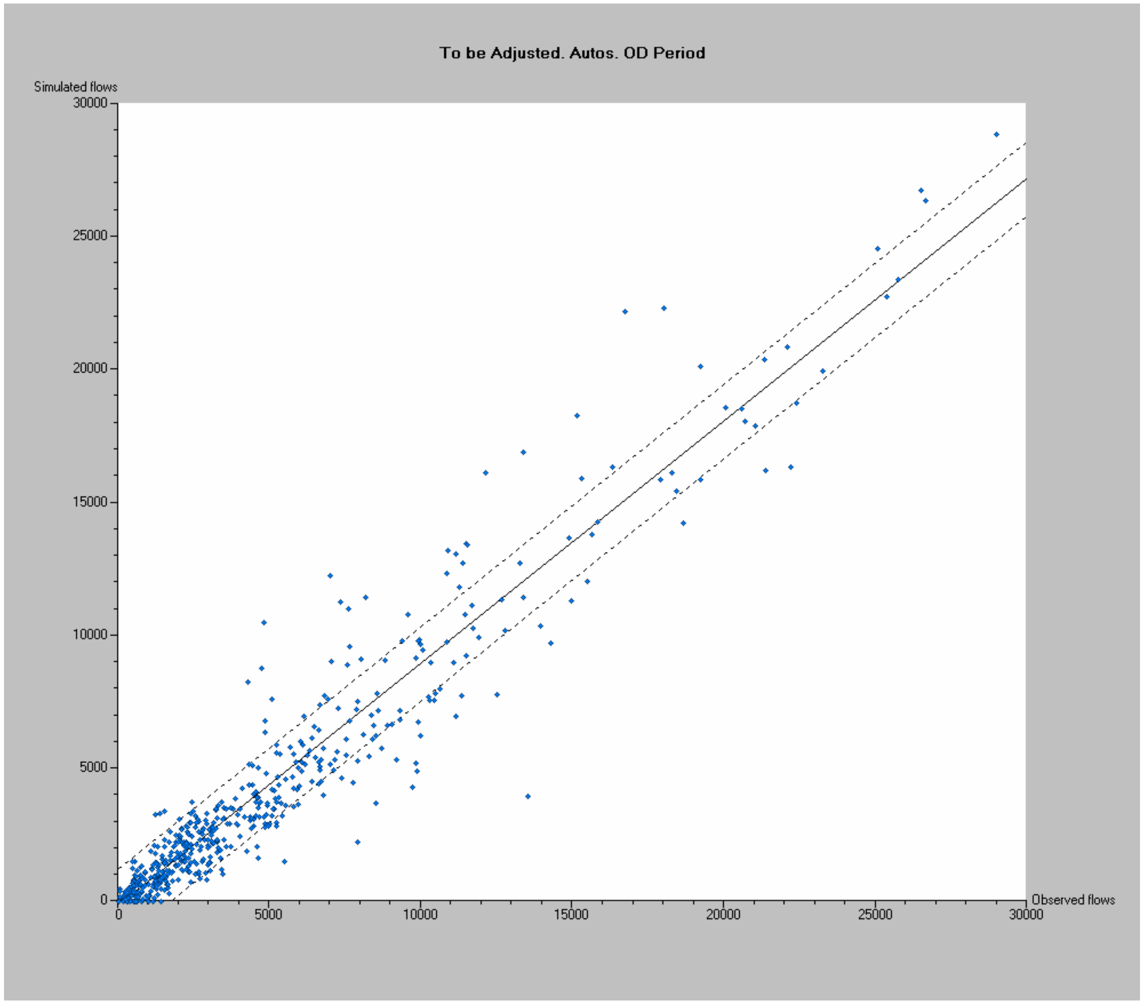
		To be adjusted	MC Adjustment	SEQ Adjustment	FIX Adjustment
NI	**Auto**	205 673	214 667	214 681	214 692
	Regular truck	5 785	6 832	6 813	6 791
	Heavy truck	6 086	6 916	6 914	6 918
AM	**Auto**	805 512	841 179	840 747	844 152
	Regular truck	21 619	25 574	25 512	26 942
	Heavy truck	12 486	14 808	14 723	15 024
OD	**Auto**	1 566 806	1 720 990	1 750 499	1 741 977
	Regular truck	68 808	89 154	89 782	92 642
	Heavy truck	33 445	39 802	39 564	39 878
PM	**Auto**	1 021 659	1 073 093	1 081 084	1 062 908
	Regular truck	19 812	24 399	24 649	24 855
	Heavy truck	11 986	14 722	14 690	15 170
ON	**Auto**	822 865	850 881	853 489	854 037
	Regular truck	38 798	46 290	46 559	46 712
	Heavy truck	22 303	23 642	23 614	23 593

Computation times (min.)

