APPLICATION OF NEW ALGORITHMS IN A TRASIT ASSIGNMENT PROBLEM FOR CONGESTED PUBLIC TRANSPORT SYSTEMS

Carlos Melo, Pontificia Universidad Católica de Chile, cfmelo@ing.puc.cl

José Enrique Fernández, Pontificia Universidad Católica de Chile, jef@ing.puc.cl

Joaquín De Cea, Pontificia Universidad Católica de Chile, jdc@ing.puc.cl

ABSTRACT

The use of a class of bush-based algorithms for the solution of user equilibrium (UE) traffic assignment problem is studied. In particular the analysis considers the use of a particular formulation of the transit assignment problem, which promises to produce highly precise solutions by exploiting the property of acyclic UE flows.
In the first stage of this work, we discuss and implement the Origin-Based Algorithm (OBA) in the diagonalization procedure for solving the transit assignment problem for congested public transport systems based on the formulation of De Cea and Fernández. A detailed analysis of the behaviour of OBA in this context is performed. It is found that the efficiency of OBA directly applied to congested transit assignment problems is not superior to traditional Frank-Wolf methods combined with diagonalization. The orientation of new developments for the improvement of OBA efficiency in the solution of congested transit assignment problems is discussed.

Keywords: user equilibrium, transit assignment, origin-based algorithm

INTRODUCTION

The transit assignment problem for congested public transport systems based on the formulation of De Cea and Fernández (1993) has asymmetric cost functions that can be solved, among others methods, by the diagonalization procedure. In this work, an implementation of the origin based assignment algorithm proposed by Bar-Gera (2002) is used to solve the convex problems generated by the diagonalization procedure, and its convergence characteristics are analyzed and compared with those obtained using the Frank-Wolfe algorithm (1956).

Previously, OBA has been incorporated into the diagonalization procedure implemented in SATURN (Hall and Van Vliet, 2002) to solve a traffic assignment problem with asymmetric cost functions. The reported levels of convergence reached with OBA are better than the ones obtained with the Frank-Wolfe algorithm and the execution times are similar. Nevertheless, the numerical precision reached with the incorporation of OBA into the diagonalization procedure is lower than the precision obtained when this algorithm is used to solve a road traffic assignment problem with separable and analytical cost functions, where a diagonalization procedure is not necessary (Bar-Gera and Van- Vliet, 2003).
In the light of the work carried by Bar-Gera and Van Vliet, it is feasible to incorporate OBA into the diagonalization procedure in order to solve the transit assignment problem considered. However, it is not possible to predict the results because this problem is different from the traffic assignment problem solved in SATURN where OBA was tested, principally due to the topology of the network where the trips are assigned.
In this work, we implement OBA in the transit assignment module for congested public transport systems and use a bus test network which allows a good analysis due to its spatial extent. We analyze in detail the behaviour of OBA in the context of the problem considered and discuss the main results found.
The article is organized as follows. In sections 2 and 3, a summary of the transit assignment problem and OBA is presented. In Section 4, some general considerations regarding the incorporation of OBA into the diagonalization procedure are discussed. In Section 5, the public transport system considered to test the implementation is described and the results obtained are analyzed.
In Section 6, a comparison with the Frank-Wolfe algorithm is made. Finally, in Section 7 the main conclusions of this work are presented.

THE TRANSIT ASSIGNMENT MODEL

The transit assignment model for congested public transport system considered in this work corresponds to the model proposed by De Cea and Fernández with minor modifications in its computational implementation. Regarding its basic definitions, a transit line or just a line is a fleet of homogeneous vehicles travelling on a specific cyclic route (or itinerary) over the network and stopping on a set of predefined nodes (transit stops) to allow the boarding and alighting of passengers. It provides a service level determined by its operating frequency, travel time between stops and service capacity. A line section is any portion of a line between two not necessarily consecutive nodes on its itinerary. A transit route is any sequence of nodes that a transit user can follow on the transit network in order to travel between any two nodes, the first one being the origin of the trip, the final being the destination and all the intermediate nodes representing transfer points. A route section is the portion of a route between two consecutive transfer nodes. Each route section is associated with a set of common lines which is the subset of lines that stop at its initial and final node and that minimize the expected total travel time (waiting time and in-vehicle travel time) (Chriqui and Robillard, 1975).
Regarding the general assumptions of the model, there will be many different transit routes composed by one or several consecutive transit sections that a passenger could use to travel

between two nodes on the transit network. Passengers select the route that minimizes their total travel time (in-vehicle plus waiting time).

It is assumed that the transit system has a limited capacity and therefore, congestion appears when passenger flow increases. The congestion phenomena is assumed to be concentrated at transit stops, where the passengers experience waiting times that depend on the total capacity of the attractive set of lines considered and the total number of passengers using that set of lines. Once a passenger boards a line, the in-vehicle travel time is assumed fixed and solely determined by the level of congestion over the road network, which is considered exogenous. When a passenger is travelling over a given transit route, at each transit stop he considers a set of common lines and he boards the first vehicle belonging to that set that has a place available. Passengers waiting at the same transit stop are classified in different groups, depending on their next transfer node, given that each of them has a different set of attractive lines; therefore, the services for which each group is waiting is different, with different capacities, occupancies and waiting times. Finally, the common lines set for any pair of transfer nodes depend on the level of congestion in the system.

In order to model the transit assignment problem, the road network *G(N, A)*, with node set *N* and link set *A*, is transformed into the virtual network *G(N, S)*, with link set *S* corresponding to the set of route sections over the network, with element *s*. An example of a network *G(N, S)* is illustrated in Figure 1, for a case with two lines. It can be noted that a large number of links are generated for each line and also that a large number of links reach each node (for example, see node C in Figure 1) which will have important implications in the implementation of OBA as discussed below in Section 4.

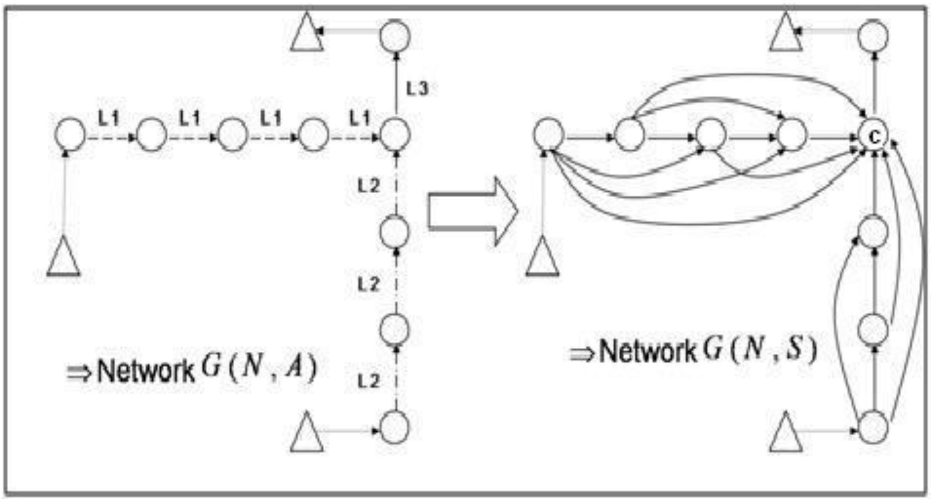

Figure 1 – Example of a network *G(N, S)*

The following notation is used:

