

Equilibrium traffic assignment on large Virtual Networks: Implementation issues and limits for multi-modal freight transport

Bart Jourquin*** and Sabine Limbourg*

* Facultés Universitaires Catholiques de Mons (FUCaM),

Group Transport & Mobility (GTM)

Mons

Belgium

e-mail: gt&m@fucam.ac.be

** Universiteit Hasselt

Diepenbeek

Belgium

EJTIR, 6, no. 3 (2006), pp. 205-228

Received: February 2005

Accepted: March 2006

Multi-Modal freight models are traditionally built following the well known “four steps model” in which generation, distribution, modal-split and assignment are seen as separated modules. An alternative approach, now implemented in some software, is to represent the multi-modal network by means of a “mono-modal” one, in which each particular transport operation (loading or unloading operation, transshipments ...) is represented by a dedicated “virtual link”, that represents a specific operation in the transportation chain.

This approach, promoted by several authors, often referenced to as “super networks” or “virtual networks”, is proven to give interesting results, but has the drawback to generate much larger networks than the pure geographic representation of the studied area. It has also some kind of “hidden trap”, linked to transport distances, that will be presented in this paper and that can only be solved using appropriate assignment techniques.

This paper presents some results obtained on a large multi-modal network, using different equilibrium assignment algorithms, in order to test their ability to give an appropriate solution to the “distance trap”. It however concludes that the implementation of classical equilibrium assignment techniques leads to solutions that are barely different from the one obtained by a simple all-or-nothing assignment, opening the way to alternative multi-flow solutions.

Keywords: freight transport, multimodal, assignment, GIS.

1. Introduction

Until a few years ago, transport models were essentially focused on passengers' flows. More recently, some freight specific network models have been developed but, like passengers models, they are essentially analysing networks from a "link" point of view rather than from a "node" point of view. Even if some of them deal to some extent with the different operations performed at the nodes, i.e. loading /unloading, transshipment or simple transit, their output still targets mainly transport flows on the networks.

As a result of a trend towards economic globalisation, just in time deliveries and road transports expansion, a great deal of attention is paid nowadays to reorganising the networks and intermodal transports, developing new technical bundling concepts in order to substitute transport solutions with less negative external effects to direct road transport. Thus, there is a need for a better modelling of the functions assumed by nodes, i.e. terminals and transshipment platforms, because the costs of the operations performed at these nodes are important in the total cost of transport. Indeed, a geographical multimodal transport network is not only made of links like roads, railways or waterways, on which vehicles move but also of connecting infrastructures like terminals or logistics platforms that exist at the nodes. To analyse transport operations over the network, costs or weights must be attached to the links over which goods are transported as well as to the connecting points where goods are handled. However, most of these transport or handling infrastructures can be used in different ways and with different costs. For example, boats of different sizes and operating costs can use the same waterway; at a terminal a truck's load can be transhipped on a train, bundled with some others on a boat or simply unloaded as it reaches its destination. Normally, the costs of these alternative operations are different. In order to model this, one of the solutions is to represent each kind of operation in a node as a specific link of a "virtual network", for which a relevant cost is then computed.

Beyond this brief introduction, this paper will present an overview of the "virtual network" methodology implemented in specific software and its "hidden trap". It will then discuss the most used assignment techniques and how they can be applied to virtual networks. Finally, the performances of these methods will be discussed on a test and a large real-world network.

2. Virtual networks and software implementation

2.1 The building of a virtual network: an intuitive approach

A simple geographic network does not provide an adequate basis for detailed analyses of transport operations as the same infrastructure, link or node, can be used in different ways. To solve this problem, the basic idea, initially proposed by Harker (1987) and Crainic et al. (1990), is to create a virtual link with specific costs for a particular use of an infrastructure. The concept of "supernetworks" proposed by Sheffi (1985), that proposed "transfer" links between modal networks also provides a somewhat similar framework. The NODUS software proposes a methodology and an algorithm that creates in a systematic and automatic

way a complete virtual network with all the virtual links corresponding to the different operations, which are feasible on every real link or node of a geographic network. This systematic and automatic approach, built upon a special codification of the virtual node labels, is probably the biggest benefit over other software tools such as STAN (Crainic et al. (1990)), in which most of the tasks that are possible at a given node are to be introduced “by hand”. The software, which is completely geographically referenced in its latest version, and its underlying methodology were already discussed in Jourquin (1995) and Jourquin and Beuthe (1996). More recent works (see for instance Tavasszy (1996)) include some discussion about virtual networks and the software that implements them.

The methodology can be presented first in an intuitive way by using the example of a simple multi-modal network, as shown in figure 1. This network consists of 4 nodes (a, b, c, d) and 5 links (1 to 5). The “W” links represent waterways and the “R” links rail tracks. The numbers after these letters correspond to the transportation means that can be used on the links. So, “W1” represents a waterway that can only support small barges and “W2” a waterway that can be used by both small and large barges.

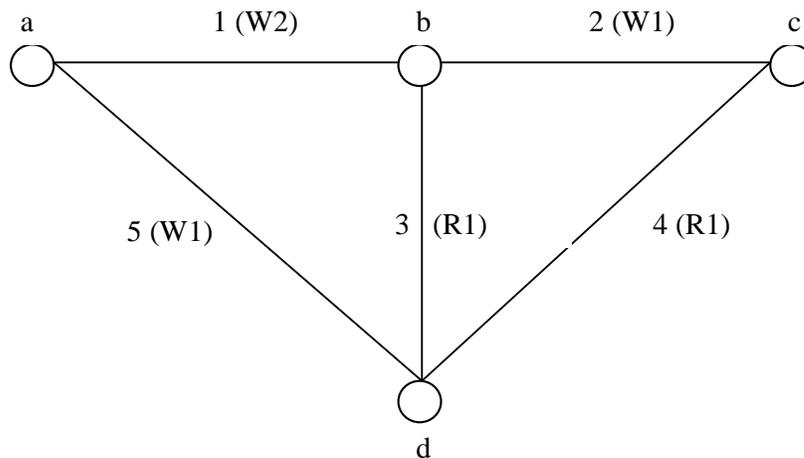


Figure 1. Multimodal network

To go from node “a” to “d”, it could be that route through links 1+3 and using large barges and trains is less expensive than route 5, using exclusively small barges. Indeed, computing costs and routes on this kind of networks is not immediate:

- a) Different costs can be assigned on a single link, depending on which transportation means is used. In this example, the use of a small barge on link 1 has a different cost than the use of a large barge on the same link;
- b) The same is true for the nodes: in the given example, the simple transit of a small barge from link 1 to link 2 can be done at no cost, but the transshipment from a large barge onto a train that will go on link 3 represents an important cost.

This problem can be usefully handled on the corresponding virtual network illustrated in Figure 2, provided that the relevant costs are attached to each of the virtual links. As can be seen, the solution involves the creation of a set of virtual nodes and a set of virtual links connecting these nodes. Each real link has been split in as many virtual links as there are possible uses; their end-nodes are connected by new virtual links corresponding to simple transit or transshipment operations at the real nodes.

In this way, this network with multiple means use is represented by a unique but more complex network on which each link corresponds to a unique operation with a specific cost. Then, one cheapest path can be computed by means of an algorithm such as the one proposed by Dijkstra (1959). The resulting solution is an exact solution, taking all the possible choices into account.

