Assignment techniques on Virtual Networks

Performance considerations on large multi-modal networks
European Regional Science Association

ERSA 2003 Congress – University of Jyväskylä - Finland

Bart JOURQUIN[I, II] and Sabine LIMBOURG[I]

[II] Facultés Universitaires Catholiques de Mons (FUCaM),
  Group Transport & Mobility (GTM)
  151 Ch. de Binche, B-7000, Mons, Belgium
  Tel : 32-65-323211, Fax : 32-65-315691, gt&m@fucam.ac.be
[II] Limburgs Universitair Centrum (LUC)
  Universitaire Campus, gebouw D, B-3590 Diepenbeek, Belgium
  Tel : 32-11-268600, Fax : 32-11-268700

ABSTRACT

Multi-Modal freight models are traditionally built following the well known "fours steps model" in which generation, distribution, modal-split and assignment are seen as separated modules. An alternative approach, implemented in some softwares, is to represent the multi-modal network by means of a mono-modal one, representing each particular transport operation (loading or unloading operation, transhipments, ...) by a dedicated "virtual link".

This approach is proven to give interesting results, but as the drawback to generate much larger networks as their geographic representation. These huge networks (often larger than several hundreds of thousands links) made it difficult to implement loop-based equilibrium assignment techniques.

The increasing computation power of recent hardware make it now much more easy to test several alternative assignment techniques.

This paper presents some results obtained on a large multi-modal network, using different equilibrium assignment algorithms.

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 deals to some extend with the different operations performed at the nodes, i.e. loading /unloading, transhipment 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 reorganisation of networks, intermodal transports and new technical bundling concepts in order to substitute to direct road transport alternative transport solutions with less negative external effects. Thus, there is a need for a better modelling of the functions assumed by nodes, i.e. terminals and transhipment platforms.

Beyond this brief introduction, this paper will present an overview of the Virtual Network methodology implemented in the Nodus software developed at FUCaM-GTM. 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 small and large networks.

2   NODUS and Virtual networks

A geographical multimodal transport network is made of links like roads, railways or waterways, on which vehicles moves; at its nodes it is made also of connecting infrastructures these links like terminals or logistics platforms where goods are loaded, unloaded, transhipped or processed in different ways. To analyse transport operations over the network, costs or weights must be attached to these geographical links over which goods are transported and to the connecting points where goods are handled. However, most of these 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 reached its destination. Normally, the costs of these alternative operations should be different.

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 transports operations where the same infrastructure is 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 cost for a particular use of an infrastructure. The NODUS software proposes a methodology and an algorithm which 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. The software and its underlying methodology are discussed in Jourquin (1995) and Jourquin & Beuthe (1996). This permits to apply the methodology to extensive multimodal netwoks.

The solution can be presented first in an intuitive way by using the example of a simple waterways network, as shown in Figure 1.

This network consists of 4 nodes and 5 links. 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 support small barges and "W2" a waterway that can be used by both small and large barges.

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

- 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 "W1" has a different cost than the use of a large barge on the same link;
- the same is true for the nodes: in the given example, the simple transit of a small barge come from link 1 and going on link 2 can be done at no cost, but the transhipment from a large barge coming from link 1 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 1, 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 transhipment 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) for low density graphs. The resulting solution is an exact solution, taking all the possible choices into account.

We can now demonstrate how the virtual network is built on the basis of a rather simple, though somewhat complex notation, which provides a convenient way to link cost functions to virtual links. Table 1 enumerates the elements of the real network.

Figure 1: Multimodal network
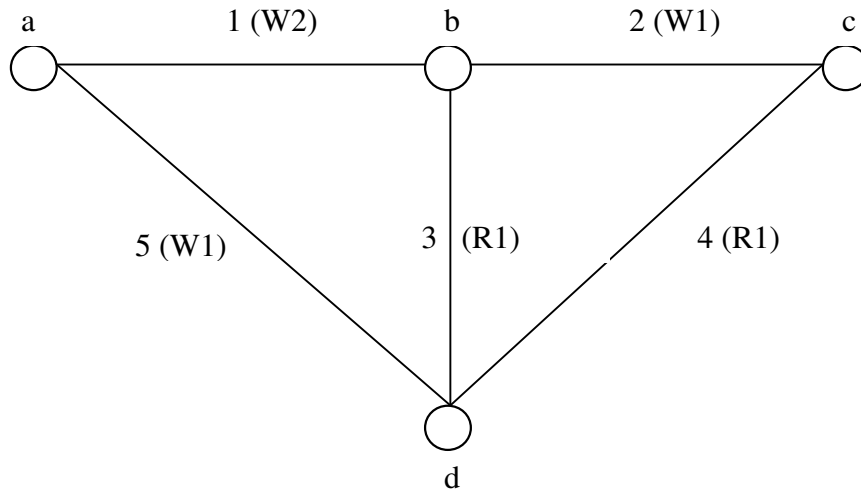
In a first step, the virtual links corresponding to the real links, i.e. rail tracks, etc., must be generated. These are defined in Table 2 by their end-nodes, the notation of which indicates successively the node, the real link, the mode and the means.