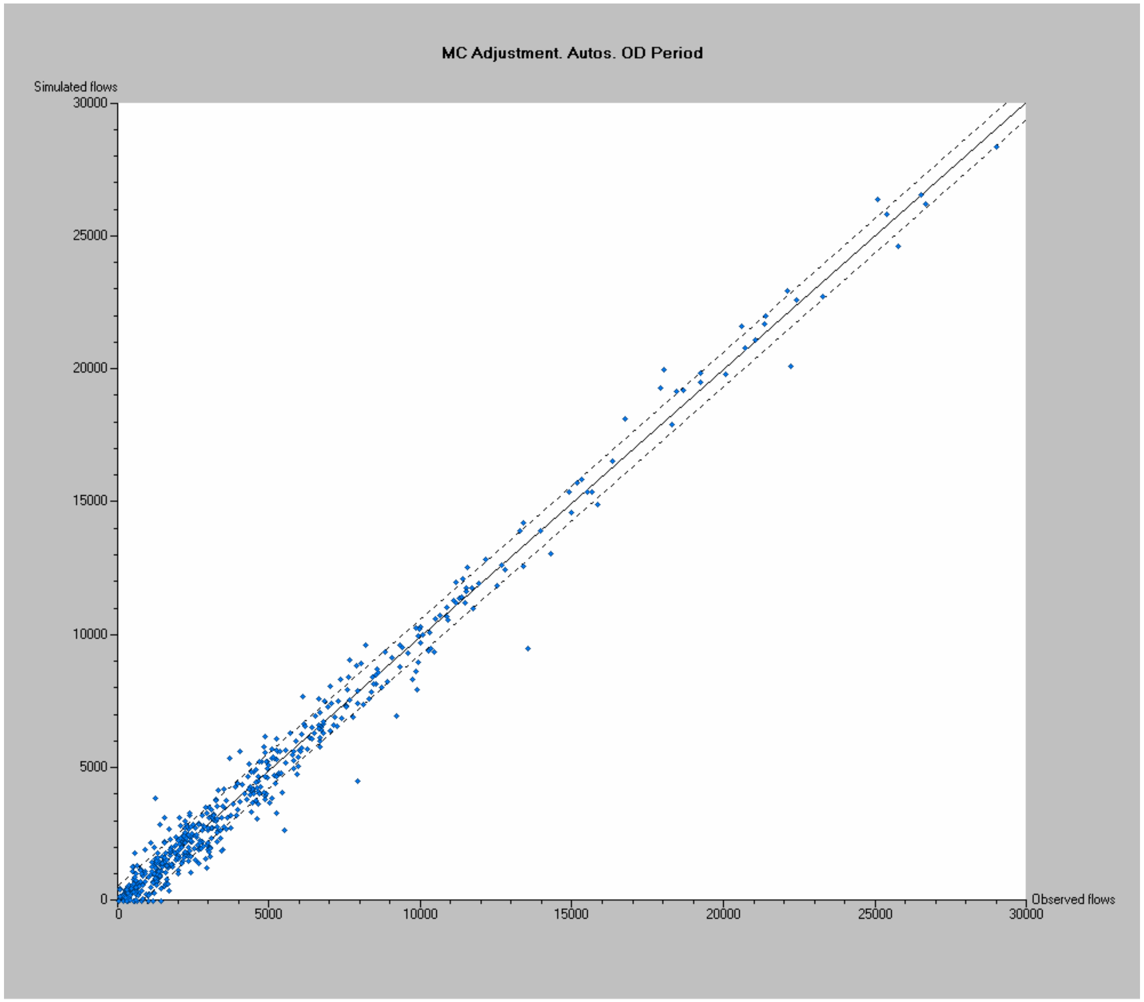
On Dell Intel Core 2 Duo 2.15 Mhz, WinXP

Period	Mod. MC Adj.	Mod. SEQ Adj.	Mod. FIX Adj.
NI	30	58	21
AM	548	1 018	189
OD	426	835	177
PM	524	882	170
ON	163	319	67

