European Journal of Operational Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Minimum weight clustered dominating tree problem

Pablo Adasme[a], Rafael Castro de Andrade[b,*]

[a] *Universidad de Santiago de Chile, Departamento de Ingeniería Eléctrica, Avenida Victor Jara 3519, Santiago 9160000, Chile*
[b] *Universidade Federal do Ceará, Departamento de Estatística e Matemática Aplicada, Campus do Pici, bloco 910, Fortaleza, Ceará 60440-900, Brasil*

ARTICLE INFO

Article history:
Received 22 September 2020
Accepted 10 August 2022
Available online xxx

Keywords:
Combinatorial optimization
Clustered network design
Valid inequalities

ABSTRACT

We discuss minimum weight clustered dominating trees that find applications in the wireless sensor network design based on a clustered independent set structure. A cluster consists of a master sensor and the sensors belonging to its sensing radius. Masters collect, filter, and transmit the sensed data to a central sensor responsible for processing all sensed information. The data sent from a cluster to the central sensor follow a unique path. It alternates between a master and a bridge node, in this order. A bridge allows data communication between two neighboring clusters. The larger the distance between masters and bridges, the higher the energy consumption for data transmission. To reduce energy consumption and increase the network lifetime, we investigate a clustered tree structure of minimum total link distances. We propose hop- and flow-based models, introduce valid inequalities for them, and discuss five exponential families of cuts when embedded into the branch-and-cut framework of the CPLEX solver. Our models benefit properly of the CPLEX Benders' decomposition. We also highlight the differences between the topology of the clustered tree of minimum cost and the one of the (non-clustered) minimum dominating tree of the corresponding instances in terms of cost and number of solution nodes.

1. Introduction

Wireless sensor networks (WSN) attract the interest from academia and industry (Ayaz et al., 2018; Khan et al., 2016; Park et al., 2018), mainly for applications with highly-correlated data (Vlajic & Xia, 2006). Indeed, we explore them in military communications, emergency and disaster recovery, intelligent home security, transportation, agriculture, and smart cities. Sensor networks offer vast potential for future technologies using the internet of things paradigm (Khan et al., 2016). Wireless devices (sensors, nodes) cover distinct points of a given area. They have limited battery capacity, central processing unit, and physical data storage memory. It is essential to efficiently handle sensors' energy resources to maximize network lifetime while maintaining data communication and connectivity requirements. Organizing a clustered hierarchical structure where sensors have distinct roles (master, slave, or bridge nodes) can save energy (Santos et al., 2009, 2012, 2016; Vlajic & Xia, 2006). Master nodes form an independent set of the given WSN and define clusters of sensors. Slave sensors belong to clusters defined by the coverage area of their master sensors. Slaves reduce their energy consumption by staying in a slow energy consumption state until they have to report new data to their master nodes. Masters stay active and, after processing and eliminating repeated data of their clusters, transmit the sensed information to a central sensor. To allow data communication among neighboring clusters, a sensor belonging to the sensing radius of their master nodes must also stay active. It performs a bridge role to allow data communication while keeping a distance of two hops (in the number of connections) between the master nodes of two neighboring clusters.

In practice, in addition to requiring more energy consumption, masters are more expensive than bridge nodes. Thus, it is interesting to cover the area to be monitored while distributing them in such a way that two masters do not belong to the sensing radius one of the other. We assume that the area is vast enough to require various clusters to cover it.

Large distances from non-master sensors to the cluster head (master node) possibly allow the transmission of multiple repeated data. The smaller are these distances, the less energy the sensors waste unnecessarily. In this sense, Vlajic & Xia (2006) considers a cluster's organization where master sensors are at most two hops away from any node of their clusters. In this work, we are interested in the case where non-master sensors are at one hop away from its cluster head and compare it w.r.t. a non-clustered dominating tree topology. This choice for the size (in number of hops) of a cluster is to save more energy. As an example, consider a

[*] Corresponding author.
E-mail addresses: pablo.adasme@usach.cl (P. Adasme), rafael.andrade@ufc.br (R.C.d. Andrade).

https://doi.org/10.1016/j.ejor.2022.08.014
0377-2217/© 2022 Elsevier B.V. All rights reserved.

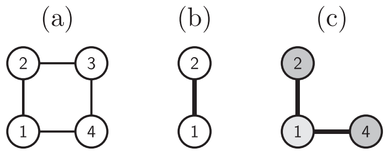

Fig. 1. The trees in (b) and (c) are a DT and a CT of the graph in (a), respectively.

sensor B positioned two hops away from the master node in the up-to-two-hops topology, namely 'master - sensor A - sensor B', and assume both A and B sensors detect the same event. Then, sensor B would take two hops to transmit its data to the master because the sensor A has not the ability to check and eliminate repeated data. Master and bridge sensors must form a connected network, minimizing the sum of the weights (energy consumption) of its connections (edges) for the data communication. Edge weights correspond to, for instance, the Euclidean distance between two sensors belonging to the sensing radius one of the other. Master nodes must eliminate repeated data of their clusters before sending them to a central node. Thus, bridge sensors are neither allowed to directly exchange data among them, nor to collect data from slave sensors. The network topology is a clustered tree of minimum weight.

Adasme et al. (2017) investigate the difference by considering the overall network link distances to evaluate WSN energy consumption instead of hop-count measures, and study a weighted dominating tree topology. In this structure, the active (dominating) nodes define a tree and every non-active (slave or dominated) node is adjacent to at least one dominating node. The dominating tree (DT) and the clustered tree (CT) concepts are distinct. Every CT is a DT, but a DT is not necessarily a CT. If we allow the master and bridge nodes to communicate among them with no distinction of their role in a CT, then there exists no difference between these two structures. For instance, in Fig. 1, the one-edge tree in (b) is a DT of the graph in (a). It does not have the required structure to be a CT. Indeed, one of the end-nodes of edge $\{1, 2\}$ must be the cluster head, e.g., node 2. In this case, it needs two hops to reach node 4 passing by the active node 1. Thus, violating the rule stating that every node (bridge or dominated) of a cluster is at a distance of one hop away from its master node. In this example, every CT of the graph in (a) must have two clusters to cover all its nodes as is the case of the tree in (c). Masters and bridges are in dark and light gray colors, respectively.

If we consider positive link distances and that the end-nodes of an edge belonging to the tree cannot be both bridges or masters, then the minimum weight DT (MDT) gives a lower bound on the solution value of the minimum weight CT (MCT). For instance, in Fig. 1, if all edges have unitary costs, then the DT in (b) has cost 1 while the CT in (c) has cost 2. This motivates us to compare (non-clustered) MDT and (clustered) MCT solution structures.

To the best of our knowledge, this is the first work handling weighted CT structures. In the literature, we find only a mixed-integer linear programming (MILP) model for a related non-weighted clustered WSN design problem (Santos et al., 2012). The authors propose a MILP formulation for the clustered WSN problem based on the average number of hops from master nodes to a central sensor. Similarly to (Adasme et al., 2018), we consider the total link distances instead of the number of hops from active nodes to the central sensor as this leads to a more realistic evaluation in terms of energy consumption and network lifetime.

Thus, we propose two MILP models for this variant of the clustered WSN problem. The first one (Desrochers & Laporte, 1991) uses improved Miller–Tucker–Zemlin (MTZ) inequalities (Miller et al., 1960). Whereas the second one is a network flow-based formulation (Gavish & Graves, 1978). For each model, we derive valid inequalities that allow reducing running times (CPU times, for short) for most of our instances of the problem. We highlight the importance of the proposed inequalities, including symmetry-breaking inequalities and the ones strengthening lower and upper bounds of the hop-count and arc-flow variables of our models. Note that any node can be the root of a tree and the hop-count increases by one unit every time we move in the path from the root to every other node of the tree. On the other hand, if we send some integer amount of flow from the root to the remaining nodes, with every other node retaining one unit of flow, the amount of flow in the path from the root to a given node decreases in one unit every traversed arc of this path. We can explore these facts only in hop- or flow-based models, thus justifying their choice in this work. We also show the impact of using five exponential families of linear inequalities (cuts) in a branch-and-cut framework for the hop-based model. Additionally, we further solve the proposed models by using an automatic Benders' decomposition module of a commercial MILP solver. This allows to reduce CPU times of both models when considering an extensive set of numerical experiments performed with instances presenting up to 150 nodes. We also implement the cut-and-branch algorithm of Adasme et al. (2018) to obtain the corresponding MDT solution of our instances to compare both MDT and MCT structures in terms of costs and the number of nodes belonging to the corresponding solutions.

2. Problem formulations

Initially, consider the following preliminary concepts. $G = (V, E)$ represents an undirected and connected graph. V represents the sensors (nodes). The edge set E contains all pairs $\{u, v\}$ such that sensors u and v belong to the sensing radius one of the other. Let $c_{uv} \geq 0$ represent the edge weight (cost, distance) connecting sensors u and v. Because we assume all sensors are identical, hereafter the cost related to the bidirectional communication between two sensors u and v, connected by an edge $\{u, v\} \in E$, is the same, i.e., $c_{uv} = c_{vu}$. A set $S \subseteq V$ is dominating if, for every $v \in V$, then either $v \in S$ or v is adjacent to some $u \in S$; otherwise, it is a non-dominating set. A node belonging (resp. not belonging) to a dominating set is a dominating (resp. non-dominating) node. A set $S \subseteq V$ is independent if, for every pair $u, v \in S$, then $\{u, v\} \notin E$. Let $N(u) = \{v \in V \mid \{v, u\} \in E\}$ denote the neighborhood of node $u \in V$. The closed neighborhood of $v \in V$ is defined as $N[u] = N(u) \cup \{u\}$. We call $S \subset V$ a node cut set if the exclusion of S from G generates a disconnected graph. If $\{v\}$ is a unitary node cut set, then $v \in V$ is a cut node of G. A non-empty set of edges $(S, S') = \{\{u, v\} \in E \mid u \in S, v \in S'\}$, with $\emptyset \neq S,\ S' \subset V$ and $S \cap S' = \emptyset$, is an edge cut if its exclusion from G disconnects it.

A tree T of G is an acyclic connected subgraph. It is dominating if its nodes form a dominating set of G. If we choose a node s of a tree T to be its root, then the hop-count $\pi(u)$ is the number of hops (hereafter given in the number of connections) in the path from a node u of T to s, with $\pi(s) = 0$. The center of a tree is a node minimizing the maximum path (in number of hops) from it to the remaining nodes of the tree. A node incident to exactly one edge in a tree is a leaf. Otherwise, it is an internal node. A tree is trivial if it is a single isolated node.