Table 1: Real network

Link	Node 1	Node 2	Type of link
1	A	B	W2
2	B	C	W1
3	B	D	R1
4	D	C	R1
5	A	D	W1

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

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

Figure 2: Partial virtual network
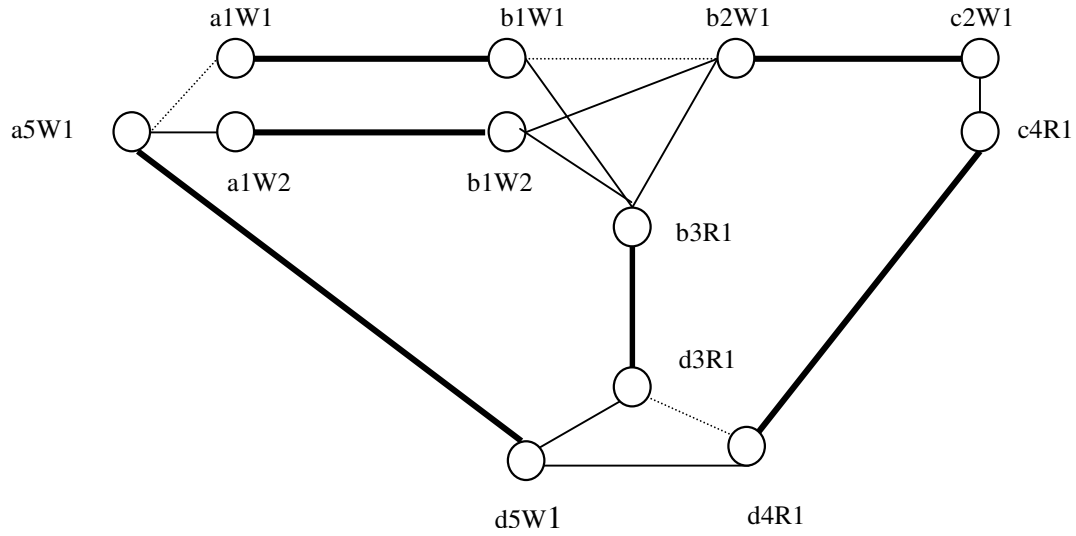
In a second step, these virtual links must be connected by transit or transhipment virtual links. To keep things as simple as possible, we just enumerate in Table 3 the connecting virtual links related to node "b". They can be viewed in Figure 2, where the virtual network resulting from that procedure is presented when it is applied to all the nodes. In this network, the boldfaced links represent the links of the real network, possibly split up. The dotted links represent the transit virtual links, while the transhipment links are indicated by a thin continuous line.

This network is not yet complete because it does not contain entry and exit nodes in/from the network. This can be done by the creation of additional virtual nodes associated to loading and unloading operations at nodes where they 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.

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 that problem, all 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. This is illustrated for real node b in figure 3. The virtual network requires the development of four types of cost functions, which are associated with specific virtual links through their notation:

- The use of a travelling cost function is indicated by a difference between the virtual link's indices denoting the real end-nodes;
- Transit costs are applied when the link's indices denoting the two connected real links vary, whereas the mode and means indices remain the same;
- For transhipment costs, the link's indices of the connected real links should vary, as well as the mode and/or means indices;
- For loading/unloading costs, one of the two indices of the real links is "00000".

Those "doubled" virtual nodes also make it possible to avoid "unwanted movements", like an unloading followed by a loading to circumvent a forbidden transhipment operation.

Figure 3: Detailed virtual network at node
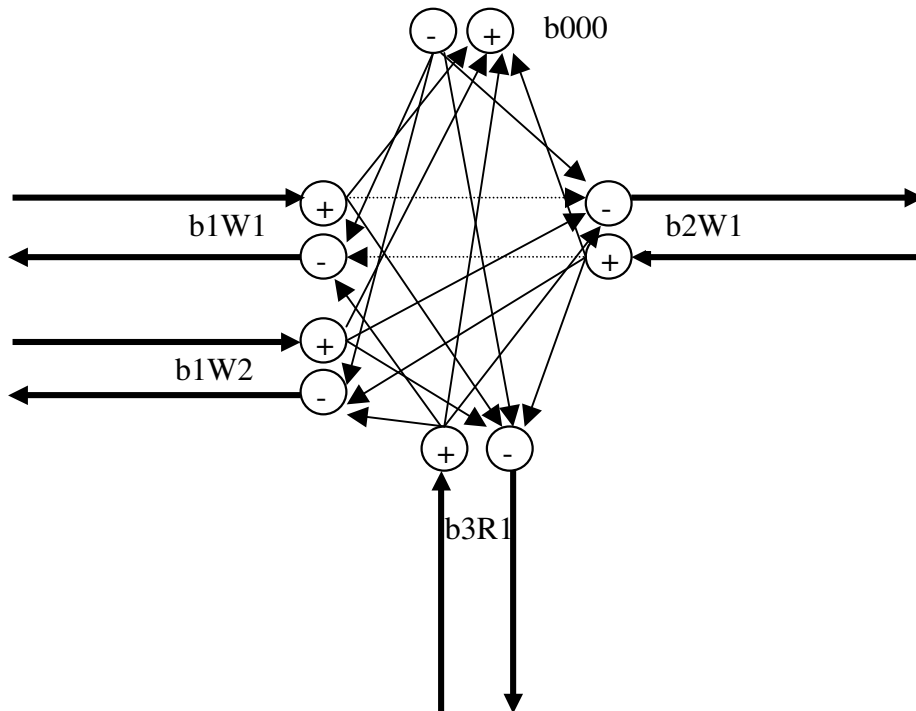
Obviously, the codification used in the preceding figures is not suitable for real applications; using a single letter as node label would limit the size of the real network to 26 nodes! That's why the virtual nodes are coded in the following way: a plus or minus sign, 5 digits for the node number, 5 digits for the link number, 2 digits for the mode and 2 digits for the means. Each label is thus represented by a 14 digits number preceded by a sign.