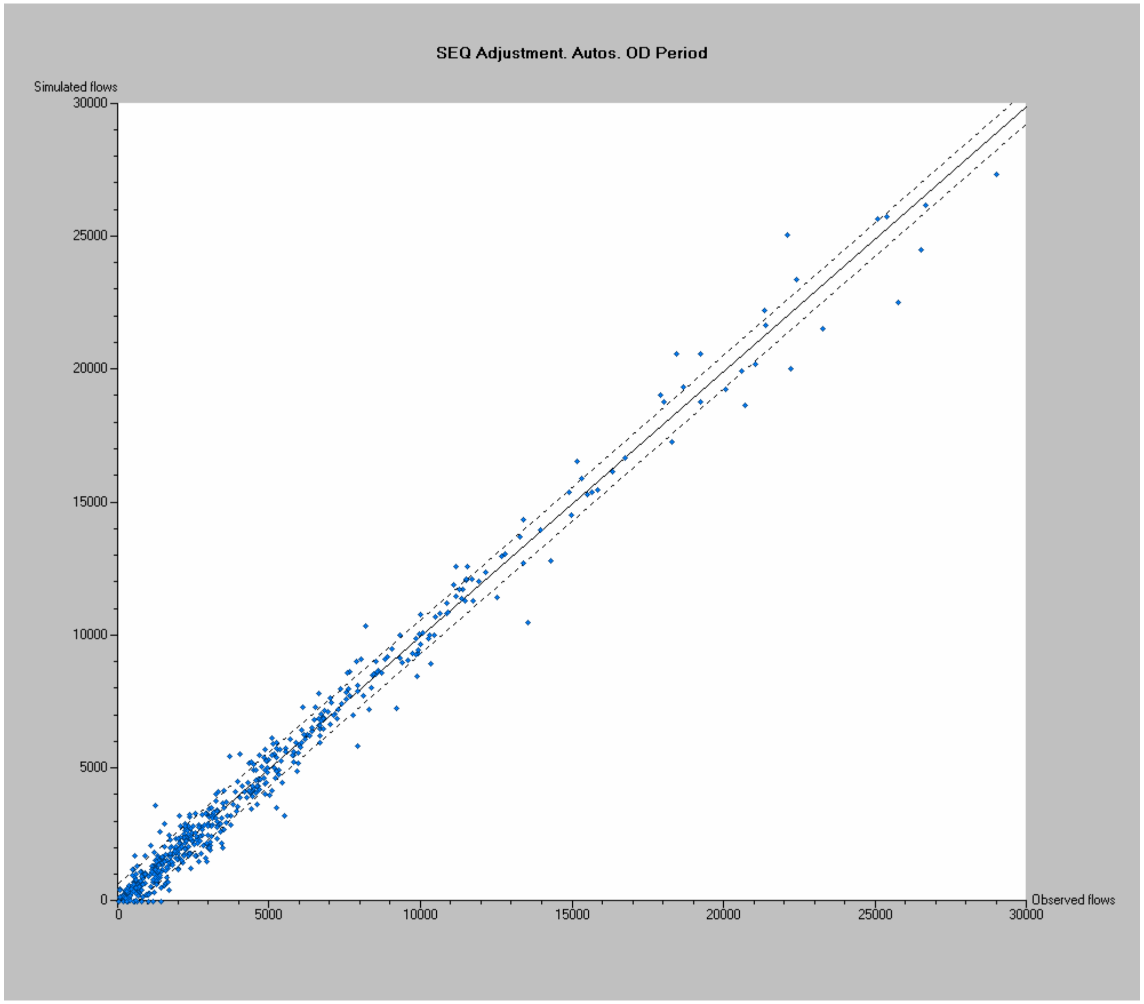
Auto Demand to be Adjusted



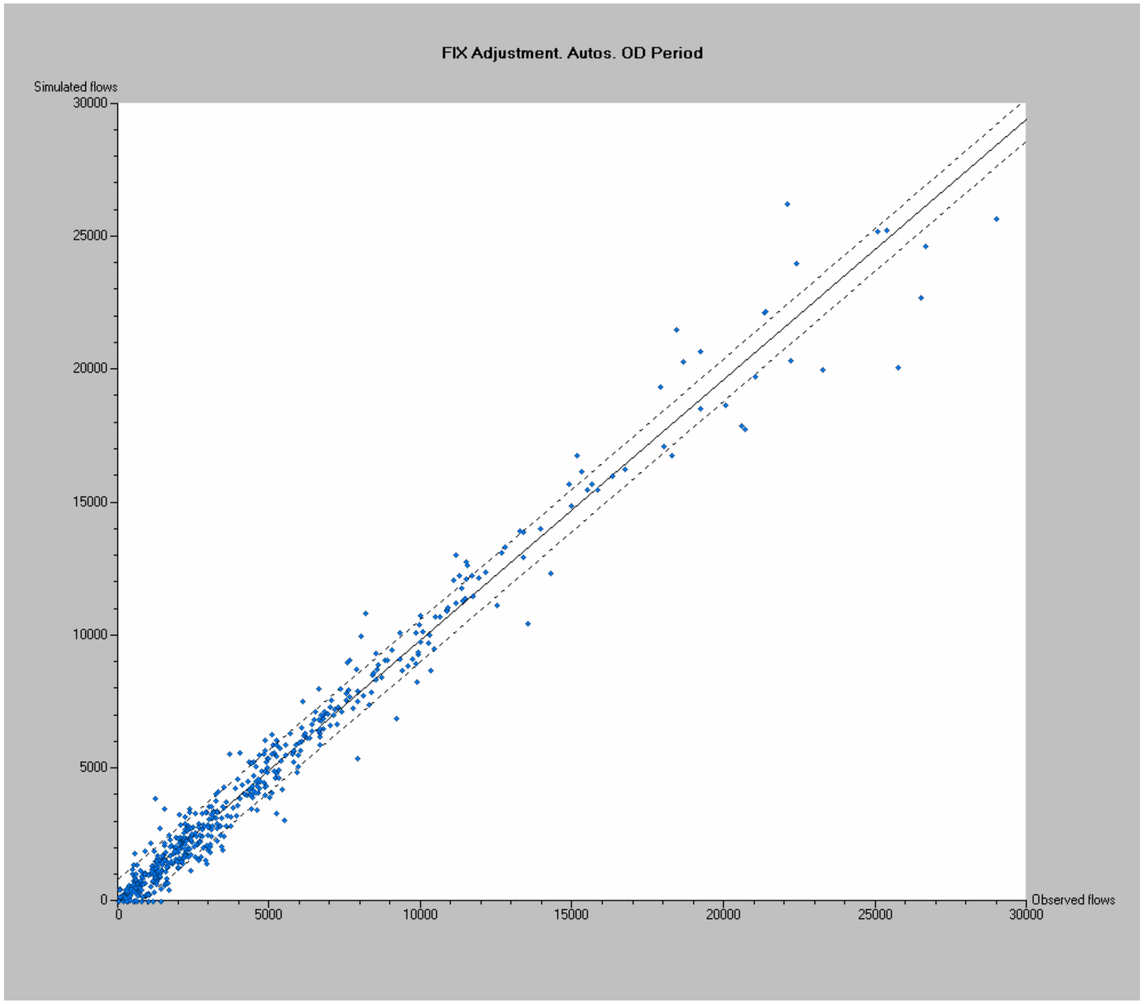
MC Adjustment of Auto Demand