Consider a directed graph $\hat{G} = (V, A)$ obtained from $G = (V, E)$. The arc set A is such that if edge $\{u, v\} \in E$, then arcs (u, v) and (v, u) belong to A, with costs $c_{uv} = c_{vu}$ being equal to the cost of edge $\{u, v\}$. An s-rooted arborescence is a directed graph in which there is exactly one directed path from s to every other node.

2.1. A hop-based model

Let an arborescence T of $\hat{G}$ be represented by variables $x \in \{0, 1\}^{|A|}$, where $x_{uv} = 1$ if arc (u, v) belongs to T, and $x_{uv} = 0$, otherwise. Consider variables $y_v \in \{0,1\}$, for all $v \in V$, where $y_v = 1$ if v is a master node of T; and $y_v = 0$, otherwise. Similarly, let us define variables $z_v \in \{0,1\}$, for all $v \in V$, where $z_v = 1$ if v is a bridge node of T; and $z_v = 0$, otherwise. For all $v \in V$, let us define the variables $\pi_v \in \mathbb{R}_+$ to represent the hop-count (in number of arcs) from the root node to v in T. For nodes not belonging to T, their distance to the root node is irrelevant. Thus, we propose the following MILP model

$$(M) \quad \min \sum_{(u,v)\in A} c_{uv} x_{uv} \tag{1}$$

$$s.t. \quad \sum_{(u,v)\in A} x_{uv} = \sum_{v\in V}(z_v + y_v) - 1, \tag{2}$$

$$\pi_v - \pi_u - (|V|-1)x_{uv} - (|V|-3)x_{vu} \geq 2 - |V|, \quad \forall (u,v) \in A, \tag{3}$$

$$\sum_{u\in N(v)} x_{uv} \leq y_v + z_v, \quad \forall v \in V, \tag{4}$$

$$\sum_{u\in N[v]} y_u \geq 1, \quad \forall v \in V, \tag{5}$$

$$y_u + y_v \leq 1, \quad \forall \{u,v\} \in E, \tag{6}$$

$$y_v + z_v \leq 1, \quad \forall v \in V, \tag{7}$$

$$z_u + z_v + x_{uv} + x_{vu} \leq 2, \quad \forall \{u,v\} \in E, \tag{8}$$

$$2x_{uv} \leq z_u + z_v + y_u + y_v, \quad \forall (u,v) \in A, \tag{9}$$

$$z, y \in \{0,1\}^{|V|}, x \in \{0,1\}^{|A|}, \pi \in [0, |V|-2]^{|V|}. \tag{10}$$

In model (M), constraint (2) establishes that the number of arcs is one unit less than the sum of the master and bridge nodes in a solution T. Constraints (3) are strengthened MTZ subtour elimination constraints (Desrochers & Laporte, 1991). They establish that if arc (u,v) belongs to T, then π_v is at least one unit larger than π_u. Alternatively, if arc (v,u) belongs to T, then π_v is at least $\pi_u - 1$. By constraints (4) and (7), there is at most one arc entering every node belonging to T. Constraints (5) impose that either a node or one of its neighbors is a dominating master node. Constraints (6) ensure that the two extremities of an edge cannot be both master nodes. Constraints (5) and (6) came from the independent dominating set polytope (Mahjoub & Mailfert, 2006). Notice that the master nodes form an independent set. Constraints (7) establish that $v \in V$ is either a master or a bridge node; otherwise, it does not belong to T. Constraints (8) impose that if one of the arcs (u,v) or (v,u) belongs to T, then both of its extremities cannot be bridge nodes. By constraints (9), if an arc (u,v) belongs to T, then one of its extremities will be a bridge and the other one a master node. Since constraints (7) impose that at most one of the variables y_w and z_w is 1, for every $w \in V$, then constraints (4) and (9) also impose that if u (resp. v) is a dominated node, then no arc leaves u (resp. enters v) in the solution. The domain of the variables is given in constraints (10).

Proposition 1. *Model (M) correctly gives a minimum cost clustered arborescence of $\hat{G}$.*

Proof. Let $(\bar{z}, \bar{y}, \bar{x}, \bar{\pi})$ be an optimal solution of (M), with the non-null integer components of $\bar{z}$, $\bar{y}$, and $\bar{x}$ representing the sets $\bar{B}$ of bridges, $\bar{M}$ of masters, and $\bar{A}$ of arcs in the solution, respectively. Clearly, $\bar{M}$ is an independent set of $\hat{G}$ and every node $v \in V$ is a master, a bridge, or a dominated node, with $\bar{M} \cap \bar{B} = \emptyset$, $|\bar{M} \cup \bar{B}| = |\bar{M}| + |\bar{B}|$ and, by (2), $\sum_{(u,v)\in \bar{A}} \bar{x}_{uv} = |\bar{M}| + |\bar{B}| - 1$. If an arc (u,v) belongs to the solution, by (9) and (7), then $u \in \bar{B}$ and $v \in \bar{M}$ or $u \in \bar{M}$ and $v \in \bar{B}$, i.e., by (8) u and$v$ cannot be both bridge nodes. Notice that there is no arc (u,v) incident to a node u not belonging to the solution. Now, we show that $(\bar{z}, \bar{y}, \bar{x}, \bar{\pi})$ induces a unique component of $\hat{G}$. Initially, note that the number of components in any solution of model (M) is finite. Notice also that there is no isolated node in the solution. If it were the case, an isolated node w would contribute with one unit to the number of arcs in the solution, where no arc is incident to w. All the arcs in $\bar{A}$ belong to the component $\bar{M} \cup \bar{B} \setminus \{w\}$. Consequently, $\sum_{(u,v)\in \bar{A}} \bar{x}_{uv} = \sum_{(u,v)\in A} \bar{x}_{uv} = \sum_{v\in V}(\bar{z}_v + \bar{y}_v) - 1 = \sum_{v\in \bar{M}\cup\bar{B}}(\bar{z}_v + \bar{y}_v) - 1 = \sum_{v\in \bar{M}\cup\bar{B}\setminus\{w\}}(\bar{z}_v + \bar{y}_v) = |\bar{M} \cup \bar{B}| - 1$.

In this case, the component $\bar{M} \cup \bar{B} \setminus \{w\}$ has $|\bar{M} \cup \bar{B}| - 1$ arcs. Thus, it must contain a cycle because it is well known that the maximum number of arcs to connect a set of $|\bar{M} \cup \bar{B}| - 1$ nodes without forming a cycle is $|\bar{M} \cup \bar{B}| - 2$. By (3) and (4), the solution must not contain a cycle. Therefore the solution does not contain isolated master and bridge nodes. We also claim that the solution does not contain disjoint non-trivial arborescences. To see this, assume w.l.g. that a feasible solution contains exactly two arborescences, say $T_1 = (V_1, A_1)$ and $T_2 = (V_2, A_2)$. The set of master and bridge nodes and the set of arcs of the solution are $V_1 \cup V_2$ and $A_1 \cup A_2$, respectively, with $|A_1| = |V_1| - 1$ and $|A_2| = |V_2| - 1$. Because T_1 and T_2 are disjoint, the number of arcs in the solution is the sum of the arcs of these arborescences, i.e., $|A_1| + |A_2| = |V_1| - 1 + |V_2| - 1 = |V_1| + |V_2| - 2 < |V_1| + |V_2| - 1$. Thus, violating (2) and contradicting the assumption that the solution is feasible for (M). Therefore, the solution cannot contain disjoint non-trivial arborescences. Consequently, all masters and bridges are connected and form a single arborescence of $\hat{G}$. Finally, by (5), every node belongs to the solution or, if not, it is adjacent to some master node. Therefore, every feasible solution satisfies the required clustered topology and, by the objective sense in (1), it is of minimum cost. Finally, the edges associated with the arcs of the solution arborescence for model (M) clearly form a minimum cost CT of G. Thus, concluding the proof. □

2.1.1. Valid inequalities for model (M)

In the following paragraphs, we discuss valid inequalities for model (M). We assume there is no dominating set of cardinality 1. Let $A(S) \subset A$, with $S \subset V$, denote the arc set with both extremities in S. Initially, we lift the cover inequalities (5) as in Gendron et al. (2014) to obtain

$$\sum_{a\in A(N[u])} x_a \leq \sum_{v\in N[u]} (y_v + z_v) - 1, \quad \forall u \in V. \tag{11}$$

Note that $\sum_{v\in N[u]}(y_v + z_v)$ represents the number of solution nodes in component $N[u]$. Thus, inequalities (11) correspond to the well-known subtour elimination constraints $\sum_{a\in A(S)} x_a \leq |S| - 1$ defined for a set $S = N[u] \subset V$, for every $u \in V$. However, note that $\sum_{a\in A(S)} x_a \leq \sum_{v\in S}(y_v + z_v) - 1$ is not valid for every subset S of V. Indeed, if we consider $\bar{S}$ a subset of dominated nodes of a feasible solution, then the right-hand side of the corresponding constraint (11) for this component is -1. Thus, this constraint cannot be satisfied because of the non-negativity of the x variables.

We adapt them for every subset $S \subset V$ according to

$$\sum_{a\in A(S)} x_a \leq \sum_{v\in S\setminus\{u\}} (y_v + z_v), \quad \forall S \subset V, \ \forall u \in S. \tag{12}$$

Alternatively, we can look for violated cut-set inequalities (Gendron et al., 2014). If some component $S \subset V$ defines a node cut, then its exclusion from G makes it disconnected. If neither S nor $V \setminus S$ are dominating sets, then there exists at least one edge of $(S, V \setminus S)$ in any feasible solution of (M), i.e.,

$$\sum_{\{u,v\}\in(S,V\setminus S)} (x_{uv} + x_{vu}) \geq 1, \ \forall \text{ cut } (S, V \setminus S) \times \text{ of } G\text{: } S \text{ and } V \setminus S \text{ not dominating sets.} \quad (13)$$

We can also strengthen the linear relaxation of (M) by observing that if a node belongs to the solution then some arc must be adjacent to it. Depending on whether this node is a bridge or a master, we have

$$y_v + 2z_v \leq \sum_{u\in N(v)} (x_{uv} + x_{vu}), \quad \forall v \in V. \quad (14)$$

Moreover, for node cuts and closed neighborhoods of nodes in V, we explore straightforward versions of (13) as follows. Let $\mathcal{C}$ be the set of cut nodes of G and C^S denote the set of connected components obtained by removing the nodes of S from G, except for those components which are dominated by S. Then

$$\sum_{S\in C^{\{w\}}} \sum_{\{u,v\}\in(\{w\},S)} (x_{uv} + x_{vu}) \geq |C^{\{w\}}|, \quad \forall w \in \mathcal{C}, \quad (15)$$

$$y_u + z_u = 1, \quad \forall u \in \mathcal{C}, \quad (16)$$

$$\sum_{S\in C^{N[u]}} \sum_{\{u,v\}\in(N[u],S)} (x_{uv} + x_{vu}) \geq |C^{N[u]}|, \quad \forall u \in V. \quad (17)$$

Observe that (15) and (17) are linear combinations of the corresponding inequalities in (13) related to the components obtained with the exclusion of the corresponding set of nodes. Using one inequality (13) for every non-dominated component gives better linear relaxed solutions than their linearly combined versions (15) and (17) to solve our instances. The overall impact on CPU times is almost the same for both cases. Thus, we opted for using (15) and (17) in our models and use them only when all components obtained with the exclusion of the corresponding set of nodes are not dominated by them. In Annex D, we show the impact on the linear relaxed solution values and CPU times when using these inequalities, linearly and not linearly combined, for the case where not all components are non-dominated.