2.2   Cost Functions on Virtual Networks

After this overview of the basic methodology, it is necessary to explain how the cost functions are connected with the virtual network and which are their characteristics. As usual in transportation analysis (see, for example, Kresge and Roberts (1971), or Wilson and Bennet (1985)), "generalised cost", which allows to integrate all factors relevant for transport decision making in terms of monetary units, are used. The concept can be defined in different ways according to whether it is the point of view of the shipper which is taken or the one of the carrier, and according to the unit of reference which is used, i.e. tons, tons-km, vehicles, distances, 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 virtual network requires the development of four types of cost functions, which are associated with specific virtual links through their notation:

- The use of a travelling cost function is indicated by a difference between the virtual link's indices denoting the real end-nodes;
- Transit costs are applied when the link's indices denoting the two connected real links vary, whereas the mode and means indices remain the same;
- For transhipment costs, the link's indices of the connected real links should vary, as well as the mode and/or means indices;
- For loading/unloading costs, one of the two indices of the real links is "00000".

These four types of function are made of the following elements:

- All the costs related to moving a vehicle between a trip's origin and destination, like labour, fuel, insurance, maintenance costs, or tariffs;
- The cost of inventory of the goods during transportation;
- Handling and storage costs or tariffs, including packaging, loading and unloading and services directly linked to a transport.

All residual indirect costs like general administrative services which may be assigned to transports on an average basis.

2.3   Computing shortest paths on networks

The contemporary transport systems are used in an intensive way and are often congested at 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. 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 as well as the flows supported by the network (capacity). The assignment methods try to model the traffic's distribution over a network taking into account a set of constraints, such as capacity and transport costs, in order to obtain an equilibrium. This type of problems can be solved by means of optimisation methods.

We are interested here in the methods intended to model flows of commodities on large multimodal networks, using virtual network, which, as explained earlier, make it possible to combine modal choice and assignment in a single step. 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.

All the equilibrium assignment methods are based on the "all-or-nothing" 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.

As shown in Table 4, the Dijkstra's algorithm, with a binary heap implementation, seems to be the best algorithm to be used on virtual networks (which is a graph with low density, as the number of links is far below the square of the number of nodes).

The Dijkstra's algorithm solves the problem of the shortest path from an origin to all the possible destinations. The algorithm of Johnson finds all the shortest paths between all the pairs of nodes of the graph, it using the algorithm of Bellman-Ford and Dijkstra.

If we take the detailed virtual network generated from the Belgian network, we have approximately 300000 virtual links (M) and 100000 virtual nodes (N), of which 1500

(X) are potential (un)loading nodes. The latest characteristic involves that the Dijkstra's algorithm is the most suitable, because only a small amount of nodes are relevant to compute paths from, because they are potential origins.

Table 4: Complexity of shortest-path algorithms

	Used priority queue		
	Linear (worst case)	**Binary heap (worst case)**	**Fibonacci heap (amortized analyze[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] A amortised analysis guarantees an average performance for each operation in the worst case.

Among the various implementations of this algorithm, the binary heap implementation gives the best results, because our graph is not dense (M is much lower than N^2). The Fibonacci heap implementation gives smaller execution times only in the most unfavourable cases (15% of the cases tested on our networks) and is, on average, 50% slower than the binary heaps for our problems.

We have also introduced a stop criterion in the Dijkstra's algorithm. It stops indeed, as soon as the algorithm determined the shortest path to all the destinations to reach starting from a given origin. This improves the computing time by more than 50%, because the flows starting from a node are, in most cases, sent to only a few destinations that are relatively close to the origin.

2.4   Cost functions and congested links

The All-Or-Nothing (AON) algorithm presents however some limits because the flow of transport between two given nodes should be distributed between various alternate routes. The multi-flow algorithms make it possible to calculate those paths, but they don't take into account any constraint. However, certain axes of the network suffer from congestion and this congestion also lead in the search of alternate routes, in order to balance the traffic on the network.

