European Journal of Operational Research 204 (2010) 20–34

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

A node rooted flow-based model for the local access network expansion problem

Margarida Corte-Real [a,*], Luís Gouveia [b]

[a] FEG, Universidade Católica Portuguesa, Rua Diogo Botelho, 1327, 4169-005 Porto, Portugal
[b] CIO and DEIO, F.C., Universidade de Lisboa, Bloco C6, Piso 4, 1749-016 Lisboa, Portugal

ARTICLE INFO

Article history:
Received 31 March 2009
Accepted 2 October 2009
Available online 12 October 2009

Keywords:
Network flows
Local access network
Capacitated Minimum Spanning Tree Problem
Valid inequalities

ABSTRACT

In this paper, we present a new formulation for the local access network expansion problem. Previously, we have shown that this problem can be seen as an extension of the well-known Capacitated Minimum Spanning Tree Problem and have presented and tested two flow-based models. By including additional information on the definition of the variables, we propose a new flow-based model that permits us to use effectively variable eliminations tests as well as coefficient reduction on some of the constraints. We present computational results for instances with up to 500 nodes in order to show the advantages of the new model in comparison with the others.

1. Introduction

A local access network is composed by a set of customer nodes, a central office and several links that allow the transmission of the traffic from one customer to another through the central office. The network has a tree structure that connects the customers to the central office, located at the root of the tree. This work addresses the local access network expansion problem (LANEP), where an existing network must be adapted to traffic increase either by expanding the capacity of the links and/or by installing concentrators in some nodes. Concentrators are electronic devices that compress the traffic and can be installed in the network in order to eliminate or reduce the need of the links expansion. Balakrishnan et al. (1991) and, more recently, Carpenter and Luss (2006) have presented the main characteristics of several problems arising in the context of access network planning. Some variants of the local access network expansion problem have also been considered in the literature, with different kinds of concentrators and transmission links and for single and multi-period versions (see, for instance, Bienstock (1993), Lee (1993), Balakrishnan et al. (1995), Cho and Shaw (1996, 1998), and Flippo et al. (2000) for single-period versions of the problem and Jack et al. (1992), Shulman and Vachani (1993), Gendreau et al. (2006) and Kouassi et al. (2009) for multi-period versions).

The problem we address here is the expansion of a local access network with a fixed tree topology. We consider the single time period version of the problem, where we want to determine, with minimum cost, which links need to be expanded and/or where the concentrators should be located in order to guarantee that the node demands can be sent to (or from) the central office. We consider the version of the problem with uncapacitated concentrators. It was shown in Corte-Real and Gouveia (2007) that the LANEP could be seen as an extension of the well-known Capacitated Minimum Spanning Tree Problem (CMSTP) (see, for instance, Gavish (1993) and Gouveia and Lopes (2000)) and, as a consequence, two flow-based models were proposed and tested for the LANEP. These two models, denoted by *FA* and *FD*, are characterized respectively by aggregated and disaggregated flows that circulate in the links and are augmented versions of models originally developed for the CMSTP. In this paper we review the *FA* model and present a new flow-based model that adds information on the first node of each feasible path in the graph (that corresponds to the central office or to a concentrator). As we shall show, this additional information permits us to create a model with a linear programming relaxation that is substantially tighter than the linear programming relaxation of the *FA* model. Furthermore, the new information on the variables permits us to use effectively variable elimination tests as well as coefficient reduction on some of the constraints of the new model. These features lead to more solution methods for the problem which are, in general, more efficient than the ones based on the previous *FA* and *FD* models.

* Corresponding author. Tel.: +351 226 196 200.
E-mail addresses: mcortereal@porto.ucp.pt, margarida.cortereal@crb.ucp.pt (M. Corte-Real), legouveia@fc.ul.pt (L. Gouveia).

0377-2217/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2009.10.001

The paper is organized as follows. Section 2 defines the problem and presents the assumptions that are usually given for the problem. In Section 3 we review how the LANEP can be seen as an extension of the CMSTP and review the aggregated flow model *FA* as well as some valid inequalities presented in Corte-Real and Gouveia (2007). We also introduce new inequalities that permit us to improve the linear programming relaxation of the *FA* model. In Section 3.3 we make a brief reference to the disaggregated flow model *FD*. In Section 4 we present the new model, the Node Rooted Aggregated Flow model denoted by *NRFA*, and develop several sets of valid inequalities that are based on the definition of new variables and that improve considerably the linear programming relaxation of the original model. We analyse the inequalities introduced for the *FA* model and show that several of them are redundant in the linear programming relaxation of the new model. Section 5 gives some computational results to evaluate the efficiency of the proposed models: in Section 5.1 some pre-processing techniques for variables elimination and coefficient reduction are described and Section 5.2 presents several computational results to compare the three classes of models from a set of tests with 100, 200 and 500 nodes. The last section presents some conclusions.

2. Problem description

A local access network is composed by a set of customer nodes, a central office and several links that allow the transmission of the traffic from one customer to another through the central office. The network has, in general, a fixed tree structure (neither nodes nor edges can be added) where the central office is the root of the tree and all the other nodes represent the customer nodes. Concentrators that compress the traffic can be installed in the customer nodes. The edges of the network correspond to the existent links between the customer nodes. In the local access network expansion problem, due to traffic demand increase, we need to expand the given network either through the installation of concentrators at the nodes or through the expansion of the capacity on the links. To each link we associate an available capacity that identifies the amount of information that can circulate initially on the link, a maximum capacity that corresponds to the maximum amount of traffic that will be able to circulate on the link after expanding the network and the costs of its expansion. To each node, we associate its demand and the costs of locating a concentrator.

Let $T = (N, E)$ be the tree defining the local access network, let $N = \{1, \ldots, n\}$ be the corresponding node set (node 1 is the root of the tree and represents the central office node) and let E be the edge set of the tree (corresponding to the existing links between customer nodes). For each node j, let d_j be the corresponding demand and let F_j and c_j be, respectively, the fixed and variable costs of installing a concentrator at that node. For each edge (i,j), we associate the available capacity B_{ij} and the maximum capacity M_{ij} which corresponds to an upper bound on the traffic that will be able to circulate on the edge after expanding the network (this value can be equal to the total demand of the tree). We also define a fixed cost G_{ij} and a variable cost e_{ij}, corresponding to the expansion of the link (i,j).

Given the demand in each node, the capacities and costs related to each edge and the concentrators costs, the local access network expansion problem is to determine, with minimum cost, which edges need to be expanded and/or where the concentrators should be located in order to guarantee that the node demands can be sent to (or from) the root node. We consider the following assumptions, already stated in, Balakrishnan et al. (1995), Cho and Shaw (1996, 1998), Flippo et al. (2000) and Corte-Real and Gouveia (2007):

- The information is concentrated once, either by one concentrator or by the central office (the central office can be viewed as a concentrator with zero costs);
- The demand of each node is unsplitted, that is, all the demand is processed by only a single concentrator or by the central office;
- If the demand of a node i is processed by a given concentrator, the demand of every node in the path from node i to the concentrator is also processed by the same concentrator (called contiguity restriction);
- The demand processed by a given concentrator is transmitted to the central office through the original links of the network (and the capacity used is considered insignificant) or by a direct link to the central node (in this case, the corresponding transmission cost is incorporated in the fixed cost of the concentrator).

The problem has been extensively studied by Balakrishnan et al. (1995), which state that the problem is NP-hard. They present a method that combines lagrangian relaxation with a dynamic programming algorithm in order to generate upper and lower bounds on the optimal value of the problem. Later on, Flippo et al. (2000) presented a pseudo-polynomial dynamic programming algorithm for the problem which depends on a constant value B that represents an upper bound on the capacity of the concentrators. Their algorithm solves the problem in $O(nB^2)$ time and $O(nB)$ storage space (where n refers to the size of the network) and considers more general cost structures for links expansion and concentrators installation. For the case with uncapacitated concentrators that we are considering, the parameter B is set equal to the total demand (as the authors of that paper suggest). The computational experience shown in Corte-Real and Gouveia (2007) gives empirical evidence that the flow-based models should be, in general, preferred when compared with the dynamic programming approach for the uncapacitated version of the problem.

3. Graph transformation and flow-based models

3.1. Graph transformation and the FA model (Corte-Real and Gouveia, 2007)

In this section, we review the approach described in Corte-Real and Gouveia (2007). With the inclusion of an additional node 0 and edges $(0,j)$ for each node $j \in N$ (denoted by *auxiliary edges*), we transform the problem under study into an extension of the CMSTP. Given the tree T, node and edge sets N and E as presented in the previous section, we define a new graph T_0, obtained from T by adding the new node and the new edges, as described in Fig. 1. Let N_0 denote the node set in T_0 and let E_0 denote the corresponding edge set.

When an edge $(0,j)$ is included in the solution in T_0, it means that a concentrator is located at node j. The total demand served by that concentrator in T is represented by the flow in the edge $(0,j)$ in T_0. Since the inclusion of each one of these edges in T_0 corresponds to the installation of one concentrator in T, the corresponding available edge capacity B_{0j} must be unlimited and these edges are not included in the set of edges to be expanded. For $j \in N$, we also define the coefficient M_{0j} as an upper bound on the traffic that can be served by the concentrator located at j. Two costs are associated with the new edges $(0,j)$: one corresponding to the fixed cost of installing one concentrator

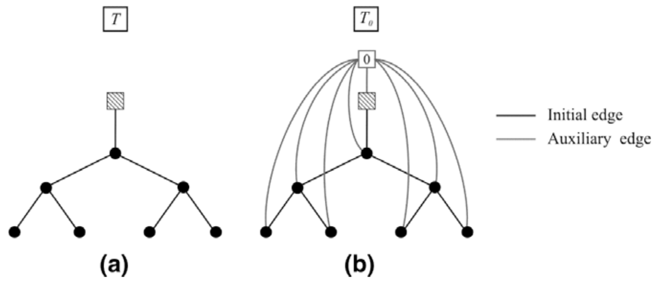


Fig. 1. Structure of: (a) T and (b) T_0.

at j, F_j, and the other corresponding to the variable cost c_j, which depends on the amount of flow that circulates through the edge. Fig. 2 illustrates a feasible solution in T and the corresponding feasible solution in T_0. As we can see, this solution is a spanning tree in the new graph.

The representation of the problem given by the new graph permits us to model the problem as a capacitated network flow model in T_0. Since a powerful modelling construct to improve formulations for several network design problems is to "direct the given network" (see, for instance, Goemans and Myung (1993) and Magnanti and Wolsey (1995)), we model our problem in a directed graph $D_0 = (N_0, A_0)$, where N_0 denotes the set of nodes and A_0 the set of all arcs, as described next: each edge (i,j) in E_0 is replaced by two arcs, $\langle i,j\rangle$ and $\langle j,i\rangle$, with the same parameters as the original edge (the exception are edges of the form $(1,j)$ and $(0,j)$ that are replaced only by one single arc, $\langle 1,j\rangle$ and $\langle 0,j\rangle$, respectively, since we consider that the traffic flows from the central office/concentrators to the nodes). The arcs of the form $\langle 0,j\rangle$, for each node $j \in N$, are designated as *auxiliary arcs*. The set A corresponds to the set of arcs without the auxiliary arcs.

Based on this representation, Corte-Real and Gouveia (2007) has described an adaptation for the new problem of a single-commodity model originally proposed for the CMSTP. We define, for each arc $\langle i,j\rangle \in A_0$, the binary variable x_{ij} indicating whether or not the arc $\langle i,j\rangle$ is included in the solution and a nonnegative variable y_{ij}, indicating the amount of flow that circulates in the arc. For each arc $\langle i,j\rangle \in A$, let z_{ij} denote the binary variable indicating whether or not the arc $\langle i,j\rangle$ is expanded and s_{ij} a nonnegative variable indicating the added capacity to the arc. The basic directed single-commodity flow formulation, denoted by FA_0, is as follows:

$$\text{Minimize} \quad \sum_{j=1}^{n} F_j x_{0j} + \sum_{j=1}^{n} c_j y_{0j} + \sum_{\langle i,j\rangle \in A} G_{ij} z_{ij} + \sum_{\langle i,j\rangle \in A} e_{ij} s_{ij}$$

$$\text{subject to} \quad \sum_{i:\langle i,j\rangle \in A_0} x_{ij} = 1 \quad \forall j \in N, \tag{1}$$

$$\sum_{i:\langle i,j\rangle \in A_0} y_{ij} - \sum_{i:\langle j,i\rangle \in A} y_{ji} = d_j \quad \forall j \in N, \tag{2}$$

$$z_{ij} \leqslant x_{ij} \quad \forall \langle i,j\rangle \in A, \tag{3}$$

$$y_{ij} \leqslant B_{ij} x_{ij} + s_{ij} \quad \forall \langle i,j\rangle \in A, \tag{4}$$

$$y_{0j} \leqslant M_{0j} x_{0j} \quad \forall j \in N, \tag{5}$$

$$s_{ij} \leqslant (M_{ij} - B_{ij}) z_{ij} \quad \forall \langle i,j\rangle \in A, \tag{6}$$

$$x_{ij} \in \{0,1\}, y_{ij} \geqslant 0 \quad \forall \langle i,j\rangle \in A_0; \quad z_{ij} \in \{0,1\}, \; s_{ij} \geqslant 0 \quad \forall \langle i,j\rangle \in A. \tag{7}$$

Constraints (1) ensure that each node, except node 0, has only one incident arc and constraints (2) are the flow conservation constraints, guaranteeing that each node j receives d_j units. Constraints (3) ensure that an arc in A is expanded only if it is used. For each one of these arcs, constraints (4) guarantee that the flow in any arc is less or equal than the available capacity plus the added capacity. For the auxiliary arcs $\langle 0,j\rangle$, constraints (5) state that the flow value is less or equal than the maximum capacity M_{0j}. Constraints (6) define an upper bound on the added capacity to the expanded links (the maximum demand that can circulate in the arc minus the corresponding available capacity). The last set represents the variables domain. The objective function states that we want to minimize the sum of the concentrator costs together with the link expansion costs. Note that constraints (4) and (6) together with (3) guarantee that

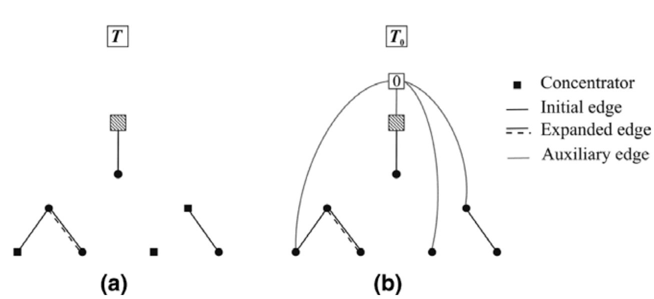


Fig. 2. Feasible solution in: (a) T and (b) T_0.

the constraints $y_{ij} \leqslant M_{ij}x_{ij} \quad \forall \langle i,j \rangle \in A$ are verified and that is why they were not explicitly included. These constraints together with (5), (1), (2), and the nonnegativity constraints (7), guarantee that the set of arcs defined by the x variables equal to 1 define a spanning tree in the graph D_0.