It is now possible to demonstrate how virtual networks are built on the basis of a rather simple, though somewhat complex notation, which provides a convenient way to link cost functions to virtual links. Table1 enumerates the elements of the real network.

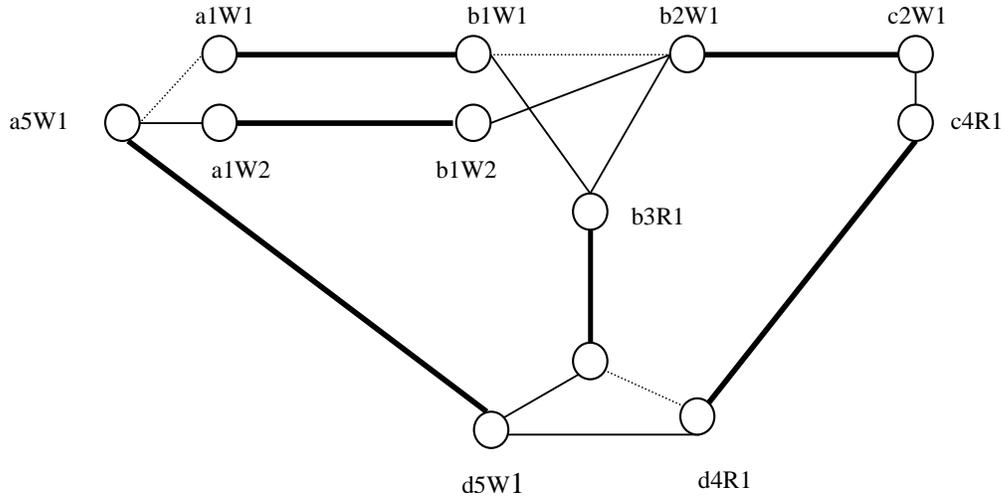


Figure 2. Partial virtual network

Table 1. Real network

Link	Origin node	Destination node	Type of link
1	a	b	W2
2	b	c	W1
3	b	d	R1
4	d	c	R1
5	a	d	W1

In a first step, the virtual links corresponding to the real links, i.e. rail tracks, etc., must be generated. These are defined in table 2 by their end-nodes, the notation of which indicates successively the real node, the real link, the mode and the means they refer to.

Table 2. Travelling links

Real links	End-nodes of virtual links	
1	a1W1	b1W1
	a1W2	b1W2
2	b2W1	c2W1
3	b3R1	d3R1
4	d4R1	c4R1
5	a5W1	d5W1

In a second step, these virtual links must be connected by transit or transshipment virtual links. To keep things as simple as possible, we just enumerate in table 3 the virtual links related to node “b”. They can be viewed in figure 2. In this network, the boldfaced links represent the links of the real network, possibly split up in several virtual links the dotted links represent the simple transit virtual links, while the transshipment links are indicated by a thin continuous line.

Table 3. Connecting virtual links to node b

Real node	End-nodes of virtual links	
B	b1W1	b2W1
	b1W2	b2W1
	b1W1	b3R1
	b1W2	b3R1
	b2W1	b3R1
	b2W1	b3R1

In general, the weight given on a link can very well vary with the direction it is used: loading and unloading operations for instance don’t have necessarily the same cost, and the cost of going upstream on a river is normally higher than going downstream. To solve this problem related to the asymmetry of cost functions, all the virtual nodes are doubled at generation time by adding a + or a – sign to their code; by the same token, all links are split into two oriented arrows connecting these new nodes, as illustrated in figure 3. Those “doubled” virtual nodes and oriented links also permits to avoid “unwanted movements”, like an unloading followed by a loading operation to circumvent a forbidden transshipment operation.

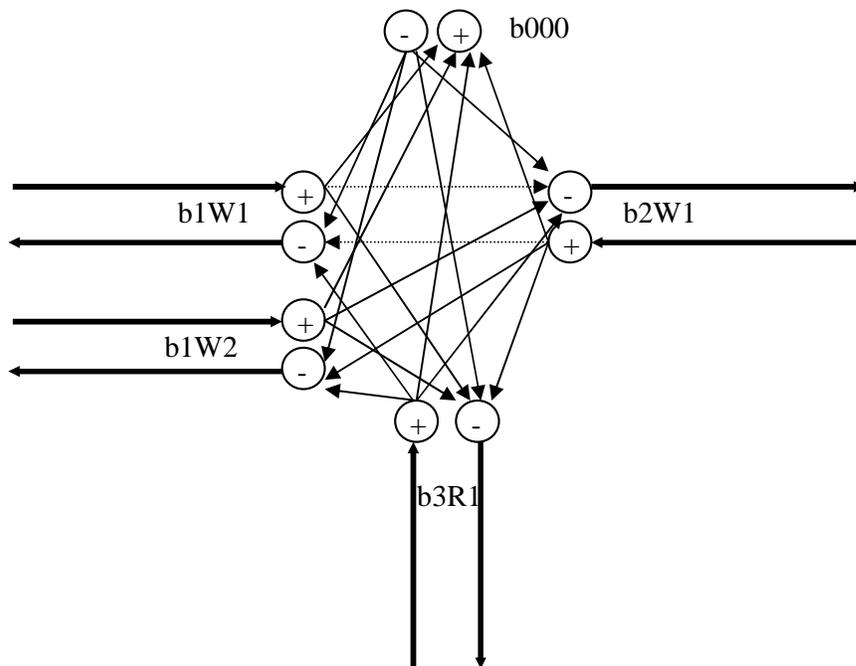


Figure 3. Detailed virtual network at node

Such a network can be completed by introducing entry and exit nodes to the network. This can be done by the creation of additional virtual nodes associated to loading and unloading operations at nodes where these operations are possible. Those entry or exit points in the virtual network are referenced by adding "000" to the real node number. They must also be connected to other nodes by appropriate virtual links. These additional virtual nodes and links appear in the upper part of figure 3.

In order to implement the above code convention, the virtual nodes are coded in the following way: a plus or minus sign, 5 digits for the real node number, 5 digits for the real link number, 2 digits for the transport mode and 2 digits for the transport means. Each label is thus represented by a signed 14 digits number. A virtual link can thus be simply characterized by its origin and destination virtual nodes.

After this overview of the basic methodology, it is necessary to explain the characteristics of the cost functions and how they are connected to the virtual network. As usual in transportation analysis (see, for example, Kresge and Roberts (1971), or Wilson and Bennet (1985)), the "generalised cost" concept is used, which allows to integrate all factors relevant for transport decision making in terms of monetary units. The concept can be defined in different ways according to whether it is the point of view of the shipper or the carrier which is taken, and according to the used unit of reference, i.e. tons, tons-km, vehicles, etc. The specific cost functions which compose the generalised cost, obviously, must be coherent across modes and means, but their functional forms can be freely chosen.

The four types of virtual links require specific cost functions, containing the following elements:

- a) All the costs related to moving a vehicle, such as labour, fuel, insurance, maintenance costs, or tariffs;
- b) The inventory costs of the goods during transportation and other time related costs;
- c) Handling and storage costs or tariffs, including packaging, loading and unloading and services directly linked to transport.
- d) All residual indirect costs like general administrative services which may be assigned to transport on an average basis.