To develop some of the following linear inequalities, note that every node of a tree can be its root. Thus, there are many symmetric solutions represented in different ways depending on the root choice. To reduce the space of feasible solutions of model (M), we can impose the arborescence root to be a master or a bridge node. Although there are more masters than bridge nodes in a CT, we decided arbitrarily to adopt a master node to be the root to develop linear inequalities for model (M) and give, in case the reader be interested in trying them, the corresponding inequality based on a bridge root. In some situations (see Proposition 2), obtaining the equivalent linear inequality for model (M) leads to a weaker inequality.

Initially, we can impose that any bridge node has at least one arc entering it

$$z_v \leq \sum_{u\in N(v)} x_{uv}, \quad \forall v \in V. \quad (18)$$

Thus, eliminating symmetric solutions corresponding to feasible arborescences rooted at them. For a bridge root, the equivalent of (18) would be $y_v \leq \sum_{u\in N(v)} x_{uv}$, for all $v \in V$.

We can reduce the space of feasible solutions of model (M) even further by letting the arborescence root be a master and internal node. Indeed, let L^1 be the leaves set of G. If $L^1 \neq \emptyset$, then for all $l \in L^1$ and its unique neighbor v in V, l cannot be a bridge node and either l or v must be a master node. If v is a bridge, then arc (v, l) must belong to the solution. Consequently,

$$z_l = 0, \quad \forall l \in L^1, \quad (19)$$

$$y_v + y_l = 1, \quad \forall l \in L^1, \ v \in N(l), \quad (20)$$

$$2z_v \leq y_l + x_{vl}, \quad \forall l \in L^1, \ v \in N(l). \quad (21)$$

Moreover, we can place the root node as near as possible to the center of the tree.

Proposition 2. *Let* $p = \lfloor \frac{|V|}{2} \rfloor$ *and* $|V| \geq 4$. *There is an optimal solution of model* (M) *where*

$$\pi_v + z_v \leq p, \quad \forall v \in V. \quad (22)$$

Proof. The largest path (in hop-count) $P = (m_1, b_1, m_2, b_2, \ldots, m_q, b_q, m_{q+1})$ in any feasible solution of model (M) occurs between two master leaf nodes m_1 and m_{q+1}, for some integer $q \geq 1$. It has an odd number of nodes and alternates masters m_k and bridges b_t nodes, for $1 \leq k \leq q+1$ and $1 \leq t \leq q$, with $2q+1 \leq |V|$. Suppose that G contains a feasible solution path P with $2q+1 = |V|$ odd (resp. with $|V|$ even and $2q+1 = |V|-1$). Let the node in position $\lceil \frac{2q+1}{2} \rceil = q+1$ (resp. $q = \frac{|V|}{2}$, for $|V|$ even) be the root of the arborescence with respect to P. The distance from the root to any other node in P is at most q, for $|V|$ odd, and $q-1$, for $|V|$ even. In both cases, the distance from the root to any other node in P is not larger than $\lfloor \frac{|V|}{2} \rfloor$. This bound remains valid if the solution of model (M) is a tree because any path in this solution has at most $|V|$ nodes. Finally, as bridge nodes cannot be leaves in a CT, the result follows. □

As a consequence of Proposition 2, we can strengthen constraints (3) according to

$$\pi_v - \pi_u - px_{uv} - (p-2)x_{vu} \geq 1 - p, \quad \forall (u, v) \in A. \quad (23)$$

We can find similar ideas related to the symmetry-breaking inequalities (22) and (23) in Akgün & Tansel (2011).

The following set of inequalities (24) is also related to the fact that nodes not belonging to the solution have no arc adjacent to them, i.e.,

$$x_{uv} + x_{vu} \leq y_v + z_v, \quad \forall v \in V, \ \forall u \in N(v). \quad (24)$$

Inequalities (24) are not the unique way to represent this idea. The following inequalities can also avoid arcs adjacent to dominated nodes.

$$x_{uv} \leq y_v + z_v, \ \forall v \in V, \ \forall u \in N(v), \quad (25)$$

$$x_{uv} \leq z_u + z_v, \ \forall \ (u, v) \in A, \quad (26)$$

$$x_{uv} \leq y_u + y_v, \ \forall \ (u, v) \in A. \quad (27)$$

Inequalities (25) are weaker than inequalities (24). However, the latter ones allow obtaining stronger linear relaxed solution values for model (M) than using jointly inequalities (26) and (27).

The following inequalities do also explore structural properties of the problem solution. Because intersecting clusters share a bridge connecting their master nodes and there are no cycles in the solution, the number of masters is at least one unit more than the one of bridges.

$$\sum_{v\in V} y_v \geq \sum_{v\in V} z_v + 1. \quad (28)$$

Also, note that not all nodes belonging to the neighborhood of a dominated or a bridge node in the solution can be bridge ones. Thus, the maximum number of bridge nodes that are neighbors to any node is

$$\sum_{u \in N(v)} z_u - y_v \leq |N(v)| - 1, \quad \forall v \in V. \tag{29}$$

Now, we develop valid bounds related to the hop-count variables π. We assume that the root arborescence is a master node. Thus, bridge nodes are at an odd distance of at least one arc from the root node and masters are at an even distance from the root node. Consequently, the following trivial inequalities are valid for model (M)

$$\sum_{v \in V} \pi_v \geq \sum_{v \in V} z_v + 2 \sum_{v \in V} y_v - 2, \tag{30}$$

$$\pi_v \geq z_v \quad \forall v \in V, \tag{31}$$

$$\pi_v \geq 2 \sum_{u \in N(v)} x_{uv} - z_v \quad \forall v \in V. \tag{32}$$

The equivalent of (30) in the case of a bridge root is $\sum_{v \in V} \pi_v \geq 2 \sum_{v \in V} z_v + \sum_{v \in V} y_v - 2$. This inequality is weaker than inequality (30). Similarly, the corresponding inequalities for (31) and (32) are $\pi_v \geq y_v$, and $\pi_v \geq 2 \sum_{u \in N(v)} x_{uv} - y_v$, for all $v \in V$, respectively.

Finally, we close this section with a new π lower bound for the bridge nodes.

Proposition 3. *Valid lower bounds on the π variables related to bridge nodes can be strengthened based on the arc extremities in the solution as*

$$\pi_v \geq 2 \sum_{k \in N(u)} x_{ku} + (y_u + z_v - 1) - 2(1 - x_{uv}), \quad \forall (u, v) \in A. \tag{33}$$

Proof. Suppose first that $x_{uv} = 0$. Thus, the only way to get a positive right-hand side, being equal to 1, is to have an arc entering u and $y_u = z_v = 1$. The bound is valid because v is a bridge (see (31)). Case $x_{uv} = 1$ and $y_u = z_v = 1$, if: (i) u is the master root node, then the right-hand side, that is 1, is valid because v is a bridge; (ii) u is a master non-root node, then there must exist an arc entering u, leading to a right-hand side being equal to 3 that is stronger than the one obtained by (32) for the bridge node v. If $x_{uv} = 1$ and $y_u = z_v = 0$, then u is a bridge node, and there must exist an arc entering it, giving a right-hand side equal to 1 that is weaker than the one obtained by (32) for the master node v. □

In case of a bridge root, the equivalent of (33) is $\pi_v \geq 2 \sum_{k \in N(u)} x_{ku} + (z_u + y_v - 1) - 2(1 - x_{uv})$, for all $(u, v) \in A$.

2.2. A flow-based model

In this section, we develop a flow-based model for the problem. Let $\tilde{G} = (V \cup \{r\}, A \cup A_r)$ be an expanded digraph obtained from $\hat{G} = (V, A)$, where we add to $\hat{G}$ a dummy node r and a set of arcs A_r with null costs from r to every node $v \in V$. Let $N^+(v) = \{u \mid (v, u) \in A \cup A_r\}$ and $N^-(v) = \{u \mid (u, v) \in A \cup A_r\}$) be the positive and the negative neighborhood of a node v of $\tilde{G}$, respectively, with $N(v) = N^-(v) \cup N^+(v)$ and $N[v] = N(v) \cup \{v\}$. The idea is to construct an arborescence rooted at r, expanding all master and bridge nodes, having exactly one arc leaving r. This allows eliminating the unique arc that is adjacent to r in any arborescence of $\tilde{G}$, thus providing an arborescence of $\hat{G}$ with the latter characteristics. In order to represent an arborescence of $\tilde{G}$, we expand the vector of arc variables to $x \in \{0, 1\}^{|A \cup A_r|}$ and keep variables $y \in \{0, 1\}^{|V|}$ and $z \in \{0, 1\}^{|V|}$ previously defined. We also define continuous non-negative flow variables $f \in [0, |V| - 2]^{|A \cup A_r|}$, where f_a denotes the amount of flow on arc $a \in A \cup A_r$. If we represent the number of master and bridge nodes in the solution by a non-negative variable K, then the idea (Gavish & Graves, 1978) is to send K units of flow from r to the nodes in V, one unit of flow to each one of them.

$$(F) \quad \min \sum_{(u,v) \in A} c_{uv} x_{uv}$$

$$s.t. \quad \sum_{v \in V} (y_v + z_v) = K, \tag{34}$$

$$\sum_{(u,v) \in A} x_{uv} = K - 1, \tag{35}$$

$$\sum_{a \in A_r} f_a = K, \tag{36}$$

$$\sum_{u \in N^-(v)} f_{uv} - \sum_{u \in N^+(v)} f_{vu} = y_v + z_v, \quad \forall v \in V, \tag{37}$$

$$x_{uv} \leq f_{uv} \leq (|V| - 2) x_{uv}, \quad \forall (u, v) \in A \cup A_r, \tag{38}$$

$$\sum_{v \in V} x_{rv} = 1, \tag{39}$$

$$\sum_{u \in N^-(v)} x_{uv} = y_v + z_v, \quad \forall v \in V, \tag{40}$$

$$y_u + y_v \leq 1, \quad \forall \{u, v\} \in E, \tag{41}$$

$$y_v + z_v \leq 1, \quad \forall v \in V, \tag{42}$$

$$z_u + z_v + x_{uv} + x_{vu} \leq 2, \quad \forall \{u, v\} \in E, \tag{43}$$

$$2x_{uv} \leq z_u + z_v + y_u + y_v, \quad \forall (u, v) \in A, \tag{44}$$

$$\sum_{u \in N[v] \setminus \{r\}} y_u \geq 1, \quad \forall v \in V, \tag{45}$$

$$z, y \in \{0, 1\}^{|V|}, \ x \in \{0, 1\}^{|A \cup A_r|}, \ f \in [0, |V| - 2]^{|A \cup A_r|}, K \geq 0. \tag{46}$$