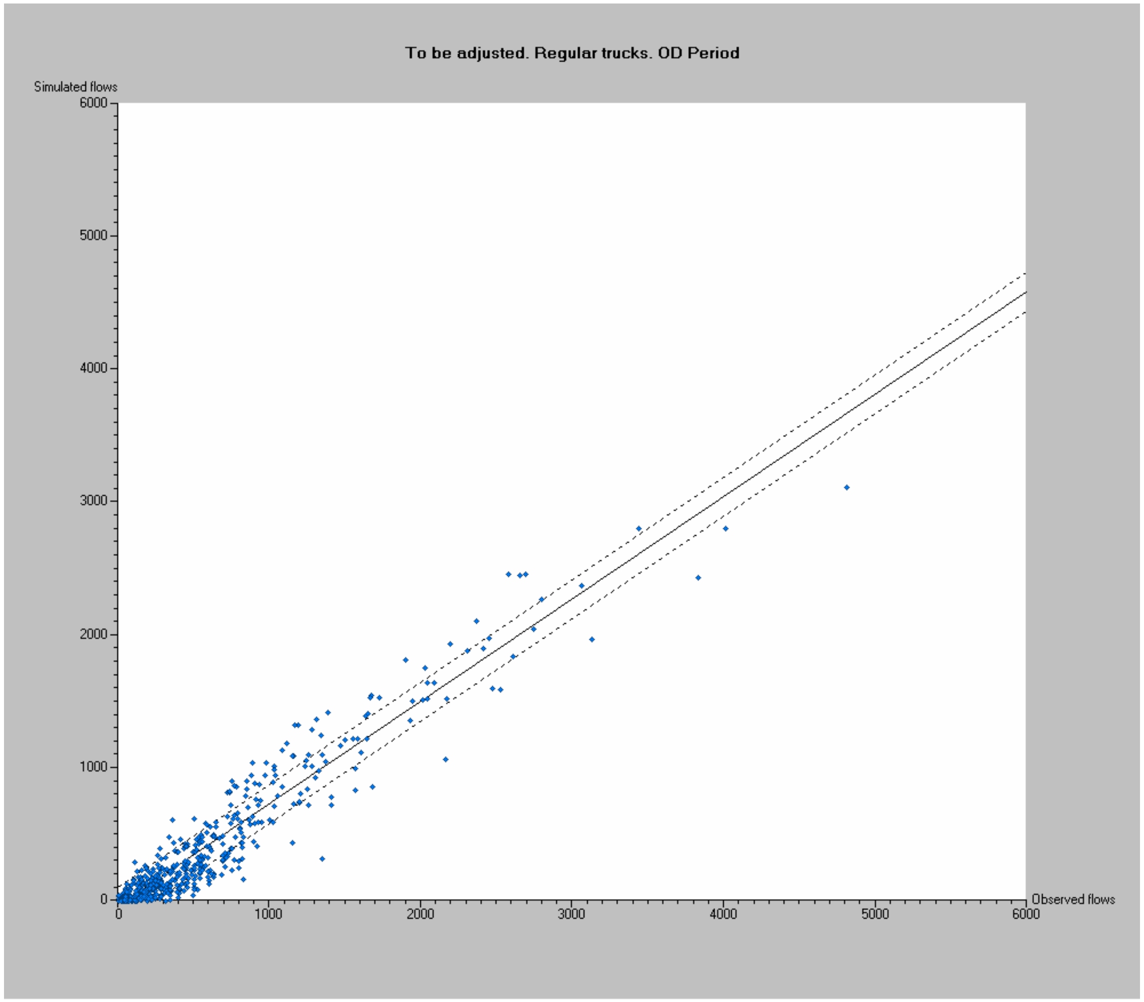
SEQ Adjustment of Auto Demand



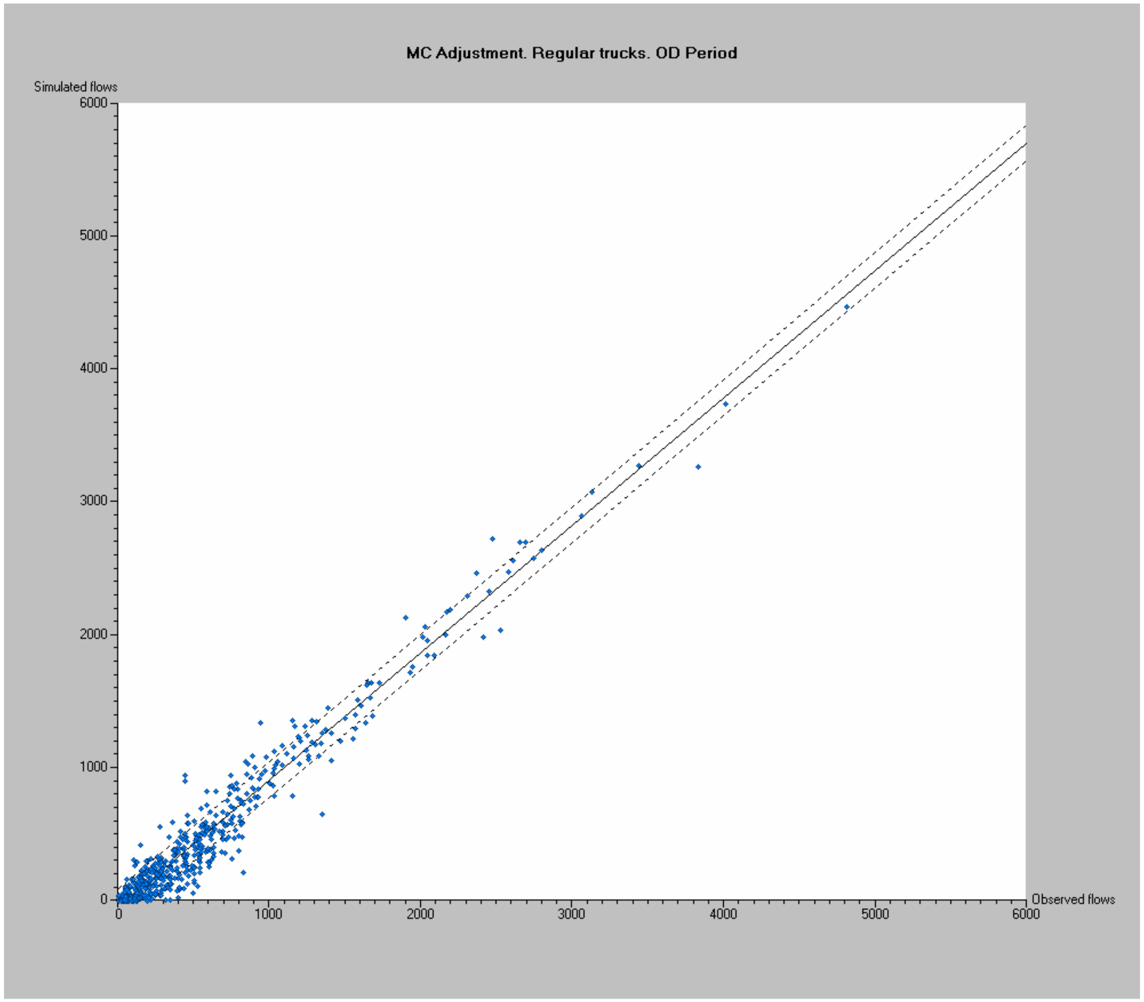
SEQ Adjustment of Auto Demand



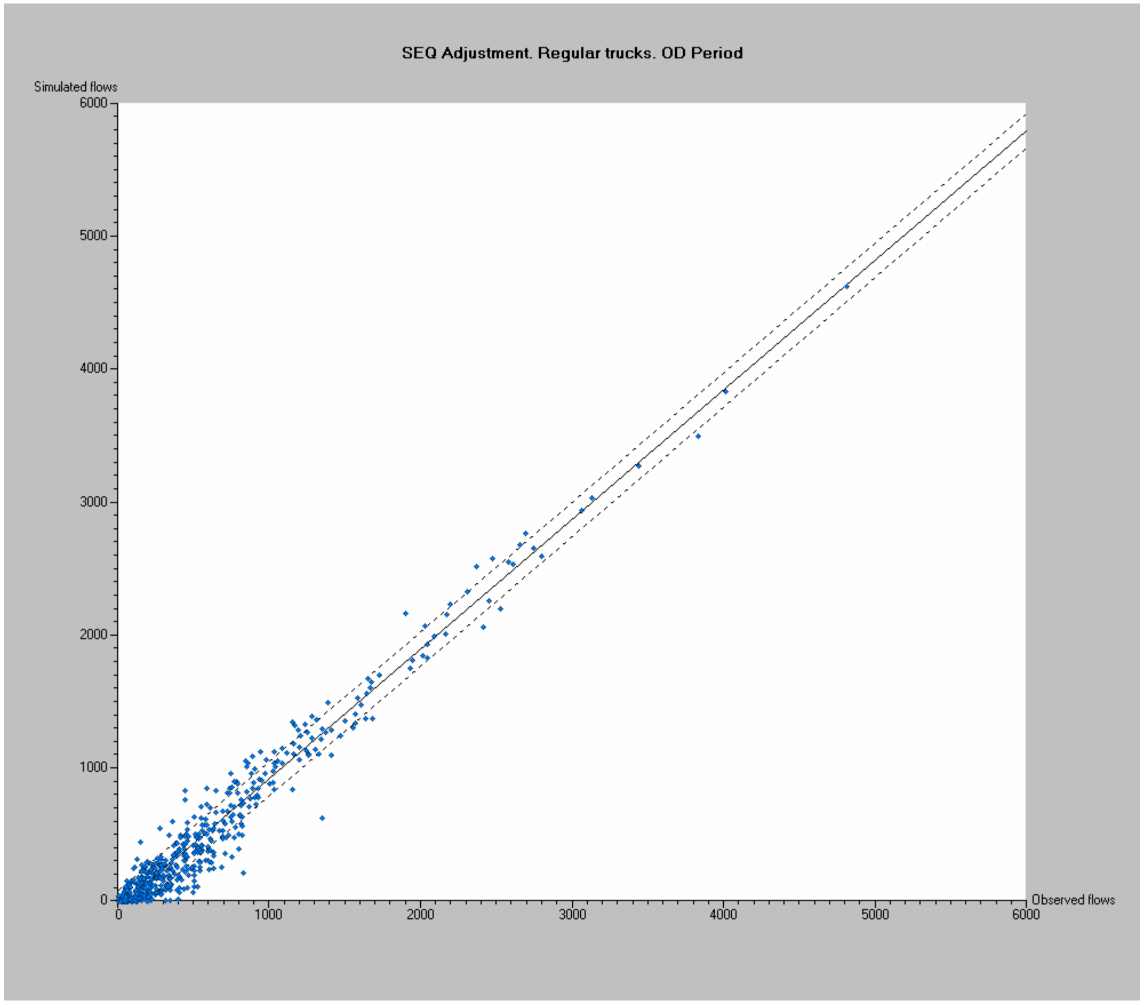