In order to improve the optimal solution of the linear programming relaxation of FA_0, several sets of valid inequalities were described in Corte-Real and Gouveia (2007). As these inequalities are needed for comparing with some of the inequalities for the new *NRFA* model, we make a brief review of them.

As noted in Corte-Real and Gouveia (2007), subtour elimination constraints are satisfied by the solutions of the problem. In particular, the linear programming relaxation of the model was substantially improved by adding the simple 2-subtour elimination constraints:

$$x_{ij} + x_{ji} \leqslant 1 \quad \forall (i,j) \in E, \; i,j \neq 1. \tag{8}$$

Another interesting class of inequalities is based on the concept of Saturated Nodes (originally introduced in Balakrishnan et al. (1995)). We recall that a *node j is saturated* if the total demand in the subtree rooted at node j, which is denoted by $D(j)$, is greater than the available capacity in the arc that connects the predecessor of j, a_j, to j. Let N_s be the set of saturated nodes and let $T(j)$ be the subtree rooted at node j. Two sets of constraints, based on the concept of saturated nodes, are as follows:

$$\sum_{k \in T(j)} x_{0k} + z_{a_j j} \geqslant 1 \quad \forall j \in N_s, \tag{9}$$

$$(D(j) - B_{a_j j}) \left(\sum_{k \in T(j)} x_{0k} \right) + s_{a_j j} \geqslant (D(j) - B_{a_j j}) \quad \forall j \in N_s. \tag{10}$$

For each saturated node j, constraints (9) guarantee that either we install at least one concentrator in the subtree rooted at j or we expand the arc that converges into node j; constraints (10) ensure that, if no concentrator is installed in that subtree, the expanded capacity must be greater or equal than the difference of the total demand $D(j)$ and the available capacity of the arc.

3.2. New inequalities – inequalities that give lower and upper bounds on the value of the arcs

In this paper, we introduce new inequalities that are lower and upper bounding inequalities on the value of the arc flows and that generalize, in some sense, simple lower and upper bounding flow inequalities.

Since the value of the flow on arc $\langle i,j \rangle$ must satisfy at least the demand of that node, we have the following well-known lower bounding inequalities, already used in Corte-Real and Gouveia (2007):

$$y_{ij} \geqslant d_j x_{ij} \quad \forall \langle i,j \rangle \in A_0. \tag{11}$$

Similar and more general inequalities, originally described in Gouveia and Lopes (2000), that consider one or all the arcs diverging from node j are also valid to the problem and were used in Corte-Real and Gouveia (2007). Here, we present stronger versions of these inequalities:

$$y_{ij} \geqslant (d_j + d_k)x_{ij} + d_k(x_{jk} + x_{kj} - 1) \quad \forall \langle i,j \rangle \in A_0, \quad \langle j,k \rangle \in A \quad (k \neq i), \tag{12}$$

$$y_{ij} \geqslant \left(d_j + \sum_{k:\langle j,k \rangle \in A; k \neq i} d_k \right) x_{ij} + \sum_{k:\langle j,k \rangle \in A; k \neq i} d_k(x_{jk} + x_{kj} - 1) \quad \forall \langle i,j \rangle \in A_0. \tag{13}$$

These inequalities differ from the versions presented in Corte-Real and Gouveia (2007) and Gouveia and Lopes (2000) due to the extra x_{kj} term involved in the right hand-side summation terms.

Result 3.1. Inequalities (12) and (13) are valid for the LANEP.

Proof. For simplicity we only proof the validity of (12) since the proof of (13) is similar. Since (12) is known to be valid when $x_{kj} = 0$, we only need to address the case with $x_{kj} = 1$. But then, by (1), $x_{ij} = 0$ and $y_{ij} = 0$ by constraints $y_{ij} \leqslant M_{ij}x_{ij} \quad \forall \langle i,j \rangle \in A_0$. The inequality becomes $x_{jk} + x_{kj} \leqslant 1$ which is valid. □

It can be easily proved that the inclusion of (13) does not imply that constraints (11) and (12) become redundant in the linear programming relaxation of FA_0 or that the inclusion of (12) implies the redundancy of (11). Thus, using both sets together with (11) might lead to better linear programming bounds and this is confirmed by our results.

We introduce, next, new inequalities that are upper bounding analogues of the previous inequalities. In order to introduce these general upper bounding inequalities, we define more precisely the M_{ij} coefficients that appear in constraints (6). Since, for each arc $\langle i,j \rangle$ the corresponding coefficient represents the maximum capacity that can circulate in arc $\langle i,j \rangle$, its value can be made equal to the total demand of the nodes that can be served through the arc (see Corte-Real and Gouveia (2007)). If some nodes are disconnected from the arc by the elimination of some arcs from the graph, this coefficient value can be further reduced.

In a similar way as was previously developed for the lower bounding inequalities, we develop upper bounding inequalities related to the flow in each arc $\langle i,j \rangle$. Consider the following inequalities:

$$y_{ij} \leqslant (M_{ij} - M_{jk})x_{ij} + M_{jk}x_{jk} \quad \forall \langle i,j \rangle \in A_0, \quad \langle j,k \rangle \in A (k \neq i). \tag{14}$$

As done previously with (12) to obtain (13), we can adapt constraints (14) in order to include all the diverging arcs from node j leading to

$$y_{ij} \leqslant d_j x_{ij} + \sum_{k:\langle j,k \rangle \in A; k \neq i} M_{jk}x_{jk} \quad \forall \langle i,j \rangle \in A_0. \tag{15}$$

Result 3.2. Inequalities (14) and (15) are valid for the LANEP.

Proof. Again, we only provide the proof of the validity of (14) since the proof of (15) is similar. Consider the following three cases: (i) if $x_{ij} = x_{jk} = 1$, we get $y_{ij} \leqslant M_{ij}$, which is valid as referred before (part (a) of Fig. 3); (ii) if $x_{ij} = 1$ and $x_{jk} = 0$, $y_{ij} \leqslant M_{ij} - M_{jk}$, which is valid because the M_{jk} coefficient includes the demand of all nodes of the "subtree rooted in k" and these nodes cannot be served through arc $\langle i,j\rangle$; $M_{ij} - M_{jk}$ is then an upper bound on the flow value that circulates in $\langle i,j\rangle$ (part (b) of Fig. 3); (iii) if $x_{ij} = 0$, we get $y_{ij} = 0$ from $y_{ij} \leqslant M_{ij}x_{ij} \quad \forall \langle i,j\rangle \in A_0$, and thus $x_{jk} \geqslant 0$, which is valid from the nonnegativity constraints (7). □

Similarly to what was stated before for the lower bounding inequalities, the inclusion of (15) in FA_0 formulation does not make (14) redundant in the linear programming relaxation of FA_0.

Finally, we can create constraints that are similar to the previously given constraints $s_{ij} \leqslant (M_{ij} - B_{ij})z_{ij} \quad \forall \langle i,j\rangle \in A$ (6):

$$s_{ij} \leqslant (M_{ij} - M_{jk} - B_{ij})z_{ij} + M_{jk}x_{jk} \quad \forall \langle i,j\rangle, \langle j,k\rangle \in A \quad (k \neq i), \tag{16}$$

$$s_{ij} \leqslant (d_j - B_{ij})z_{ij} + \sum_{k:\langle k,j\rangle \in A; k \neq i} M_{jk}x_{jk} \quad \forall \langle i,j\rangle \in A, \tag{17}$$

which consider, respectively, one or all the arcs diverging from node j.

Result 3.3. Inequalities (16) and (17) are valid for the LANEP.

Proof. Similarly to previous results, we only prove the validity of (16). Consider the following three cases: (i) if $z_{ij} = 1$ and $x_{jk} = 1$, the inequality becomes $s_{ij} \leqslant M_{ij} - B_{ij}$, which is valid due to (6); (ii) if $z_{ij} = 1$ and $x_{jk} = 0$, we get $s_{ij} \leqslant M_{ij} - M_{jk} - B_{ij}$; since the arc $\langle j,k\rangle$ is not in the solution, the maximum value of the flow that can circulate in $\langle i,j\rangle$ is given by $M_{ij} - M_{jk}$ and the added capacity to the arc must be less or equal than this maximum flow value minus the available capacity; (iii) if $z_{ij} = 0$, we get $s_{ij} = 0$ from (6) and (7) and $M_{jk}x_{jk}$ is always nonnegative. □

The following result shows that inequalities (16) and (17) imply constraints (14) and (15), respectively. Thus, only the inequalities (14) and (15) corresponding to auxiliary arcs are considered for augmenting the FA_0 formulation.

Result 3.4. (i) For each arc $\langle i,j\rangle \in A$ and each arc $\langle j,k\rangle \in A$ (with $k \neq i$), inequality (16) implies inequality (14) and (ii) for each arc $\langle i,j\rangle \in A$, inequality (17) implies inequality (15).

Proof. (i) Let $\langle i,j\rangle$ and $\langle j,k\rangle$ be two arcs in A (with $k \neq i$). By combining (4) for the arc $\langle i,j\rangle$ with (16), for $\langle i,j\rangle$ and $\langle j,k\rangle$, we obtain $y_{ij} \leqslant B_{ij}x_{ij} + (M_{ij} - M_{jk} - B_{ij})z_{ij} + M_{jk}x_{jk}$. By combining this inequality with (3) for the same $\langle i,j\rangle$, we obtain $y_{ij} \leqslant (M_{ij} - M_{jk})x_{ij} + M_{jk}x_{jk}$ as desired. The proof of (ii) is similar. □

To illustrate that the inclusion of several of these valid inequalities can improve the linear programming relaxation of FA_0, we introduce the following example where a local access network with 10 nodes is considered.

Example 3.1. Consider the local access network with 10 nodes denoted by *tree 10* and shown in Fig. 4, whose structure is represented in (a) and the associated parameters in (b).

Table 1 depicts the linear programming bound obtained with FA_0 formulation and the same formulation augmented with the different sets of the inequalities presented before. Several combinations were performed but, for simplicity, we have chosen the eight formulations presented in the table and easily identified by the designation at the top of each column. Table 1 includes two more columns, the first one characterizes the instance considered and the last one gives the optimal cost. We have also included the results taken from the instances *tree 10_Fx2* and *tree 10_Bx2*, obtained from *tree 10* by, respectively, doubling the fixed cost of installing the concentrators and doubling the available capacity of the links.

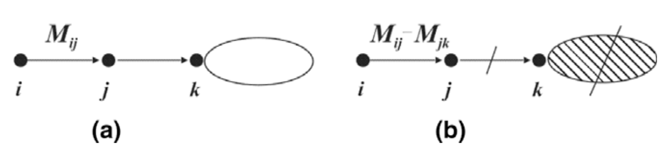
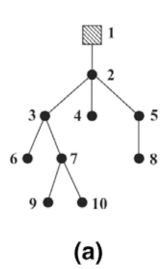


Fig. 3. Upper bound on the flow value in arc $\langle i,j\rangle$: (a) M_{ij}, if $\langle j,k\rangle$ is in the solution (b) $M_{ij} - M_{jk}$, otherwise.



Nodes	2	3	4	5	6	7	8	9	10
d_j	7	9	15	9	8	5	9	6	8
F_j	200	200	20	20	200	200	200	200	200
c_j	20	20	20	20	20	20	20	20	20

Edges	(1,2)	(2,3)	(2,4)	(2,5)	(3,6)	(3,7)	(5,8)	(7,9)	(7,10)
B_{ij}	5	10	10	5	5	10	2	5	5
G_{ij}	50	50	50	50	50	50	50	50	50
e_{ij}	20	20	20	20	20	20	20	20	20

(b)

Fig. 4. Instance *tree10*: (a) network configuration and (b) parameters.

Table 1
Linear programming bound of FA_0 formulation with the inclusion of different sets of valid inequalities.

Instance	FA_0	FA_0 + (8) + (9) + (10)	FA_0 + (11)	FA_0 + (11) + (12)	FA_0 + (11) + (12) + (13)	FA_0 + (11) + (12) + (13) + (14) + (15)	FA_0 + (11) + (12) + (13) + (14) + (15) + (16) + (17)	FA_1	Opt.
tree10	1522.6	2041.1	2012.4	2130.7	2161.7	2169.2	2170.5	2170.5	2280
tree10_Fx2	1616.7	2303.7	2123.6	2301.1	2504.9	2520.5	2521.9	2527.1	2620
tree10_Bx2	1396.2	1693.2	1544.9	1562.2	1597.5	1658.3	1665.6	1704.6	1740

The results on the Table 1 show that the several sets of inequalities previously discussed improve the cost of the linear programming solution presented in column "FA_0". The second formulation represents the FA_0 formulation augmented with the 2-subtour elimination constraints and the constraints based on the saturated nodes. The next three show the effect of the inclusion of the lower bounding inequalities: the simple ones (11) and the more general inequalities (12) and (13). In the next two columns we include the lower and upper bounding inequalities (in the last case, the upper bounding inequalities (14) and (15) are included only for the auxiliary arcs). The last column "FA_1" gives the FA_0 formulation augmented with all the inequalities presented.

3.3. Disaggregated flow model – FD (Corte-Real and Gouveia, 2007)

We now make a brief reference to the *FD* model, the multi-commodity version of the *FA* model presented in Corte-Real and Gouveia (2007). As it is well-known, the linear programming relaxation of a single-commodity formulation can be strengthened by reformulating as a so-called multi-commodity formulation where the flow variables are indexed by source and/or destination (see, for instance, Magnanti and Wong (1984) and Rardin and Choe (1979)). The disaggregated flow model *FD* uses flow variables f_{ij}^p indexed by destination, for each arc $\langle i,j\rangle$ and node p, that specify the amount of flow in $\langle i,j\rangle$ sent from node 0 to p. We denote by FD_0 the initial formulation, adapted directly from FA_0 with the new flow variables and the inclusion of the inequalities $f_{ij}^p \leqslant d_p x_{ij}\ \forall\langle i,j\rangle \in A_0, p \in N$, that characterizes the multi-commodity models (the reader is referred to Corte-Real and Gouveia (2007) for a detailed description of the *FD* model). The FD_1 formulation is the formulation FD_0 augmented with similar inequalities to the ones presented before, non-redundant however to the linear programming relaxation of FD_0. In Section 5 we include the results obtained with these two formulations.

4. Node rooted aggregated flow model – NRFA

4.1. The model

The idea of the new model is to add information about the first node on each feasible path leaving the root. Recall that, for any two nodes p and $j (p,j \neq 0)$, if the demand of node j is processed by the concentrator located at node p in T (Fig. 5a), the demand of every node in the path from the concentrator to node j is also processed by the same concentrator (by the contiguity restriction) and the path from p to j must be in the solution in D_0 (Fig. 5b).