In model (F), constraint (34) ensures that variable K is equal to the sum of bridge and master nodes. Constraint (35) states that the number of arcs is one unit less than the number of nodes in the solution (excluding the arc leaving the root node r). Constraint (36) imposes that K flow units leave from r to the remaining nodes in a solution T. The flow conservation constraints (37) state that if a node $v \in V$ belongs to T, then it retains one unit of flow. It corresponds to the difference between the incoming and outgoing flows of node v. Constraints (38) impose that if an arc belongs to T, then some flow must traverse it; otherwise, its flow is null. Constraint (39) states that exactly one arc leaves the root node r. Unlike model (M), the solution arborescence T is imposed to be rooted at r. Consequently, constraints (40) impose that exactly one arc enters every other node in T. Notice that node r belongs to the negative neighborhood of every node in V. Constraints (41)–(44), and (45), have the same interpretation as for constraints (6)–(9), and (5) in model (M), respectively. Finally, constraints (46) are domain constraints for the decision variables. Essentially, model (F) differs from model (M) in the way we model a

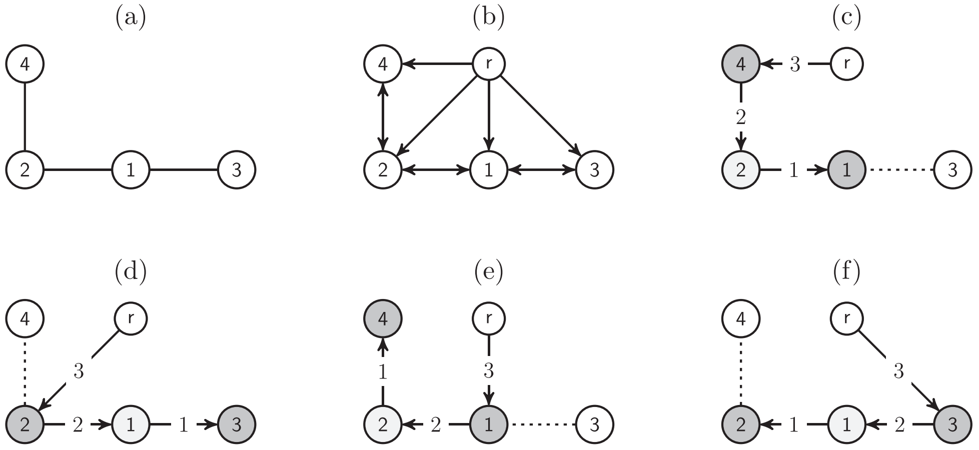

Fig. 2. In (c), (d), (e), and (f), we have all feasible arborescences rooted at the dummy node r for the digraph in (b) that is obtained from the graph in (a).

feasible solution by using flows (Gavish & Graves, 1978) to ensure the connectivity between the dummy root node r and the remaining nodes of the arborescence. Note that only one null cost arc will leave the dummy node r to reach exactly one of the original nodes in the solution of model (F). Thus, excluding this arc from an optimal solution, we obtain an arborescence of the same cost that is feasible for model (M). Therefore, it must be also of minimum cost for this model.

2.2.1. Valid inequalities for model (F)

We introduce valid inequalities for model (F) based on the flow structure. Initially, notice that at least two units of flow must arrive at every bridge node. One unit must be delivered to it, whilst a second one must be delivered to a master node that is adjacent to this bridge.

$$2z_v \leq \sum_{u \in N^-(v)} f_{uv}, \quad \forall v \in V. \tag{47}$$

We can strengthen arc flow lower bounds in (38) depending on the end-node u of a solution arc (u, v). If u is a bridge node, then at least one unit of flow must be sent to node v through this arc. If u is a master node, then v is a bridge node, and there exists another master node adjacent to and reached by v in the solution. Thus, at least two units of flow must traverse arc (u, v).

$$2x_{uv} - z_u \leq f_{uv} \quad \forall\ (u, v) \in A. \tag{48}$$

We also consider upper bounds on the arc flow variables. Notice that for a given node v, constraints (37) lead to $\sum_{u \in N^-(v)} f_{uv} = \sum_{u \in N^+(v)} f_{vu} + y_v + z_v$. If arc $(u, v) \in A$ belongs to the solution, then $x_{uv} = 1$ and $y_u + z_u = 1$. Since $f \geq \mathbf{0}$, we have that $f_{uv} \leq \sum_{k \in N^-(u)} f_{ku} - 1$. Thus,

$$f_{uv} + x_{uv}(|V| - 1) - \sum_{k \in N^-(u)} f_{ku} \leq |V| - 2, \quad \forall\, (u, v) \in A. \tag{49}$$

In Fig. 2, we show an example where the upper bound on f_{uv} in (49) is tight. Values in the arcs indicate their flow. Dotted lines connect a dominated node to a master one. The maximum arc flow referred to the edges of the graph in Fig. 2(a) is 2. The next inequalities in Propositions 4 and 5 do also provide upper bounds on the arc flow variables.

Similarly, as we do for model (M), we develop symmetry-breaking inequalities by imposing that each feasible solution should have a master node directly reached by the dummy node r, then we obtain

$$x_{rv} \leq y_v, \ \forall v \in V \ \text{ or, consequently, } x_{rv} + z_v \leq 1, \ \forall v \in V. \tag{50}$$

Proposition 4. *A valid upper bound on the arc flow is*

$$f_{uv} + 2(y_u - x_{ru}) + z_u \leq K - 1, \quad \forall\, (u, v) \in A. \tag{51}$$

Proof. When u is a master node reached by the arc leaving r, this bound is clearly valid. We consider three cases: (i) if u is a master node distinct of the arc's head leaving r, then there exists at least a bridge node u' between u and r as well as a master node u'' connected to u'. In this case, u', u'', and u retain one unit of flow. Therefore, the maximum flow on arc $(u, v) \in A$ is $K - 3$; (ii) if u is a bridge node not directly reached by r, then there exists at least a master node adjacent to u and distinct of v. Thus, the maximum flow leaving u by arc (u, v) is $K - 2$; and (iii) if u does not belong to T, then (51) becomes redundant. □

Proposition 5. *The arc flow is limited above according to*

$$f_{uv} + y_v + 2z_v - 2x_{ru} \leq K - 1, \quad \forall\, (u, v) \in A. \tag{52}$$

Proof. If v is a master, and u a bridge node not reached by r, then at least a flow unit must be delivered to the master node adjacent to u, and a second flow unit should be delivered to u. Thus, any flow incoming to v by arc (u, v) is at most $K - 2$. If v is a bridge node, we have two cases. First, if node u is reached by r, then it retains one unit of flow. Thus, at most, the $K - 1$ remaining flow units can traverse arc (u, v). Second, if node u is a master node not reached by r, then another node is reached by r. Also, there are at least a master and a bridge node in the path from r to u. These three nodes jointly retain three flow units. Thus, no more than $K - 3$ units of flow can leave from u to v through the arc (u, v). Finally, if v does not belong to the solution, then this inequality is also valid. □

Finally, notice that constraints (11), (14)–(17), (19)–(21), (24), (28), and (29) are also valid for model (F).

So far, we have discussed valid linear inequalities for models (M) and (F). However, using all of them is not effective because we observe that doing this requires larger CPU times for solving our instances. Based on an exhaustive set of numerical experiments (see Annex C), we suggest using the following combinations of linear inequalities for models (M) and (F). Hereafter, we denote them by $(M)^+$ and $(F)^+$, respectively.

Model $(M)^+$: constraints of model $(M)^+$ (11), (14)–(17), (19)–(24), (28)–(33).

Model $(F)^+$: constraints of model (F)+ (11), (14)–(17), (19)–(21), (24), (28), (29), (48).

2.3. Cutting-plane strategies

We also explore five exponential families of cuts for the problem. We implement them as user-cut callbacks in the CPLEX branch-and-cut (B&C) framework in every node of the B&C tree. We generate these cuts based on the digraph $\bar{G} \subseteq G$ induced by the non-null variables $\bar{x}$ of the linear relaxed solution $(\bar{z}, \bar{y}, \bar{x}, \bar{\pi})$ of a B&C node. The B&C algorithm requires large CPU times to find MCT solutions for our instances. Thus, in this work we discuss the B&C results only for model (M^{+}). Moreover, numerical experiments indicate to be more advantageous (w.r.t. CPU times) to generate one cut for each class of cuts for every call of their separation procedures.

The first two separation procedures are related to generalized sub-tour elimination constraints (GSEC) (12) and cut-set inequalities (13). We use a heuristic procedure to separate violated GSECs of $\bar{G}$ at every node of our B&C algorithm and separate violated cut-set inequalities (13) only when $\bar{G}$ has two or more components.

The self-explaining Algorithms 1 and 2 are procedures to separate violated inequalities (12) and (13), respectively, when $\bar{G}$ is a set $\bar{C}$ of disjoint components. The set of nodes $S \subset V$ inducing a component $S \in \bar{C}$ represents this component. $A(S)$ represents the set of arcs with both extremities in S, and $S^{\downarrow} = \{s_1, s_2, \dots, s_{|S|}\}$ denotes a non-increasing order of the nodes in S w.r.t. the sum of the $\bar{x}$ arc variables values adjacent to them. $S^{\downarrow}_k$, $k := 1, \dots, |S|$, denotes the subset of the first k elements of $S^{\downarrow}$.

Algorithm 1 has complexity $O(|V|^2)$. The complexity of Algorithm 2 is $O(|\bar{C}| \times |V| \times |\tilde{S}|)$, where $\tilde{S}$ is the set of nodes of the component in $\bar{C}$ with the maximum number of nodes. We check if $\tilde{S}$ and $V \setminus \tilde{S}$ are not dominating sets of G in $O(|V| \times |\tilde{S}|)$.