2.2 Computing shortest paths on networks

The contemporary transport systems are used in an intensive way and are often congested in various degrees, particularly in urban areas. Transport models have to determine how the traffic is distributed over the transport network of which the structure and the capacity are known. This is the assignment problem. The results of the assignment, which depend on the sophistication of the implemented method, include an estimate of flows, travel duration and/or corresponding costs, for each link of the network. Unless no capacity constraints are taken into account (All-Or-Nothing assignment), an assignment consists in the distribution of the traffic on a network considering the demand for trips as well as the capacity of the network; the assignment methods distribute the traffic over a network in order to obtain an equilibrium solution. This type of problems can be solved by means of optimisation methods. Other assignment techniques try to spread the flow over several alternatives routes, trying to take into account behavioural considerations.

This paper deals with the methods intended to model flows of commodities on large inter-urban, regional and international multimodal freight networks, like those generated by the virtual network methodology.

With the exception of the new “origin-based assignment”, developed by Bar-Gera and Boyce (2003), which for a given origin considers all the destinations together and optimises the routes serving them¹, all the equilibrium assignment methods are based on the “All-or-Nothing” (AoN) algorithm. The principle of this algorithm is to compute the minimum weight path between each pair of origin and destination, and to allocate the total demand to be transported between these nodes onto this single path. The AoN assignment is itself based on shortest path computations. The efficiency of this latest algorithm is crucial and it is important to choose the one that gives the fastest results for a given problem. The algorithm of Dijkstra solves the problem of the shortest path from an origin to all the possible destinations. The one developed by Johnson finds all the shortest paths between all the pairs of nodes of the graph, combining the algorithm of Bellman-Ford and Dijkstra. The implementation of the algorithm in the computer software, using efficient data structures, also has an important effect on the performances.

For solving the shortest path problem Zhan and Noon (1998) have tested 15 algorithms, among which Bellman-Ford-Moore, Dijkstra, Pape, or Pallottino on real road networks; they conclude that the implementation of Pape is probably the most efficient but that it is worthwhile to consider Dijkstra’s implementations in case a subset only of the destinations from a given node is to be computed. That is exactly the case we have to cope with, as our origin-destination (O-D) matrixes do not contain a demand for each possible O-D pair.

The efficiency of an algorithm can be measured using the notion of time complexity², which can be defined as the number of steps needed to solve an instance of a problem, expressed as a function of its size of.

If a shortest path algorithm solves, in the worst case, a problem with N nodes in a time $c.N^2$, where c is independent of the input size, the required computational effort is $O(N^2)$. This is the case of Dijkstra’s algorithm whereas Johnson’s is $O(N^3)$. These complexities are obtained if the input data is stored in linear arrays. These algorithms can be improved (faster search and insert operations) by using binary heaps instead of linear arrays. Using these heaps, the complexities become respectively $O(M \log_2 N)$ and $O(NM \log_2 N)$. Fibonacci heaps are even more efficient than binary heaps, decreasing the time complexity of the shortest paths algorithms.

The detailed virtual network generated from the European network that we used has more than 500.000 virtual links (M) and 140.000 virtual nodes (N), of which about 1500 (X) are potential (un)loading nodes. As explained earlier, this latest characteristic indicates that Dijkstra’s algorithm is probably the most suitable, since paths must be computed from only a small number of potential origins. As shown in table 4, the algorithm of Dijkstra, with a heap implementation, seems to be the most efficient when applied on a virtual network (which is a low-density graph, since the number of links is much smaller than the square of the number of nodes).

However, despite the fact that, from a theoretical perspective, the Fibonacci heap implementation should perform better than a binary heap based implementation, it appears that the later performs better on the problems that are dealt with here. Indeed, the Fibonacci heap implementation gives smaller execution times only in the most unfavourable cases. During the numerous tests we have performed, these cases represent only about 15% of the test sets. On

¹ Unfortunately, this method is not yet enough documented to be easily implemented in our software

² A complete presentation of the time complexity computation of the algorithms can for instance be found in Cormen et al. (2001), chapter 24.

average, we found out that the computing time for a complete O-D matrix using a Fibonacci heap is about 50% slower than with a binary heap. This is due to the fact that the computer cost of dealing with the Fibonacci heap appears often to be higher than the search cost in simpler heaps.

Table 4. Complexity of shortest-path algorithms

	Data structure		
	Linear (worst case)	Binary heap (worst case)	Fibonacci heap (amortised analysis ¹)
Dijkstra's algorithm Executed X times	$O(XN^2)$	$O(XM \log_2 N)$	$O(XN \log_2 N + XM)$
Johnson's algorithm	$O(N^3)$	$O(NM \log_2 N)$	$O(N^2 \log_2 N + NM)$

Given:

N number of nodes

M number of links

X number of nodes that are a potential origin or destination

¹ An amortised analysis differs from an average case analysis because it guarantees an average performance for the worst cases.

Finally, in order to obtain still faster results, we have introduced a stop criterion in Dijkstra's algorithm. Indeed, it stops as soon as the paths to all the relevant destinations are computed instead of doing the work for all possible destinations. This improves the computing time by more than 50% on real cases, because the commodities are sent to only a few destinations in most cases. Moreover, these destinations are often relatively close to this origin.

2.3 A promising methodology with a hidden trap

The used cost functions for the different transport operations can be relatively detailed and complex, and for instance take into account the nature of the transported goods. This can lead to the use of different transportation modes for different commodities. Nevertheless, the aggregation level of the origin-destinations matrixes that can be produced for very large areas is such that one cannot guess that, for a given category of commodities, everything is transported by the same transportation mode. Indeed, the demand at the European level for instance is often only available at the NUTS1 or NUTS2 level. Even when more sophisticated techniques are used to obtain city to city relations, it is a nonsense to consider that everything that is sent from a given city to another uses the same, unique, transport mode: a factory located nearby a railway station will most probably more often use railway transport than another, located in the same city, but far from the station. Finally, even for a given point-to-point relation between two factories, some internal logistic considerations make it sometimes useful to use alternative transportation modes.

In the classical models, the modal choice is applied as a separate step. In other words, in such an approach, all the quantities for a given mode are separately assigned on their respective modal networks in order that each mode is used on short and long distances with no particular limitations.

However, on "virtual networks", an assignment doesn't search for a cheapest physical itinerary, but for a cheapest path that includes all the (un)loading, transit and transshipment operations, so that the simple notion of transportation mode loses a lot of its original meaning, cre-

ating a new problem. This will now be illustrated by means of a simple example. In table 5, a small demand matrix is described, that contains data for two transportation modes. The last column is obviously not an input data but a simple computation based on the two previous columns. From this table, it is easy to conclude that 17 tons are transported by mode 1 and 16 by mode 2. The tons.km for the two modes are respectively equal to 3850 and 5400.

Table 5. Input data for simple example

Mode	Distance	Tons	Tons.km
1	100	4	400
1	150	7	1050
2	200	6	1200
2	300	4	1200
1	400	6	2400
2	500	6	3000

This example clearly illustrates that distance cannot be considered as the only explanatory variable and that there exists some unobserved factors that explain the mode (and route) choice for individual shipments. This is a classical point that has often legitimated the use of random utility models to describe actual flows on a network. In passenger transport models, the use of such a stochastic approach is useful because the behaviour of the human being during his journeys can be influenced by a lot of, sometimes subjective, factors. This is particularly true in urban networks. This is less the case for long haul freight transport. However, in large scale models, the details of the network is not sufficient to identify the exact location of each firm and the modal networks it is connected to. Finally, the nature of the demand matrixes is so that, even in the best cases, information is only available from city to city and not from a particular firm to another. All these factors make the demand table 1, in which the different transport modes are used on different distance classes, an often encountered and plausible scenario.