The information about the first node on each feasible path permits us to define new variables and a new coefficient M_{ij}^p that corresponds to an upper bound on the required capacity for arc $\langle i,j\rangle$, when the directed path rooted at node p includes arc $\langle i,j\rangle$. For some values of p this bound may be smaller than the one given by M_{ij} and used in the previous model, leading to an improved linear programming relaxation. Recall that the upper bound M_{ij} on the capacity required on arc $\langle i,j\rangle$ is defined by the sum of the demands of the nodes that can be served through the arc. In this case, the concentrator that sends this traffic is not known in advance and the M_{ij} value is the sum of the node demands reachable from arc $\langle i,j\rangle$ (Fig. 6a). To define the new coefficients, let $p \in N$ and $\langle i,j\rangle \in A$ such that the directed path from p to j includes arc $\langle i,j\rangle$. The value of the maximum flow that can circulate on arc $\langle i,j\rangle$ sent by node p is equal to the sum of the demand of the nodes reachable from arc $\langle i,j\rangle$ that can be served by the concentrator located at p. This value will be denoted by M_{ij}^p. If, by the elimination of some arcs, some paths or some node to concentrator assignments, we can show that node p cannot serve some of those nodes in any optimal solution, then the M_{ij}^p value for some indexes p may become less than the M_{ij} coefficient (Fig. 6b). For the auxiliary arc $\langle 0,j\rangle$, we define as before the coefficient M_{0j} as the total demand that can be served by the concentrator located at j.

The model uses binary variables x_{ij}^p indicating whether or not arc $\langle i,j\rangle \in A$ is included in a directed path rooted at node p (if $x_{ij}^p = 1$, nodes i and j are assigned to the concentrator located at node p). These variables are defined for each pair $(p,\langle i,j\rangle)$ only if the directed path from p

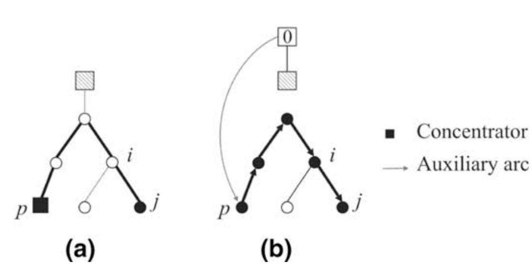


Fig. 5. (a) Concentrator at node p process node j demand in T (b) $\langle 0,p\rangle$ is in the solution in D_0 and node p process node j demand and all the demand of the nodes from the path from p to j.

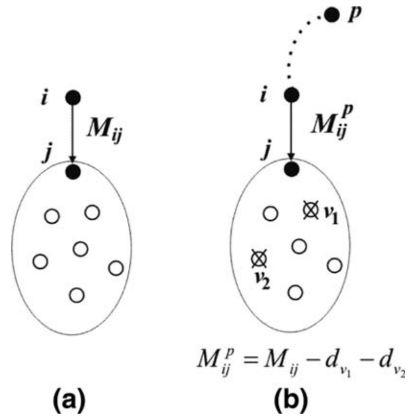


Fig. 6. (a) M_{ij} coefficient (b) M_{ij}^p coefficient, assuming that p cannot serve nodes v_1 and v_2 demand.

to j traverses arc $\langle i,j\rangle$. In a similar way, we define nonnegative flow variables y_{ij}^p indicating the amount of flow sent from the concentrator at node p that circulates in arc $\langle i,j\rangle$. For the auxiliary arcs, we define the binary variables x_{0j} indicating whether or not a concentrator is located at node j and the flow variables y_{0j} indicating the total demand served by that concentrator. To model the expansion of the links, we define the binary variables z_{ij}^p indicating whether or not arc $\langle i,j\rangle$, belonging to the directed path rooted at p, is expanded and the nonnegative variables s_{ij}^p denoting the respective expanded amount of flow. Note that the new and the variables of the *FA* model are related as follows:

$$\sum_{p=1}^{n} x_{ij}^p = x_{ij}, \quad \sum_{p=1}^{n} y_{ij}^p = y_{ij}, \quad \sum_{p=1}^{n} z_{ij}^p = z_{ij} \text{ and } \sum_{p=1}^{n} s_{ij}^p = s_{ij}. \tag{18}$$

The summation terms in these expressions can be simplified by noting that the index "p" is not considered whenever i or j cannot be served by the concentrator located at node p.

The formulation, denoted by *NRFA*, involving the new variables is as follows:

$$\text{Minimize} \quad \sum_{j=1}^{n} F_j x_{0j} + \sum_{j=1}^{n} c_j y_{0j} + \sum_{\langle i,j\rangle \in A} \sum_{p=1}^{n} G_{ij} z_{ij}^p + \sum_{\langle i,j\rangle \in A} \sum_{p=1}^{n} e_{ij} s_{ij}^p$$

$$\text{subject to} \quad \sum_{i:\langle i,j\rangle \in A_0} \sum_{p=1}^{n} x_{ij}^p = 1 \quad \forall j \in N, \tag{19}$$

$$\sum_{i:\langle i,j\rangle \in A_0} y_{ij}^p - \sum_{i:\langle j,i\rangle \in A} y_{ji}^p = d_j, \quad \sum_{i:\langle i,j\rangle \in A_0} x_{ij}^p \ \forall j \in N \ \forall p \in N, \tag{20}$$

$$z_{ij}^p \leqslant x_{ij}^p \quad \forall \langle i,j\rangle \in A \quad \forall p \in N, \tag{21}$$

$$y_{ij}^p \leqslant B_{ij} x_{ij}^p + s_{ij}^p \quad \forall \langle i,j\rangle \in A \ \forall p \in N, \tag{22}$$

$$y_{0j} \leqslant M_{0j} x_{0j} \quad \forall j \in N, \tag{23}$$

$$s_{ij}^p \leqslant \left(M_{ij}^p - B_{ij}\right) z_{ij}^p \quad \forall \langle i,j\rangle \in A \ \forall p \in N, \tag{24}$$

$$x_{ij}^p \in \{0,1\}, \ y_{ij}^p \geqslant 0 \ \forall \langle i,j\rangle \in A_0, \ p \in N; \ z_{ij}^p \in \{0,1\}, \ s_{ij}^p \geqslant 0 \ \forall \langle i,j\rangle \in A, \ p \in N. \tag{25}$$

We omit the description of the objective function and constraints in the new model since it can be taken from the description of the constraints in the *FA* model and the constraints (18) linking the two sets of variables. Note the new coefficients M_{ij}^p appearing in constraints (24). Recall that these coefficients may become smaller than the corresponding coefficient in the *FA* model when it is known that node p (the corresponding concentrator) will not serve some nodes in the tree. Although the new model includes many more variables and constraints than the *FA* model, the proposed pre-processing permits us to eliminate several directed paths and consequently eliminate many variables and constraints as well as reduce the value of the M_{ij}^p coefficient (in fact, it was the pre-processing leading to a reduction in the M_{ij}^p coefficients that has motivated this model).

Result 4.1. The linear programming relaxation value of the *NRFA* formulation is greater or equal than the linear programming relaxation bound of the FA_0 formulation.

Proof. The result follows simply from the fact that by using (18), the constraints of the *NRFA* model imply the constraints of the *FA* model (note that $M_{ij}^p \leqslant M_{ij}, \ \forall p \in N$ and all arcs $\langle i,j\rangle$). The objective functions are the same after using (18). □

The computational results show that in general, the *NRFA* model already produces better lower bounds. However, the linear programming relaxation of the new model can be substantially improved by using the information attached to the new variables in order to derive new sets of inequalities:

$$x_{pk}^p \leqslant x_{0p} \quad \forall \langle p,k\rangle \in A, \tag{26}$$

$$x_{ij}^p \leqslant x_{p_i i}^p \quad \forall \langle i,j\rangle \in A \quad \forall p \in N \setminus \{i\}, \tag{27}$$

where node p_i is the predecessor of node i in the directed path from p to i. The constraints in the first set guarantee that if the arc $\langle p,k\rangle$ is served by node p (it means that the nodes associated to the arc are served by node p) a concentrator must be located at p and the constraints in the second set state, explicitly, the contiguity restriction for the arc $\langle i,j\rangle$ and the arc $\langle p_i,i\rangle$. The proof of the following result follows immediately from these arguments.

Result 4.2. Inequalities (26) and (27) are valid for the LANEP.

Note that (26) and (27) guarantee that $x_{ij}^p \leqslant x_{0p}, \forall\langle i,j\rangle \in A, \forall p \in N$, which, in turn, guarantee that a concentrator will be located at node p if this node serves arc $\langle i,j\rangle$. The valid inequalities (26) and (27) are included in the *NRFA* formulation in order to improve its linear programming relaxation. We denote by $NRFA_0$ the formulation obtained in this way.

4.2. Valid inequalities "adapted" from the FA model

In this subsection, we describe another interesting feature of the new formulation. Namely, we introduce valid inequalities that are similar to the ones introduced for the *FA* model and show that several of them are redundant in the linear programming relaxation of the $NRFA_0$ model.

4.2.1. 2-subtour elimination constraints

Using the relation between the new and old variables, the 2-subtour elimination constraints (8) can be rewritten as

$$\sum_{p=1}^{n} x_{ij}^p + \sum_{p=1}^{n} x_{ji}^p \leqslant 1 \quad \forall (i,j) \in E,\ i,j \neq 1. \tag{28}$$

Result 4.3. Inequalities (28) are redundant to the linear programming relaxation of $NRFA_0$.

Proof. Let $(i,j) \in E$ with $i,j \neq 1$. Using constraints (19) for node j, the left-hand side of (28) can be rewritten as $\sum_{p=1}^{n} x_{ij}^p + \sum_{p=1}^{n} x_{ji}^p = 1 - x_{0j} - \sum_{k:\langle k,j\rangle\in A;k\neq i}\sum_{\substack{p=1\\ p:k\in P_{pj}}}^{n} x_{kj}^p + x_{ji}^j + \sum_{\substack{p=1\\ p:j\in P_{pi};p\neq j}}^{n} x_{ji}^p$, where P_{pv} denotes the directed path from node p to node v.

Now, if we add constraints (27) for $\langle j,i\rangle$ and all $p \neq j$ such that j is in the path from p to i, we obtain $\sum_{\substack{p=1\\ p:j\in P_{pi};p\neq j}}^{n} x_{ji}^p \leqslant \sum_{\substack{p=1\\ p:j\in P_{pi}}}^{n} x_{p_j j}^p = \sum_{k:\langle k,j\rangle\in A;k\neq i}\sum_{\substack{p=1\\ p:k\in P_{pj}}}^{n} x_{kj}^p$, and constraint (26) implies $x_{ji}^j \leqslant x_{0j}$. Combining these two inequalities with the previous expression gives the desired inequality. $\square$

4.2.2. Inequalities based on saturated nodes

The inequalities based on saturated nodes can be simply rewritten for the new model by using the relation between the new and old variables (for this case, we were unable to write disaggregated constraints as we did for many other inequalities): thus, for each $j \in N_s$, we consider

$$\sum_{k\in T(j)} x_{0k} + \sum_{p=1}^{n} z_{a_j j}^p \geqslant 1, \tag{29}$$

$$(D(j) - B_{a_j j})\left(\sum_{k\in T(j)} x_{0k}\right) + \sum_{p=1}^{n} s_{a_j j}^p \geqslant \quad D(j) - B_{a_j j}. \tag{30}$$

4.2.3. Lower and upper bounding inequalities

We now analyse the lower bounding inequalities introduced before. These inequalities can be adapted to the new model by using the relation between the new and old variables or by considering their disaggregated versions. We consider the disaggregated versions to each of the inequalities presented before. We shall prove that several of the adapted inequalities are redundant to the linear programming relaxation of $NRFA_0$.

Before presenting these inequalities and the redundancy of the results we make the following important observation.

Observation: Let $p,j \in N$: (i) If $p \neq j$, then $\sum_{k:\langle k,j\rangle\in A_0} x_{kj}^p = x_{ij}^p$ and $\sum_{k:\langle k,j\rangle\in A_0} y_{kj}^p = y_{ij}^p$, where $\langle i,j\rangle$ denotes the arc $\langle p_j,j\rangle$; (ii) If $p = j$, then $\sum_{k:\langle k,j\rangle\in A_0} x_{kj}^p = x_{0j}$ and $\sum_{k:\langle k,j\rangle\in A_0} y_{kj}^p = y_{0j}$ (note that in (i) we are considering the directed path from p to j which has only one incident arc in j and (ii) represent the situation where node j is served by a concentrator located at this node).

The disaggregated version of constraints (11)–(13) are, respectively, constraints:

$$y_{ij}^p \geqslant d_j x_{ij}^p \quad \forall\langle i,j\rangle \in A_0\ \forall p \in N, \tag{31}$$

$$y_{ij}^p \geqslant (d_j + d_k)x_{ij}^p + d_k(x_{jk}^p - 1) \quad \forall\langle i,j\rangle \in A_0,\ \langle j,k\rangle \in A(k \neq i)\ \forall p \in N, \tag{32}$$

$$y_{ij}^p \geqslant \left(d_j + \sum_{k:\langle j,k\rangle\in A;k\neq i} d_k\right)x_{ij}^p + \sum_{k:\langle j,k\rangle\in A;k\neq i} d_k(x_{jk}^p - 1) \quad \forall\langle i,j\rangle \in A_0\ \forall p \in N. \tag{33}$$

Note that we cannot include in (32) and (33) the x_{kj}^p term (as we did for the aggregated versions in *FA*) because this variable is not defined in the new model. Note also that as observed before, these constraints are defined only for specific triples (or m-tuples) of nodes.

Result 4.4. Inequalities (31)–(33) are redundant in the linear programming relaxation of $NRFA_0$.

Proof. We start by proving that constraints (31) are redundant to the linear programming relaxation of $NRFA_0$. Let $p \in N$ and $\langle i,j\rangle \in A_0$. Assume $p \neq j$. Using the observation, the flow conservation constraints (20) become $y_{ij}^p = d_j x_{ij}^p + \sum_{k:\langle j,k\rangle\in A;k\neq i} y_{jk}^p$. The nonnegativity of the flow variables gives the desired inequality $y_{ij}^p \geqslant d_j x_{ij}^p$. A similar argument is used when $p = j$, leading to $y_{0j} \geqslant d_j x_{0j}$.

With respect to (32), note also that, for $p \neq j$ and $k \neq i$, the expression $y_{ij}^p = d_j x_{ij}^p + \sum_{v:\langle j,v\rangle \in A; v \neq i} y_{jv}^p$ can be rewritten as $y_{ij}^p = d_j x_{ij}^p + y_{jk}^p + \sum_{v:\langle j,v\rangle \in A; v \neq i,k} y_{jv}^p$ and $y_{ij}^p \geqslant d_j x_{ij}^p + y_{jk}^p$. Using (31) in the previous expression (note that j is in the path from node p to node k) we obtain $y_{ij}^p \geqslant d_j x_{ij}^p + d_k x_{jk}^p$. Since $d_k\left(x_{ij}^p - 1\right) \leqslant 0$, the previous inequality implies $y_{ij}^p \geqslant d_j x_{ij}^p + d_k x_{jk}^p + d_k\left(x_{ij}^p - 1\right)$, which is the same as the desired inequality $y_{ij}^p \geqslant (d_j + d_k) x_{ij}^p + d_k\left(x_{jk}^p - 1\right)$. A similar analysis holds when $p = j$.