1. *W*: Set of network origin-destination (O-D) pairs, with element $w = (i, j)$
2. *R:* Set of routes in *G* available to transit users, with element *r*
3. R_w: Set of feasible routes associated with O-D pair *w*

4. A_s: Set of all transit lines going from the origin node to the destination node of route section *s*

5. δ_{sr}: *Transit* link-route incidence matrix, with a value of 1 if route section *s* belongs to route *r* and 0 otherwise

6. *i(s)*: Origin node of route section *s*

7. f_l: Nominal frequency of line *l*

8. k_l: Capacity of line *l*

9. $f_s = \sum_{l \in \bar{A}_s} f_l$: Total nominal frequency on route section *s*

10. $K_s = \sum_{l \in \bar{A}_s} k_l$: Total capacity on route section *s*

11. *T*: Set of O-D transit demands

12. T_w: Transit demand between O-D pair *w*

13. $\bar{A}_s$: Set of common lines associated with route section *s*

14. h_r: Transit passenger flow over route *r*

15. c_s: Travel cost experienced by transit users on route section *s*

16. $\bar{t}_s$: In-vehicle travel cost including fare on route section *s*

17. $C_r = \sum_{s \in S} \delta_{sr} \cdot c_s$: Travel cost experienced by transit users over route *r*

18. v_l^r : Passengers travelling on line *l* over route section *r*

19. $\bar{v}_{il}$: Passengers taking line *l* before, and alighting after, node *i(s)*

20. $f_l^{'s}$: Effective frequency of line *l* at node *i(s)*

21. $f_s^{'} = \sum_{l \in \bar{A}_s} f_l^{'s}$: Total effective frequency on route section *s*

22. V^s: Total number of passengers boarding the route section *s* at node *i(s)*

23. V_{is}^{+} : Total number of passengers boarding at node *i(s)* all other route sections that use lines contained in route section *s*

24. $\overline{V}_{is}$: Total number of passengers boarding all the lines belonging to route section *s* at a node before *i(s)* and alighting after *i(s)*

25. S_{is}^{+}: The set of route sections going out of node *i(s)*, with the exception of route section *s*

26. $\overline{S}_{is}$: The set of route sections with initial node before *i(s)* and final node after *i(s)*

27. *V*: Route section flow, with element V^s

28. *C(V)*: Route section cost set, with element c_s

Passengers boarding route section *s* al transit stop *i(s)*, compete for the same common transit capacity with the following flow of passengers called "competing flow":

$$V_s = \sum_{l \in s} \left(\sum_{r \in S_{is}^{+}} v_l^r + \sum_{r \in \overline{S}_{is}} v_l^r \right) = V_{is}^{+} + \overline{V}_{is} \tag{1}$$

The effective frequency of line *l* belonging to the common lines set $\overline{A}_s$ is defined as:

$$f_l^{'s} = \frac{\max \left(k_l - \overline{v}_{il} \, , \varepsilon \right)}{k_l} \cdot f_l \tag{2}$$

where *ε* is a small positive constant. In this way the effective frequency of route section *s* considers available places in vehicles that arrive at transit stop *i(s)*. Considering what has been previously presented and taking an approach similar to that used to predict equilibrium conditions on congested road networks, the following general form for route section cost function is assumed:

$$c_s = \overline{t}_s + \left(\frac{\alpha}{f_s^{'}} \right) + \beta_s \cdot \left(\frac{V^s + V^{s}}{K_s} \right)^n \tag{3}$$

With:

$$\beta_s = \frac{\beta_0}{\sum_{l \in L} f_l^{'s}} \tag{4}$$

where *α*, *n* and β_0 are positive calibration parameters. Notice that, in the uncongested case, $f_s^{'} = f_s$ and $c_s = \overline{t}_s + \alpha / f_s$

The proposed functional form considers congestion at the transit stops correctly, by increasing the waiting times as total passengers flow increases (for more details see De Cea et al, 1998). Finally, it is easy to see that the incorporation of the concept of competing flow generates asymmetric interactions between route sections because the Jacobian of the cost function vector *C* is not symmetric in general.

After the definitions and assumptions made, the transit assignment problem can be formulated as the following variational inequality problem:

$$C\left(V^*\right)\left(V - V^*\right) \geq 0, \forall V \in \Omega \tag{5}$$

where the feasible set is defined by:

$$\Omega = \left\{ \begin{array}{c} \sum_{r \in R_w} h_r = T_w, \forall w \in W \quad (a) \\ \sum_{r \in R} \delta_{sr} \cdot h_r = V^s, \forall s \in S \quad (b) \\ h_r \geq 0, \forall r \in R \quad (c) \\ v_l^r = \frac{f_l^{'}}{f_s^{'}} \cdot V^s, \forall s \in S, \forall l \in A_s \quad (d) \end{array} \right\} \tag{6}$$

Notice that, the number of passengers using each line in each route section (v_l^r) is proportional to its relative effective frequency within the corresponding route section.

A feasible flow set is considered to be in equilibrium if it satisfies the following Wardrop conditions over the transit network *G(N, S)*:

$$C_r \begin{cases} = C_w^*, \forall r \in R_w / h_r > 0, \forall w \in W \\ \geq C_w^*, \forall r \in R_w / h_r = 0, \forall w \in W \end{cases} \tag{7}$$

where C_w^* is the equilibrium travel cost over all used routes that connect *w*.

To solve the variational inequality defined by (5), where the Jacobian of the cost functions *C(V)* is asymmetric, it is natural to use the well know diagonalization procedure (Florian and Spiess, 1982). In each iteration, based on the current solution *V*, the non-diagonal elements of the cost vector *C(V)* are fixed, in order to generate a symmetric problem. This problem differs from the classical separable traffic assignment problem by the presence of equations (6d) in the feasible set Ω, which involve the variables v_l^s. Nevertheless, these restrictions are completely separable from the generated problem because the variables v_l^s do not appear in the objective function which depends only on variables V_s. As a result, the problem generated can be solved as a standard traffic assignment problem with variables V_s over the network *G(N, S)*. Once an equilibrium solution is obtained variables v_l^s can be calculated using equation (6d). Therefore, the algorithm used in practice can be summarized as follows:

1. Step 0: (Initialization) Find an initial feasible solution (V^0, v^0). Make $k = 1$.

2. Step 1: (Diagonalization) In iteration *k*, using the current solution (V^k, v^k) the diagonalized cost vector is generated calculating for each route section *s* its common lines set $\overline{A}_s$, its parameter β_s and its competing flow V_s.

3. Step 2: (Resolution) Solve the convex problem generated to obtain a route flow solution (F^k). Make $X^k = \left(\dfrac{V^k + F^k}{2} \right)$.

4. Step 3: (Termination) If the desired convergence level evaluated at the solution (X^k) is reached, stop and consider (X^k) as the equilibrium solution. Otherwise, do (V^{k+1}) = (X^k) and return to step 1.

In the algorithm described, the initial solution for a new diagonalization is obtained making the average of the initial solution of the current diagonalization and the solution of the convex problem generated. This is a way of improving the convergence of the diagonalization procedure. Eventually, there could be other ways of averaging those solutions.

THE ORIGIN BASED ALGORITHM

The origin-based algorithm proposed by Bar-Gera (2002) (usually identified as OBA) is able to find in reasonable execution times an equilibrium solution that satisfies Wardrop's first principle for the traffic assignment problem with separable cost functions. In this case the solution precision is limited only by the numerical accuracy of the computer used. This marks an important difference with respect to the popular Frank-Wolfe algorithm (1956) that can only reach a limited accuracy due to the well-known "jumping" convergence phenomenon (Patriksson, 1994).