Consider that the two transportation modes have linear cost functions, so that $cost = A + B \cdot distance$. The values for A are 2 for mode 1 and 55 for mode 2. The values for B are set to 0.5 and 0.3 respectively.

Figure 4 illustrates these two cost functions, and it can clearly be seen that mode 1 will be chosen for the three first elements of the demand matrix, because they are related to distances that are shorter than the break-even distance. Mode 2 will be used for the remaining entries of the matrix. Thus, 17 tons will be assigned to mode 1 and 16 to mode 2, which are the expected figures. But only 2650 tons.km will be transported by mode 1 and 6600 by mode 2.

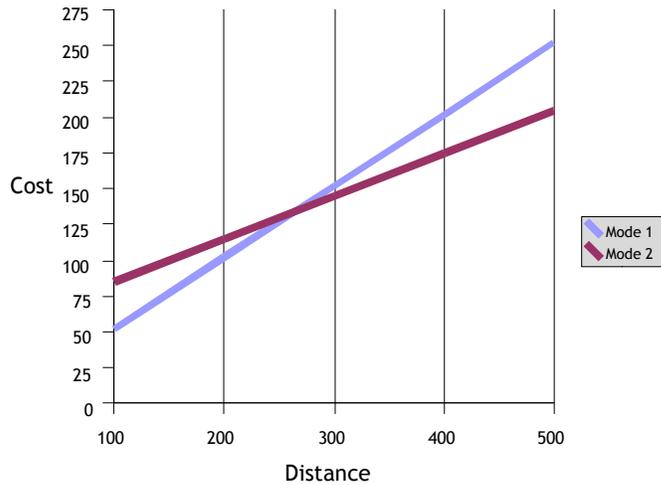


Figure 4. Original cost functions

The cost functions can obviously be recalibrated in order to try to obtain a better fit in terms of tons.km. For instance, figure 5 illustrates the cost functions when the values of A and B are respectively fixed to 52 and 0.35 for mode 2, the values for mode 1 remaining unchanged. In this case, the four first lines of the demand matrix will be assigned to mode 1, for a total of 3850 tons.km, which is perfectly right. But now, 21 tons will be transported by mode 1, which is too much.

This simple example shows that, using a simple All-Or-Nothing assignment, it is impossible to calibrate an assignment on both the transported quantities (tons) and the flows (tons.km) when virtual networks are used to avoid separate modal-choice and assignment models. The only way to solve this “distance trap” is to implement techniques that don’t, for a given distance (and origin-destination pair), assign all the demand on the same path (and thus maybe same mode) on a virtual network. The first techniques one might think to are the equilibrium assignment algorithms, which spread the flow over several alternative routes, according to the flow on the used links. Their implementation will now be discussed. This is the main topic of this paper.

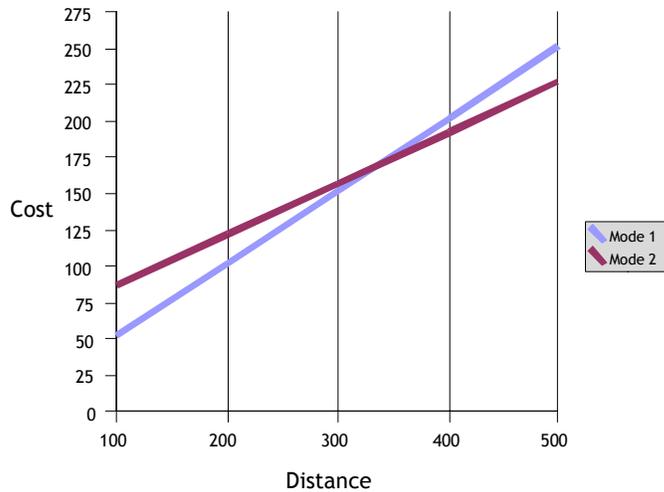


Figure 5. Modified cost functions

2.4 Cost functions and congested links

As just discussed, a simple implementation of the AoN algorithm presents some limits because it is often observed that the flow of transport between two given nodes is distributed over various alternative routes. Two main reasons can explain this phenomenon: the capacity constraints of the network and the fact that all the users don't have the same perception of the costs of the different alternative routes. Both reasons induce a spreading of the flow over several routes. But do the equilibrium assignment techniques give an appropriate answer to the "distance trap" problem?

The equilibrium algorithms take into account the variation of the transportation costs according to the assigned flow, considering that the distribution of the traffic over the network is the result of an interaction between the supply and the demand for transport. They try to implement the second equilibrium condition of Wardrop (1952), who stated that:

Under equilibrium conditions traffic should be arranged in congested networks in such a way that the average (or the total) travel cost is minimised.

It is important to note that the implementation of these methods on a virtual network makes it possible to observe transfers of flow not only between different routes, but also between various transportation modes. Indeed, in opposition to the classical four stages models (generation, distribution, modal-split and assignment), virtual networks, which explicitly decompose all the tasks in the transportation chain, combine modal split and assignment in a single step. Thus, an alternative route could very well be used, totally or partially, by another mode and/or means of transport than the ones that were used on the initial route, without any explicit modal-split. This is probably the most important contribution of the use of equilibrium assignment methods on virtual networks and this will be illustrated in the next section.

The discrete choice models often used in the classical four stages approach include an error term describing non-explained variation, as well as random coefficients explaining variations of preferences. The supporters of these methods often argue that these methods avoid that two competing routes with almost the same costs would be assigned all-or-nothing if there are no capacity problems. This problem is at least partially addressed in our model, because the demand is split into 10 categories of commodities, each of them having its own set of cost functions. Moreover, the speed-flow curve is not flat as long as the capacity of a link is not reached, making it possible that the flow is spread over several routes, even if saturation is never observed on any link. Finally, the introduction of random coefficients in order to take into account behavioural aspects (probability to choose a given route) is much more linked to stochastic (equilibrium) traffic assignment models, which are not discussed in this paper.

The equilibrium assignment models require cost functions which are related to the flow on the network. Such a relation can be generally expressed as: $C_a=C_a(\{V\})$.

In principle, the cost on link “a” should be a function of the flow V on the total network and not only on the link itself, because the flow on a given link is also influenced by the neighbourhood. This is usually simplified by considering that $C_a=C_a(V_a)$, i.e. that the cost on link “a” is a function of the flow V_a on it. A good cost function must be realistic, non-decreasing, monotonous, continuous and differentiable and should not generate infinite costs if the flow is equal or higher than the capacity in order to always ensure a solution.

Many functions were proposed to describe this relation. Ortúzar and Willumsen (1990) give a good overview of them. The most used are:

- a) Smock (1962) : $C = C_0 e^{V/K}$ where C is the cost for a given flow V , C_0 is the cost at free flow and K the capacity of the link.
- b) Overgaard (1967) generalises the previous function: $C=C_0 \alpha^{\beta(V/K)}$
- c) The Bureau of Public Roads (BPR, 1964), USA, proposes the standard function which is most often: $C = C_0(1 + \alpha(V / K)^\beta)$

We used latest function with $\alpha= 0.15$ and $\beta= 4$. These values may seem to be doubtful because they assume that travel time increases maximally with 15% and because the beta value 4 may only apply to multi-lane motorways. However, we are interested here only in inter-urban, regional and international traffic, which mainly uses highways (where available), and for which the demand is only available on a yearly basis, making it difficult to correctly model what happens during the peak hours. This problem will be more discussed in section 3.2.