The proof of (33) is similar and we omit it from the paper. □

The disaggregated version of the constraints (14) and (15) are, respectively, the constraints:

$$y_{ij}^p \leqslant \left(M_{ij}^p - M_{jk}^p\right) x_{ij}^p + M_{jk}^p x_{jk}^p \quad \forall \langle i,j\rangle \in A_0,\ \langle j,k\rangle \in A(k \neq i)\ \forall p \in N, \tag{34}$$

$$y_{ij}^p \leqslant d_j x_{ij}^p + \sum_{k:\langle j,k\rangle \in A; k \neq i} M_{jk}^p x_{jk}^p \quad \forall \langle i,j\rangle \in A_0\ \forall p \in N. \tag{35}$$

Result 4.5. Inequalities (34) and (35) are redundant in the linear programming relaxation of $NRFA_0$.

Proof. Note, first, that constraints $y_{ij}^p \leqslant M_{ij}^p x_{ij}^p\ \forall \langle i,j\rangle \in A_0\ \forall\ p \in N$ (36) are guaranteed by constraints (22), (21) and (24), for $\langle i,j\rangle \in A$, and by (23) for the auxiliary arcs. We start by proving the redundancy of (35). Combining the flow conservation constraints $y_{ij}^p = d_j x_{ij}^p + \sum_{k:\langle j,k\rangle \in A; k \neq i} y_{jk}^p$ with constraints (36) for the arcs $\langle j,k\rangle \in A, k \neq i$, we get $y_{ij}^p \leqslant d_j x_{ij}^p + \sum_{k:\langle j,k\rangle \in A; k \neq i} M_{jk}^p x_{jk}^p$ which is constraint (35) for $\langle i,j\rangle \in A_0$ and $p \in N$.

With respect to constraints (34), let $p \in N, \langle i,j\rangle \in A_0$ and $\langle j,k\rangle \in A(k \neq i)$. If $p \neq j$, the constraints (35) and the fact that $M_{ij}^p = d_j + \sum_{v:\langle j,v\rangle \in A; v \neq i} M_{jv}^p$ permit us to rewrite constraint (35) for $\langle i,j\rangle \in A_0$ and $p \in N$ as follows $y_{ij}^p \leqslant \left(M_{ij}^p - M_{jk}^p - \sum_{v:\langle j,v\rangle \in A; v \neq k,i} M_{jv}^p\right) x_{ij}^p + M_{jk}^p x_{jk}^p + \sum_{v:\langle j,v\rangle \in A; v \neq k,i} M_{jv}^p x_{jv}^p$. By rearranging, we obtain $y_{ij}^p \leqslant \left(M_{ij}^p - M_{jk}^p\right) x_{ij}^p + M_{jk}^p x_{jk}^p + \sum_{v:\langle j,v\rangle \in A; v \neq k,i} M_{jv}^p \left(x_{jv}^p - x_{ij}^p\right)$. Constraint (27) states that $\sum_{v:\langle j,v\rangle \in A; v \neq k,i} M_{jv}^p \left(x_{jv}^p - x_{ij}^p\right) \leqslant 0$ leading to $y_{ij}^p \leqslant \left(M_{ij}^p - M_{jk}^p\right) x_{ij}^p + M_{jk}^p x_{jk}^p$ which is (34) for $\langle i,j\rangle \in A_0, \langle j,k\rangle \in A$ and $p \in N$. If $p = j$ the proof is similar. □

The following result follows from the previous results.

Result 4.6. Inequalities (31)–(35) rewritten with the variables of the *NRFA* model by using (18) are redundant in the linear programming relaxation of $NRFA_0$.

Finally, the disaggregated version of the constraints (16) and (17) are, respectively, the constraints:

$$s_{ij}^p \leqslant \left(M_{ij}^p - M_{jk}^p - B_{ij}\right) z_{ij}^p + M_{jk}^p x_{jk}^p \quad \forall \langle i,j\rangle,\ \langle j,k\rangle \in A(k \neq i)\ \forall p \in N, \tag{37}$$

$$s_{ij}^p \leqslant (d_j - B_{ij}) z_{ij}^p + \sum_{k:\langle j,k\rangle \in A; k \neq i} M_{jk}^p x_{jk}^p \quad \forall \langle i,j\rangle \in A\ \forall p \in N. \tag{38}$$

Our computational results will show that these constraints are not redundant in the linear programming relaxation of the $NRFA_0$ model. We denote by $NRFA_1$ the formulation $NRFA_0$ augmented with (29), (30), (37), and (38). We present next the linear programming bounds obtained with *NRFA* and $NRFA_0$ and the same formulation augmented with the several sets of inequalities in order to show that the inclusion of the different sets improve the cost of the linear programming solution to the instances presented in Example 3.1.

Example 4.1. Consider the local access network shown in Fig. 4 and the three instances used in Example 3.1. Table 2 depicts the linear programming bounds obtained with the *NRFA* formulation, the $NRFA_0$ formulation and this formulation augmented with different sets of the inequalities presented before, for instances *tree10*, *tree10_Fx2* and *tree10_Bx2*. The table description is omitted because is similar to Table 1.

The results show that the value of the linear programming relaxation of the *NRFA* formulation is quite improved by adding constraints (26) and (27), which are specially tailored to the new model.

5. Computational tests

5.1. Pre-processing

Pre-processing is an efficient way of reducing the size of the models. In some cases, pre-processing leads to an improvement on the value of the corresponding linear programming relaxation and to the reduction of the time used to obtain the solutions since eliminating a variable is equivalent to adding the inequality that sets to zero the value of the same variable. In this section, we show how to reduce some data coefficients and how to eliminate some variables from the models.

Table 2
Linear programming bound of *NRFA* formulation. $NRFA_0$ formulation and $NRFA_0$ formulation augmented with the inclusion of different sets of valid inequalities.

Instance	NRFA	$NRFA_0$	$NRFA_0$ + (29) + (30)	$NRFA_0$ + (37) + (38)	$NRFA_1$	Opt.
tree10	2012.4	2220.2	2220.2	2221.3	2221.3	2280
tree10_Fx2	2123.6	2528.9	2528.9	2573.5	2573.5	2620
tree10_Bx2	1544.9	1682.7	1699.1	1691.8	1705.2	1740

5.1.1. Coefficient reduction

Recall that the M_{ij} coefficient represents the maximum flow that can circulate in an arc $\langle i,j\rangle$ and that the M_{ij}^p coefficient represents a similar value when the arc is used in the directed path rooted at node p. As referred, the initial values of these coefficients are set equal to the total demand of the nodes reachable through arc $\langle i,j\rangle$. We present next several properties that permit us to define a better upper bound on the value of these coefficients.

By examining the costs of concentrators and links, it is possible to eliminate some node to concentrator assignments. Given $p,j \in N$, Balakrishnan et al. (1995) show how to determine a lower bound to the cost of assign node j to a concentrator located at node p, denoted by L_{pj}, and an upper bound to the cost of installing a concentrator at j, denoted by U_{jj} (see Balakrishnan et al. (1995) for details).

Property 5.1. Let $p,j \in N$. If $U_{jj} < L_{pj}$, node p will not serve node j in any optimal solution.

Based on the elimination of node to concentrator assignments, it is possible to reduce or set equal to zero the values of the M_{ij} and M_{ij}^p coefficients. The next two properties show how this can be done.

Property 5.2. Let $\langle i,j\rangle \in A$ and $p \in N$ such that the path P_{pj} includes arc $\langle i,j\rangle$: (i) The total demand of the nodes that can be served by a concentrator located at node p through arc $\langle i,j\rangle$ is an upper bound to the M_{ij}^p coefficient; (ii) The value $\max_{p\in N} M_{ij}^p$ is an upper bound to the M_{ij} coefficient.

A particular case of the previous property is as follows:

Property 5.3. Let $\langle i,j\rangle \in A$: (i) If any node p cannot serve node j through arc $\langle i,j\rangle$, then $M_{ij} = 0$ and $M_{ij}^p = 0$, for all $p \in N$; (ii) If some node p cannot serve node j through arc $\langle i,j\rangle$, then $M_{ij}^p = 0$.

Note that in the context of the variables of the *FA* models, we can only set to zero the M_{ij} coefficient if *any* node p cannot serve j through that arc.

Based on the fixed and variable costs of installing a concentrator at node j and of expanding the arc $\langle i,j\rangle$, Balakrishnan et al. (1995) determine an upper bound to the value M_{ij} by calculating the amount of flow from which it is cheaper to locate a concentrator at node j rather than expanding the arc $\langle i,j\rangle$. This value is easily adapted for the M_{ij}^p coefficients. After using this property, some of the values M_{ij} and M_{ij}^p may become smaller than the sum of the demands of the nodes reachable after arc $\langle i,j\rangle$ (or reachable after $\langle i,j\rangle$ and in paths served by a concentrator in p in the case of the M_{ij}^p coefficients). In this case, the following property permits us to calculate these values in a different manner.

Property 5.4. Let $\langle i,j\rangle \in A_0$ and $\langle j,k\rangle \in A$, for all $k \in N$ and $k \neq i$: (i) The value $d_j + \sum_{k:\langle j,k\rangle\in A;k\neq i} M_{jk}$ is an upper bound to M_{ij}; (ii) Let $p \in N$. The value $d_j + \sum_{k:\langle j,k\rangle\in A;k\neq i} M_{jk}^p$ is an upper bound to M_{ij}^p.

5.1.2. Variable elimination

We first show how coefficient reduction permit us to eliminate arcs and paths from the graph D_0.

Property 5.5. Let $j \in N$ and $\langle i,j\rangle \in A_0$: (i) If $M_{ij} < d_j$ or $M_{ij}^p < d_j$, for all $p \in N$, the arc $\langle i,j\rangle$ will not be used in any feasible solution; (ii) If $M_{ij}^p < d_j$, for some $p \in N$, the arc $\langle i,j\rangle$ will not be used in the directed path rooted at p in any feasible solution (note that (i) and (ii) result from constraints (11) and (31), respectively).

We can generalize this property for the directed paths in the graph by considering the total demand of a sequence of nodes located "after" node j. Recall that P_{jk} is the directed path from node j to node k.

Property 5.6. Let $\langle i,j\rangle \in A_0, k \in N (k \neq j)$ and let us assume that the path P_{ik} includes arc $\langle i,j\rangle$. Let D_{jk} be the total demand from the path P_{jk}: (i) If $M_{ij} < D_{jk}$ or $M_{ij}^p < D_{jk}$, for all $p \in N$, then neither the path P_{ik} nor any other path that includes P_{ik} can be included in a feasible solution; (ii) If $M_{ij}^p < D_{jk}$, for some $p \in N$, then neither the path P_{ik} rooted at p nor any path rooted at p that includes P_{ik} can be included in a feasible solution.

Note that if a path P_{jk} is eliminated from the graph D_0 than the assignment of node k to a concentrator located in j is also eliminated. Finally, the next two properties show how to use the previous properties to eliminate variables from the models.

Property 5.7. Let $\langle i,j\rangle \in A$: (i) If the arc $\langle i,j\rangle$ will not be used in any feasible solution, then the variables x_{ij}, y_{ij}, z_{ij} and s_{ij} can be eliminated from the *FA* models as well as the variables $x_{ij}^p, y_{ij}^p, z_{ij}^p$ and s_{ij}^p, for all $p \in N$, can be eliminated from the *NRFA* models; (ii) If for some $p \in N$ the arc $\langle i,j\rangle$ will not be used in the directed path rooted at p in any feasible solution, then the variables $x_{ij}^p, y_{ij}^p, z_{ij}^p$ and s_{ij}^p, for the same indexes p, can be eliminated from the *NRFA* models.

This property permit us to eliminate many variables from the models (this is more notorious in the *NRFA* model). The value of the M_{ij} and M_{ij}^p coefficients (before or after reduction) also permits the elimination of the variables related to the expansion of the links.

Property 5.8. Let $\langle i,j\rangle \in A$: (i) If $M_{ij} \leqslant B_{ij}$, then the arc $\langle i,j\rangle$ will not be expanded in any feasible solution and the variables z_{ij} and s_{ij} can be eliminated from the *FA* models as well as the variables z_{ij}^p and s_{ij}^p, for all $p \in N$, can be eliminated from the *NRFA* models; (ii) If $M_{ij}^p \leqslant B_{ij}$, for some $p \in N$, the variables z_{ij}^p and s_{ij}^p, for the same indexes p, can be eliminated from the *NRFA* models.

Note also that the elimination of arcs/paths can also lead to a further reduction of several M_{ij} and M_{ij}^p coefficients (if the corresponding nodes are no longer reachable from an arc $\langle i,j\rangle$, the maximum flow value in this arc can be reduced). Thus, the properties here suggested may and should be used more than once and iteratively until no further reduction or elimination is possible. The time used in the pre-processing is less than 1 s for instances with 100 and 200 nodes and, for instances with 500 nodes, the time is less or equal than to 3 s.

5.2. Computational results

We have compared three classes of flow-based formulations for the local access network expansion problem: *FA* and *FD* models (aggregated and disaggregated flow models, respectively) which were previously developed and the new *NRFA* model. Note, however, that the *FA* model was significantly enhanced with the inclusion of the general lower and upper bounding inequalities. In this section, we present some computational results to assess their efficiency in solving instances of the problem and, in particular, we want to emphasize the advantages of developing the new model.

In order to evaluate and compare the linear programming relaxations of our models, we used instances with 100, 200 and 500 nodes. Some of these instances were already considered in Corte-Real and Gouveia (2007). Here we have included as well the 500-node instances and different scenarios for the smaller instances. The whole set includes two 100-node tree topologies, two 200-node tree topologies and two 500-node tree topologies, with two types of tree structure, which differ on the number of sons for each node. In the first case, this number is less or equal than 3 (trees 101, 201 and 501) and, in the second, is less or equal than 10 (trees 102, 202 and 502). For each tree, we have generated different set of parameters in order to represent three alternatives to expand the network, denoted by A, B and C, where A represents the situation that favours concentrator installation, B favours link expansion and C represents a more balanced alternative. Also, the number of links that need to be expanded, decreases from A to C. For each one of these 18 cases, we also varied some parameters in order to create different scenarios, considering several values for the node demands, links available capacities and costs. Thus, for each tree topology and each alternative (A, B or C), we have considered 14 more additional cases: we have divided the demand of each node by 5 and 2 and we have multiplied it by 2 (instances denoted by P/5, P/2 and P $\times$ 2, respectively); we have divided the available capacity of each arc by 2 and we have multiplied it by 2 and 5 (instances denoted by B/2, B $\times$ 2 and B$\times$5, respectively); we have divided the variable cost of the links by 2 and multiplied by 2, 5 and 20 (instances denoted by e/2, e $\times$ 2, e $\times$ 5 and e $\times$ 20, respectively) and we have divided the fixed cost of concentrators by 2 and multiplied it by 2, 5 and 20 (instances denoted by F/2, F $\times$ 2, F $\times$ 5 and F $\times$ 20, respectively). The results presented in the following tables correspond to average values obtained with three instances, the three of them generated with the same characteristics. All the tests were run in an Intel Core 2 Duo, 2.00 GHz, personal computer with 2 GB of RAM, and the CPLEX package, version 10.2, has been used to obtain the linear programming bounds and the integer optimal solutions.