The equilibrium algorithms try to model this phenomenon in transport systems. These models, which take into account the variation of the transportation generalized costs according to the assigned flows, consider that the distribution of the traffic on the network is the result of an interaction between the supply and the demand of transport.

They try to approach the equilibrium conditions stated formally by Wardrop in 1952. The second principle (or the design principle) states 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 fact of using these methods on a virtual network makes it possible to observe transfers of flow between the various transportation modes. Indeed, in opposition to the classical four stages models (generation, distribution, modal-split and assignment), and as specified higher, the virtual network combines the modal split and the assignment in a single step. Thus, an alternative route can very well be used, totally or partially, by another mode and/or means of transport. This is probably the principal contribution of the use of equilibrium assignment methods on virtual networks and will be illustrated in the next section.

The traffic assignment models which take the effects of the congestion into account require a relation between the cost and flow on the network, which can be described in a general way like : $C_a=C_a(\{V\})$. The cost on link "a" should be a function of flow V on the network and not only on the link itself. This is 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 have the following properties: it must be realistic, non decreasing, monotonous, continue, differentiable and should not generate infinite costs if the flow is equal or higher than the capacity.

Many functions (Ortúzar and Willumsen (1990)) were proposed to describe this relation. The most used are :

- Smock (1962) : $C = C_0 \; e^{V/K}$

  where C : cost for a given flow V
  C_0 : cost at free flow

  K : capacity of the link

- Overgaard (1967) generalises the previous function : $C=C_0 \; \alpha^{\beta(V/K)}$

- The Bureau of Public Roads (1964), USA, proposes the standard function which is certainly the most used: $C = C_0\left(1+\alpha(V/K)^{\beta}\right)$

We used this latest function with $\alpha= 0.15$ and $\beta= 4$, which are the most used values for the European networks.

2.5   Flow equilibrium algorithms

Various assignment algorithms which try to obtain a distribution of flows on the network approaching the equilibrium conditions stated by Wardrop can be found in the literature.

The implementation of these algorithms on a virtual network is not immediate. Indeed, in the reality, 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 that are defined. It is thus necessary to consolidate the flows obtained on the virtual links before obtaining flows indeed assigned on the real links.

A first technique, based on the method of the successive averages (MSA) was implemented in Nodus. During the initialisation step, for each 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 :

$$V_a^n = (1 - \lambda^n)V_a^{n-1} + \lambda^n F_a^n \text{, where } \lambda^n = \frac{1}{n}$$

The algorithm of Frank-Wolfe (FW) (1956) is very similar to the previous one. It differs from it only by the way λ is computed : instead of being fixed at *1/n*, as it was the case in the method of the successive averages, it is calculated to optimise the displacement in the descent direction F^n–V^{n-1}. After each iteration, one have to compute the depth of the descent:

$$\lambda^n \Leftarrow \min \sum_a \int_0^{V_a} C_a(V)dV$$

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

The effects of the congestion can also be taken into account by an incremental assignment (INC). At each iteration, only a restricted proportion of the demand matrix is assigned on the network. The incremental method 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. We calculated these factors as follows:

If n represents the number of iterations, then, $p_i = \dfrac{n-i+1}{n*(n+1)/2}$ where i=1,2, …,n

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

The main disadvantage of the incremental method is that once that a flow is assigned on a link, it is not more possible to withdraw a part of it in order to assign it to another link. This will be illustrated in the next section.

The incremental method can also be associated the algorithm of Frank-Wolfe: to accelerate the convergence of the algorithm of Frank-Wolfe, the incremental method is used to obtain the initial flows (Inc+FW).

When should the iterations be stopped? Except with regard to the incremental method, for which the number of iterations must be fixed a priori, we use the stop-rule of Le Blanc *et al* (1975) :

$$\text{Stop if } \max \frac{(|V_a^{(n)} - V_a^{(n-1)}|)}{V_a^n} < \varepsilon \,,$$

because the shape of the cost-flow curve indicates that flows are better indicators of the differences between successive iterations than costs as it is discussed in Roy Thomas' book (1991). We used ε =0,05, which is a very low value which ensures to be very close to the real equilibrium.

3   Some results on small and large networks

These various algorithms were implemented in the new version of the Nodus software, currently under development. This version (5.0) uses a file format that is compatible with Arcinfo. Nodus uses the BBN Technologies' OpenMapTM package, that is an Open Source JavaBeansTM based programmer's toolkit, providing the means to allow users to see and manipulate various geospatial information. Entirely written in Java, Nodus will be available on a wide range of computer platforms.

3.1 A study case on a small network

A first exercise was performed on a mono-modal road network of 17 nodes and 36 links. In this network, 7 nodes can be used as potential origins and/or destinations. In a first step, a small random Origin-Destination (O/D) matrix was assigned onto the network. For this matrix, and after an All-Or-Nothing assignment, no link appears 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 All-Or-Nothing assignment is obviously the fastest assignment method. The difference on the total costs on the network (see Table 5) is explained by the fact that, in the iterative methods, one uses costs calculated on the flows obtained during the preceding iteration. Indeed, these methods use a stop criteria which compares two successive iterations. A minimum of two iterations is thus necessary, and all these additional iterations assign flows on a, at least partially, loaded network. The same results are obtained for the MSA, FW and Inc + FW methods because in the MSA method with two iterations, λ is fixed, by definition to 0,5, and that a the same value is obtained for λ in the two FW based methods.