Regular Trucks to be Adjusted



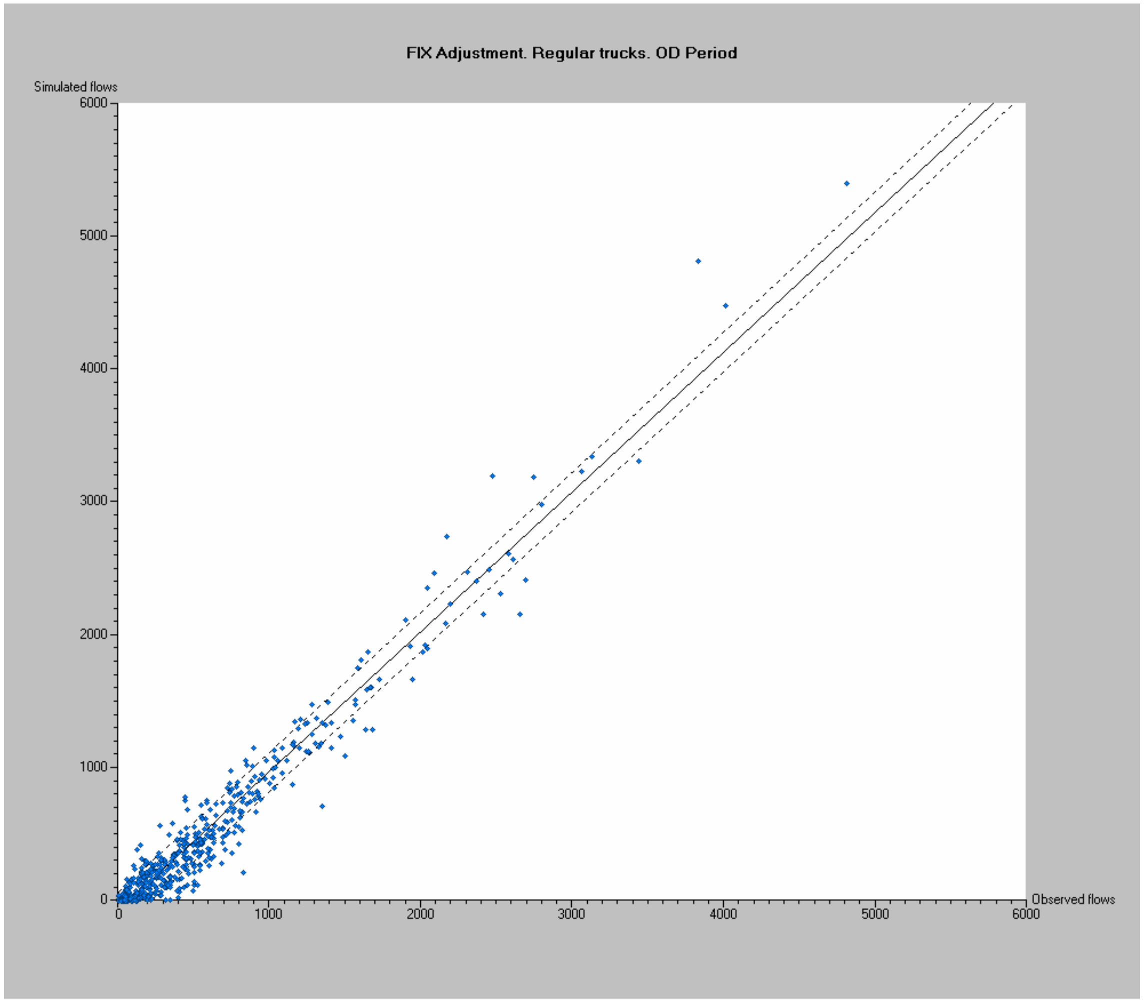
MC Adjustment of Regular Trucks Demand



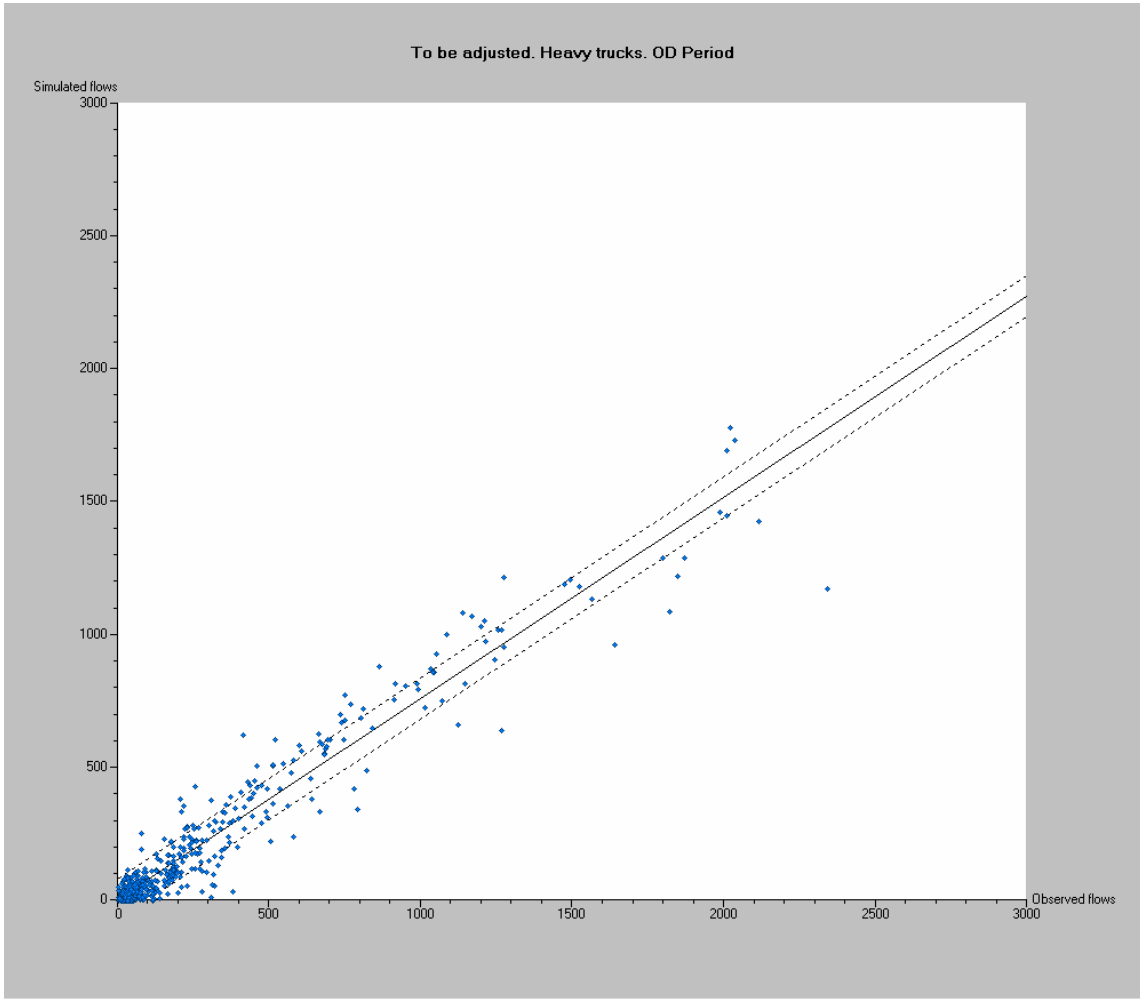