Before presenting and comparing in detail the results for the three classes of models, we first introduce Table 3 in order to illustrate the effect of the pre-processing. This table shows the gaps for the instances with 100 nodes given by the optimal linear programming bound for formulations $FA_0, FA_1, NRFA_0$ and $NRFA_1$ described before, with and without the pre-processing, and the corresponding CPU times (the results correspond to the average of the three alternatives). The first two columns specify the problem instances. The next eight columns present the gaps given by the value [(OPT − LB)/OPT] * 100 (OPT is the value of the integer optimal solution and LB is the value of the lower-bound given by the optimal linear programming solution of the model indicated at the top of the column). Next to this value we present two other values in parentheses: the first one is the CPU time needed to solve the linear programming relaxation and the second one is the CPU time needed to obtain the optimal value (time is given in seconds).

The results strongly indicate that pre-processing reduces significantly the gaps, in particular for the FA_0 formulation: the maximum reduction obtained is of 10.89% for FA_0, 0.87% for FA_1, 1.20% for $NRFA_0$ and 1.22% for $NRFA_1$. Note, also, that the pre-processing has a great impact in the CPU times of the $NRFA$ model because it eliminates many paths, leading to a substantial number of variables that were eliminated from the model. This in turn, leads to a greater reduction in the M_{ij}^p coefficients. As stated before, it was this pre-processing leading to a reduction in the M_{ij}^p coefficients that has motivated the idea for creating the new model.

In order to compare the models, Tables 4–6 depict the gaps given by the optimal linear programming bound of the formulations $FA_0, FA_1, FD_0, FD_1, NRFA_0$ and $NRFA_1$, obtained with the pre-processing described. Each table corresponds to instances with the same number of nodes and the two structures used. Each column is easily identified by the corresponding name at its top. We present, in the same way as before, the gap value and, in parentheses, the corresponding CPU times. The designation $^-$ indicates that we were not able to solve the integer model within the time limit of 2 h.

The results indicate that the $NRFA$ models produce, in general, the best linear programming bounds and that the FA models produce, as expected, the worst. However, we note that for the alternatives A and B the three classes of models produce, in general, small gaps for all the instances giving some practical confirmation that the previous transformation of the LANEP into a CMSTP with additional constraints was worth developing (the exceptions are the cases with costs variation opposite to the network expansion type and whose necessity of expansion is reduced). We can also see that the three classes of models permit, in general, to obtain very quickly the optimal solutions to all instances with 100 and 200 nodes and to the instances with 500 nodes for alternatives A and B. With respect to the 500 nodes instances and for alternative C, the models required in general more time. We identify cases where FA models use less time than the others as well as the $NRFA$ models. Note that, in some problem instances, the $NRFA_1$ formulation produced the optimal solution in seconds while the FA_1 and FD_1

Table 3
Average results for instances with 100 nodes, with and without pre-processing, for FA and $NRFA$ models.

Instance		FA_0		FA_1		$NRFA_0$		$NRFA_1$	
		without PP	with PP	without PP	with PP	without PP	with PP	without PP	with PP
100 nodes	P/5	16.64(0 + 1)	12.40(0 + 0)	6.66(0 + 1)	6.39(0 + 1)	4.21(9 + 12)	4.09(1 + 4)	3.76(11 + 23)	3.65(1 + 4)
	P/2	14.06(0 + 0)	9.33(0 + 0)	4.54(0 + 1)	4.37(0 + 0)	4.03(9 + 15)	3.80(0 + 2)	3.82(12 + 27)	3.60(1 + 2)
	P	10.68(0 + 0)	6.12(0 + 0)	2.95(0 + 1)	2.78(0 + 0)	2.48(9 + 10)	2.14(0 + 1)	2.31(12 + 20)	1.99(0 + 2)
	Px2	7.53(0 + 0)	3.43(0 + 0)	1.78(0 + 0)	1.52(0 + 0)	1.15(5 + 4)	0.77(0 + 0)	1.13(10 + 11)	0.76(0 + 1)
	F/2	8.35(0 + 0)	4.85(0 + 0)	2.96(0 + 0)	2.74(0 + 0)	2.57(8 + 11)	2.17(0 + 1)	2.44(13 + 22)	2.05(0 + 2)
	Fx2	14.04(0 + 0)	8.19(0 + 0)	3.05(0 + 1)	2.88(0 + 0)	2.29(8 + 10)	2.09(0 + 1)	2.12(12 + 21)	1.93(0 + 2)
	Fx5	19.13(0 + 0)	11.30(0 + 0)	3.19(0 + 1)	2.90(0 + 0)	1.58(8 + 9)	1.45 (1 + 1)	1.44(14 + 17)	1.32(1 + 2)
	Fx20	27.77(0 + 1)	16.88(0 + 0)	5.33(0 + 1)	4.49(0 + 1)	1.39(9 + 11)	1.33(2 + 2)	1.29(15 + 30)	1.25(3 + 9)
	e/2	8.87(0 + 0)	5.93(0 + 0)	2.44(0 + 0)	2.33(0 + 0)	1.89(9 + 8)	1.77(0 + 1)	1.72(13 + 18)	1.62(0 + 2)
	ex2	11.60(0 + 0)	5.40(0 + 0)	3.00(0 + 0)	2.79(0 + 0)	2.62(7 + 8)	2.07(0 + 1)	2.45(10 + 22)	1.89(0 + 2)
	ex5	13.53(0 + 0)	6.04(0 + 0)	3.87(0 + 1)	3.57(0 + 0)	3.24(6 + 11)	2.50(0 + 1)	3.11(10 + 20)	2.30(0 + 2)
	ex20	15.39(0 + 0)	6.54(0 + 0)	4.77(0 + 1)	3.90(0 + 0)	4.05(5 + 11)	2.85(0 + 2)	3.88(9 + 17)	2.66(0 + 2)
	B/2	9.82(0 + 0)	4.50(0 + 0)	1.86(0 + 0)	1.62(0 + 0)	1.15(5 + 5)	0.84(0 + 0)	1.09(11 + 13)	0.81(0 + 1)
	Bx2	11.17(0 + 0)	7.68(0 + 0)	4.49(0 + 0)	4.31(0 + 0)	3.90(9 + 12)	3.65(0 + 2)	3.70(12 + 21)	3.49(0 + 3)
	Bx5	9.81(0 + 0)	7.60(0 + 0)	5.39(0 + 1)	5.21(0 + 0)	4.00(8 + 12)	3.82(0 + 2)	3.77(11 + 18)	3.59(0 + 2)

Table 4
Results for instances with 100 nodes with pre-processing. for structures 101 and 102.

Instance		FA_1	FA_1	FD_1	FD_1	$NRFA_1$	$NRFA_1$	Instance		FA_1	FA_1	FD_1	FD_1	$NRFA_1$	$NRFA_1$
101A	P/5	2.09 (0 + 0)	1.02 (0 + 0)	0.78(0 + 0)	0.71(0 + 0)	0.75 (0 + 0)	0.69(0 + 0)	102A	P/5	3.58(0 + 0)	1.50(0 + 1)	1.08(0 + 1)	1.02 (0 + 2)	1.08(0 + 1)	1.02(0 + 1)
	P/2	0.39 (0 + 0)	0.17 (0 + 0)	0.20(0 + 0)	0.15(0 + 0)	0.20 (0 + 0)	0.14 (0 + 0)		P/2	0.78(0 + 0)	0.38(0 + 0)	0.32 (0 + 0)	0.32(0 + 0)	0.32(0 + 0)	0.32 (0 + 0)
	P	0.02 (0 + 0)	0.01 (0 + 0)	0.01 (0 + 0)	0.01(0 + 0)	0.01 (0 + 0)	0.01 (0 + 0)		P	0.17 (0 + 0)	0.06(0 + 0)	0.06 (0 + 0)	0.03(0 + 0)	0.06(0 + 0)	0.03 (0 + 0)
	P×2	0.00 (0 + 0)	0.00 (0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00 (0 + 0)	0.00(0 + 0)		P×2	0.03(0 + 0)	0.01(0 + 0)	0.01(0 + 0)	0.01 (0 + 0)	0.01 (0 + 0)	0.01(0 + 0)
	F/2	0.02 (0 + 0)	0.01 (0 + 0)	0.01 (0 + 0)	0.01(0 + 0)	0.01 (0 + 0)	0.01 (0 + 0)		F/2	0.17 (0 + 0)	0.06(0 + 0)	0.06 (0 + 0)	0.03(0 + 0)	0.06(0 + 0)	0.03 (0 + 0)
	F×2	0.09 (0 + 0)	0.02 (0 + 0)	0.03(0 + 0)	0.02 (0 + 0)	0.03 (0 + 0)	0.02(0 + 0)		F×2	0.23(0 + 0)	0.06(0 + 0)	0.06 (0 + 0)	0.03(0 + 0)	0.06(0 + 0)	0.03 (0 + 0)
	F×5	0.36 (0 + 0)	0.03 (0 + 0)	0.04(0 + 0)	0.03 (0 + 0)	0.04 (0 + 0)	0.03(0 + 0)		F×5	0.55(0 + 0)	0.11 (0 + 0)	0.14(0 + 0)	0.10 (0 + 0)	0.14 (0 + 0)	0.10(0 + 0)
	F×20	2.40 (0 + 0)	0.14 (0 + 0)	0.14 (0 + 0)	0.12(0 + 0)	0.14 (0 + 0)	0.11 (0 + 0)		F×20	3.53(0 + 0)	0.25(0 + 0)	0.04 (0 + 0)	0.03(0 + 0)	0.04(0 + 0)	0.03 (0 + 0)
	e/2	0.14(0 + 0)	0.07 (0 + 0)	0.08(0 + 0)	0.07 (0 + 0)	0.08 (0 + 0)	0.06(0 + 0)		e/2	0.19 (0 + 0)	0.02(0 + 0)	0.03 (0 + 0)	0.02(0 + 0)	0.03(0 + 0)	0.02 (0 + 0)
	e×2	0.02 (0 + 0)	0.01 (0 + 0)	0.01 (0 + 0)	0.01(0 + 0)	0.01 (0 + 0)	0.01 (0 + 0)		e×2	0.10 (0 + 0)	0.05(0 + 0)	0.03 (0 + 0)	0.03(0 + 0)	0.03(0 + 0)	0.03 (0 + 0)
	e×5	0.07 (0 + 0)	0.01 (0 + 0)	0.01 (0 + 0)	0.01(0 + 0)	0.01 (0 + 0)	0.01 (0 + 0)		e×5	0.08(0 + 0)	0.04(0 + 0)	0.02 (0 + 0)	0.02(0 + 0)	0.02(0 + 0)	0.02 (0 + 0)
	e×20	0.01(0 + 0)	0.00 (0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00 (0 + 0)	0.00(0 + 0)		e×20	0.10 (0 + 0)	0.08(0 + 0)	0.09 (0 + 0)	0.08(0 + 0)	0.08(0 + 0)	0.09 (0 + 0)
	B/2	0.01(0 + 0)	0.00 (0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00 (0 + 0)	0.00(0 + 0)		B/2	0.06(0 + 0)	0.01(0 + 0)	0.02 (0 + 0)	0.01 (0 + 0)	0.02(0 + 0)	0.01(0 + 0)
	B×2	0.26 (0 + 0)	0.13 (0 + 0)	0.13 (0 + 0)	0.11 (0 + 0)	0.13 (0 + 0)	0.11 (0 + 0)		B×2	0.56(0 + 0)	0.36(0 + 0)	0.28 (0 + 0)	0.28(0 + 0)	0.28(0 + 0)	0.28 (0 + 0)
	B×5	1.06(0 + 0)	0.71 (0 + 0)	0.43(0 + 0)	0.41(0 + 0)	0.40 (0 + 0)	0.38(0 + 0)		B×5	1.71(0 + 0)	1.12 (0 + 0)	0.81(0 + 1)	0.77(0 + 1)	0.81 (0 + 0)	0.77 (0 + 1)
101B	P/5	1.26(0 + 0)	1.03 (0 + 0)	0.96(0 + 1)	0.90 (0 + 1)	0.90 (0 + 2)	0.88(0 + 2)	102B	P/5	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)
	P/2	2.47 (0 + 0)	2.01 (0 + 0)	1.62 (0 + 1)	1.61 (0 + 1)	1.60 (0 + 2)	1.59 (0 + 3)		P/2	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)
	P	1.35(0 + 0)	1.10(0 + 0)	0.78(0 + 0)	0.77 (0 + 1)	0.78 (0 + 2)	0.77(0 + 3)		P	0.03(0 + 0)	0.00(0 + 0)	0.03(0 + 0)	0.00(0 + 0)	0.03(0 + 0)	0.00 (0 + 0)
	P×2	0.40 (0 + 0)	0.24 (0 + 0)	0.15 (0 + 0)	0.15(0 + 0)	0.15 (0 + 0)	0.15 (0 + 1)		P×2	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)
	F/2	1.35(0 + 0)	1.10(0 + 0)	0.78(0 + 1)	0.77 (0 + 0)	0.78 (0 + 2)	0.77(0 + 3)		F/2	0.03(0 + 0)	0.00(0 + 0)	0.03 (0 + 0)	0.00(0 + 0)	0.03(0 + 0)	0.00 (0 + 0)
	F×2	1.36(0 + 0)	1.10(0 + 0)	0.79(0 + 1)	0.77 (0 + 1)	0.79 (0 + 3)	0.77(0 + 3)		F×2	0.03(0 + 0)	0.00(0 + 0)	0.02 (0 + 0)	0.00(0 + 0)	0.02(0 + 0)	0.00 (0 + 0)
	F×5	1.37(0 + 0)	1.10(0 + 0)	0.79(0 + 1)	0.77 (0 + 1)	0.79 (0 + 2)	0.77(0 + 3)		F×5	0.04(0 + 0)	0.00(0 + 0)	0.02 (0 + 0)	0.00(0 + 0)	0.02(0 + 0)	0.00 (0 + 0)
	F×20	1.45(0 + 0)	1.10(0 + 0)	0.79(0 + 1)	0.77 (0 + 1)	0.79 (0 + 2)	0.77(0 + 4)		F×20	0.05(0 + 0)	0.00(0 + 0)	0.02 (0 + 0)	0.00(0 + 0)	0.02(0 + 0)	0.00 (0 + 0)
	e/2	0.12(0 + 0)	0.02 (0 + 0)	0.02(0 + 0)	0.01(0 + 0)	0.02 (0 + 0)	0.01 (0 + 0)		e/2	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)
	e×2	1.33(0 + 0)	1.15(0 + 0)	0.78(0 + 0)	0.78 (0 + 0)	0.77 (0 + 1)	0.77(0 + 1)		e×2	0.08(0 + 0)	0.11 (0 + 0)	0.00 (0 + 0)	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 1)
	e×5	3.28 (0 + 0)	2.73 (0 + 0)	2.23(0 + 1)	2.20 (0 + 0)	2.15 (0 + 0)	2.13 (0 + 1)		e×5	3.89(0 + 0)	2.38(0 + 0)	1.04(0 + 1)	1.04 (0 + 1)	1.04 (0 + 2)	1.04(0 + 2)
	e×20	4.73 (0 + 0)	3.79 (0 + 0)	2.71 (0 + 0)	2.71(0 + 0)	2.59 (0 + 0)	2.59(0 + 0)		e×20	8.87(0 + 0)	4.87(0 + 0)	3.44 (0 + 2)	3.37(0 + 2)	3.43(0 + 4)	3.36 (0 + 5)
	B/2	0.41(0 + 0)	0.24 (0 + 0)	0.15 (0 + 0)	0.15(0 + 0)	0.15 (0 + 0)	0.15 (0 + 0)		B/2	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)
	B×2	2.46 (0 + 0)	2.01 (0 + 0)	1.62 (0 + 1)	1.61 (0 + 1)	1.60 (0 + 2)	1.59 (0 + 4)		B×2	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)
	B×5	1.20(0 + 0)	1.02 (0 + 0)	0.94(0 + 1)	0.89 (0 + 1)	0.88 (0 + 1)	0.87(0 + 1)		B×5	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)
101C	P/5	32.24(0 + 0)	15.87 (0 + 1)	6.74(0 + 5)	5.50 (0 + 4)	6.45 (0 + 7)	5.19 (1 + 5)	102C	P/5	35.21 (0 + 0)	18.90 (0 + 1)	15.37 (1 + 3)	14.10(1 + 4)	15.37 (5 + 13)	14.10 (4 + 15)
	P/2	22.66(0 + 0)	10.81(0 + 1)	8.85(0 + 1)	8.08 (0 + 2)	8.45 (0 + 2)	7.85(0 + 2)		P/2	29.71 (0 + 0)	12.85 (0 + 1)	12.26(0 + 5)	11.68(0 + 4)	12.21 (2 + 7)	11.69 (3 + 11)
	P	15.04(0 + 0)	8.36 (0 + 1)	6.67(0 + 1)	5.95 (0 + 1)	5.88 (0 + 0)	5.40(0 + 0)		P	20.09(0 + 0)	7.15(0 + 0)	6.09 (0 + 4)	5.76(0 + 3)	6.06(0 + 6)	5.75 (1 + 5)
	P×2	7.83 (0 + 0)	4.71 (0 + 0)	2.60(0 + 0)	2.58(0 + 0)	2.51 (0 + 0)	2.48(0 + 0)		P×2	12.35 (0 + 0)	4.18(0 + 0)	1.88(0 + 1)	1.94 (0 + 1)	1.87 (0 + 2)	1.84(0 + 2)
	F×2	12.26(0 + 0)	7.92 (0 + 0)	6.60(0 + 1)	6.00 (0 + 0)	5.76 (0 + 0)	5.39(0 + 0)		F×2	15.30 (0 + 0)	7.38(0 + 0)	6.41(0 + 3)	6.12 (0 + 2)	6.36(0 + 6)	6.07 (1 + 7)
	F×2	20.62(0 + 0)	9.11(0 + 1)	6.99(0 + 1)	6.36 (0 + 1)	6.36 (0 + 0)	5.84(0 + 1)		F×2	26.82(0 + 0)	6.96(0 + 0)	5.26 (0 + 4)	4.83(0 + 5)	5.25 (1 + 4)	4.93 (2 + 5)
	F×5	30.66 (0 + 1)	10.68 (0 + 1)	5.80(0 + 3)	5.58 (0 + 2)	4.96 (0 + 1)	4.82(0 + 2)		F×5	34.79(0 + 0)	5.50(0 + 0)	2.76 (1 + 3)	2.22(1 + 2)	2.74(5 + 5)	2.22 (8 + 5)
	F×20	47.23(0 + 2)	16.48(0 + 3)	4.82(1 + 6)	4.47 (1 + 6)	4.68 (1 + 7)	4.43(1 + 14)		F×20	46.63(0 + 0)	8.87(0 + 1)	2.79 (2 + 3)	2.63(2 + 2)	2.28(8 + 4)	2.16(16 + 36)
	e/2	15.78(0 + 0)	7.70 (0 + 1)	6.32(0 + 1)	5.87 (0 + 2)	5.82 (0 + 0)	5.46(0 + 1)		e/2	18.38 (0 + 0)	6.16(0 + 0)	4.77 (0 + 3)	4.24(0 + 2)	4.67 (1 + 5)	4.15(1 + 9)
	e×2	13.04(0 + 0)	7.89 (0 + 0)	5.45(0 + 0)	4.48 (0 + 0)	4.79 (0 + 0)	3.90(0 + 0)		e×2	17.84 (0 + 0)	7.65(0 + 1)	6.94 (0 + 3)	6.71 (0 + 4)	6.84(0 + 5)	6.62 (0 + 7)
	e×5	12.16 (0 + 0)	7.89 (0 + 0)	5.16 (0 + 0)	4.00 (0 + 0)	4.55 (0 + 0)	3.41 (0 + 0)		e×5	16.75 (0 + 0)	8.30(0 + 1)	7.34 (0 + 4)	7.28(0 + 4)	7.24(0 + 5)	7.18(0 + 7)
	e×20	10.34(0 + 0)	5.75 (0 + 0)	3.86(0 + 0)	2.81(0 + 0)	3.38 (0 + 0)	2.33(0 + 0)		e×20	15.20 (0 + 1)	8.81(0 + 1)	7.78 (0 + 6)	7.74(0 + 7)	7.62(0 + 6)	7.58 (0 + 6)
	B/2	10.38(0 + 0)	5.18 (0 + 0)	3.18 (0 + 0)	3.15(0 + 0)	2.92 (0 + 0)	2.85(0 + 0)		B/2	16.18(0 + 0)	4.32(0 + 0)	1.85(0 + 2)	1.84 (0 + 1)	1.94 (0 + 2)	1.83(0 + 3)
	B×2	17.34(0 + 0)	9.42 (0 + 0)	7.97(0 + 1)	7.09 (0 + 1)	7.56 (0 + 1)	6.89(0 + 1)		B×2	25.46(0 + 0)	13.87 (0 + 1)	12.48(0 + 5)	12.06 (0 + 5)	12.33 (1 + 7)	12.06(1 + 10)
	B×5	16.82(0 + 0)	11.10 (0 + 1)	7.26(0 + 1)	6.90 (0 + 1)	7.26 (0 + 2)	6.88(0 + 3)		B×5	24.79(0 + 0)	17.31 (0 + 0)	13.57(0 + 3)	12.64 (0 + 3)	13.57 (1 + 10)	12.64(2 + 6)