In OBA, the main solution variables are the origin-based approach proportions α_{ap} defined for every link *a* and every origin *p*, such that for every origin *p* the sum of the origin based proportions over all links ending at node *i* is equal to one. The variable α_{ap} corresponds to the proportion of the flow originated at *p* that arrives at the node *i* by the link *a*.

A key point in the algorithm is that for every origin an a-cyclic restricting subnetwork A_p is chosen, such that the approach proportions for the links that are not included in A_p are equal to zero. The restriction to solutions that are a-cyclic allows the definition of a topological order of the nodes in every subnetwork such that every link in the restricting subnetwork goes from a node of lower topological order to a node of higher topological order. Therefore, most of the computations are done in a single pass over the nodes, either in ascending or descending topological order, in a time which is a linear function of the number of links in the network. This computational efficiency is the main reason for restricting the process to a-cyclic solutions.

In solving the traffic assignment problem, the algorithm starts with trees of minimum cost routes as restricting subnetworks, leading to an all-or-nothing assignment.

Then, the algorithm considers all origins in a sequential order, updating its restricting subnetwork and the origin-based approach proportions. To update a restricting subnetwork A_p, unused links are removed ($\alpha_{ap} = 0$); ζ_i the maximum cost from the origin *p* to node *i* in the

restricting subnetwork is computed for all nodes, and all links [*k, i*] such that $\zeta_k < \zeta_i$ are added to the restricting subnetwork. Due to the quick stabilization of the restricting subnetworks, it is useful to update origin-based approach proportions while keeping the restricting-networks fixed. This is done by introducing inner iterations.
To update the origin-based approach proportions within a given restricting subnetwork, a search direction based on shifting flow from high cost alternatives to low cost alternatives is used. In addition to current costs, estimates of cost derivatives are used to improve the search direction in a quasi-Newton fashion. The second-order search direction is viewed as desirable flow shifts which are scaled by a step size between zero and one, and then truncated to guarantee feasibility. This technique is referred to as the boundary search procedure, since it tends to choose solutions along the boundary although it does consider interior points as well. In order to guarantee descent of the objective function, and convergence of the algorithm, the search considers step size values of 1, ½, ¼, …, and finds the first value of the sequence that has "positive social pressure", which corresponds to the sum of the "social pressure" from all users. The basic idea of this concept is that every user that shifts from route r_1 to route r_2 applies pressure (positive or negative) which is equal to his/her gain (or loss) according to the difference in route costs that results from the shift (Kupiszewska and Van Vliet, 1999).
Therefore, OBA has the following general structure:

Initialization:

For every origin *p*:

> A_p = Tree of minimum cost routes from *p* (free flow conditions)
>
> f_p = All-or-nothing assignment using A_p

Main Loop:

For a number of main iterations:

> For every origin *p*:
>
> > Update restricting subnetwork A_p
> >
> > Update origin-based link flows f_p
> >
> > For a number of inner iterations:
> >
> > > For every origin *p*:
> > >
> > > > Update origin-based link flows f_p

GENERAL CONSIDERATIONS

When OBA is used in the diagonalization procedure as described before, in any given iteration *k*, it is necessary to average two solutions of approach proportions α_{ap}, in order to create a new solution α'_{ap} that generates the total flow solution X^k that enters as initial solution of the diagonalization $k+1$: i. the initial solution of diagonalization k, used to calculate the total flow solutions V^k and ii. the solution F^k of the convex problem generated in diagonalization k. In general, this is difficult to accomplish due to the definition of variables α_{ap}. Therefore, the total flow solution X^k obtained as the average of solutions V^k and F^k is used only as auxiliary variables to update the cost functions of the next diagonalization, whose initial solution is the solution of approach proportions α_{ap} associated to the total flow solution F^k. This can be done because OBA has the ability to quickly adjust the values of variables α_{ap} to revised cost conditions (Bar-Gera and Van Vliet, 2003). This way, the inconsistency between cost functions and solutions α_{ap} is overcome and well converged solutions can be retained. This problem does not exist when using the Frank-Wolfe algorithm, because the solution X^k is used for updating the cost functions and also constitutes the initial solution for the next diagonalization iteration.

Moreover, it is not possible to predict the behavior of OBA in the diagonalization method described above due to the topology of the network *G(N, S)* where the trips are assigned. Because of the route section concept, this network has more links than road networks and a very different distribution of the incoming links to its nodes. In a road network, on average there are only between two to three incoming links to each node, but in a network *G(N, S)*, although some nodes can present few incoming links, there are **many nodes with hundreds and even thousands of incoming links**. This characteristic of the transit network services, affects significantly to OBA because its operation is mainly based on analyzing the incoming links to every node in every subnetwork. On the other hand, if a public transport system has a reduced number of transfers, the majority of the routes will only have one route section (O-Ds will be linked by only one route section). In this case, the behaviour of OBA is uncertain because it will have to shift flows between incomings links to nodes but, the majority of the nodes will not have incoming links with flow or will have only one link with flow. In this way, only in a few nodes flow shifts will occur.

Therefore, in comparison with a road network, in a transit network for every node of every subnetwork, OBA will analyze on average a significantly larger number of incoming links, but only in a few of them flow shifts will occur.

OBA PRACTICAL IMPLEMENTATION AND TEST RUNS

The practical implementation of OBA into the diagonalization method was done using a software implemented in the C language and the tests were performed in a computer with processor XEON, 1.8 Ghz, 1.5 GB RAM, 2 GB SWAP, with Linux Red Hat 7.2 operating system.

The behaviour of OBA in the diagonalization procedure is evaluated in terms of the convergence level reached measured by the Relative Excess Cost (REC) indicator defined as:

$$REC = \frac{\sum_{s\in S} c_s \cdot \ V^s - Y^s}{\sum_{s\in S} c_s \cdot V^s} \tag{8}$$

where Y^s is the route section flow vector of an all-or-nothing assignment based on the route section cost vector c_s. The set *S* in expression (8) includes all route sections and all access links.
The test network used to evaluate OBA is a test network, corresponds to a public transport system that runs over a road network with 792 nodes (219 zones and 573 transition nodes) and 2217 links (810 access links and 1407 road links). The transit system has 155 lines that use 935 road links, with 476 transit stops connected to 215 zones centroids. The corresponding virtual network *G(N, S)* generated has 691 nodes (215 zones and 476 transition nodes) and 59130 links (377 access links and 58753 routes sections). The O-D matrix assigned to this network has a total of 88,200 trips corresponding to people travelling during the morning peak hour.
A test of the diagonalization procedure was carried out using different number of OBA iterations within each diagonalization. Figure 2 presents the convergence characteristics obtained for the best case (the one with the best convergence) obtained when seven main iterations of OBA were performed per diagonalization, without using inner iterations.