Algorithm 1 Separation procedure for GSECs (12).

```
Input: Graph $G$ and the set $\bar{C}$ of components of $\bar{G}$.
Output: A possible violated inequality(12) in $\bar{G}$.
for all $S \in \bar{C}$, where $3 \le |S| \le |V|/5$ do
    compute the weights of the nodes of $S$ and sort themto obtain $S^{\downarrow}$;
    for $k \leftarrow 3$ to $|S|$ do
        $\bar{x}(S^{\downarrow}_k) \leftarrow \sum_{(i,j) \in A(S^{\downarrow}_k)} \bar{x}_{ij}$;   $Sum_{\bar{Y}\bar{Z}}(S^{\downarrow}_k) \leftarrow \sum_{i \in S^{\downarrow}_k} \bar{y}_i + \sum_{i \in S^{\downarrow}_k} \bar{z}_i$;
        for $h \leftarrow 1$ to $k$ do
            if $\bar{x}(S^{\downarrow}_k) > Sum_{\bar{Y}\bar{Z}}(S^{\downarrow}_k) - \bar{y}_h - \bar{z}_h$ then
                add the corresponding inequality (12) for $S^{\downarrow}_k$ tomodel $(M^{+})$;
                return;   // We generate only one cut at a time
            end if
        end for
    end for
end for
```

Algorithm 2 Separation procedure for cut-set inequalities (13).

```
Input: Graph $G$ and the set $\bar{C}$ of components of $\bar{G}$.
Output: A possible violated inequality(13) in $\bar{G}$.
for all $S \in \bar{C}$ do
    if $S$ and $V \setminus S$ are not dominating sets of $G$ then
        add the corresponding inequality (13) for$(S, V \setminus S)$ to model$(M^{+})$;
        return;
    end if
end for
```

We also consider two families of valid inequalities for the problem which are related to the independent dominating set polytope Mahjoub & Mailfert (2006). First, we extend the idea of inequality (6) for every clique C of G, i.e., at most one node of a clique can be a master node

$$\sum_{j \in V(C)} y_j \le 1, \quad \text{for every clique } C \text{ of } G, \tag{53}$$

and the second family is related to odd cycle inequalities. The number of master nodes in every odd cycle C of G is limited above according to

$$\sum_{j \in V(C)} y_j \le \lfloor \frac{|V(C)|}{2} \rfloor, \quad \text{for every odd cycle } C \text{ of } G. \tag{54}$$

We can determine maximum-sized cliques in a unit disk graph in polynomial time with the algorithm proposed in Clark et al. (1990) which has a complexity of $O(|V(\bar{G})|^{4.5})$ w.r.t. the number of nodes of $\bar{G}$. A maximum (in size) clique of $\bar{G}$ not necessarily violates (53) (in node weights). To determine violated cliques of $\bar{G}$ based on node weights $\bar{y}$, we propose a heuristic procedure that allows separating them. The procedure is depicted in Algorithm 3.

Algorithm 3 Separation procedure for clique inequalities (53).

```
Input: Graph $G$, $\bar{y}$ values, and themaster nodes $\bar{Y}$ in $\bar{G}$.
Output: A possible violated clique inequality(53) in $\bar{G}$.
for $k \leftarrow 1$ to $|\bar{Y}| - 3$ do
    let $\bar{Y}^{>}(k) \leftarrow \{m_j \in \bar{Y} \mid j > k,\ \{m_k, m_j\} \in E\}$be an ordered set w.r.t. node indexes;
    set $sum(k) \leftarrow \bar{y}_{m_k} + \sum_{m_j \in \bar{Y}^{>}(k)} \bar{y}_{m_j}$;
    if $sum(k) > 1$ then
        initialize clique as $C \leftarrow \{m_k, v_1\}$;   // $v_1$ is thefirst element of $\bar{Y}^{>}(k)$
        set the weight of $C$ as $WeightIn(C) \leftarrow \bar{y}_{m_k} + \bar{y}_{v_1}$;
        set the weight of $\bar{Y}^{>}(k) \setminus C$ as$WeightOut \leftarrow sum(k) - \bar{y}_{m_k} - \bar{y}_{v_1}$;
        for $h \leftarrow 2$ to $|\bar{Y}^{>}(k)|$ do
            if $C \cup \{v_h\}$ forms a clique then
                $C \leftarrow C \cup \{v_h\}$;
                $WeightIn(C) \leftarrow WeightIn(C) + \bar{y}_{v_h}$;
                if $WeightIn(C) > 1$ then
                    for all $u \in V \setminus \bar{Y}$ do
                        if $C \cup \{u\}$ forms a clique then
                            expand clique $C \leftarrow C \cup \{u\}$;
                        end if
                    end for
                    add the clique inequality (53) w.r.t. $C$to model $(M^{+})$;
                    return;
                end if
            end if
            $WeightOut \leftarrow WeightOut - \bar{y}(v_h)$;
            if $WeightIn(C) + WeightOut \le 1$ then
                break loop;   // We cannot form a violated clique with the remainingnodes $v_h$
            end if
        end for
    end if
end for
```

We consider the set of master nodes of $\bar{G}$ as an ordered set $\bar{Y} = \{m_1, m_2, \dots, m_{|\bar{Y}|}\}$ w.r.t. node indexes, where $\bar{Y}$ is formed by the non-null entries of $\bar{y}$. The procedure intends to form a clique starting at a master node $m_k \in \bar{Y}$ and considering master nodes $m_j \in \bar{Y}$ adjacent to m_k in G, with $j > k$. We adapt the edge cover heuristic of Marzi et al. (2019) to obtain Algorithm 3. Its complexity depends on the number of masters of $\bar{G}$ and on the maximum degree, say $\bar{\Delta}$, of a master node of $\bar{G}$. Checking if a given closed neighborhood of master nodes of $\bar{G}$ induces a clique C can be done in $O(|\bar{\Delta}|^2)$. To expand a violated clique C with nodes in $V \setminus \bar{Y}$, this can be done in $O(|\bar{\Delta}| \times |V|)$. This leads to a final complexity of order $O(|\bar{Y}| \times |\bar{\Delta}|^2 \times |V|)$.

To separate odd cycles of $\bar{G}$ violating (54), we adopt the procedure proposed by de Vries & Perscheid (2021) based on shortest paths obtained from a product-graph $G[\bar{Y}] \times \bar{H}$, where $\bar{H} = (V_{\bar{H}}, E_{\bar{H}})$ is a graph with two nodes $V_{\bar{H}} = \{0, 1\}$ and one edge $E_{\bar{H}} = \{\{0, 1\}\}$. $G[\bar{Y}]$ is the sub-graph of G induced by the master nodes $\bar{Y}$. The weight of an edge $\{u, v\} \in E_{G[\bar{Y}]}$ of $G[\bar{Y}]$ is defined as $1 - \bar{y}_u - \bar{y}_v$. The weight of a cycle of this graph is the sum of its edge weights. The product graph $G[\bar{Y}] \times \bar{H}$ has the node set $\bar{Y} \times V_{\bar{H}}$ and the edge set $E_{G[\bar{Y}]\times\bar{H}} = \{\{(u, 0), (v, 1)\}, \{(v, 0), (u, 1)\} \mid \{u, v\} \in E_{G[\bar{Y}]}\}$. The weights of the edges in $E_{G[\bar{Y}]\times\bar{H}}$, associated with an edge $\{u, v\} \in E_{G[\bar{Y}]}$, are equal to the one of edge $\{u, v\}$. We compute shortest paths in $G[\bar{Y}] \times \bar{H}$ between nodes $\{u, 0\}$ and $\{u, 1\}$, with $u \in \bar{Y}$, by using the algorithm of Dijkstra (1959). These shortest paths induce shortest odd cycles in $G[\bar{Y}]$ (see Grötschel et al., 1988 for further details) and an odd cycle violates (54) if the sum of its edge weights is less than 1. By applying a depth first search (DFS) to any shortest path in $G[\bar{Y}] \times \bar{H}$, even if it is a closed walk in $G[\bar{Y}]$, we obtain the corresponding odd cycle in $G[\bar{Y}]$ (see de Vries & Perscheid, 2021).

Algorithm 4 is the procedure based on de Vries & Perscheid (2021) to separate odd cycles violating inequality (54). Its complexity $O(|\bar{Y}| \times |E_{G[\bar{Y}]}| \times log \bar{Y}|)$ depends basically on the number of nodes of $G[\bar{Y}] \times \bar{H}$ that is of order $O(|\bar{Y}|)$ and on the complexity $O(|E_{G[\bar{Y}]}| \times \log|\bar{Y}|)$ of a priority queue-based implementation of Dijkstra's algorithm.

Algorithm 4 Separation procedure for odd cycle inequalities (54).

```
Input: Graphs $G$ and $\bar{H}$, $\bar{y}$ values, and themaster nodes $\bar{Y}$ in $\bar{G}$.
Output: A violated odd cycle inequality (54) in $\bar{G}$ if it exists.
create the induced graph $G[\bar{Y}]$ and the product graph$G[\bar{Y}] \times \bar{H}$;
for all $u \in \bar{Y}$ do
    let $\mathcal{P}_u \leftarrow Dijkstra(G[\bar{Y}] \times \bar{H}, \{u, 0\}, \{u, 1\})$;  // shortest path from $\{u, 0\}$to $\{u, 1\}$ in $G[\bar{Y}] \times \bar{H}$
    determine the corresponding odd cycle $C$ in $G$, i.e., $C \leftarrow OddCycleDFS(\mathcal{P}_u)$;
    if $Weight(C) < 1$ then
        add the odd cycle inequality (54) w.r.t. $C$to model $(M^+)$;
        return;
    end if
end for
```

The last exponential family of cuts is related to flow cuts adapted to the MCT problem. Initially, in any feasible arborescence of model $(M)^+$, we have that

$$\sum_{i\in S,\, j\in V\setminus S} x_{ij} + \sum_{j\in V\setminus S, i\in S} x_{ji} \geq y_u + z_u + y_v + z_v - 1, \quad \forall v \in V, \forall S \subset V \setminus \{v\}, \forall u \in S, \tag{55}$$

are valid inequalities for model $(M)^+$ and can be separated in polynomial time. We observe that these inequalities are also valid for model $(F)^+$. They are sufficient to model a CT without using hop variables π or flow variables f in models (M) and (F), respectively. Although this leads to a new formulation, we use them as cuts in the CPLEX B&C algorithm. For this purpose, we construct a flow network $\bar{F} = (V_{\bar{F}}, A_{\bar{F}})$ based on the linear relaxed solution of a B&C node. The node set $V_{\bar{F}}$ is composed of nodes related to the non-null entries of $\bar{y}$ or $\bar{z}$. The arc set is $A_{\bar{F}} = \{(u, v), (v, u) \mid \bar{x}_{uv} > 0 \text{ or } \bar{x}_{vu} > 0,\ u, v \in V_{\bar{F}}\}$. The flow capacities of arcs (u, v) and (v, u) of A_F are both equal to $(\bar{x}_{uv} + \bar{x}_{vu})/2$. By doing this, we ensure that the maximum flow between two distinct nodes u and v of $V_{\bar{F}}$ has the same value in both directions (from u to v and from v to u) in $\bar{F}$. Consequently, the minimum cuts $(S, V \setminus S)$ and $(V \setminus S, S)$ associated with the maximum flows from u to v and from v to u, respectively, correspond to the same node partition as required in both summations on the left-hand side of (55). Therefore, it is sufficient to calculate the maximum flow between two nodes in one direction, say from u to v. If the double of its value is less than $\bar{y}_u + \bar{z}_u + \bar{y}_v + \bar{z}_v - 1$, then we have a violated inequality (55). In this case, we determine the minimum cut $(S, V \setminus S)$ associated with this maximum flow, with $u \in S$ and $v \in V \setminus S$, and add the corresponding inequality (55) to model $(M)^+$. Note that when $\bar{y}_u + \bar{z}_u + \bar{y}_v + \bar{z}_v - 1 < 0$ it is not necessary to calculate the maximum flow between nodes u and v because every flow has at least a null value. On the opposite, when $\bar{y}_u + \bar{z}_u + \bar{y}_v + \bar{z}_v - 1 > 0$, we observed (in our numerical experiments) a large number of violated flow cuts in the linear relaxed solutions of the B&C nodes. Consequently, the CPU times increase drastically when solving our instances. To reduce the number of these cuts in the B&C search tree, we arbitrarily calculate maximum flows only between pairs of master nodes in the relaxed node solutions for which the $\bar{y}$ values are equal to 1. The maximum flow algorithm we use is due to Ford & Fulkerson (1956).