SEQ Adjustment of Heavy Trucks Demand



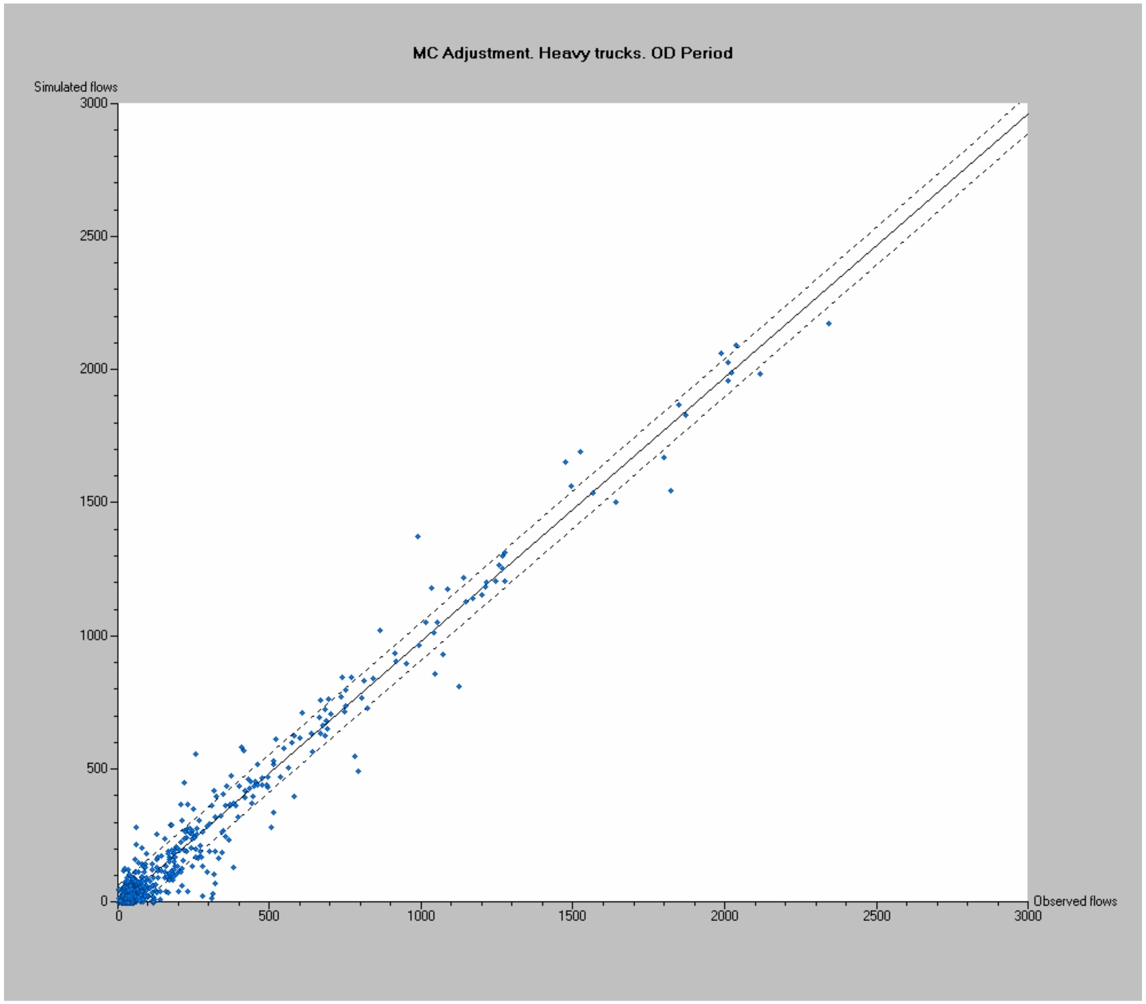
FIX Adjustment of Auto Demand



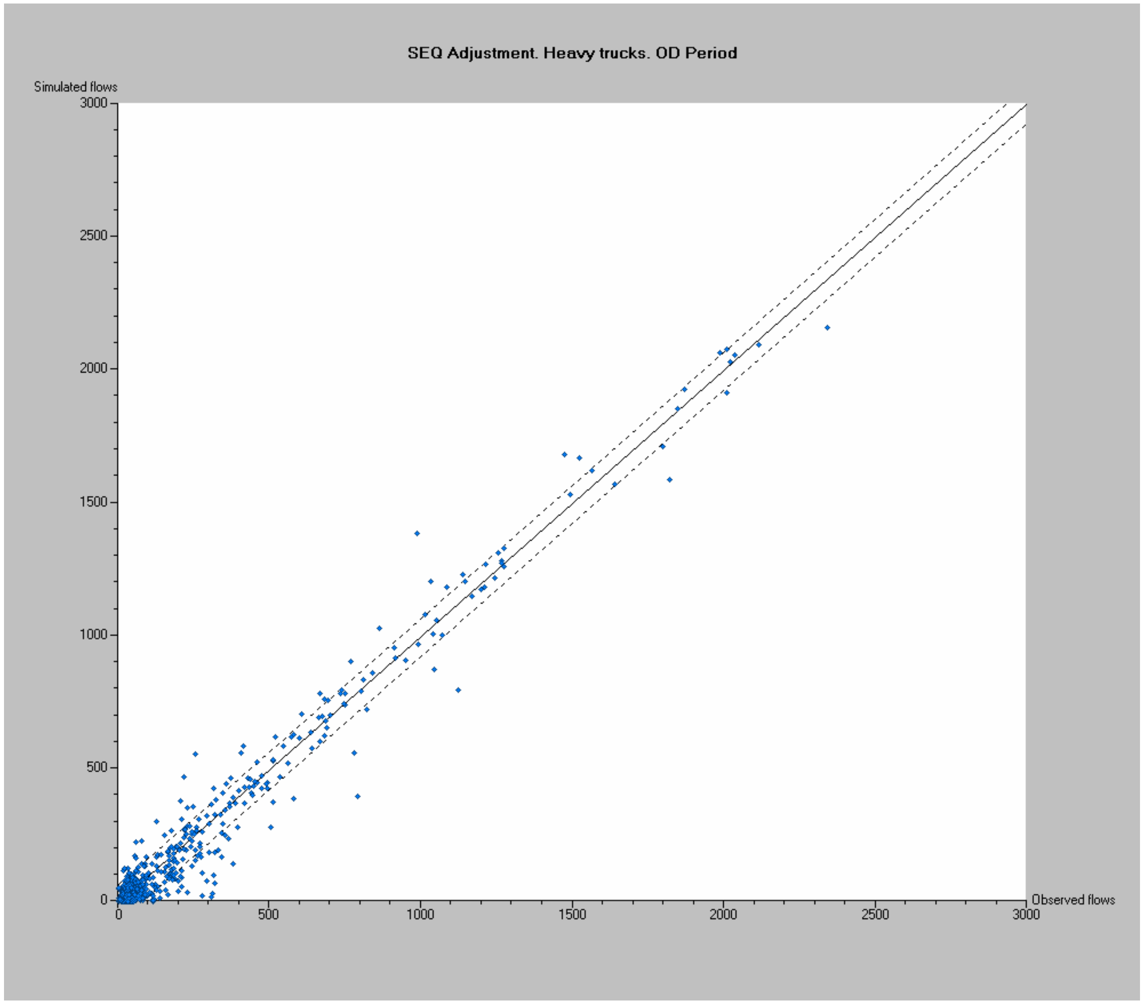