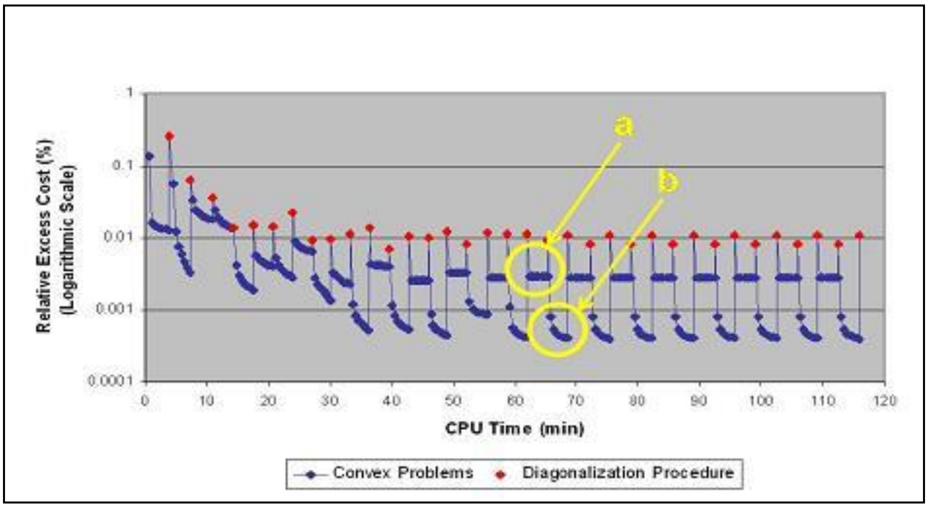

Figure 2 – Convergence of the diagonalization procedure (35 diagonalizations) and the convex problem generated (using 7 iterations of OBA)

For a given iteration of the diagonalization procedure, the seven blue points[1] represent the convergence indicator's values within each OBA iteration in the convex problem generated, and the red point that follows represents the value of the convergence indicator obtained once the convex problem has been solved. Notice that this red point convergence value is different than the convergence value of the last iteration of the algorithm (blue point immediately before); this is due to the use of updated cost functions to adequately evaluate the convergence indicator of the diagonalization procedure. As can be seen, after the twelfth convex problem generated (39 minutes), an oscillatory behaviour is observed in the value of the convergence indicator. In an iteration (identified by "a" in the Figure) the corresponding convex problem presents the same REC value for all the 7 iterations and in the next one (identified by "b") a certain reduction is obtained for each iteration. This will be explained below in this Section. As can be seen in the Figure, after 40 minutes of operation, the

[1] Notice that the seven points appear close together in the figure, looking as a curved short line

convergence indicator of the diagonalization procedure begins to oscillate between the values 0.011% and 0.008%, being unable to improve those values.

Theoretically, OBA should be able to find the unique solution to every convex problem generated with a level of convergence similar to that observed in the traffic assignment problem with separable cost functions, where the limitation is only given by the numerical precision of the computer used. Nevertheless, this does not happen. To explain this phenomenon, the solution of the convex problems is analyzed in detail.

Due to the difficulty of carrying out a detailed analysis for each of the 215 subnetworks, considered in the example used, the analysis made for an "average" subnetwork. The "average" subnetwork has the topological characteristics obtained averaging the topological characteristics of all the individual subnetworks. In particular the average subnetwork has an origin that generates a number of trips, corresponding to the average of trips generated by all the origins; this are directed to the 214 "average" destinations, that is the same number for all subnetworks. Every "average" destination receives a number of trips calculated as the average over all origins. Bar-Gera in his work (Bar-Gera, 1999) used a different but equivalent approach to analyze the subnetworks: he carried out an analysis in total terms, adding the results of all subnetworks.

First, in Figure 3 we analyze for every 7 iterations of OBA performed in the 35 diagonalizations completed the evolution of the number of total links present, total links removed and total links added to the average subnetwork. Second, in Figures 4 and 5, the disaggregation of the total links added in terms of basic links and non-basic links is analyzed. Third, the evolution of the number of links with flow, disaggregated in route sections and access links, is presented in Figure 6.

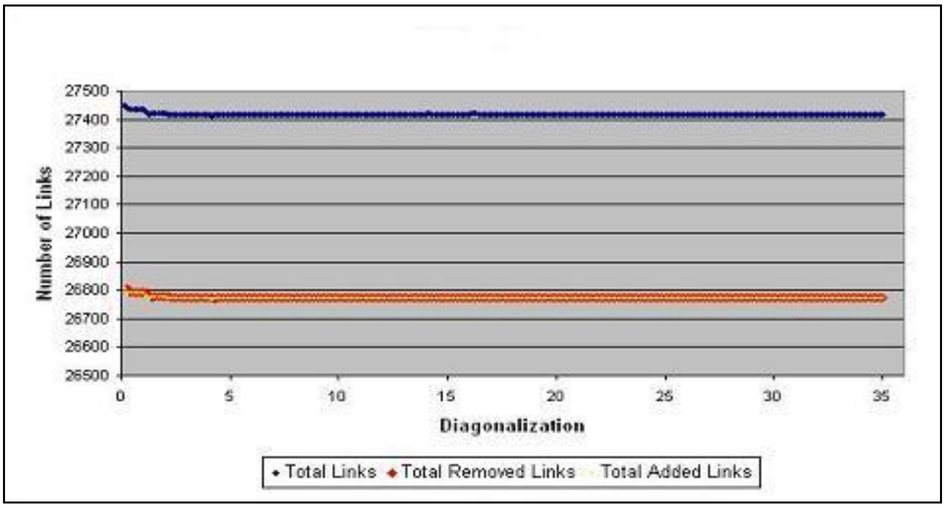

Figure 3 – Analysis of the subnetwork structure

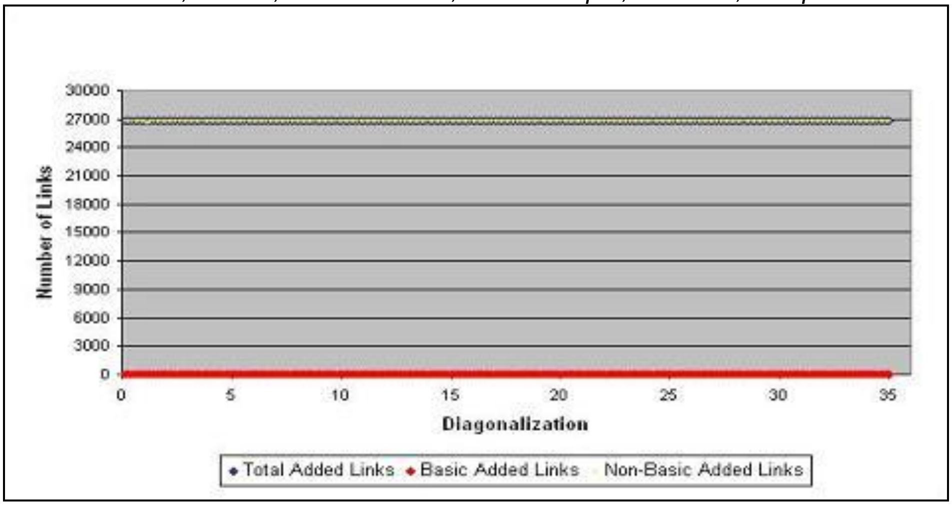

Figure 4 – Analysis of the subnetwork structure (cont.)