In Algorithm 5, we present the procedure to separate violated flow cut inequalities (55). The complexity of the Ford & Fulkerson (1956) algorithm is known to be pseudo-polynomial of order $O(\,\Gamma \times |A_{\bar{F}}|\,)$, where Γ is a positive constant which depends on the arc capacities. This complexity dominates the one required to obtain the minimum cut based on the maximum flow by applying a DFS procedure. The number of pairs of distinct nodes in Y_1 is of order $O(|Y_1|^2)$. Thus, the complexity of Algorithm 5 is $O(\,\Gamma \times |A_{\bar{F}}| \times |Y_1|^2\,)$.

Algorithm 5 Separation procedure for flow cut inequalities (55).

```
Input: Graph $G$, graph $\bar{G}$, $\bar{x}$ and$\bar{y}$ values.
Output: A possible violated flow cut inequality(55) in $\bar{G}$.
determine the flow network $\bar{F}$;
define $\bar{Y}_1 \leftarrow \{v \in V \mid \bar{y}_v = 1\}$ and if $|\bar{Y}_1| < 2$, then return;
for all pairs of distinct master nodes $u, v \in \bar{Y}_1$ do
    $MaxFlow \leftarrow FordFulkerson(\bar{F}, u, v)$;  // maximum flow from $u$ to $v$ in $\bar{F}$
    if $MaxFlow < 0.5$ then
        determine node sets $S$ and $V \setminus S$ related to the minimum cutw.r.t. $MaxFlow$;
        add the flow cut inequality (55) to model $(M^+)$;
        return;
    end if
end for
```

We explore the above algorithms as user cut callbacks in the CPLEX B&C framework as follows. At every relaxed B&C node solution, we call them in the following sequence. First, we apply Algorithm 4 to check for violated odd cycle inequalities (54). Then, we apply Algorithm 3 to check for violated clique inequalities (53). Additionally, if the graph $\bar{G}$ has only one component, then we apply Algorithm 5 to separate flow cut inequalities (55). Otherwise, we apply in sequence Algorithms 2 and 1 to separate violated cutset and GSECs inequalities related to inequalities (13) and (12), respectively. We can apply an upper limit on the number of generated cuts. We notice, from our numerical experiments, that this sequence of calls to the B&C procedures allow to further save CPU times.

2.4. Benders' decomposition approaches

In this work, we explore the automatic Benders decomposition strategy of CPLEX to solve models $(M)^+$ and $(F)^+$. CPLEX gives the possibility of defining the master problem of the decomposition considering only integer variables while the sub-problem is defined by using the remaining continuous variables of each model. Putting all proposed inequalities for both models in the decomposition schemes is not usual. Thus, to give an idea of the master and sub-problem formulations related to the decomposition schemes for these models, we develop the master and sub-problem formulations only for models (M) and (F).

Concerning model (M), fixing the values of the integer variables $(\bar{x}, \bar{y}, \bar{z})$, the constraints involving the continuous variables π define the sub-problem of its decomposition scheme as

$$(S_M) \quad \min 0$$

$$s.t. \quad \pi_v - \pi_u \geq 2 - |V| + (|V|-1)\bar{x}_{uv} + (|V|-3)\bar{x}_{vu}, \quad \forall (u,v) \in A, \tag{56}$$

$$\pi_u \geq 0, \quad \forall u \in V, \tag{57}$$

$$-\pi_u \geq 2 - |V|, \quad \forall u \in V, \tag{58}$$

where $\bar{x}_{uv}$, for every arc $(u,v) \in A$, represents a constant parameter. We associate a dual non-negative variable α_{uv}, for all $(u,v) \in A$, γ_u and β_u, for all $u \in V$, with constraints (56)–(58), respectively. The Lagrangian function $L(\pi, \alpha, \gamma, \beta)$ related to (S_M) can be written as

$$L(\pi,\alpha,\gamma,\beta) = \sum_{(u,v)\in A} (2 - |V| + (|V|-1)\bar{x}_{uv} + (|V|-3)\bar{x}_{vu} \\ -\pi_v + \pi_u)\alpha_{uv} - \sum_{u\in V} \pi_u \gamma_u + \sum_{u \in V}((2-|V|) + \pi_u)\beta_u. \tag{59}$$

The dual of sub-problem (S_M) is $\inf_\pi L(\pi, \alpha, \gamma, \beta)$. It can be easily solved analytically, since a linear function is bounded below only when it is identically zero Boyd & Vandenberghe (2013). Consequently, we can impose that the coefficient of each π_v variable be equal to zero, for all $v \in V$. This allows to obtain the following equivalent dual problem for (S_M).

$$D(S_M) \max \sum_{(u,v)\in A} (2 - |V| + (|V|-1)\bar{x}_{uv} + (|V|-3)\bar{x}_{vu})\alpha_{uv} \\ +(2-|V|)\sum_{v \in V} \beta_v \tag{60}$$

$$s.t. \sum_{v:(u,v)\in A} (\alpha_{uv} - \alpha_{vu}) + \beta_u \geq 0, \quad \forall u \in V, \tag{61}$$

$$\beta \in [0, \infty)^{|V|}, \alpha \in [0,\infty)^{|A|}. \tag{62}$$

Based on the solution of problem $D(S_M)$, we define the master problem for model (M) as

$$(M_M) \min w \tag{63}$$

$$s.t. \ (2), (4) - (9) \ and \\ \sum_{(u,v)\in A} \alpha^t_{uv}(2 - |V| + (|V|-1)x_{uv} + (|V|-3)x_{vu}) \\ +(2-|V|)\sum_{u\in V}\beta^t_u \leq 0, \quad t \in \mathcal{R} \tag{64}$$

$$\sum_{(u,v)\in A} c_{uv}x_{uv} + \sum_{(u,v)\in A} \alpha^t_{uv}(2 - |V| + (|V|-1)x_{uv} + (|V|-3)x_{vu}) \\ +(2-|V|)\sum_{u\in V}\beta^t_u \leq w, \quad t \in \mathcal{V} \tag{65}$$

$$z, y \in \{0,1\}^{|V|}, x \in \{0,1\}^{|A|}, w \in \mathbb{R}_+, \tag{66}$$

where there exists a feasibility cut (64) associated with every ray $t \in \mathcal{R}$ related to unbounded solutions of $D(S_M)$ and an optimality cut (65) associated with every vertex $t \in \mathcal{V}$ related to limited optimal solutions of $D(S_M)$.

Concerning the decomposition approach for model (F), analogously as for model (M), given $(\bar{x}, \bar{y}, \bar{z})$, we determine the sub-problem for model (F) as

$$(S_F) \min 0$$

$$s.t. \sum_{v\in V}(\bar{y}_v + \bar{z}_v) = K, \tag{67}$$

$$\sum_{(u,v)\in A} \bar{x}_{uv} = K - 1, \tag{68}$$

$$\sum_{(r,v)\in A_r} f_{rv} = K, \tag{69}$$

$$\sum_{u\in N^-(v)} f_{uv} - \sum_{u \in N^+(v)} f_{vu} = \bar{y}_v + \bar{z}_v, \quad \forall v \in V, \tag{70}$$

$$\bar{x}_{uv} \leq f_{uv}, \quad \forall (u,v) \in A \cup A_r, \tag{71}$$

$$f_{uv} \leq (|V|-2)\bar{x}_{uv}, \quad \forall (u,v) \in A \cup A_r, \tag{72}$$

$$K \geq 0. \tag{73}$$

To obtain the dual of (S_F), we associate free dual variables σ_1, σ_2, σ_3, and φ with constraints (67)–(70), respectively. Then, we associate non-negative dual variables σ_4, τ, and δ with constraints (71)–(73), respectively. The dual of (S_F) is written as

$$D(S_F) \max \sum_{(u,v)\in A\cup A_r} \tau_{uv}\bar{x}_{uv} - (|V|-2)\sum_{(u,v)\in A\cup A_r}\delta_{uv}\bar{x}_{uv} - \sum_{v\in V}\varphi_v\bar{y}_v \\ - \sum_{v\in V}\varphi_v\bar{z}_v - \sigma_1(\sum_{v\in V}(\bar{y}_v + \bar{z}_v)) - \sigma_2 - \sigma_2\sum_{(u,v)\in A}\bar{x}_{uv} \tag{74}$$

$$s.t. \ \sigma_1 + \sigma_2 + \sigma_3 - \sigma_4 = 0, \tag{75}$$

$$\varphi_v - \varphi_u - \tau_{uv} + \delta_{uv} = 0, \quad \forall (u,v) \in A, \tag{76}$$

$$-\sigma_3 - \tau_{rv} + \delta_{rv} + \varphi_v = 0, \quad \forall v \in V, \tag{77}$$

$$-\tau_{vr} + \delta_{vr} - \varphi_v = 0, \quad \forall v \in V, \tag{78}$$

$$\sigma_1, \sigma_2, \sigma_3 \in \mathbb{R}, \ \varphi \in \mathbb{R}^{|V|}, \ \sigma_4 \in \mathbb{R}_+, \ \tau \in \mathbb{R}_+^{|A\cup A_r|}, \ \delta \in \mathbb{R}_+^{|A \cup A_r|}. \tag{79}$$