Table 5
Results for instances with 200 nodes with pre-processing. for structures 201 and 202.

Instance		FA_1	FA_1	FD_1	FD_1	$MRFA_1$	$NRFA_1$	Instance		FA_1	FA_1	FD_1	FD_1	$NRFA_1$	$NRFA_1$
201A	P/5	2.32 (0 + 11)	1.26(0 + 6)	0.87(0 + 2)	0.74 (0 + 2)	0.84 (0 + 0)	0.68 (0 + 0)	202A	P/5	2.88 (0 + 17)	1.21 (0 + 8)	0.84(0 + 78)	0.81(0 + 83)	0.82 (0 + 3)	0.79(0 + 5)
	P/2	0.57 (0 + 0)	0.20(0 + 0)	0.23(0 + 0)	0.17 (0 + 0)	0.22 (0 + 0)	0.17(0 + 0)		P/2	0.82 (0 + 0)	0.30(0 + 0)	0.29(0 + 0)	0.25 (0 + 0)	0.28 (0 + 0)	0.24(0 + 0)
	P	0.08 (0 + 0)	0.03(0 + 0)	0.03(0 + 0)	0.03 (0 + 0)	0.03 (0 + 0)	0.03 (0 + 0)		P	0.16(0 + 0)	0.06(0 + 0)	0.05(0 + 0)	0.05 (0 + 0)	0.05 (0 + 0)	0.05(0 + 0)
	P×2	0.01(0 + 0)	0.01(0 + 0)	0.01 (0 + 0)	0.01(0 + 0)	0.01(0 + 0)	0.01(0 + 0)		P×2	0.04 (0 + 0)	0.01 (0 + 0)	0.01 (0 + 0)	0.01(0 + 0)	0.01(0 + 0)	0.01(0 + 0)
	F/2	0.07 (0 + 0)	0.03(0 + 0)	0.03(0 + 0)	0.03 (0 + 0)	0.03 (0 + 0)	0.03 (0 + 0)		F/2	0.13(0 + 0)	0.05(0 + 0)	0.05(0 + 0)	0.05 (0 + 0)	0.05 (0 + 0)	0.05(0 + 0)
	F×2	0.12(0 + 0)	0.03(0 + 0)	0.03(0 + 0)	0.03 (0 + 0)	0.03 (0 + 0)	0.03 (0 + 0)		F×2	0.28 (0 + 0)	0.06(0 + 0)	0.05(0 + 0)	0.05 (0 + 0)	0.05 (0 + 0)	0.05(0 + 0)
	F×5	0.36 (0 + 0)	0.07(0 + 0)	0.04(0 + 0)	0.03 (0 + 0)	0.03 (0 + 0)	0.03 (0 + 0)		F×5	0.60 (0 + 0)	0.09(0 + 0)	0.06(0 + 0)	0.06 (0 + 0)	0.06 (0 + 0)	0.05(0 + 0)
	F×20	3.28 (0 + 0)	0.26(0 + 0)	0.18 (0 + 0)	0.15 (0 + 0)	0.18(0 + 0)	0.12(0 + 0)		F×20	3.07 (0 + 0)	0.23(0 + 0)	0.06(0 + 0)	0.05 (0 + 0)	0.06 (0 + 0)	0.04(0 + 0)
	e/2	0.19(0 + 0)	0.07(0 + 0)	0.07(0 + 0)	0.06 (0 + 0)	0.07 (0 + 0)	0.06 (0 + 0)		e/2	0.21(0 + 0)	0.04(0 + 0)	0.06(0 + 0)	0.04 (0 + 0)	0.06 (0 + 0)	0.04(0 + 0)
	e×2	0.04 (0 + 0)	0.01(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00 (0 + 0)	0.00 (0 + 0)		e×2	0.13(0 + 0)	0.07(0.0)	0.06(0 + 0)	0.06 (0 + 0)	0.06 (0 + 0)	0.06(0 + 0)
	e×5	0.02 (0 + 0)	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00 (0 + 0)	0.00 (0 + 0)		e×5	0.06 (0 + 0)	0.04(0 + 0)	0.04(0 + 0)	0.04 (0 + 0)	0.04 (0 + 0)	0.04(0 + 0)
	e×20	0.02 (0 + 0)	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00 (0 + 0)	0.00 (0 + 0)		e×20	0.03 (0 + 0)	0.02(0 + 0)	0.02(0 + 0)	0.02 (0 + 0)	0.02 (0 + 0)	0.02(0 + 0)
	B/2	0.01(0 + 0)	0.01(0 + 0)	0.01 (0 + 0)	0.01(0 + 0)	0.01(0 + 0)	0.01(0 + 0)		B/2	0.05 (0 + 0)	0.02(0 + 0)	0.02(0 + 0)	0.02 (0 + 0)	0.02 (0 + 0)	0.01(0 + 0)
	B×2	0.36 (0 + 0)	0.15(0 + 0)	0.14 (0 + 0)	0.13 (0 + 0)	0.13(0 + 0)	0.12(0 + 0)		B×2	0.64 (0 + 0)	0.24(0 + 0)	0.22(0 + 0)	0.20 (0 + 0)	0.22 (0 + 0)	0.20(0 + 0)
	B×5	1.13(0 + 2)	0.72(0 + 1)	0.52(0 + 1)	0.46 (0 + 1)	0.51(0 + 0)	0.45 (0 + 0)		B×5	1.32(0 + 3)	0.78(0 + 4)	0.51 (0 + 17)	0.48 (0 + 27)	0.50 (0 + 3)	0.48(0 + 3)
201B	P/5	3.19(0 + 0)	3.07(0 + 1)	2.41 (0 + 5)	2.40 (0 + 6)	2.39 (1 + 14)	2.38 (1 + 10)	202B	P/5	0.20 (0 + 0)	0.18 (0 + 0)	0.18 (0 + 1)	0.18(0 + 1)	0.19(1 + 0)	0.18(1 + 1)
	P/2	0.83 (0 + 0)	0.67(0 + 1)	0.54(0 + 2)	0.46 (0 + 2)	0.54 (0 + 2)	0.46 (0 + 7)		P/2	0.06 (0 + 0)	0.05(0 + 0)	0.05(0 + 1)	0.05 (0 + 1)	0.05 (0 + 1)	0.05(0 + 1)
	P	0.40 (0 + 0)	0.32(0 + 0)	0.15 (0 + 1)	0.15 (0 + 1)	0.15(0 + 3)	0.15(0 + 3)		P	0.33 (0 + 0)	0.02(0 + 0)	0.01 (0 + 1)	0.00 (0 + 1)	0.01(0 + 1)	0.00(0 + 1)
	P×2	0.19(0 + 0)	0.07(0 + 0)	0.02(0 + 0)	0.01(0 + 0)	0.02 (0 + 0)	0.01(0 + 1)		P×2	0.27 (0 + 0)	0.10 (0 + 0)	0.00(0 + 1)	0.00 (0 + 1)	0.00 (0 + 1)	0.00(0 + 1)
	F/2	0.40 (0 + 0)	0.32(0 + 0)	0.16 (0 + 1)	0.15 (0 + 1)	0.16(0 + 4)	0.15(0 + 2)		F/2	0.32 (0 + 0)	0.02(0 + 0)	0.01 (0 + 1)	0.00 (0 + 1)	0.01(0 + 1)	0.00(0 + 1)
	F×2	0.40 (0 + 0)	0.32(0 + 0)	0.15 (0 + 1)	0.15 (0 + 1)	0.15(0 + 4)	0.15(0 + 3)		F×2	0.34 (0 + 0)	0.02(0 + 0)	0.01 (0 + 1)	0.00 (0 + 1)	0.01(0 + 1)	0.00(0 + 1)
	F×5	0.41(0 + 0)	0.32(0 + 0)	0.15 (0 + 1)	0.14 (0 + 1)	0.15(0 + 3)	0.14(0 + 3)		F×5	0.35 (0 + 0)	0.02(0 + 0)	0.01 (0 + 1)	0.00 (0 + 1)	0.01(0 + 1)	0.00(0 + 1)
	F×20	0.47 (0 + 0)	0.42(0 + 0)	0.15 (0 + 1)	0.14 (0 + 1)	0.15(0 + 2)	0.14(0 + 3)		F×20	0.44 (0 + 0)	0.02(0 + 0)	0.00(0 + 1)	0.00 (0 + 1)	0.00 (0 + 1)	0.00(1 + 0)
	e/2	0.22 (0 + 0)	0.18(0 + 0)	0.14 (0 + 1)	0.13 (0 + 1)	0.14(0 + 2)	0.13(0 + 2)		e/2	0.01(0 + 0)	0.00(0 + 0)	0.01 (0 + 1)	0.00 (0 + 1)	0.01(0 + 1)	0.00(0 + 1)
	e×2	1.15(0 + 0)	0.93(0 + 1)	0.67(0 + 2)	0.62 (0 + 2)	0.62 (0 + 2)	0.58 (0 + 5)		e×2	0.85 (0 + 0)	0.35(0 + 0)	0.08(0 + 1)	0.05 (0 + 1)	0.08 (1 + 0)	0.05(1 + 1)
	e×5	2.49 (0 + 0)	2.11 (0 + 1)	1.44 (0 + 1)	1.42 (0 + 2)	1.33(0 + 1)	1.31 (0 + 2)		e×5	2.58 (0 + 0)	1.92 (0 + 1)	0.64(1 + 3)	0.56 (1 + 3)	0.64 (1 + 9)	0.56 (1 + 11)
	e×20	4.38 (0 + 1)	3.52(0 + 1)	3.19 (0 + 2)	3.13 (0 + 2)	2.99 (0 + 0)	2.84 (0 + 1)		e×20	3.68 (0 + 1)	2.47(0 + 1)	1.78 (1 + 8)	1.66 (1 + 8)	1.76(1 + 14)	1.65 (1 + 12)
	B/2	0.19(0 + 0)	0.07(0 + 0)	0.02(0 + 0)	0.01(0 + 0)	0.02 (0 + 0)	0.01(0 + 1)		B/2	0.28 (0 + 0)	0.10 (0 + 0)	0.00(0 + 1)	0.00 (0 + 1)	0.00 (0 + 1)	0.00(0 + 1)
	B×2	0.82 (0 + 0)	0.67(0 + 1)	0.54(0 + 2)	0.46 (0 + 2)	0.54 (0 + 2)	0.46 (1 + 4)		B×2	0.05 (0 + 0)	0.05(0 + 0)	0.05(0 + 1)	0.05 (0 + 1)	0.05 (0 + 1)	0.05(1 + 0)
	B×5	3.14(0 + 0)	3.06(0 + 1)	2.40(0 + 5)	2.39 (0 + 6)	2.38 (0 + 11)	2.38 (1 + 11)		B×5	0.19(0 + 0)	0.18 (0 + 0)	0.18 (0 + 1)	0.18(0 + 1)	0.18(1 + 0)	0.18(1 + 1)
201C	P/5	30.46 (0 + 3)	17.56 (0 + 10)	15.54 (3 + 125)	13.42(3 + 138)	14.80(9 + 116)	12.95(10 + 290)	202C	P/5	28.25(0 + 1)	7.83(0 + 1)	7.22(4 + 23)	6.36 (4 + 20)	7.20 (24 + 60)	6.36(25 + 85)
	P/2	19.89(0 + 5)	11.92 (0 + 9)	8.55(1 + 50)	7.76 (1 + 50)	7.95 (1 + 28)	7.19(1 + 24)		P/2	19.51 (0 + 0)	6.62(0 + 1)	5.55(2 + 21)	5.40 (3 + 14)	5.54 (6 + 25)	5.38(11 + 55)
	P	11.76 (0 + 3)	6.24(0 + 3)	4.92(0 + 16)	4.37 (0 + 12)	4.54 (0 + 3)	4.08 (0 + 3)		P	13.90(0 + 0)	4.25(0 + 1)	3.49 (1 + 12)	3.43 (1 + 13)	3.48 (2 + 15)	3.43(2 + 18)
	P×2	8.04 (0 + 1)	4.80(0 + 1)	4.03(0 + 1)	3.71 (0 + 1)	3.94 (0 + 0)	3.61(0 + 0)		P×2	8.26 (0 + 0)	2.72(0 + 1)	1.64 (0 + 2)	1.52(0 + 2)	1.61 (0 + 1)	1.48(0 + 4)
	F/2	9.05 (0 + 2)	5.77(0 + 3)	4.17 (0 + 6)	3.90 (0 + 7)	3.95 (0 + 2)	3.68 (0 + 2)		F/2	10.60(0 + 1)	4.14 (0 + 1)	3.58 (1 + 14)	3.50 (1 + 17)	3.57 (1 + 11)	3.49(2 + 16)
	F×2	16.22(0 + 3)	6.90(0 + 6)	4.80(0 + 18)	4.24 (0 + 16)	4.45 (0 + 4)	3.87 (0 + 5)		F×2	18.98(0 + 1)	4.61 (0 + 1)	3.16 (1 + 12)	3.05 (1 + 14)	3.14(3 + 20)	3.04(5 + 42)
	F×5	24.54 (0 + 7)	8.19(0 + 7)	4.60(1 + 26)	3.84 (0 + 31)	4.17(0 + 8)	3.38 (1 + 9)		F×5	27.50(0 + 1)	6.08(0 + 2)	3.56(3 + 25)	3.38 (2 + 23)	3.56 (11 + 41)	3.37(17 + 98)
	F×20	39.94 (0 + 15)	12.68 (0 + 45)	5.01 (3 + 213)	4.42 (3 + 148)	4.49 (2 + 53)	4.06 (4 + 88)		F×20	35.04(0 + 1)	6.98(0 + 2)	1.68 (7 + 8)	1.64 (7 + 11)	1.68(25 + 43)	1.63(33 + 51)
	e/2	12.12 (0 + 2)	5.28(0 + 3)	3.97(0 + 11)	3.43 (0 + 11)	3.78 (0 + 4)	3.36 (0 + 12)		e/2	13.31 (0 + 0)	3.68(0 + 1)	2.64(1 + 8)	2.48 (1 + 8)	2.64 (3 + 14)	2.48(5 + 21)
	e×2	11.35 (0 + 4)	6.33(0 + 4)	4.61 (0 + 8)	4.26 (0 + 8)	4.46 (0 + 4)	4.10(0 + 3)		e×2	13.28(0 + 2)	4.97(0 + 2)	4.31 (1 + 70)	4.17(1 + 176)	4.31(1 + 12)	4.16 (1 + 15)
	e×5	11.15 (0 + 6)	7.49(0 + 9)	4.70(0 + 7)	4.61(0 + 6)	4.61(0 + 1)	4.53 (0 + 2)		e×5	12.10 (0 + 1)	5.46(0 + 2)	5.15 (1 + 60)	4.84 (1 + 56)	5.15(1 + 12)	4.84 (1 + 11)
	e×20	10.88(0 + 3)	7.68(0 + 4)	4.79(0 + 6)	4.55 (0 + 6)	4.71(0 + 1)	4.48 (0 + 2)		e×20	11.02 (0 + 1)	5.62(0 + 1)	5.58 (1 + 18)	5.24 (1 + 18)	5.58 (1 + 8)	5.24(1 + 13)
	B/2	11.16 (0 + 4)	5.66(0 + 2)	4.58(0 + 2)	4.27 (0 + 2)	4.37 (0 + 0)	4.04 (0 + 1)		B/2	11.88 (0 + 0)	2.94(0 + 1)	1.58 (0 + 3)	1.44(0 + 3)	1.55(0 + 2)	1.37(0 + 3)
	B×2	15.69(0 + 3)	10.70 (0 + 5)	8.09(0 + 33)	7.38 (0 + 26)	7.43 (0 + 19)	6.76 (0 + 17)		B×2	15.11(0 + 1)	6.72(0 + 1)	5.35(2 + 18)	5.19(2 + 22)	5.34 (3 + 19)	5.18(5 + 52)
	B×5	18.61 (0 + 2)	13.70 (0 + 7)	11.80(1 + 53)	11.16 (1 + 47)	11.65 (1 + 47)	11.00 (1 + 161)		B×5	18.57(0 + 1)	9.65(0 + 1)	8.81 (2 + 21)	8.66 (2 + 18)	8.78 (9 + 40)	8.64(8 + 38)

Table 6
Results for instances with 500 nodes with pre-processing. for structures 501 and 502.