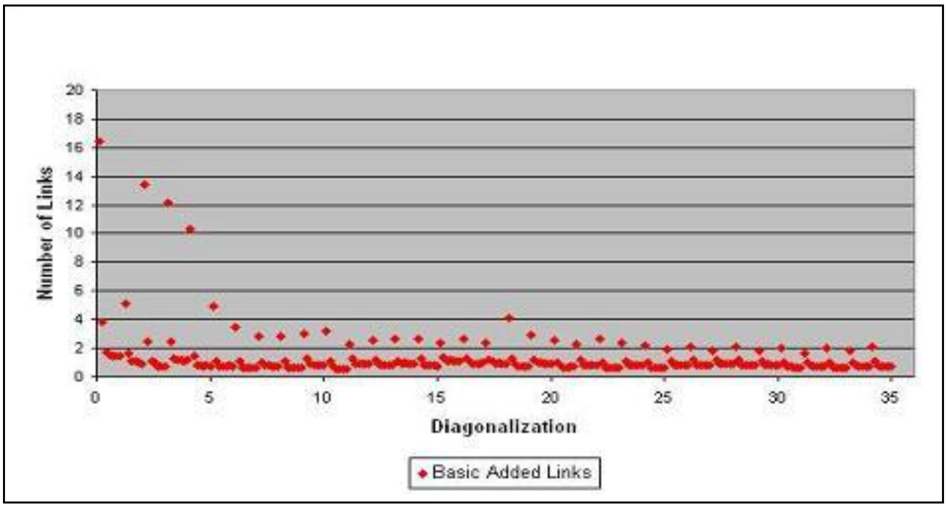

Figure 5 – Analysis of the subnetwork structure (cont.)

As shown in Figure 3, in every convex problem generated the average subnetwork has a total of 27415 links, independent of the values that may have the approach proportions calculated and the corresponding flows values obtained based on the average origin; moreover the number of links removed and added are approximately 26770, carrying out a practically complete modification of the average subnetwork. In terms of this analysis, all the convex problems generated are similar: from the second iteration of OBA, the values considered practically do not change.

As noted in Section 3, in every main iteration, the algorithm updates for every origin (in a sequential order) the restricting subnetwork and its approach proportions. The objective of the updating process of a subnetwork is to eliminate all the unused links (and the unused associated routes) and to aggregate all the links (and the associated routes) that do not violate the characteristics of the subnetwork and could eventually carry flow from its origin. The only way an added link could eventually carry flow from its origin is becoming a basic link. If it becomes a non-basic link, it will continue having an approach proportion value equal to zero and therefore will be eliminated from the subnetwork in the next main iteration.

As shown in Figures 4 and 5, practically all the links added to the average subnetwork become non-basic links, being therefore eliminated in the next iteration of the algorithm. In particular, Figure 5 shows that after the fifth convex problem generated, only 1 or 2 links of the approximately 26770 links added become basic links and could eventually receive flow from the average origin.

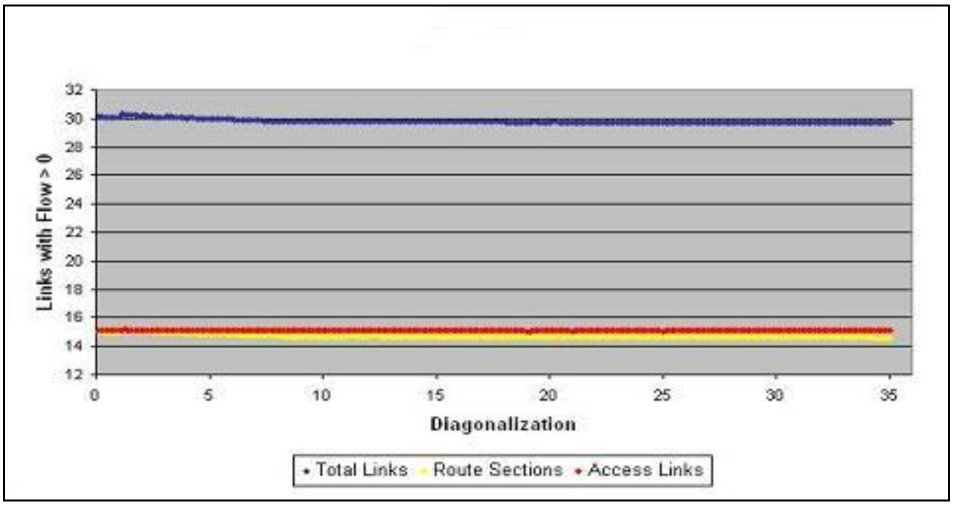

Figure 6 – Analysis of the subnetwork structure (cont.)

In Figure 6, we can observe that only 30 of the approximately 27415 links that exist in the average subnetwork have flow from the average origin; 15 are route sections and the other 15 are access links. At first sight, this result seems to be surprising.

However, the trip matrix assigned has only 6.5% of its cells with trip values greater than 0: this means that for the average subnetwork only 6.5% of the destinations demand trips (approximately 14 in 215).

To travel between the average origin and these 14 destinations, at least 15 access links (1 to leave the origin and 14 to arrive to every destination) are needed, which is exactly what happens in the average subnetwork.

On the other hand, for the transit system used in the example 92% of the trips do not carry out transfers and most trips with transfers present only one transfer. Therefore, around 92% of the trips only use 1 route section and the majority of the remaining 8% use only 2 route sections. However, this fact alone does not necessarily explain that the average subnetwork should have 15 route sections with flow.

For example, if each of these 14 destinations is connected by a unique access link with flow to the same transition node, there would be only 2 transition nodes acting as access nodes, 1 after leaving the origin and 1 before arriving to each destination, eventually connected by 1 route section. To complete the 15 route sections with flow, there would have to be a certain number of additional transition nodes with incoming and outgoing route sections with flow. In a situation like this, if most of the trips use the route section that directly connects the 2 access nodes (transit stops) and the remaining trips use the remaining route sections with flow (which involve transfers), the percentages of trips with transfers would be satisfied. In this topological situation of the route sections with flow, the only access node towards the destinations would have several incoming links with flow (see Figure 7) and the algorithm should be able to shift flow between them, eventually finding the unique solution of the convex problem generated. Even in the case of the convex problem of a diagonalization generated after the first one, where in general the link cost functions are not consistent with the solution of approach proportions, the algorithm should be able to find the unique equilibrium solution because it is able to shift flows in order to adjust the approach proportion solution. However, this topology of the links with flow is not the most probable because the destinations that demand flow are in general connected to different access nodes (in general, the destinations are spatially distant between each other). Therefore, this topology of the links with flow could be seen as a first extreme case.

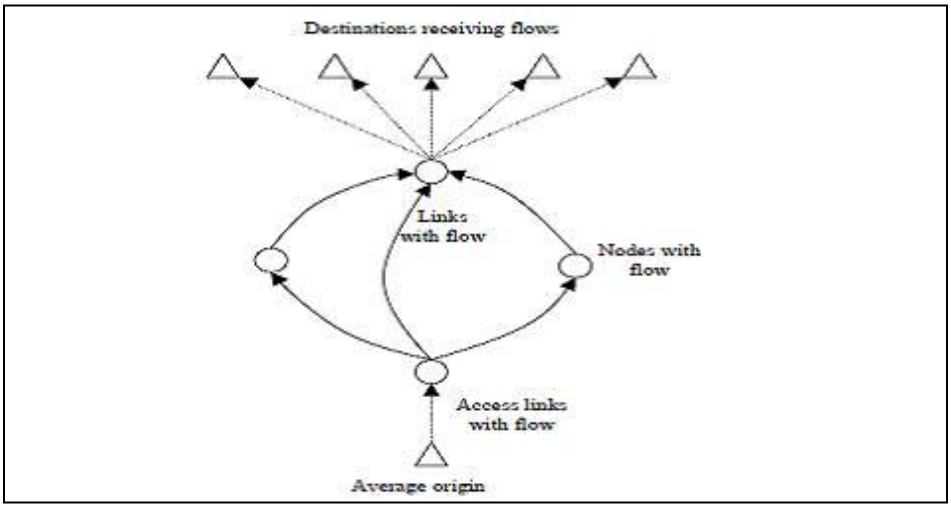

Figure 7 – First extreme case of topology of links with flow in the average subnetwork

Another extreme case would be that each of the 14 destinations demanding trips were connected with its unique access link to a different transition node.
Therefore, in order to have 15 route sections with flow, and considering that 8% of the trips require transfers, there could be 13 of the 14 destinations requiring only 1 route section to be accessed from the origin (flows without transfers) and the remaining destination requiring 2 route sections because, for example, there is no line directly connecting the origin with this destination (see Figure 8).