Finally, based on a limited or unbounded solution of the dual problem $D(S_F)$ w.r.t. a given $(\bar{x}, \bar{y}, \bar{z})$, the master problem of the Benders' decomposition for model (F) is

$$(M_F) \min w \tag{80}$$

$$s.t. \ (39) - (45), \ and \tag{81}$$

$$\sum_{(u,v)\in A\cup A_r} \tau^t_{uv}x_{uv} - (|V|-2) \sum_{(u,v)\in A\cup A_r} \delta^t_{uv}x_{uv} - \sum_{v\in V}\varphi^t_v y_v - \sum_{v\in V}\varphi^t_v z_v$$
$$-\sigma^t_1\left(\sum_{v\in V}(y_v+z_v)\right) - \sigma^t_2 - \sigma^t_2 \sum_{(u,v)\in A} x_{uv} \le 0, \quad t\in\mathcal{R}, \qquad (82)$$

$$\sum_{(u,v)\in A} c_{uv}x_{uv} + \sum_{(u,v)\in A\cup A_r} \tau^t_{uv}x_{uv} - (|V|-2) \sum_{(u,v)\in A\cup A_r} \delta^t_{uv}x_{uv} - \sum_{v\in V}\varphi^t_v y_v$$
$$-\sum_{v\in V}\varphi^t_v z_v - \sigma^t_1\left(\sum_{v\in V}(y_v+z_v)\right) - \sigma^t_2 - \sigma^t_2 \sum_{(u,v)\in A} x_{uv} \le w, \quad t\in\mathcal{V}, \qquad (83)$$

$$z, y \in \{0,1\}^{|V|}, x \in \{0,1\}^{|A\cup A_r|}, w \in \mathbb{R}_+. \qquad (84)$$

where constraints (82) and (83) represent feasibility and optimality cuts, respectively. As usual, to solve both master problems (M_M) and (M_F), we iteratively determine a ray (resp. vertex) from the set of extreme rays $\mathcal{R}$ (resp. set of vertices $\mathcal{V}$) if the solution of the corresponding sub-problem is unbounded (resp. limited).

3. Computational results

We implement models $(F)^+$ and $(M)^+$ in C++ in a 64 bits Linux Ubuntu system version 18.04 LTS and used the MIP solver of IBM ILOG CPLEX 12.8.1 in a computer Intel Octa-Core i7-7700 CPU 3.60-GigaHertz / 32 Gigabyte RAM. We adopt default parameters for the preprocessing phase of CPLEX, as well as for the use of generic cuts, the B&B node selection, the branching strategy. These models benefit from the automatic CPLEX Benders' decomposition algorithm to reduce their execution times. We also use the 'Full' default parameter for the Benders' decomposition strategy, where CPLEX automatically defines the master problem considering all integer variables, whilst it constructs each sub-problem by considering all the remaining continuous variables as it is usually performed in the classical Benders' decomposition approach. We also implement branch-and-cut (B&C) procedures as CPLEX user cut callbacks and run them at every node of the B&C tree.

We use 120 randomly-generated instances[1] ranging from 100 to 150 nodes. They are Euclidean simple connected graphs whose random node coordinates are generated in a 100 × 100 square meters box. Instances have $|V| \in \{100, 110, 120, 130, 140, 150\}$ homogeneous sensors. Sensing radius, in meters, are drawn from the interval $rad \in \{15, 20, 25, 30\}$. If the Euclidean distance between two nodes is not larger than the sensing radius, then there exists an edge connecting them. Edge density varies according to the instance sensing radius. The networks are very dense for larger sensing radii. We generate five instances for each combination of $|V|$ and rad. We consider only connected graphs.

We implement the cut-and-branch algorithm from Adasme et al. (2018) to obtain the corresponding minimum dominating tree (MDT) of our instances. The MDT solution value gives a lower bound (LB) on the optimal one of the minimum clustered tree (MCT).

3.1. Numerical results for models $(F)^+$ and $(M)^+$

We report average solution values w.r.t. MDT and MCT in Table 1. The legend in this table is as follows. 'Instances' are grouped by same number of nodes $|V|$ and sensing radius 'rad' given in meters. The group identifier has the format $|V|$-rad in the first column. For each group, the average number of edges in the second column is $\overline{|E|}$ and the average optimal solution values are in column $\overline{z}$. The next three columns related to MDT average results are $\overline{BB}$, $\overline{t(s)}$, and $\overline{LB}$. They show average values of the number of branch-and-bound nodes, CPU times (in seconds), and MDT costs, respectively. For each model $(F)^+$ and $(M)^+$, we report the average linear relaxed solution values $\overline{z^b_r}$ and $\overline{z_r}$ obtained without using valid inequalities and using them, respectively. The average CPU times to obtain the improved linear relaxed solutions are given in column $\overline{t_r(s)}$ (in seconds). We also report for these models average values of $\overline{BB}$ nodes, average CPU times $\overline{t(s)}$, and the relative average integrality gap in percentage in column $\overline{gap}$ which is calculated between the optimal and the improved linear relaxed solution of our instances for these models. CPU times equal to 0 means less than one second. In the last two columns 'Ratios', we present average relative differences, in percentage, between the corresponding CPU times and the improved linear relaxed values for each group of instances solved with models $(F)^+$ and $(M)^+$, respectively.

Table 1 reports average numerical results for our instances. All solutions w.r.t. models $(F)^+$ are optimal while two instances of the group 140-15 were not solved by model $(M)^+$ due to a CPLEX shortage of memory error. In this case, average values for this group do not consider these two instances. For some groups of instances, results w.r.t. average MDT optimal solutions do not consider instances for which an out of memory error (OM) occurred. For groups 150-30 and 110-30, we have the smallest (5.9%) and largest (17.5%) relative difference, in percentage, between average MDT solution values ($\overline{LB}$) and average MCT optimal solutions ($\overline{z}$). All average values with respect to both $\overline{z^b_r}$ and $\overline{z_r}$ for model $(M)^+$ are slightly smaller than for model $(F)^+$, where group 100-15 presents the largest ratio for $\overline{z_r}$, i.e., 0.5% (in bold). To see the impact of the proposed inequalities, for group 150-30 and model $(F)^+$, we observe the largest improvement from an average $\overline{z^b_r}$ value of 7.96 to an average $\overline{z_r}$ value of 136.61 (both values in bold), corresponding to an increase of 1, 616.0%. On the other hand, for group 100-15 and model $(M)^+$, we observe the smallest improvement from an average $\overline{z^b_r}$ value of 71.90 to an average $\overline{z_r}$ value of 332.01, corresponding to an increase of 361.8%. Group 110-30 presents the largest average relative integrality gaps for models $(F)^+$ and $(M)^+$ with values (in bold) of 31.1% and 31.4%, respectively. In terms of average CPU time ratios, for group 100-15, model $(M)^+$ requires 152.00 seconds on average (in bold), representing 62.5% more than that for model $(F)^+$. While for the remaining groups, model $(M)^+$ requires less average CPU times than model $(F)^+$. The average values of $\overline{BB}$ nodes is smaller for model $(F)^+$ for only five groups.

3.2. MCT and MDT solution structures and Benders' decomposition results

Table 2 reports average results for MCT and MDT topologies in terms of number of nodes and costs. It also reports average results for models $(F)^+$ and $(M)^+$ obtained by using the automatic Benders' decomposition approach of CPLEX. Its legend is self-explaining. We report the average number of nodes belonging to the MCT and MDT solution structures in the corresponding columns. For instance, we report the average number of master and bridge nodes in columns 'Masters' and 'Bridges', respectively, and their sum is in column 'Nodes MCT'. The average number of dominating nodes of the MDT is reported in column 'Nodes MDT'. For the decomposition approach related to models $(F)^+$ and $(M)^+$, we report the average numbers of $\overline{BB}$ nodes, the average numbers of Benders' cuts in column $\overline{Cuts}$, and the average CPU times $\overline{t(s)}$, in seconds, required to solve the instances of each group. For a given group of instances, values in bold indicate smaller values

[1] Available under request and at: https://data.mendeley.com/datasets/fzd5ndbxsy/1

Table 1
CPLEX average results for models $(F)^+$ and $(M)^+$ with all 120 instances grouped by number of nodes and sensing radius.