Table 5: Results on non congested sample network

	Iterations	Comp. Time (s)	Total cost
AON	1	2	769755,2
MSA	1	2	769755,2
	2	4	786345,4
FW	1	2	769755,2
	2	5	786345,4
Inc	1	2	769755,2
	2	4	780685,06
	3	6	783498,44
	4	9	784633,2
Inc+FW	1	9	784633,2
	2	11	786345,4

In a second exercise, all the quantities of the same O-D matrix were multiplied by four, so that the most important flow after the first AON assignment represents 70% of the link capacity. In this test case, the results obtained by the different assignment techniques are now different from the results of a simple AON. This is illustrated by Figure 4, in which the 100% cost level corresponds to an AON assignment.

A four step incremental assignment associated with a Frank-Wolfs algorithm gives the fastest results as the equilibrium is obtained twice as fast as for a simple FW. The variation of cost between the solution obtained by an AON assignment and the optimal solution is about 8%.

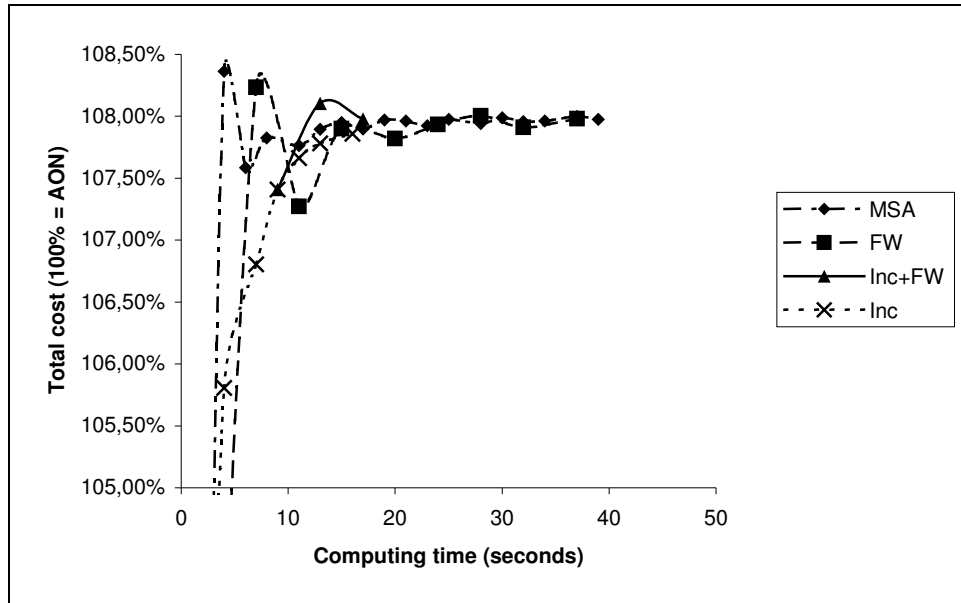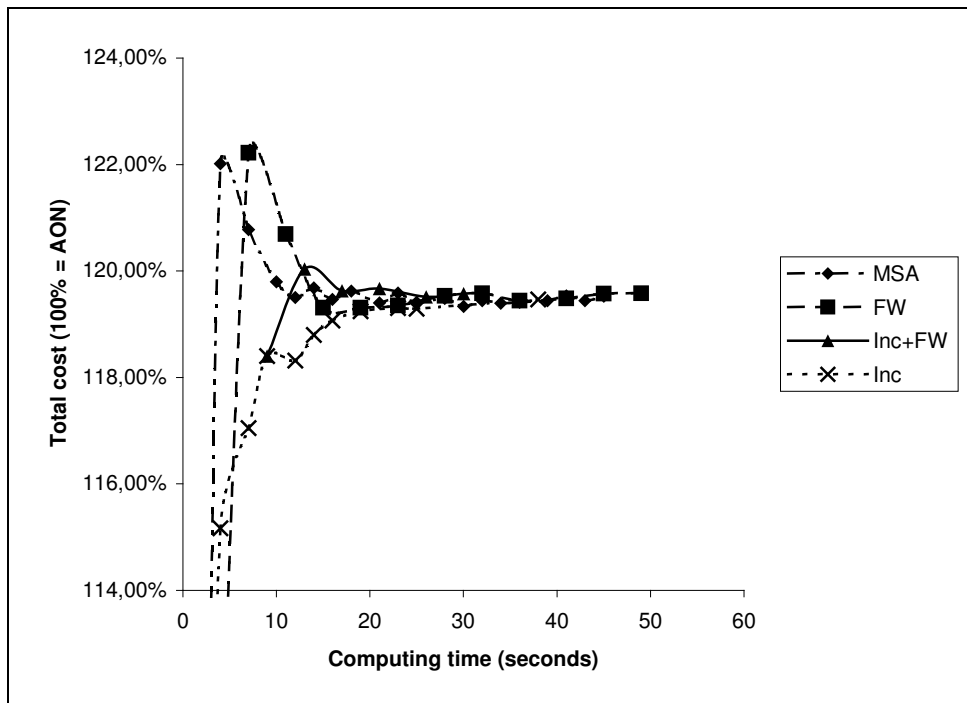*Figure 4: assignment results on lightly congested small network*