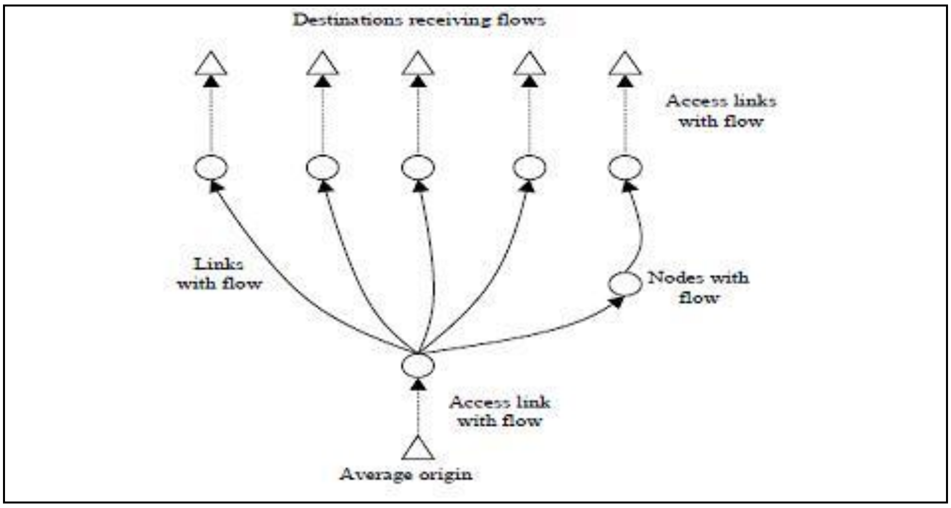

Figure 8 – Second extreme case of topology of links with flow in the average subnetwork

In a case like this, every link with flow leaving the origin arrives to a different node connected to a destination. When the diagonalization procedure is in an iteration different than the first one, for the algorithm to be able to shift flows (to adjust the approach proportions to the revised link cost functions and then eventually find the unique equilibrium solution) at least one of the links added to the average subnetwork must become a basic link of a node that has an incoming link with flow. If this does not happen, the algorithm is unable to shift flows. In addition, because of the inconsistency between the approach proportion solution and the link cost vector for every subnetwork, the all-or-nothing assignment necessary to obtain the convergence indicator of the generated convex problem should not necessarily coincide with the current total flow solution.

The reality of the public transportation trip distribution in the network used in this example is closer to the second extreme case. It is possible that some of the 14 destinations demanding trips located in the downtown (which is the area that concentrates most trips destinations) be connected to the same access node. However, given that this central area is quite large, only a few of their destinations that demand trips could be in a situation like this. The remaining destinations receiving flows, are directly connected by a unique access link from an access node. The majority of the access nodes with flow considered are in this last situation; only a few nodes that do not have a direct connection from the origin are connected by more than one route section with transfers in an intermediate node (see Figure 9).

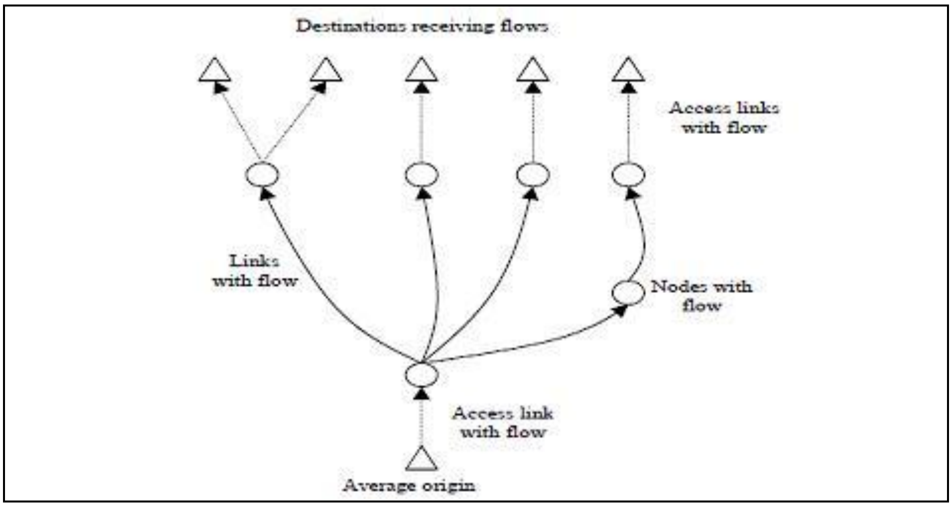

Figure 9 – Expected case of topology of links with flow in the average subnetwork

This case is more similar to the second extreme example where, every link with flow arrives to a different node. Therefore, there will be a shift in flows only if some of the links added to the average subnetwork that became basic are incoming links to nodes that have an incoming link with flow. In Figure 5, it can be seen that only 1 of the 26770 links that are added to the average subnetwork becomes a basic link, not being removed in the next iteration and being able to potentially receive flow. If this new link that became basic arrives to a node without an incoming link with flow, there will not be a flow shift and the solution of flow, based in the average origin, will not change. There will only be a flow shift if this new link that became basic arrives to a node with an incoming link with flow.

In this case, the convergence indicator should have a descent in its value, but not necessarily reaching a value equal to 0 because of the problem mentioned before.

After carrying out the above analysis, it is possible to conclude that the oscillatory phenomenon observed in the Figure 2 occurs because, in one generated convex problem, the added link that becomes basic arrives to a node without incoming links with flow; therefore, the algorithm cannot modify the solution, but in the following convex problem, the added link that becomes basic arrives to a node that has an incoming link with flow, allowing OBA to locally modify the solution.

Therefore, due to the topology of the links with flow in the subnetworks and the inconsistency between the link cost vector and the approach proportions, after some relatively small number of diagonalizations, the algorithm is in general unable to modify the solution of the convex problem generated. However, in the first generated convex problem (of the first diagonalization), there is no inconsistency between the link cost vector and the approach proportions because these solutions were created using the link cost vector of this first convex problem. Hence, without the presence of the inconsistency, the algorithm is able to find the unique equilibrium solution of this first convex problem. Figures 10, 11 and 12 show the convergence characteristics of the first convex problem solution, for an example in which 500 OBA iterations are completed. It can be clearly seen that the convergence indicator stabilizes approximately in a value of 0.011% after only 15 iterations (10 minutes) of the algorithm run.