Instance		FA_1	FA_1	FD_1	FD_1	$NRFA_1$	$NRFA_1$	Instance		FA_1	FA_1	FD_1	FD_1	$NRFA_1$	$NRFA_1$
501A	P/5	2.03 (0 + 3443)	1.01 (0 + 500)	0.86 (0 + 232)	0.71 (0 + 128)	0.84 (0 + 1)	0.68(0 + 1)	502A	P/5	2.71(0 + –)	0.82(0 + –)	0.68(1 + –)	0.66(1 + –)	0.67 (1 + 188)	0.65 (1 + 110)
	P/2	0.60 (0 + 0)	0.23(0 + 0)	0.23 (0 + 0)	0.19 (0 + 0)	0.23 (0 + 0)	0.19 (0 + 0)		P/2	0.70 (0 + 7)	0.23(0 + 4)	0.19(0 + 10)	0.18 (0 + 5)	0.19 (0 + 0)	0.18(0 + 0)
	P	0.10(0 + 0)	0.04(0 + 0)	0.04 (0 + 0)	0.03(0 + 0)	0.04 (0 + 0)	0.03(0 + 0)		P	0.08 (0 + 0)	0.04(0 + 0)	0.03 (0 + 0)	0.03(0 + 0)	0.03(0 + 0)	0.03 (0 + 0)
	P×2	0.01(0 + 0)	0.01 (0 + 0)	0.01(0 + 0)	0.01 (0 + 0)	0.01(0 + 0)	0.01 (0 + 0)		P×2	0.01(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)
	F/2	0.08 (0 + 0)	0.04(0 + 0)	0.04 (0 + 0)	0.03(0 + 0)	0.04 (0 + 0)	0.03(0 + 0)		F/2	0.07 (0 + 0)	0.03(0 + 0)	0.03 (0 + 0)	0.03(0 + 0)	0.03(0 + 0)	0.03 (0 + 0)
	F×2	0.14(0 + 0)	0.05(0 + 0)	0.06 (0 + 0)	0.04(0 + 0)	0.06 (0 + 0)	0.04(0 + 0)		F×2	0.17(0 + 0)	0.05(0 + 0)	0.04 (0 + 0)	0.04(0 + 0)	0.04(0 + 0)	0.04 (0 + 0)
	F×5	0.42 (0 + 0)	0.07(0 + 0)	0.07 (0 + 0)	0.06(0 + 0)	0.07 (0 + 0)	0.06(0 + 0)		F×5	0.48 (0 + 0)	0.08(0 + 0)	0.05 (0 + 0)	0.04(0 + 0)	0.05(0 + 0)	0.04 (0 + 0)
	F×20	3.34 (0 + 2)	0.25(0 + 1)	0.18(0 + 0)	0.16 (0 + 0)	0.17(0 + 0)	0.13 (0 + 0)		F×20	3.22 (0 + 2)	0.26(0 + 1)	0.06 (0 + 1)	0.05(0 + 1)	0.06(0 + 0)	0.04 (0 + 0)
	e/2	0.18(0 + 0)	0.07(0 + 0)	0.07 (0 + 0)	0.06(0 + 0)	0.07 (0 + 0)	0.05(0 + 0)		e/2	0.18(0 + 0)	0.07(0 + 0)	0.07 (0 + 0)	0.06(0 + 0)	0.07(0 + 0)	0.06 (0 + 0)
	e×2	0.04 (0 + 0)	0.02(0 + 0)	0.01(0 + 0)	0.01 (0 + 0)	0.01(0 + 0)	0.01 (0 + 0)		e×2	0.08 (0 + 0)	0.04(0 + 0)	0.03 (0 + 0)	0.03(0 + 0)	0.03(0 + 0)	0.03 (0 + 0)
	e×5	0.02 (0 + 0)	0.01 (0 + 0)	0.00 (0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00(0 + 0)		e×5	0.04 (0 + 0)	0.02(0 + 0)	0.02 (0 + 0)	0.02(0 + 0)	0.02(0 + 0)	0.02 (0 + 0)
	e×20	0.01(0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00(0 + 0)	0.00 (0 + 0)	0.00(0 + 0)		e×20	0.04 (0 + 0)	0.02(0 + 0)	0.01(0 + 0)	0.01(0 + 0)	0.01 (0 + 0)	0.01(0 + 0)
	B/2	0.03 (0 + 0)	0.01 (0 + 0)	0.01(0 + 0)	0.01 (0 + 0)	0.01(0 + 0)	0.01 (0 + 0)		B/2	0.02 (0 + 0)	0.00(0 + 0)	0.01(0 + 0)	0.00(0 + 0)	0.01 (0 + 0)	0.00 (0 + 0)
	B×2	0.40 (0 + 0)	0.19 (0 + 0)	0.19(0 + 0)	0.15 (0 + 0)	0.18(0 + 0)	0.15 (0 + 0)		B×2	0.46 (0 + 1)	0.20(0 + 1)	0.17(0 + 1)	0.16 (0 + 1)	0.17 (0 + 0)	0.16(0 + 0)
	B×5	0.99 (0 + 29)	0.61 (0 + 11)	0.56 (0 + 17)	0.48(0 + 14)	0.54 (0 + 0)	0.47(0 + 0)		B×5	1.08 (0 + 2527)	0.62(0 + 2661)	0.51(0 + –)	0.50(0 + –)	0.50(0 + 95)	0.49 (0 + 84)
501B	P/5	2.14(0 + 2)	1.99 (0 + 5)	1.75(5 + 115)	1.71(6 + 99)	1.69(42 + 303)	1.65 (52 + 281)	502B	P/5	2.67 (0 + 0)	2.43(0 + 1)	1.62(7 + 21)	1.61(8 + 18)	1.62 (87 + 49)	1.61 (90 + 758)
	P/2	0.72 (0 + 1)	0.63(0 + 3)	0.51(2 + 25)	0.49(1 + 27)	0.48 (4 + 24)	0.47(6 + 42)		P/2	0.70 (0 + 0)	0.46(0 + 1)	0.04 (4 + 5)	0.04(4 + 5)	0.04(39 + 14)	0.04 (71 + 24)
	P	0.30 (0 + 1)	0.22(0 + 1)	0.17(1 + 7)	0.17 (1 + 7)	0.17 (1 + 5)	0.17 (2 + 13)		P	0.45 (0 + 0)	0.13 (0 + 0)	0.02 (2 + 5)	0.00(2 + 5)	0.02(13 + 12)	0.00 (38 + 22)
	P×2	0.10(0 + 0)	0.05(0 + 1)	0.03 (0 + 3)	0.03(1 + 2)	0.03 (0 + 2)	0.03(1 + 7)		P×2	0.28 (0 + 0)	0.05(0 + 0)	0.02 (1 + 5)	0.02(1 + 5)	0.02 (42 + 26)	0.02 (77 + 39)
	F/2	0.30 (0 + 1)	0.22(0 + 1)	0.17(1 + 6)	0.17 (1 + 6)	0.17 (1 + 6)	0.17 (2 + 10)		F/2	0.44 (0 + 0)	0.13 (0 + 0)	0.02 (3 + 4)	0.00(2 + 5)	0.02 (16 + 11)	0.00 (38 + 18)
	F×2	0.31(0 + 1)	0.22(0 + 1)	0.17(1 + 6)	0.17 (1 + 6)	0.17 (1 + 4)	0.17 (3 + 13)		F×2	0.46 (0 + 0)	0.13 (0 + 0)	0.02 (3 + 4)	0.00(3 + 5)	0.02(16 + 12)	0.00 (33 + 18)
	F×5	0.32 (0 + 0)	0.22(0 + 1)	0.17(1 + 8)	0.17 (1 + 6)	0.17 (1 + 4)	0.17 (2 + 14)		F×5	0.48 (0 + 0)	0.13 (0 + 1)	0.02 (3 + 4)	0.00(2 + 4)	0.02(17 + 12)	0.00 (75 + 18)
	F×20	0.38 (0 + 1)	0.22(0 + 1)	0.17(1 + 6)	0.17 (1 + 7)	0.17 (1 + 5)	0.17 (4 + 13)		F×20	0.63 (0 + 0)	0.13 (0 + 1)	0.02 (4 + 4)	0.00(3 + 4)	0.02(17 + 12)	0.00 (28 + 21)
	e/2	0.18(0 + 0)	0.14 (0 + 1)	0.09 (1 + 7)	0.09(1 + 8)	0.09 (8 + 27)	0.09 (20 + 27)		e/2	0.03 (0 + 0)	0.01 (0 + 0)	0.01(1 + 4)	0.01(1 + 4)	0.01(7 + 12)	0.01(13 + 18)
	e×2	0.86 (0 + 1)	0.59(0 + 2)	0.51(0 + 8)	0.49(0 + 8)	0.50 (0 + 5)	0.48(1 + 7)		e×2	0.81(0 + 0)	0.34(0 + 1)	0.20 (5 + 7)	0.20(5 + 8)	0.20 (36 + 38)	0.20 (67 + 67)
	e×5	1.40(0 + 1)	0.93(0 + 2)	0.78 (0 + 4)	0.75(0 + 5)	0.74 (0 + 2)	0.72(0 + 3)		e×5	1.87(0 + 1)	1.18(0 + 2)	0.49 (6 + 18)	0.46(6 + 23)	0.48 (25 + 73)	0.46 (43 + 165)
	e×20	3.11(0 + 3)	2.04(0 + 3)	1.80(0 + 3)	1.69 (0 + 3)	1.74(0 + 1)	1.63 (0 + 1)		e×20	5.27 (0 + 9)	3.60(0 + 35)	1.45(3 + 160)	1.45 (3 + 186)	1.44 (3 + 86)	1.44(5 + 87)
	B/2	0.12(0 + 0)	0.05(0 + 1)	0.03 (0 + 3)	0.03(1 + 2)	0.03 (0 + 2)	0.03(1 + 7)		B/2	0.28 (0 + 0)	0.05(0 + 1)	0.02 (1 + 5)	0.02(1 + 5)	0.02 (37 + 22)	0.02 (88 + 37)
	B×2	0.71(0 + 1)	0.63(0 + 3)	0.50 (1 + 25)	0.49(1 + 26)	0.49 (4 + 30)	0.47(12 + 58)		B×2	0.69 (0 + 0)	0.46(0 + 1)	0.04 (3 + 5)	0.04(4 + 5)	0.04 (72 + 20)	0.04 (66 + 53)
	B×5	2.09 (0 + 2)	2.41 (0 + 7)	1.74(5 + 131)	1.71(5 + 93)	1.68(23 + 488)	1.65 (40 + 339)		B×5	2.63 (0 + 0)	2.43(0 + 1)	1.62(7 + 26)	1.61(6 + 24)	1.62 (75 + 48)	1.61 (54 + 56)
501C	P/5	31.31(0 + –)	14.27 (0 + –)	9.60 (19 + –)	7.82(19 + –)	8.90 (44 + –)	7.19 (41 + –)	502C	P(5)	32.32 (0 + 28)	12.56 (0 + 28)	9.85 (135 + 2614)	9.58(136 + 2578)	9.81 (831 + –)	8.55 (623 + –)
	P/2	21.58(0 + –)	8.61(0 + –)	6.56(2 + –)	5.77(2 + –)	5.85 (2 + 590)	5.11(2 + 968)		P/2	22.01 (0 + 673)	8.77(0 + 143)	6.10(38 + –)	5.86(38 + –)	6.10(142 + 3506)	5.95 (164 + –)
	P	13.48(0 + –)	6.69(0 + –)	5.37 (0 + –)	4.86(0 + –)	4.88 (0 + 71)	4.43(0 + 78)		P	14.74(0 + 284)	5.74(0 + 71)	3.48 (12 + –)	3.38(12 + –)	3.41 (37 + 250)	3.28 (31 + 308)
	P×2	7.61(0 + 60)	3.77(0 + 16)	2.95 (0 + 10)	2.68(0 + 21)	2.85 (0 + 1)	2.56(0 + 1)		P×2	8.48 (0 + 5)	3.36(0 + 6)	1.54(1 + 31)	1.48 (1 + 36)	1.51 (1 + 5)	1.44 (1 + 8)
	F/2	10.43(0 + –)	6.44(0 + 2143)	5.35 (0 + 3746)	4.92(0 + –)	4.75 (0 + 35)	4.37(0 + 26)		F/2	11.37 (0 + 127)	5.64(0 + 58)	3.58(7 + –)	3.50(9 + –)	3.54(13 + 134)	3.43 (13 + 231)
	F×2	17.99 (0 + –)	6.95(0 + –)	5.20(1 + –)	4.74(1 + –)	4.83 (0 + 102)	4.38(1 + 104)		F×2	19.13 (0 + 178)	5.82(0 + 67)	3.20 (21 + –)	3.08(25 + 2563)	3.16 (75 + 441)	3.04 (56 + 649)
	F×5	27.65(0 + –)	8.55(0 + –)	4.53(2 + –)	4.26(2 + –)	4.09 (1 + 224)	3.80 (2 + 640)		F×5	25.74 (0 + 107)	5.80(0 + 30)	2.60 (47 + 1756)	2.51 (33 + 1454)	2.57(138 + 530)	2.48 (178 + 717)
	F×20	42.01(0 + –)	12.81(0 + –)	3.98(14 + –)	3.57(16 + –)	3.70 (8 + 414)	3.35(14 + 2408)		F×20	36.52 (0 + 175)	8.88(0 + 16)	1.33 (185 + 182)	1.23 (159 + 200)	1.30 (387 + 533)	1.21(688 + 1061)
	e/2	13.83(0 + –)	6.11 (0 + 4282)	4.71(1 + –)	4.48(1 + –)	4.46 (0 + 151)	4.20(1 + 278)		e/2	15.16 (0 + 21)	4.86(0 + 16)	2.75 (15 + 836)	2.63(18 + 964)	2.73 (53 + 245)	2.60 (81 + 318)
	e×2	12.45(0 + –)	6.56(0 + –)	5.03 (0 + 2841)	4.41 (0 + 3227)	4.51(0 + 19)	3.86(0 + 24)		e×2	13.56(0 + 1086)	5.76(0 + 378)	3.62(7 + –)	3.50(5 + –)	3.52(13 + 125)	3.38 (12 + 182)
	e×5	10.95(0 + –)	6.23(0 + –)	4.57 (0 + 416)	3.68 (0 + 563)	4.28 (0 + 6)	3.40(0 + 8)		e×5	12.12(0 + –)	5.88(0 + 265)	3.83(5 + –)	3.76(5 + –)	3.70(5 + 68)	3.63 (5 + 107)