Figure 5: assignment results on a congested small network

For quantities ten times the one assigned during the first exercise, we have, after an AON assignment, a capacity overshooting on three links. Here also, the incremental +

Frank Wolfe assignment arrives more quickly at the equilibrium solution than any other method. The variation of cost between the solution obtained by an AON assignment and the optimal solution is about 19.5%.

3.2   An application on a real network

The "Virtual Newtork" 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 or even countries.

The following exercise, based on the network represented in Figure 6, implements the different assignment techniques discussed earlier on a real case, in which road, rail and waterborne transport can be used.

Figure 6: Real network
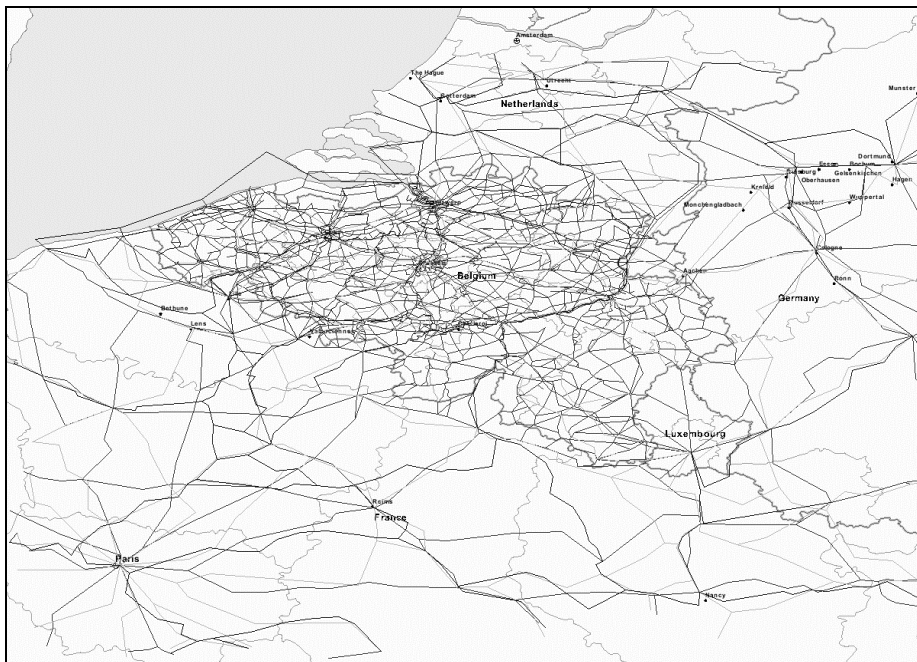
The implementation of the capacity restrained methods is not immediate, because the roads are shared by private cars and lorries.

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.

The residual capacity for the trucks was estimated in the following way :

- Average load of the lorries (including empty return trips) : 12,3 tons

- 58% of the traffic is done by small trucks (equivalent to 1,5 private vehicles)
- 42% of the traffic is performed by large trucks (equivalent to 2 private vehicles)

Thus, an average truck represents 1,71 private vehicles and transports 12,3 tons. Knowing that an average 80% of the number of vehicles observed on the network are light vehicles, the residual capacity for trucks, expressed in tons, can be estimated as 0,2 x (Capacity of the road section/ 1,17) x 12,3.

The origin destination matrixes that were used and the original model are thoroughly discussed in Geerts and Jourquin (2001). These matrixes contain annual amounts of goods that moves between the origins and destinations, and only contains inter-urban and inter-regional data. This is quite different to urban O-D matrixes for peak hours for different reasons, but mainly because no information is available on the departure and/or arrival time. Moreover, as distances are long, a single trip often occurs during different periods of the day, including peak hours.

No capacity restrictions were introduced for rail or waterborne transport, i.e., their respective cost functions are not dependent on the already assigned flows.

In this exercise, we simulated a traffic during an average peak period. To do that, the saturation of the road network was estimated on the basis of the well-known criterion of the 30th peak hour. (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, this value is proven to be 13,6% of the daily flow.

Knowing that a small truck is in operation 241 days/year and large trucks work 264 days/year, we can estimate the average workload: (58 x 241 + 42 x 264)/100 = 250,66 days.

The quantity to be assigned on the network during an average peak hour can thus be estimated to (Annual quantity/250,66) x 0,136.

Now that both the residual capacity on the network is expressed in tons and that we have O-D matrixes for an average peak hour, we can test the different assignment methods on the real network. The results are shown in Figure 7. As could be expected, the

equilibrium solution is only slightly more expensive (+1%) that the AON initial solution.