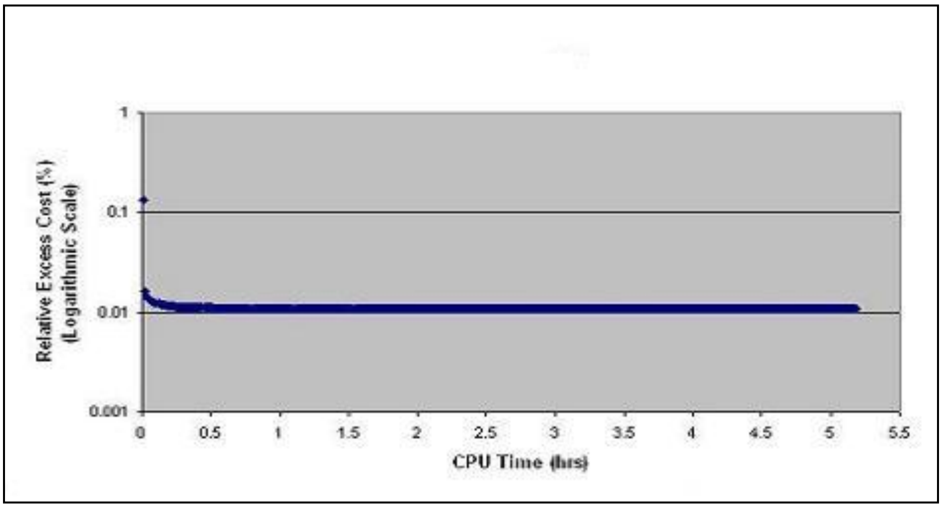

Figure 10 – Convergence of the first convex problem

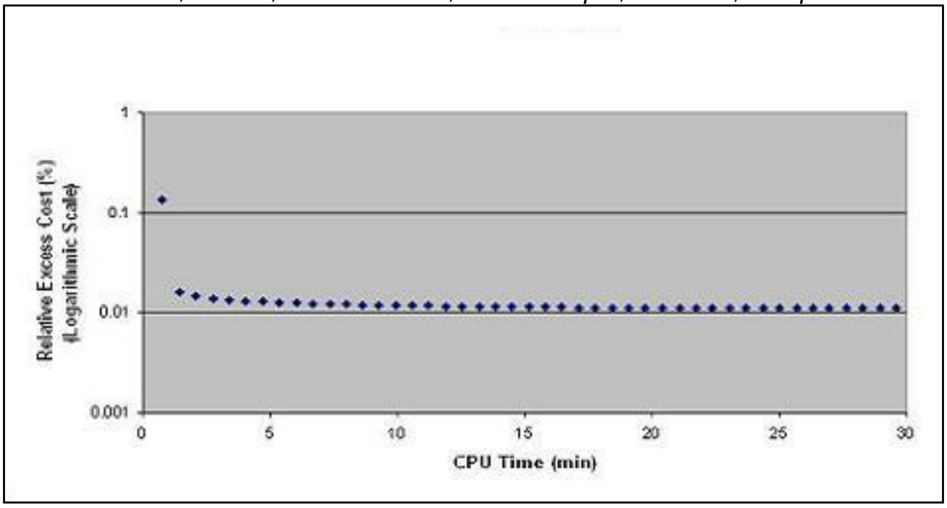

Figure 11 – Convergence of the first convex problem (cont.)

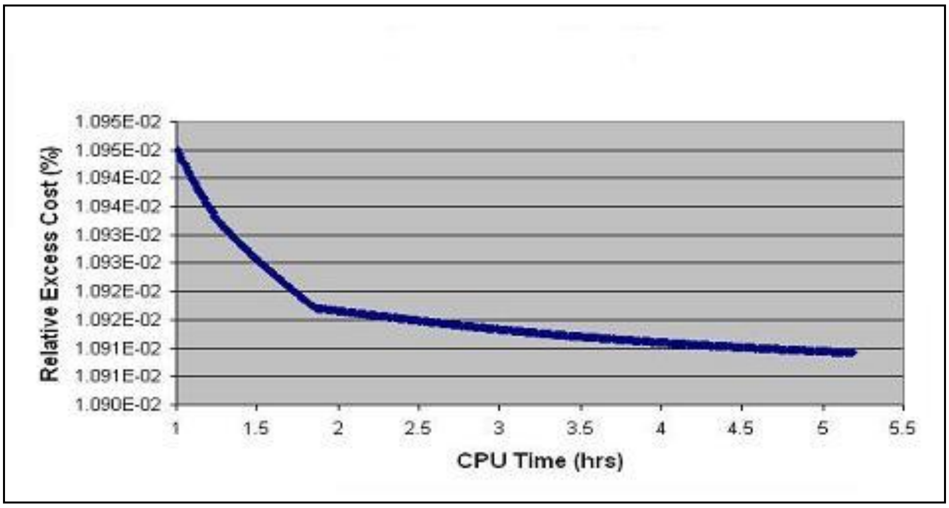

Figure 12 – Convergence of the first convex problem (cont.)

For the next 485 iterations, the rate of descent between two consecutive iterations is so small (1E-6% before the 2 hours of execution time and 1E-8% after the 2 hours) that for practical considerations, the convergence indicator does not change. In the first convex problem generated, once a certain level of convergence has been reached, the algorithm, with its current formulation, is not able to improve the solution. Hence, it cannot find the unique solution with the levels of convergence accuracy obtained for the traffic assignment problem with separable cost functions. In principle it could be thought, that the algorithm should be able to solve this network convex problem with the same accuracy in both cases (though not necessarily in the same execution time). However, after the analysis performed in this section, it seems reasonable to conclude that the particular topology of the network *G(N,S)* is the cause of the problem.

Finally, it is possible to conclude that in its current form, implemented in the diagonalization OBA has problems to improve the solution of the public transport assignment problem considered after a few iterations (only 15) are performed due to the particular topology of the public transport network *G(N, S)*.

COMPARISON BETWEEN OBA AND THE FRANK-WOLFE

After analyzing in the previous section the behaviour of OBA when used within the diagonalization procedure to solve the public transport assignment problem considered, in this section a comparison is made with the behaviour observed when using the Frank-Wolfe algorithm (Frank and Wolfe, 1956). This algorithm corresponds to the state of practice to solve the transport network assignment problem in most software used worldwide and in particular in ESTRAUS[2]. Figures 13 and 14 show the levels of convergence reached by OBA compared with those reached by the Frank-Wolfe algorithm with 2200 iterations.