Demand to be Adjusted Heavy Trucks



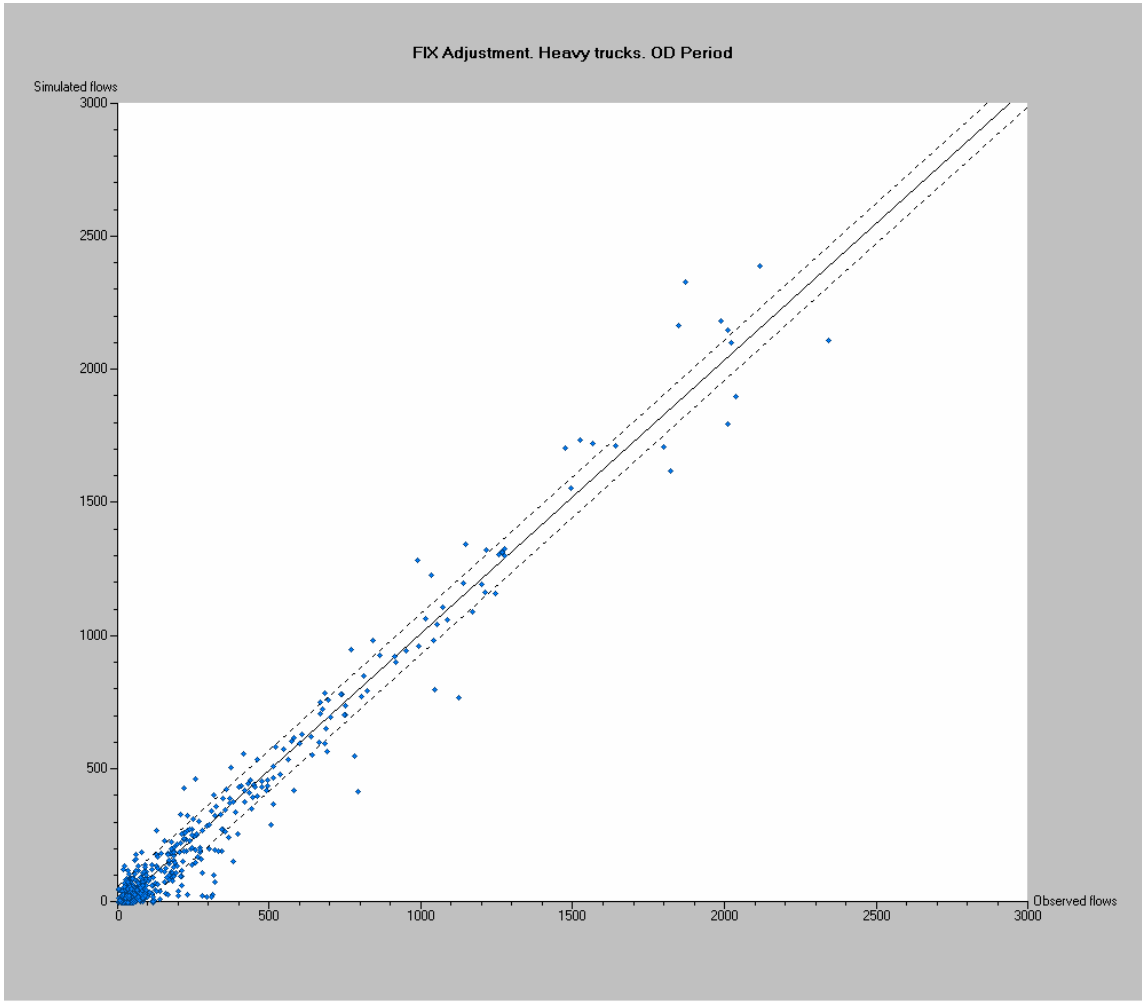
MC Adjustment of Heavy Trucks Demand



SEQ Adjustment of Heavy Trucks Demand



FIX Adjustment of Heavy Trucks Demand



Conclusion

One can do several interesting models with the multi-class assignment of Emme, if needed.