Finally, it is clear that passenger cars interact with trucks on the roads and that one cannot consider that the total capacity of the roads can be devoted to trucking only. The way this consideration is handled on a real network will also be presented in section 3.2.

Another important issue deserves more discussion here: using a speed-flow curve at a large geographic scale may not be appropriate for freight transport. Indeed, one could assume at a more strategic level that the cost-functions could be “reversed”, reflecting the fact that high demand volumes could justify higher frequencies and/or a more effective use of the vehicles, with less empty return trips and better filling rates. If this is certainly true for trucking, this is even more the case for barges and trains, which sizes and/or compositions can be adapted in the short term to the demand. To take these effects into account and implement such refinements, the information provided by simple base year freight transport flow matrixes is not enough, and additional information on shipment/consignment sizes is needed, which is rarely

the case in real world strategic models on large territories. The O-D matrixes we have used didn't contain any information about shipment sizes. Note that in the German national freight transport model, financed by the German Federal Transport Investment Plan 2003, and which was developed by the BVU Beratergruppe Verkehr + Umwelt GmbH from Freiburg, the shipment sizes are split over three classes ($\leq 15t$, $15-25t$, $\geq 25t$), making it possible to, at least partially, cope with the above discussed consideration. If such detailed data were available, a multi-class assignment could very-well be performed on virtual networks.

2.5 Flow equilibrium algorithms

Various assignment algorithms, which try to obtain an equilibrated distribution of flows on the network, can be found in the literature.

The implementation of these algorithms on virtual networks is not immediate. Indeed, congestion is observed on real links, not on virtual links. However, in a virtual network, the same real link is represented by various virtual links, according to the number of transportation means (types of vehicles with different operation costs) that are defined. It is thus necessary to consolidate the flows obtained on the virtual links related to a same real link in order to obtain the total flow on the real link.

A first technique, based on the method of the successive averages (MSA) was implemented. During the initialisation step, for each real link a , the flow V_a is set to null and its associated cost C_a is computed for a free flow situation. The process then enters in a loop that is repeated until a stop condition is satisfied. At a given iteration n , and for each link a , the cost C_a^n is computed, that depends on the flow V_a^{n-1} found on the link at iteration $n-1$. A set of auxiliary flows F_a^n is then obtained by means of an AoN assignment based on the just re-computed costs. The new flow V_a^n is then obtained:

$$V_a^n = (1 - \lambda^n) V_a^{n-1} + \lambda^n F_a^n \tag{1}$$

where $\lambda^n = \frac{1}{n}$

The algorithm of Frank-Wolfe (FW) (Frank and Wolfe, 1956) is very similar to the previous one. It differs only by the way λ is computed: instead of being fixed at $1/n$, it is calculated to optimise the displacement in the descent direction $F^n - V^{n-1}$. Hence, after each iteration, the depth of the descent must be re-computed:

$$\lambda^n \Leftarrow \min_a \sum \int_0^{V_a^n} C_a(V) dV \tag{2}$$

where $0 \leq \lambda^n \leq 1$

The effects of the congestion can also be taken into account by incremental assignment (INC). At each iteration, only a restricted proportion of the demand matrix is assigned on the network. The incremental loading charges the network gradually. The total quantity to transport is split by a factor p_i , such as $\sum_i p_i = 1$ and, during each iteration, an additional increment is loaded on the network. These factors can be calculated as:

$$p_i = \frac{n-i+1}{n^*(n+1)/2} \quad (3)$$

where $i=1,2, \dots, n$ and n is the number of iterations.

The p 's $_i$ are thus a decreasing arithmetic progression in which the difference between two successive terms is $\frac{-1}{n^*(n+1)/2}$.

The main disadvantage of the incremental method is that once a flow is assigned to a link, it is not anymore possible to withdraw (a part of) it in order to assign it to another link. Consequently, this method will not necessary ensure that the algorithm converges to an equilibrium solution as in MSA and FW (Sheffi, 1985). Note that in order to accelerate the convergence of the algorithm, the incremental method can also be associated with the Frank-Wolfe's algorithm to obtain the initial flows (Inc+FW). However, this method can be counterproductive, especially if the network is not congested.

Another important element to implement is the criteria used to stop the iterative process. Except for the incremental method, for which the number of iterations must be fixed a priori, the stop-rule of Le Blanc *et al* (1975) is used:

$$\text{Stop if } \text{gap} = \text{Max} \frac{(|V_a^{(n)} - V_a^{(n-1)}|)}{V_a^n} < \varepsilon \quad (4)$$

because flows are better indicators of the differences between successive iterations than costs (Thomas, 1991). A value of $\varepsilon = 0.05$ was used, which can be considered as being low enough to ensure a solution close to the real equilibrium. The gap formula that was implemented refers to the maximum difference between the current solution and the equilibrium solution in terms of a mathematical objective function. Other formulations are also possible: The relative gap is the ratio between the absolute gap and the current value of the objective function, expressed as a percentage; the normalized gap is computed as the absolute gap divided by the number of trips. The two latest criteria weren't implemented because their computation takes too much time on the generated virtual networks.

Beside these classical equilibrium methods, Bar-Gera (2002) shows that his Origin-Based method is faster than the Frank-Wolfe method for low target relative gaps. The core question is obviously the estimation of the optimal level of convergence needed for a given problem³. The work presented in this paper mainly concerns long distance and international transport, and the level of knowledge and information that the truck drivers can have on the trans-European network is more than probably not sufficient to make the right route choice with a probability close to one. In these circumstances, using a stop criteria with a very low value doesn't make a lot of sense, and the Frank-Wolfe implementation can certainly be considered as a good choice for the problems discussed here.

All of the algorithms outlined in this paper (AoN, Incremental, MSA and Frank-Wolfe), are implemented in a new release of the Nodus software. In addition to these implementations, this version (5.0) uses a file format that is compatible with the popular ESRI's Arcinfo geographic information system (GIS). It also uses the BBN Technologies' OpenMapTM

³ An interesting discussion about stopping criteria and rules can be found in Boyce et al. (2002).

(www.openmap.org) package, that is an Open Source JavaBeans™ (java.sun.com) based programmer's toolkit, providing the means to allow users to see and manipulate various geo-spatial information. Entirely rewritten in Java, Nodus is available for a wide range of computer platforms.

The results presented in this paper were obtained with a 2Ghz Intel Pentium IV based computer with 512 Mo RAM, Windows 2000 and Java 2, (release 1.4.1)

3. Some results on artificial and large networks

3.1 A case study on an artificial network

A first exercise was performed on an artificial road network of 17 nodes and 36 links. In this network, 7 nodes are used as potential origins and/or destinations. The incremental assignment was implemented with four iterations, while the other algorithms converge accordingly to the above described stopping rule. The network will gradually be stressed in order to observe the behaviour of the different assignment techniques. In a first step, a small O-D matrix was assigned onto the network. For this matrix, and after an AoN assignment, no link appeared to be congested.