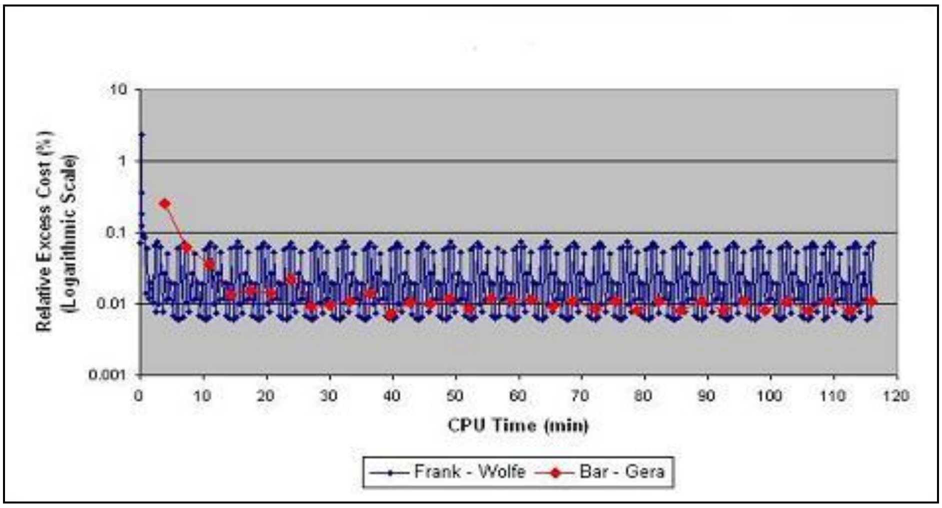

Figure 13 – Comparison of convergence results for OBA vs. F-W

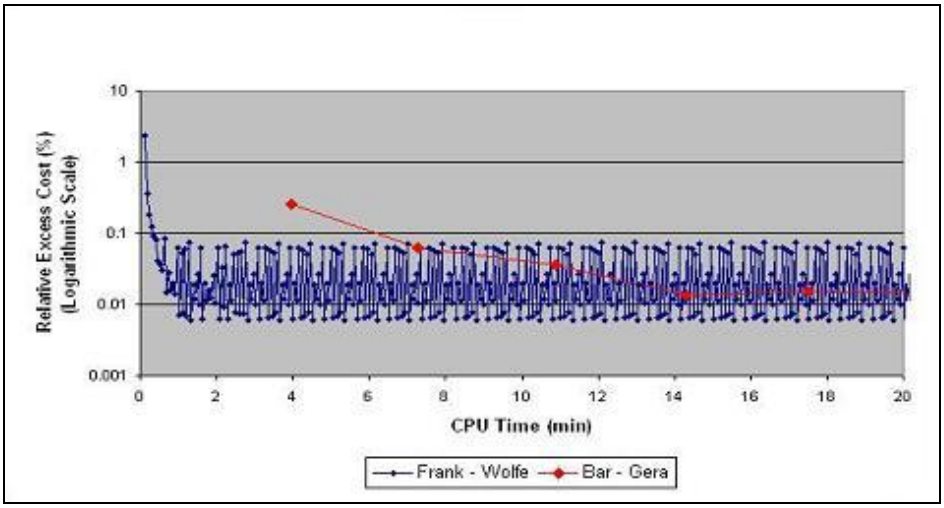

Figure 14 – Comparison of convergence results for OBA vs. F-W (cont.)

When the Frank-Wolfe algorithm is used, after 1.5 minutes of run the convergence indicator of the diagonalization procedure oscillates between an upper bound of 0.072% and a lower bound of 0.006%. This behaviour is similar to OBA, in terms of lower bound levels but with a wider oscillation in the convergence indicator. In effect, the lower bound is almost the same

[2] Chilean software for Transport Networks Planning (De Cea et al., 2003)

for both algorithms (approximately a 0.007%) while the upper bound of the Frank-Wolfe algorithm is higher (approximately 0.07% and 0.01%). Nevertheless, the Frank-Wolfe algorithm requires only 1.5 minutes to enter in its oscillation process while OBA requires about 65 minutes. Although there is a higher execution time required, an advantage of OBA is its lower level of oscillation.

CONCLUSIONS

In this work, the OBA algorithm (Bar-Gera, 2002) was implemented within the diagonalization procedure in order to solve the transit assignment problem for congested public transport systems. The implementation was based on the model by De Cea and Fernández (1993).

In the public transportation system used to test the implementation, the levels of convergence reached were not as good as the ones obtained from the traffic assignment problem with separable cost functions. The reason for this phenomenon is mainly due to the topology of the public transport network, which is significantly different from a typical road network due to the route section concept. The use of this concept produces a denser network with a significant increase in the number of incoming links to the nodes in comparison to a road network. Also the routes that connect OD pairs have very few route sections, and many times only one route section connects an OD pair without intermediate transfer nodes.

The result is that a typical restricting subnetwork created in OBA has a significantly higher ratio (number of links / number of nodes) than the typical restricting subnetwork of the traffic assignment problem with separable cost functions. Only very few of this links have flow coming from its respective origin and practically each one of them arrive to a different node.

In all diagonalizations, even though the current solution of approach proportions α_{ap} is not optimal for the cost functions of the convex problem generated, OBA, with its current formulation, has problems to shift flows and therefore the inconsistency between costs and approach proportions cannot be solved. Therefore, the convergence indicator of the convex problems and the diagonalization procedure practically cannot be improved. In principle it could be thought that OBA should be able to solve the convex problem generated in every diagonalization in the similar way as in the case of the traffic assignment problem with separable cost functions. Nevertheless, as it was shown in this paper it is possible to affirm that this is a consequence of the particular topology of the public transport network.

As a result, it is possible to state that OBA, in its current form, can be implemented into the diagonalization procedure for solving the transit assignment problem for congested public transport systems considered, but the results obtained in terms of the convergence indicator are inferior to the ones obtained in the traffic assignment problem with separable cost functions when the transit system has a reduced number of transfers. In the case analyzed, OBA reached the same level of convergence as the Frank-Wolfe algorithm, with less oscillation, but with significantly greater requirements in execution time.

However, the evaluation of the behavior of OBA in the diagonalization procedure considered when the transit system has an important number of transfers remains a subject of future investigation.

Finally, as a subject of future research, modifications to OBA should be studied regarding the theoretical and computational implementation in order to adapt the way it functions considering the specific topology of the network in the transit assignment problem for congested public transport systems presented in this work. We believe that it may be possible to alleviate some of the resulting difficulties by various adjustments to the OBA algorithm; for example, by adding new links only if they are to become basic, the question is how to determine which links "are to become basic". We expect to answer this question in a future publication based in ongoing research.

REFERENCES

Bar-Gera, H. (1999). Origin-based Algorithms for Transportation Network Modelling. Ph. D. Thesis, Civil Engineering, University of Illinois at Chicago, USA.

Bar-Gera, H. (2002). Origin-based Algorithms for the Traffic Assignment Problem. Transportation Science, 36 (4), 398-417.

Bar-Gera, H. and Van Vliet, D. (2003). The Application of Origin-based Assignment to Saturn Networks with Asymmetric Cost Functions. The Workshop on Traffic Modelling: Trends and Challenges. Sitges, 2-4 June 2003.

Chriqui, C. and Robillard, P. (1975). Common Bus Lines. Transportation Science, 9, 115-121.

De Cea, J. and Fernández, J.E. (1993). Transit Assignment for Congested Public Transport Systems: An Equilibrium Model. Transportation Science, 27, 133-147.

De Cea, J., Fernández, J.E., Decock, V., Soto, A., and Friesz, T. (2003). ESTRAUS: A computer Package for Solving Supply-Demand Equilibrium Problems on Multimodal Urban Transportation Networks with Multiple User Class. Transportation Research Board Annual Meeting. Washington, January 2003.

Florian, M. and Spiess, H. (1982). The Convergence of Diagonalization Algorithms for Asymmetric Network Equilibrium Problems. Transportation Research B, 16 (6), 477-483.

Frank, M. and Wolfe, P. (1956). An Algorithm for Quadratic Programming. Naval Research Logistic Quarterly, 3, 95-110.

Hall, M.D. and Van Vliet, D. (2002). The SATURN 10.2 User Manual, W.S, Atkins, Epsom.

Kupiszewska, D. and Van Vliet, D. (1999). 101 Uses for Path-Based Assignment. Proceedings of the 27th European Transport Annual Meeting: Transportation Planning Methods. PTRC, London.

Patriksson, M. (1994). The Traffic Assignment Problem – Models and Methods. VSP, Utrecht.