Table 6: impact on the estimated modal split

	Modal split		
Method	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 assignment methods also have an impact on the market-share of the different transportation modes (see Table 6). Indeed, a cheaper path on a virtual network can very well be found using another transportation mode.

*Figure 7: assignment results on a real case*
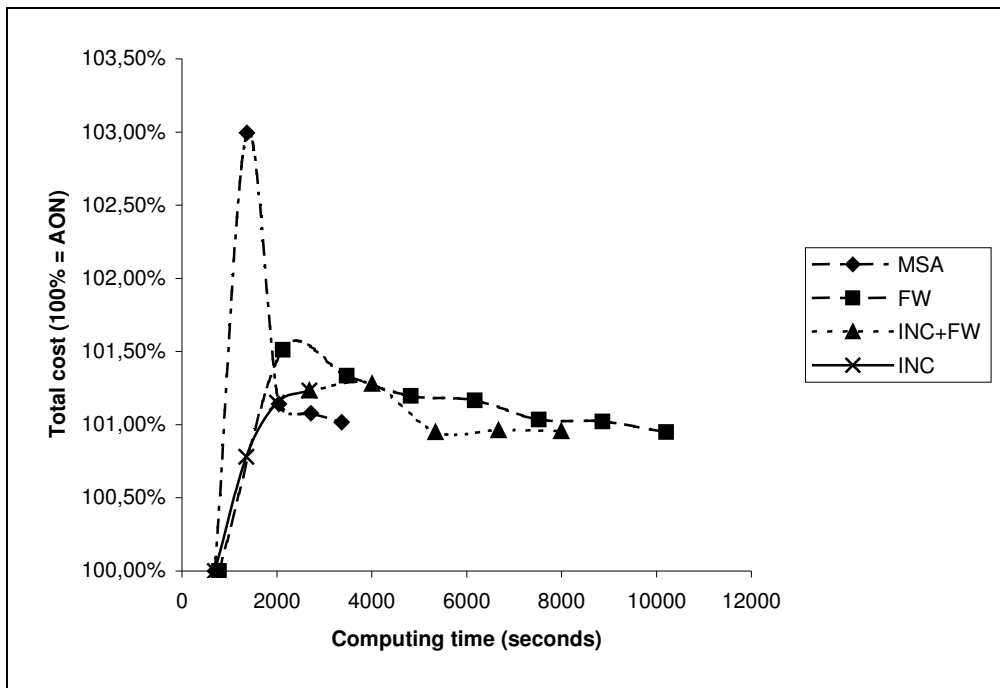
The various methods give different modal splits than the one obtained by an AON. Each time, there is a reduction of the tons.km transported by road due to the effects of congestion. For the incremental method, one notes a significant difference of the modal splits compared to the other methods. Indeed, once that a flow is affected on a link, it is not any more possible to withdraw a part of it and to assign it on another link.

The model was further more stressed, doubling the assigned quantities, to see how the different algorithms react on significantly more loaded network. The results of this exercise are presented in Figure 8 and Table 7. This time, the equilibrium solution is about 4,5% more expensive than the AON total cost.

Table 7: new modal split

Method	Modal split		
	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%

Figure 8: assignment results on a stressed real case
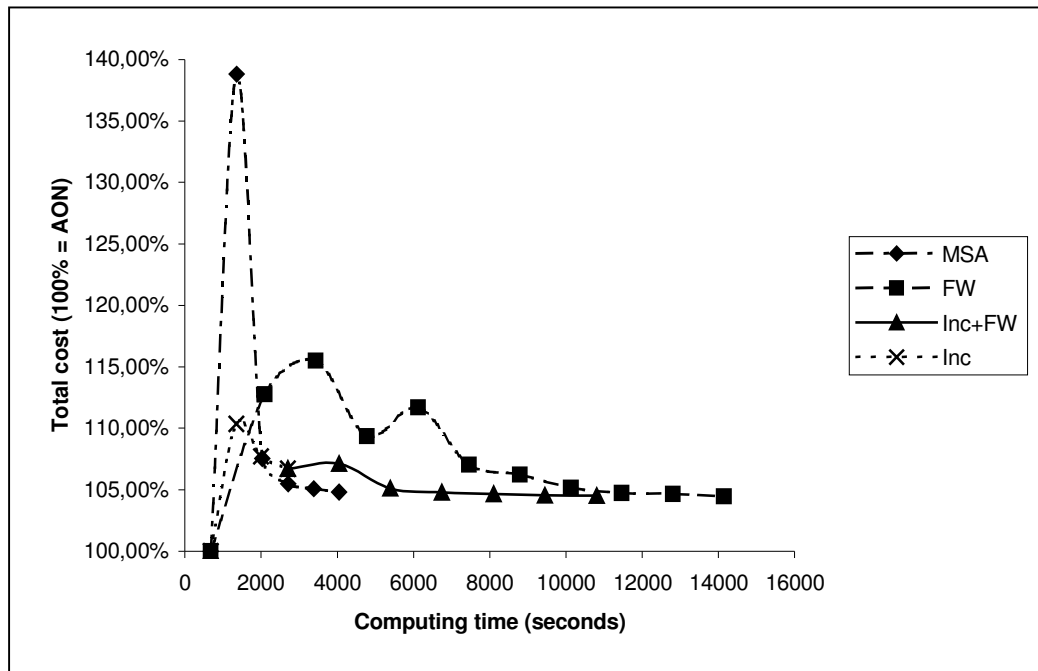
4   Conclusions