For all the tested assignment techniques, we obtained the same set of computed paths and the same total flows on the links. The AoN assignment is obviously the fastest assignment method, because it stops after only one iteration. The difference on the total costs on the network⁴ (see table 6, last column) for the different methods is explained by the fact that, in the iterative methods, the costs are always calculated on the flows obtained during the previous iteration. The total costs obtained for the MSA, FW and Inc + FW methods are similar: in the MSA method with two iterations, λ is fixed, by definition to 0,5, and the same value is obtained for λ in the two FW based methods because there is no congestion and the routes computed by these methods are equal to those obtained by the AoN assignment.

Next, all the quantities of the same O-D matrix were multiplied by four, in order to simulate a problem in which the most important flow observed on a link after the first AoN assignment represents 70% of the its capacity. In this test case, the results obtained by the different capacity constrained assignments are different from the results of a simple AoN. This is illustrated by figure 6, in which the 100% cost level corresponds to an AoN assignment. A four step incremental assignment associated with a Frank-Wolfs algorithm (Inc+FW) gives the fastest results as the convergence is obtained twice as fast as with a simple FW. The cost variation between the solution obtained by an AoN assignment and the one obtained at convergence for the other methods is about 8%.

⁴ This doesn't mean that the total cost is used as stop criteria for the convergence. The used criterion is outlined in section 2.5.

Table 6. Results on non-congested sample network

	Iterations	Comp. Time (s)	Total cost (%)
AoN	1	2	100*
MSA	1	2	+0
	2	4	+2.2
FW	1	2	+0
	2	5	+2.2
Inc	1	2	+0
	2	4	+1.4
	3	6	+1.8
	4	9	+1.9
Inc+FW	1	9	+0
	2	11	+2.2

*102.2% after re-computing costs according to assigned flows

For quantities ten times more important than those assigned during the first exercise (figure 7), a capacity overshooting on three links was observed after an AoN assignment. Here also, the incremental + Frank Wolfe assignment arrives more quickly at a solution than any other method. The cost variation between the solution obtained by an AoN assignment and the one obtained at convergence for the other methods is now about 19.5%.

The goal of this simple example based on an artificial network is to evaluate the behaviour of the several equilibrium algorithms when the network is gradually stressed. It also has the benefit to test and validate the computer implementation of these algorithms when applied to virtual networks. Obviously, such a simple network is not representative of real world situations, for which a more complex and large example will be presented in the next section. This example will not only be commented from a technical point of view. Indeed, the use of equilibrium algorithms with O-D matrixes that contain annual data for long distance transport has to be discussed more deeply.

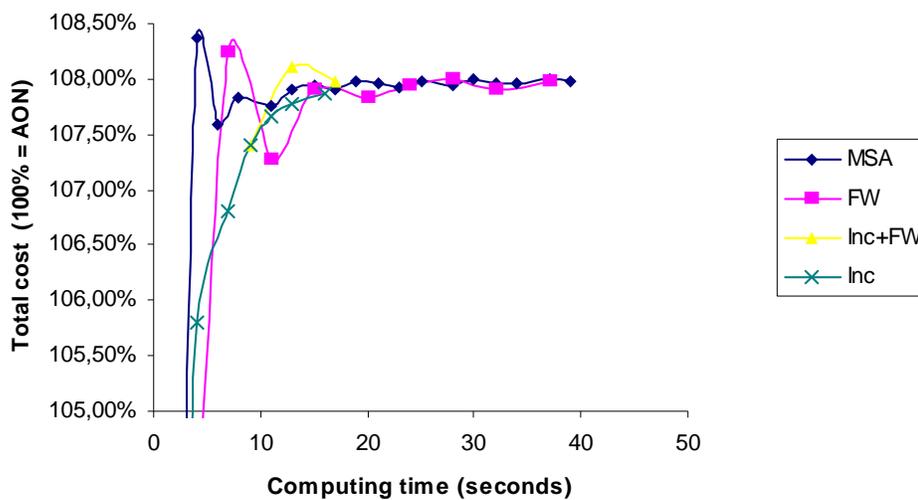


Figure 6. Assignment results on lightly congested artificial network

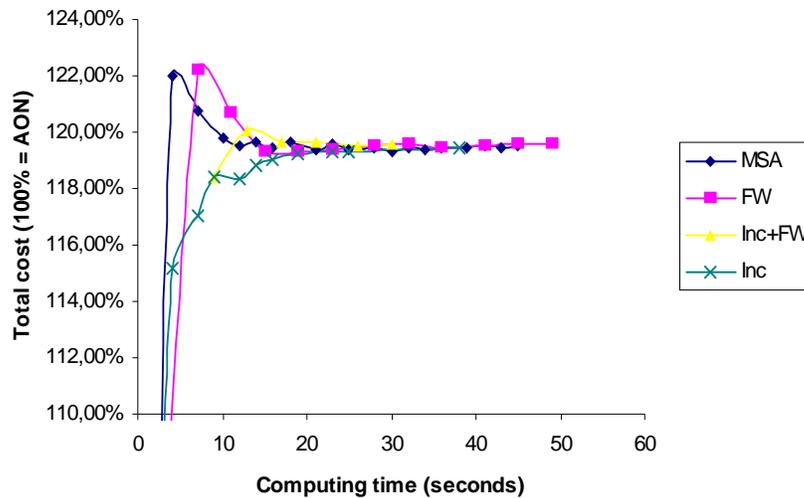


Figure 7. Assignment results on a congested artificial network

3.2 An application on a real network

The “virtual network” methodology was designed to develop multi-modal models, i.e. models in which different transportation modes can be used or combined between the origins and destinations of the O-D matrixes. Moreover, the methodology is intended to be used on large geographic networks, covering several regions and countries. In section 2.3, the “distance trap” problem was presented, which led to the conclusion that, on virtual networks, the flow has to be spread over several alternative paths. Equilibrium assignments do these kinds of things. But do they solve the “distance trap” problem?

This second exercise, based on the extensive trans-European multi-modal network, as defined by the European Conference of Ministers of Transport, tests the different assignment techniques, in which road, rail and waterborne transport can be used. The digitised network that what used contains about 17.000 links and 12000 nodes, from which about 1500 are potential origins and/or destinations.

The implementation of the capacity restrained methods is not immediate, because the roads are shared by private cars and trucks. To give a reasonable answer to this problem, the capacity of the roads was fixed to a theoretical residual capacity that is left over when the private cars are already on the network. A more refined method could be imagined, in which the private cars and the trucks are assigned at the same time and interact with each other. Such an approach is for instance presented in Nielsen *et al.* (1998), but is much more difficult to apply in inter-urban, regional or international transport problems because of the nature of the O-D matrixes⁵, since they contain annual amounts of goods on inter-urban and inter-regional and international relations. This is quite different from urban O-D matrixes set-up for peak hours, because no information is available on the departure and/or arrival time of the commodities.

⁵ The used matrixes for this exercise are thoroughly discussed in Geerts and Jourquin (2001).

Moreover, as distances are long, a single trip often occurs during different periods of the day, including peak hours. Finally, most of these traffics use the highway network, which justifies the use of the standard HCM parameters discussed in section 2.4. The model could however be refined, using α and β values for the BPR formula, which are specific to each (type of) link in the network.

In this exercise, the traffic during an average peak period was estimated. To achieve this, the saturation of the road network was estimated on the basis of the well known “30th peak hour” criterion⁶. No capacity restrictions were introduced for rail or waterborne transports, i.e., their respective cost functions are not dependent on the already assigned flows.