	e×20	9.27 (0 + 191)	5.37(0 + 73)	3.78 (0 + 92)	2.83(0 + 48)	3.57 (0 + 2)	2.64(0 + 3)		e×20	11.52(0 + –)	6.12 (0 + 217)	4.31(4 + –)	4.03(5 + –)	4.15 (4 + 45)	3.86 (4 + 48)
	B/2	10.57(0 + 142)	4.17 (0 + 51)	3.04 (0 + 16)	2.73(0 + 20)	2.87 (0 + 2)	2.54(0 + 2)		B/2	12.07(0 + 28)	3.54(0 + 18)	1.54(2 + 62)	1.47 (2 + 42)	1.52 (1 + 14)	1.42(2 + 13)
	B×2	15.11 (0 + 1201)	7.18 (0 + 978)	6.07 (1 + 1695)	5.28(1 + 1149)	5.56 (1 + 161)	4.83(1 + 580)		B×2	16.65(0 + 126)	8.02(0 + 118)	5.81(26 + –)	5.74(27 + –)	5.80(78 + –)	5.72 (83 + –)
	B×5	19.36(0 + –)	11.91(0 + –)	10.21 (4 + –)	8.81 (4 + –)	8.83 (5 + 2301)	8.54(6 + 2201)		B×5	21.82 (0 + 35)	12.11 (0 + 78)	10.72(54 + 3588)	10.44(52 + 3801)	10.68(271 + 2509)	10.40(284 + –)

formulations did not provide the solution within the given time (see, for example, instances 502A_P/5, 501C_Fx2 and 501C_Bx5). Alternative *C* is the one that clearly presents the worst results with respect to the gaps values as well as the CPU times because it represents a more balanced alternative and the need to expand the network is smaller than in the other cases. We can also see that the alternatives *C* and *B* are the most dependent on the tree structure.

When we compare the results of the augmented formulation with the original formulation, for each class, or compare the results between the augmented formulations, the results indicate that the gaps given by the FD_1 and $NRFA_1$ formulations are significantly better than the corresponding gaps from the FA_1 model. These improvements are more accentuated for the alternative *C*. When we compare the gaps between the formulations $NRFA_1$ and FD_1, the differences are less significant (less than 0.5 percentage points), with $NRFA_1$ still providing the best results. The results also show that it is the *FA* class that benefits more from the inclusion of the valid inequalities. This is expected, since many of the inequalities introduced in FA_0, leading to FA_1, are redundant in the linear programming relaxation of FD_0 and $NRFA_0$. The comparison between the FA_0 and FA_1 models indicates a maximum reduction of nearly 40% on the gap values (see instance 102C_Fx20). For the *FD* and *NRFA* classes, the comparison between the original formulation and the corresponding augmented model show a maximum reduction of 2% on the gap values.

6. Conclusion

We have proposed a new flow-based model for the local access network expansion problem. We have shown that the new model has a linear programming relaxation that dominates the linear programming relaxation of a single-commodity flow model. This result is significantly enhanced by the fact that the linear programming relaxation of the new model also implies a large set of inequalities defined in the variables space of the single-commodity flow model. Furthermore, the model is also adequate for coefficient reduction and variable elimination, permitting us to obtain an extended flow-based model with much fewer variables (in some cases, the number of variables is in the same order of the number of variables of the aggregated *FA* model) and a tighter linear programming relaxation. Although there is no dominance relationship between the linear programming relaxation of the new model and the linear programming relaxation of the previously presented disaggregated flow model, the results indicate that the new model is preferable since it leads, in general, to smaller gaps (in fact, in no case did the *FD* models produce a better gap) and to similar or faster solution times. As a conclusion, we think that these network flow-based approaches, and here significantly enhanced by the new *NRFA* model, are a good alternative to solve the local access network expansion problem.

References

Balakrishnan, A., Magnanti, T.L., Shulman, A., Wong, R.T., 1991. Models for planning capacity expansion in local access telecommunication networks. Annals of Operations Research 33, 239–284.

Balakrishnan, A., Magnanti, T.L., Wong, R.T., 1995. A decomposition algorithm for local access telecommunications network expansion planning. Operations Research 43 (1), 58–76.

Bienstock, D., 1993. Computational experience with an effective heuristic for some capacity expansion problems in local access networks. Telecommunication Systems 1, 379–400.

Carpenter, T., Luss, H., 2006. Telecommunications access network design. In: Pardalos, P.M., Resende, M.G.C. (Eds.), Handbook of Optimization in Telecommunications. Springer, pp. 313–339.

Cho, G., Shaw, D.X., 1996. Limited Column Generation for Local Access Telecommunication Network Expansion and Its Extensions – Formulations, Algorithms and Implementation. Technical Report, School of Industrial Engineering, Purdue University.

Cho, G., Shaw, D.X., 1998. Models and implementation techniques for local access telecommunication network design. In: Yu, G. (Ed.), Book Chapter in Industrial Applications of Combinatorial Optimization. ISBN 0792350731.

Corte-Real, M., Gouveia, L., 2007. Network flow models for the local access network expansion problem. Computers and Operations Research 34, 1141–1157.

Flippo, O.E., Kolen, A.W.J., Koster, A.M.C.A., Leensel, R.L.M.J., 2000. A dynamic programming algorithm for the local access telecommunication network expansion problem. European Journal of Operational Research 127, 189–202.

Gavish, B., 1993. Formulations and algorithms for the capacitated minimal directed tree problem. Journal of the Association for Computing Machinery 30 (1), 118–132.

Gendreau, M., Potvin, J.Y., Smires, A., Soriano, P., 2006. Multi-period capacity expansion for a local access telecommunications network. European Journal of Operational Research 172, 1051–1066.

Goemans, M.X., Myung, Y., 1993. A catalog of Steiner tree formulations. Networks 23, 19–28.

Gouveia, L., Lopes, M.J., 2000. Valid inequalities for non-unit demand capacitated spanning tree problems with flow costs. European Journal of Operational Research 121, 394–411.

Jack, C., Kai, S.R., Shulman, A., 1992. Design and implementation of an interactive optimization system for telephone network planning. Operations Research 40 (1), 14–25.

Kouassi, R., Gendreau, M., Potvin, J.Y., Soriano, P., 2009. Heuristics for multi-period capacity expansion in local telecommunications network. Journal of Heuristics 15 (4), 381–402.

Lee, C.Y., 1993. An algorithm for the design of multitype concentrator networks. Journal of Operational Research Society 44 (5), 471–482.

Magnanti, T.L., Wolsey, L., 1995. Optimal trees in network models. In: Handbooks in Operations Research and Management Science, vol. 7. pp. 503–615.

Magnanti, T.L., Wong, R.T., 1984. Network design and transportation planning: Models and algorithms. Transportation Science 18, 1–55.

Rardin, R., Choe, U., 1979. Tighter Relaxations of Fixed Charge Network Flow Problems. Report 3-79-18, Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta.

Shulman, A., Vachani, R., 1993. A decomposition algorithm for capacity expansion of local access networks. IEEE Transactions on Communications 41 (7), 1063–1107.