Several assignment techniques and algorithms were implemented on virtual networks to model multi-modal freight transport networks. The virtual network approach is different from the classical four steps models because it combines modal choice and assignment in a single step. The essential drawback of this methodology is that it generates huge networks, making computing times much more longer than those observed on classical "real" networks.

Modern computers now give the possibility to compute such large models, because both the power of their processor(s) and the amount of memory that is available are not any more to be compared with the hardware that was available some years ago.

All the equilibrium assignment methods that were implemented and tested in this paper are based on the All-Or-Nothing algorithm. Capacity constraints are taken into account by the incremental method, the Method of Successive Averages (MSA) and the Frank-Wolfe algorithm in order to approach the equilibrium condition stated formally by Wardrop. A variation based on the Frank-Wolfe algorithm, but using the result of an incremental solution as starting point was also tested.

One of the most interesting results obtained in this research is impact of the equilibrium models on the modal split. Indeed, as a virtual network can be seen as a mono-modal representation of a multi-modal network, an alternative path can very well involve the use of another (combination of) transportation mode. However, the incremental loading technique will not converge to the correct solution if one of the initial iterations assigns too much flow on a given link. This method can thus very well lead to incorrect modal split solution and is therefore not recommended.

It is also interesting to note that, on real case, the equilibrium solution is very close to one obtained after a single All-Or-Nothing assignment. In the case commented in this paper, the total cost on the system is just 1% higher than the initial AON cost. This is linked to the nature of the problem that is solved : one has to assign annual amounts of goods on long distances. Moreover, the trans-European network that is used doesn't contain the details of the network at the very local level, making it difficult to find close alternative routes. But isn't it a realistic option, as a truck driver coming from Paris and going to Antwerp doesn't know the details of the local networks around each city?

5 Bibliography

1. P.T. Harker. *Predicting intercity freight flows*. VNU Science press, 1987.
2. T.G. Crainic, M. Florian, J. Guélat, and H. Spiess. *Strategic planning of freight transportation: Stan, an interactive graphic system.* Transportation research record, 1283, 1990.
3. B. Jourquin. *Un outil d'analyse économique des transports de marchandises sur des réseaux multi-modaux et multi-produits: Le réseau virtuel, concepts, métho-*

des et applications. PhD thesis, Facultés Universitaires Catholiques de Mons, 1995.

4. B. Jourquin and M. Beuthe. Transportation policy analysis with a geographic information system: the virtual network of freight transportation in Europe. Transportation Research C, 4(6):359-371, 1996.

5. D.B. Johnson. *A note on Dijkstra's shortest path algorithm*. Journal A.C.M., 20:385-388, 1973.

6. E.W. Dijkstra. *A note on two problems in connection with graphs*. Numerische Mathematik, 1:269-271, 1959.

7. D.T. Kresge and P.O. Roberts. *Techniques of Transportation Planning: Systems Analysisand Simulation Models*. Brooking Institution, Washington DC, 1971.

8. A.G. Wilson and R.J. Bennet. *Mathematical Methods in Human Geography and Planning*. John Wiley & Sons, N-Y, 1985.

9. Wardrop,J.G. So*me Theoretical Aspects of Road Traffic Reasearch*. 1952 Proc. Inst. Civ. Eng.,Part II, 1, p325-378.

10. J. de D. Ortúzar and L.G. Willumsen. *Modelling Transport.* John Wiley & Sons, N-Y, 1990. p243-244

11. M. Frank and Wolfe. *An algorithm for quadratic programming*, Naval Research Logistics Quarterly, 3 (1956), p 95-110

12. Le Blanc, L.J., Morlock, E.K. and Pierskalla, W.P. *An Efficient Approach to Solving the Road Network Equilibrium Assignment Problem*. 1975 Transpn. Res., 9, p309-318.

13. Smock R.J. *An iterative assignment approach to capacity restraint on arterial networks*. Highway Reasearch Board Bulletin 156, 1962, 1-13.

14. Overgaard K.R. *Urban transportation planning : traffic estimation*. Traffic Quarterly, XXVI(2),197-218.

15. Bureau of Public Roads *Traffic Assignment Manual*. Urban Planning Division, US Department of Commerce, Washington DC, 1964.

16. Roy Thomas. *Traffic Assignment Techniques*. Avebury Technical, 1991. p122-127

17. Geerts J.F. and Jourquin B.**,** *Freight Transportation Planning on the European Multimodal Network: the case of the Walloon Region*, European Journal for Transport Infrastructure Research, Vol 1, No. 1, 2001.