The O-D matrixes contain demands for 10 categories of goods (NST-R 0-9), and specific cost functions were developed for each of these 10 classes. This exercise is thus a multi-class assignment. Obviously, not only the flows for the different transportation means usable on a real link are consolidated, but also the flows obtained for the different classes of the demand matrix are summed before new cost values are computed for the next iteration of the assignment procedure.

Another critical point about static asymmetric multi-class assignments must be raised here. As explained by Toint and Wynter (1996), at congestion, incoherent results occur for some existing formulations of multi-class assignments. In particular, they show that, under heavy congested situations, the difference between the speeds of the classes of vehicles (cars and trucks for instance) doesn't approach zero as expected. This problem is however avoided in this exercise for two main reasons:

- Cars are not assigned along with trucks, but the latest are assigned on a network on which only a residual capacity remains available (which is, however, also a limit of our model);
- The remaining vehicles classes (trucks) circulate at roughly the same speed.

Once the residual capacities on the network and the O-D matrixes for an average peak hour are known, the different assignment methods can be tested. The results are shown in figure 8. It is worthwhile to note that the found solution is only slightly more expensive (+1%) than the AoN initial solution. One could argue that is due to the α and β parameters used in the speed-flow functions. However, several other factors can explain this:

- The used O-D matrixes contain total quantities on an annual basis, and it is difficult to estimate what is transported at a given time of a day.
- The implemented assignment algorithms are “static” and the sense that the vehicles are not tracked during their journey. This is a problem for the assignment on long distances, because often, several hours or even days (including peak and off-peak periods), pass between the starting and the arrival time. The assignments don't take into account the actual location of the vehicles at a given time.
- Congestion is mostly observed in and around (large) urban areas. The transportation networks that are located in such areas only represent a small part of the total trans-European network.

⁶ Traffic observed at the 30th rank, according to the classification of the observed peak hours during one year, ordered by decreasing value. This method consists in comparing the flow of the 30th peak hour with the theoretical capacity of the considered road section. This method, simple in its design, has the following particular advantage: the flow of the 30th peak hour represents a quasi constant share – whatever the place or time – of the average daily flow. For the Belgian network for instance, this value is known to be 13,6% of the daily flow.

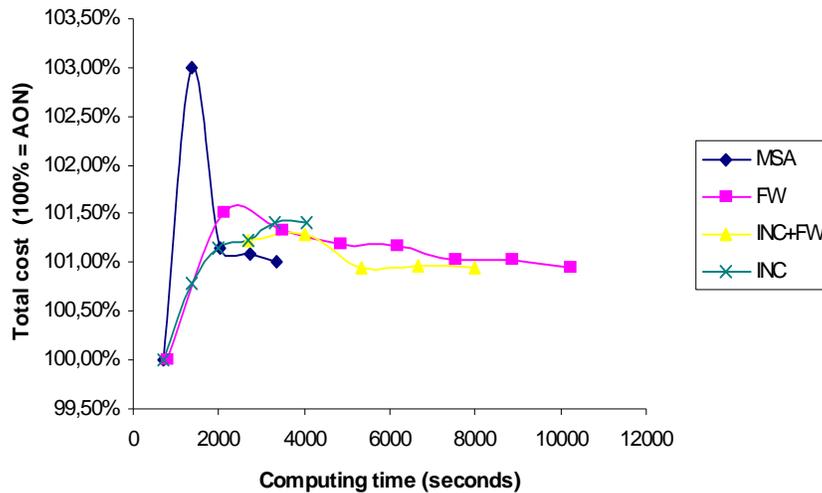


Figure 8. Assignment results on a real case

The assignment methods also have an impact on the market-share of the different transportation modes (see table 7). This is what we are looking for in order to solve the “distance trap”. Indeed, a cheaper path on a virtual network can very well be found using another transportation mode. Furthermore, there is a reduction of the tons-km transported by road due to capacity restrictions in all cases. The change in modal share is more important after an incremental assignment because, once a flow assigned on a link, it is not anymore possible to withdraw it in order to re-assign it on another link. This obviously constrains the search for an optimal solution.

Table 7. Impact on the estimated modal split

Method	Variation of modal split induced by equilibrium solutions		
	Road (%)	Rail (%)	Water (%)
MSA	-2,56	+ 0,80	+ 5,75
FW	-3,23	+ 1,15	+ 7,09
Inc+FW	-3,34	+ 1,16	+ 7,16
Inc (n=4)	-1,62	+ 0,35	+ 3,64

The model was further stressed, doubling the assigned quantities, to see how the different algorithms react on a significantly more loaded network. The results of this exercise are presented in figure 9 and table 8. This time, the solution is about 4,5% more expensive than the AoN total cost. But, again, the same arguments than those explained earlier can be used. More generally, an important question can be raised here: are equilibrium algorithms, even when used on virtual networks, realistic enough to be used in conjunction with O-D matrixes that contain annual data for long distance freight transport?

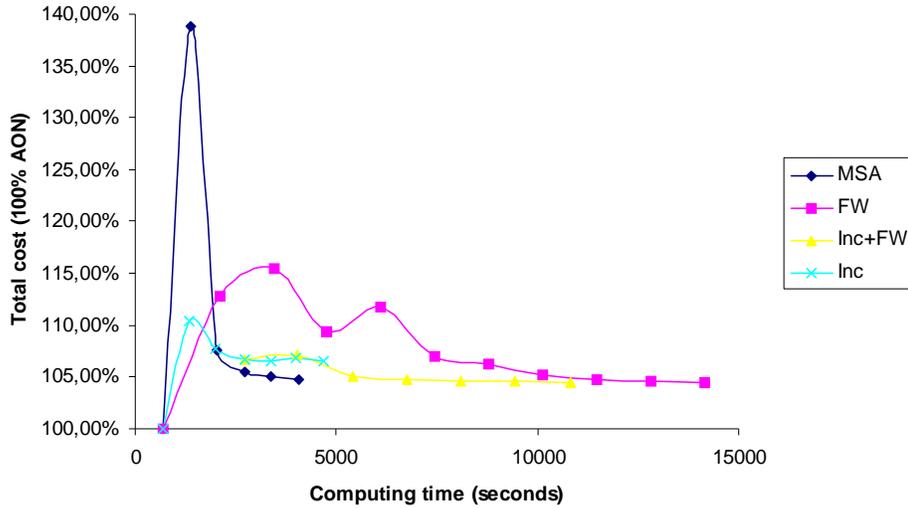


Figure 9. Assignment results on a stressed real case

Table 8. Modal split on the stressed network

Method	Variation of modal split induced by equilibrium solutions		
	Road (%)	Rail (%)	Water (%)
MSA	- 4,21	+ 3,77	+ 8,60
FW	- 4,89	+ 4,27	+9,79
INC+FW	- 4,79	+ 4,01	+ 9,19
INC (n=4)	- 2,30	+ 1,38	+ 4,70

3.3 Limits and perspectives

The equilibrium methods were implemented in order to try to ensure that the total flow to assign between a given origin and destination is spread over several routes (and thus modes if virtual networks are used). Unfortunately, it appears that capacity restrictions alone are not an explanatory factor that is strong enough to reach the expected result, for all the reasons that were just outlined. In other words, it becomes clear that equilibrium assignment techniques don't give an adequate solution to the "distance trap" and, therefore, other approaches must be explored.

Our conclusion is that obtaining a set of credible alternative routes, i.e. (nearby) non overlapping routes, is only possible if multi-flow algorithms are used. Their implementation and use on virtual networks is discussed and illustrated in Jourquin (2006). These algorithms are much more promising to solve the "distance trap", as can be seen in figure 10. The first map illustrates the result of an AoN assignment, and the second the output of a multi-flow algorithm designed for virtual networks. In the latest, the two available modes (road and rail) are used on almost every OD pair, even if no separate modal-split module is used.



Figure 10. Sample assignments of an OD matrix

4. Conclusions

Transportation models are often based on the classical four stages approach, in which generation, distribution, modal-split and assignment are seen as separated steps. Another approach, promoted by different authors, is to create an alternative network, that not only represents the geographic components of the network, but also the different operations that are possible during the transportation chain. These “super networks” or “virtual networks” open interesting perspectives, because both the modal choice and the assignments steps of the classical approach are performed at the same time.

Unfortunately, these virtual networks contain a hidden trap, as it is nearly impossible to calibrated the models on both the transported quantities and the flows (expressed in tons.km). This problem, which is referred to in this paper as the “distance trap”, can only be solved if the flow that must be sent from an origin point to some destination is spread over several routes and transportation modes.

This paper examines if the implementation of equilibrium assignment algorithms is able to provide an adequate answer to the distance trap problem, as they spread the flows over several alternative routes according to the flows on the different links of the network. The way several classical algorithms were implemented on virtual networks is explained, discussed and illustrated on a small artificial and a large real network.

It appears that the use of equilibrium assignment procedures don’t give satisfactory results on large-scale networks such as the trans-European multi-modal freight network. Indeed, the solutions found after convergence were very close to the one obtained after a single All-Or-Nothing assignment. This is mostly linked to the nature of the problem that is solved. Indeed, the assignment of annual amounts of goods on long distances doesn’t give information on the quantities transported at different periods of a day, particularly during the peak hours.

The results may also be influenced by the fact that we couldn’t explicitly take into account the effect of flexible frequencies and improved efficiency for O-D pairs between which a high demand exists, just because no information was available on shipment sizes. This could be resolved if statistical distributions of shipment sizes were applied to our O-D matrixes.

However, it seems clear that equilibrium assignment techniques, used on inter-urban, regional or international freight networks are not able to enough reflect the modal and route behaviours, and that other deterministic or stochastic multi-flow algorithms must be implemented and tested on virtual networks. Obtaining a set of credible alternative routes, i.e. (nearby) non overlapping routes, is only possible if multi-flow algorithms are used. Their implementation on virtual networks is discussed in Jourquin (2006).

Acknowledgments

This research was funded by the *Fond Special de la Recherche Inter-Universitaire* (FSRIU, Belgium). We are grateful to D. Boyce for his constructive remarks on the preliminary version of this paper. We also thank two referees for their useful critical comments.

References

- Bar-Gera, H. and Boyce, D. (2003). Origin-based algorithms for combined travel forecasting models. *Transportation Research B*, vol. 37, no. 5, pp. 405-422.
- Bar-Gera, H. (2002). Origin-Based Algorithm for the Traffic Assignment Problem', *Transportation Science*, vol. 36, no. 4, pp. 398-417.
- Boyce, D., Ralevic-Dekic, B. and Bar-Gera, H. (2002). Convergence of traffic assignments: How much is enough? *Journal of Transportation Engineering*, vol. 130, no. 1, pp. 49-55.
- Bureau of Public Roads (BPR)(1964). *Traffic Assignment Manual*. Urban Planning Division, US Department of Commerce, Washington DC.
- Crainic, T.G., Florian, M., Guélat, J. and Spiess, H. (1990). Strategic planning of freight transportation: Stan, an interactive graphic system. *Transportation research record*, vol. 1283, pp. 97-124.
- Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Cliff Stein, C. (2001). *Introduction to Algorithms* (Second Edition). MIT Press and McGraw-Hill.
- Dijkstra, E.W. (1959). A note on two problems in connection with graphs. *Numerische Mathematik*, vol. 1, pp. 269-271.
- Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, vol. 3, pp. 95-110.
- Geerts, J.F. and Jourquin, B. (2001). Freight Transportation Planning on the European Multimodal Network: the case of the Walloon Region. *European Journal for Transport Infrastructure Research*, vol. 1, no. 1.
- Harker, P.T. (1987). *Predicting intercity freight flows*. VNU Science Press.
- Johnson, D.B. (1973). A note on Dijkstra's shortest path algorithm. *Journal A.C.M.*, vol. 20, no. 3, pp. 385-388.
- Jourquin, B. (2006). A multi-flow multi-modal assignment procedure on large freight transportation networks. *Studies in Regional Science*, vol. 35, no. 4 (in press).

- Jourquin, B. (1995). Un outil d'analyse économique des transports de marchandises sur des réseaux multi-modaux et multi-produits: Le réseau virtuel, concepts, méthodes et applications. PhD Thesis, Facultés Universitaires Catholiques de Mons, Belgium
- Jourquin, B. and Beuthe, M. (1996). Transportation policy analysis with a geographic information system: the virtual network of freight transportation in Europe. *Transportation Research C*, vol. 4, no. 6, pp. 359-371.
- Kresge, D.T. and Roberts, P.O. (1971). *Techniques of Transportation Planning: Systems Analysis and Simulation Models*. Brooking Institution, Washington D.C.
- Le Blanc, L.J., Morlock, E.K. and Pierskalla, W.P. (1975). An Efficient Approach to Solving the Road Network Equilibrium Assignment Problem. *Transportation Research*, vol. 9, no. 5, pp. 309-318.
- Nielsen, O.A., Frederiksen, R.D. and Simonsen, N. (1998). Stochastic User Equilibrium Traffic Assignment with Turn-delays in Intersections. *International Transactions in Operational Research*, vol. 5, no. 6., pp. 555-568.
- Ortúzar, J.D. and Willumsen, L. (1990). *Modelling Transport*. John Wiley & Sons, New-York
- Overgaard, K.R. (1967). Urban transportation planning: traffic estimation. *Traffic Quarterly*, vol. 26, no. 2, pp. 197-218.
- Sheffi, Y. (1985). *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall, Englewood Cliffs, New Jersey.
- Smock, R.J. (1962). An iterative assignment approach to capacity restraint on arterial networks. *Highway Research Board Bulletin*, vol. 156, pp. 1-13.
- Tavasszy, L. (1996). *Modelling European Freight transport Flows*. PhD thesis, Technical University of Delft, The Netherlands.
- Thomas R. (1991). *Traffic Assignment Techniques*. Avebury Technical, England
- Toint, P. and Wynter, L. (1996). Asymmetric Multiclass Traffic Assignment: A coherent formulation. *Proceedings of the 13th International Symposium on Transportation and Traffic Theory*, ed. J.-B. Lesort, Pergamon, Exeter
- Transportation Research Board (2000). *Highway Capacity Manual*.
- Wardrop, J.G. (1952). Some Theoretical Aspects of Road Traffic Research. *Proceedings of the Institute of Civil Engineers*, Part II, vol. 1, pp. 325-378.
- Wilson, A.G. and Bennet, R.J. (1985). *Mathematical Methods in Human Geography and Planning*. John Wiley & Sons, New York
- Zhan, F.B. and Noon, C.E. (1998). Shortest Path Algorithms: An Evaluation using Real Road Networks. *Transportation Science*, vol. 32, no. 1, pp. 65-73.