Instances			MDT solution			Model $(F)^+$						Model $(M)^+$						Ratios$^+$	
$\|V\|$-rad	$\overline{\|E\|}$	$\bar{z}$	$\overline{BB}$	$\overline{t(s)}$	$\overline{LB}$	$\overline{z_r^b}$	$\overline{z_r}$	$\overline{t_r(s)}$	$\overline{BB}$	$\overline{t(s)}$	$\overline{gap}$	$\overline{z_r^b}$	$\overline{z_r}$	$\overline{t_r(s)}$	$\overline{BB}$	$\overline{t(s)}$	$\overline{gap}$	$\overline{t(s)}$	$\overline{z_r}$
100-15	297.40	374.25	242,756.80	449.00	351.98	71.90	332.01	0.20	14,569.40	**158.00**	12.7	70.66	330.39	0.40	95,657.60	256.80	13.3	62.5	**0.5**
100-20	508.60	274.58	295,598.00	464.00	249.80	38.52	226.55	0.60	46,783.20	1,248.60	21.2	37.82	226.01	0.60	44,654.60	399.60	21.5	−68.0	0.2
100-25	764.80	210.19	37,727.40	157.20	190.70	20.55	167.32	0.80	14,009.40	1,491.20	25.6	20.28	167.00	0.80	19,985.60	413.00	25.9	−72.3	0.2
100-30	1,038.80	175.38	44,566.40	168.00	160.25	12.49	140.25	0.80	6,846.00	1,226.00	25.1	12.11	140.03	2.80	7,952.20	290.20	25.2	−76.3	0.2
110-15	372.60	357.67	344,698.00	989.40	333.12	67.82	313.02	0.00	37,824.20	528.60	14.3	66.99	312.42	0.40	55,716.20	259.00	14.5	−51.0	0.2
110-20	632.20	279.30	860,669.20	2,152.20	258.35	37.40	235.61	0.80	82,678.60	3,436.40	18.5	36.67	235.01	1.00	123,524.80	1,458.60	18.8	−57.6	0.3
110-25	901.00	213.67	141,371.60	685.40	191.52	21.07	168.83	1.00	21,366.00	2,695.00	26.6	20.67	168.59	1.40	44,268.60	1,131.60	26.7	−58.0	0.1
110-30	1,290.40	181.73	705,511.25	4,168.25	154.68	11.71	138.57	1.40	21,338.40	5,106.40	**31.1**	11.44	138.33	4.80	18,364.00	965.20	**31.4**	−81.1	0.2
120-15	438.40	349.00	801,018.60	6,332.00	326.30	59.02	303.91	0.60	47,886.00	1,417.80	14.8	58.13	303.05	0.60	96,933.80	760.40	15.2	−46.4	0.3
120-20	717.20	273.39	807,032.20	3,938.80	254.27	30.61	228.02	0.60	54,690.80	4,298.20	19.9	30.27	227.60	1.00	132,653.20	2,074.20	20.1	−51.7	0.2
120-25	1,115.80	209.22	186,065.33	1,365.67	187.94	14.99	167.83	1.00	34,123.40	7,350.20	24.7	14.79	167.61	4.40	42,717.60	1,703.60	24.8	−76.8	0.1
120-30	1,507.20	166.61	79,961.25	549.00	150.82	10.32	136.22	1.00	5,903.60	3,083.20	22.3	10.22	136.03	7.00	7,002.60	638.40	22.5	−79.3	0.1
130-15	494.80	355.84	1,225,595.40	12,230.00	332.83	62.46	306.51	0.60	149,516.60	5,742.80	16.1	61.78	305.97	0.80	469,776.80	4,310.60	16.3	−24.9	0.2
130-20	876.80	257.67	1,244,602.00	7,268.40	241.20	27.97	213.98	1.20	124,785.60	15,227.60	20.4	27.56	213.73	1.60	117,942.00	3,130.20	20.6	−79.4	0.1
130-25	1,298.80	199.91	473,248.00	4,285.50	188.52	15.41	167.09	1.40	12,000.80	4,340.80	19.6	15.27	166.89	6.40	14,171.00	1,062.20	19.8	−75.5	0.1
130-30	1,844.60	171.46	432,627.40	7,083.20	156.53	8.81	133.73	2.00	11,858.00	9,296.80	28.2	8.68	133.55	10.40	19,265.40	2,378.40	28.4	−74.4	0.1
140-15	615.00	348.87	441,644.60	7,281.60	320.19	57.08	293.73	0.60	333,914.20	18,569.60	18.8	56.38	293.28	1.00	638,959.67	9,212.67	19.0	−50.4	0.2
140-20	1,004.20	264.28	703,807.33	5,722.00	244.93	27.40	216.91	1.40	401,747.60	81,812.80	21.8	27.06	216.53	3.60	365,108.80	12,889.40	22.1	−84.2	0.2
140-25	1,502.60	212.75	84,850.00	859.75	190.08	16.68	169.81	1.80	58,938.60	32,282.00	25.3	16.49	169.55	8.80	82,353.60	7,428.80	25.5	−77.0	0.2
140-30	2,143.00	167.02	125,499.50	2,263.50	148.61	8.85	130.86	1.80	10,829.60	13,445.40	27.6	8.75	130.69	15.00	18,049.00	3,449.80	27.8	−74.3	0.1
150-15	686.20	361.42	1,193,482.33	33,403.00	332.99	51.40	307.74	1.00	395,922.40	44,240.80	17.4	50.91	307.02	1.20	451,042.20	11,639.00	17.7	−73.7	0.2
150-20	1,162.00	258.51	2,026,764.50	33,921.50	238.35	25.32	211.85	1.80	267,524.60	73,786.40	22.0	25.03	211.54	4.60	220,112.40	11,602.00	22.2	−84.3	0.1
150-25	1,738.20	201.33	1,300,479.75	21,911.50	174.75	14.84	163.96	2.00	25,958.20	19,416.20	22.8	14.75	163.74	13.40	70,860.80	7,393.80	23.0	−61.9	0.1
150-30	2,426.60	165.97	303,524.00	7,151.00	156.74	**7.96**	**136.61**	2.60	5,116.40	6,896.40	21.5	7.89	136.39	22.20	7,441.80	1,708.00	21.7	−75.2	0.2

Table 2
Average number of nodes in the MCT and MDT solutions and Benders' decomposition results for models $(F)^+$ and $(M)^+$ with instances grouped by number of nodes and sensing radius.

	MCT and MDT average number of nodes				Benders model $(F)^+$			Benders model $(M)^+$			Ratio$^+$ %
Instances	Masters	Bridges	Nodes MCT	Nodes MDT	$\overline{BB}$	$\overline{Cuts}$	$\overline{t(s)}$	$\overline{BB}$	$\overline{Cuts}$	$\overline{t(s)}$	$\overline{t(s)}$
100-15	20.20	15.40	35.60	**34.40**	11,018.80	**6.60**	**45.00**	84,824.00	632.80	136.20	−66.96
100-20	12.20	9.20	21.40	**18.80**	18,880.00	**7.20**	129.00	19,337.60	15.80	**124.80**	3.37
100-25	8.20	5.60	13.80	**12.40**	12,574.80	**2.60**	200.00	12,572.40	9.40	**193.00**	3.63
100-30	6.00	4.00	10.00	**9.80**	5,751.80	0.40	**151.00**	6,722.60	**0.00**	152.00	−0.66
110-15	19.40	15.80	35.20	**32.20**	26,239.80	**11.00**	108.20	24,942.80	66.20	**79.40**	36.27
110-20	12.40	9.00	21.40	**21.20**	42,482.80	**4.00**	432.20	36,494.00	45.20	**316.20**	36.69
110-25	8.60	5.00	13.60	**13.20**	27,163.80	**3.60**	567.00	22,357.60	4.40	**367.80**	54.16
110-30	6.20	4.00	**10.20**	10.25	12,503.20	0.60	**421.80**	11,510.40	**0.00**	438.00	−3.70
120-15	19.00	14.40	33.40	**31.80**	45,301.00	**11.40**	278.00	42,239.00	41.00	**209.40**	32.76
120-20	12.20	8.20	20.40	**20.20**	53,505.60	**8.40**	687.00	39,937.60	28.20	**478.80**	43.48
120-25	8.00	5.00	**13.00**	13.33	21,224.80	**2.60**	**582.00**	26,552.60	3.20	636.00	−8.49
120-30	5.80	3.20	9.00	**8.75**	7,068.00	0.60	**363.00**	7,664.40	**0.00**	378.60	−4.12
130-15	19.80	15.20	35.00	**33.60**	122,373.40	**25.00**	977.40	124,376.60	178.80	**605.00**	61.55
130-20	11.40	8.00	19.40	**19.00**	108,797.60	**9.00**	2,322.80	100,733.60	12.60	**1,255.00**	85.08
130-25	8.00	4.20	**12.20**	12.75	13,155.80	0.80	583.20	9,412.00	**0.00**	**491.60**	18.63
130-30	5.80	3.80	9.60	**9.20**	14,913.60	0.20	**947.20**	13,632.40	**0.00**	1,068.60	−11.36
140-15	20.40	14.80	35.20	**32.00**	395,283.60	**45.60**	5,987.20	1,407,748.60	593.40	**5,969.60**	0.29
140-20	12.20	9.20	21.40	**19.33**	194,397.40	**9.60**	6,034.80	131,748.40	26.40	**2,814.60**	114.41
140-25	8.40	5.00	**13.40**	13.75	44,435.60	1.00	**2,583.80**	53,438.20	**0.00**	2,947.00	−12.32
140-30	6.00	3.60	9.60	**9.00**	12,974.00	0.00	**1,343.00**	12,221.60	0.00	1,702.60	−21.12
150-15	21.20	15.40	36.60	**33.67**	461,451.40	**65.60**	22,227.20	634,546.40	113.40	**6,271.20**	254.43
150-20	12.20	7.80	20.00	**18.50**	166,891.40	**12.00**	8,822.40	211,856.20	401.20	**6,090.60**	44.85
150-25	8.20	4.80	13.00	**11.25**	33,672.20	0.80	2,561.00	27,708.80	**0.00**	**2,458.80**	4.16
150-30	5.60	4.00	**9.60**	9.75	7,215.60	0.00	**899.20**	6,235.00	0.00	928.20	−3.12

when comparing the average number of MDT and MCT nodes, and when comparing columns $\overline{Cuts}$ and $\overline{t(s)}$. CPLEX gives the sum of optimality and feasibility cuts related to the second (branch-and-cut) phase of the Benders' decomposition. In the first phase, CPLEX applies the Benders' decomposition to solve the linear relaxation of the given formulation. The number of Benders' cuts in this phase is not included in the total number of cuts of the second phase that CPLEX reports. It does not report optimality and feasibility cuts separately. In column 'Ratio', we give the relative difference, in percentage, between the corresponding average CPU times $\overline{t(s)}$.

In Table 2, we observe that group 140-30 presents the largest difference (equal to 3.2) between the average number of MCT and MDT solution nodes. There are differences of less than one node on average for 13 groups of instances. For five groups, the average number of MDT nodes is larger than the MCT one. As observed for the average MCT and MDT costs in Table 1, it seems that the MCT and MDT topology differences in number of nodes are not so significant. Concerning the Benders' decomposition results, all instances are solved to optimality with both models $(F)^+$ and $(M)^+$. Compared to average CPU times in Table 1, the largest reductions concerns group 140-20. This group presents reductions of average CPU times going from 81,812.80 and 12,889.40 seconds to 6,034.80 and 2,814.60 seconds for models $(F)^+$ and $(M)^+$, respectively. Average CPU times obtained with the Benders' decomposition approach are smaller for model $(M)^+$ for 15 groups, being group 150-15 the one with largest average CPU times ratio. In general, the automatic CPLEX Benders' decomposition presents a larger average number of cuts for model $(M)^+$ than for model $(F)^+$. We observe a similar behavior for the average number of B&B nodes. The maximum average number of Benders' cuts is obtained with the group 100-15 when using model $(M)^+$.

3.3. Branch-and-cut results

The next set of experiments in Table 3 concerns the use of our procedures to separate violated GSEC (12), Cut-set (13), Flow-cut (55), Clique (53), and Odd-cycle (54) inequalities. We explore these cuts in the CPLEX branch-and-cut approach as user cut callbacks according to the algorithms presented in Section 2.3. For convenience, we evaluate their use only for the 100 nodes instances (grouped by same sensing radius) using model $(M)^+$. We consider two situations in the B&C algorithm. In the first one, we allow CPLEX to generate up to 1001 cuts considering simultaneously five families of cuts. The corresponding average results for this situation are in the rows containing the label of the groups. In these rows, considering all instances of each group, we report average values for the number of each type of cut given in the columns with their respective names. Average values for the 'Total' of cuts, CPU times, and the number of CPLEX B&C (BC) nodes. In the second situation, we solve these instances considering individually each family of cuts. In this case, we do not impose a limit on the maximum number of cuts being generated. We report the average number of cuts for all five families in the rows labeled 'Individually' in their corresponding columns. The respective average CPU times, in seconds, are in the rows labeled '$t(s)$'. For the second situation, CPLEX presents out of memory error for one instance of group 100-15 and in this case average values do not consider this instance. Finally, we separate instances groups with dashed lines. We highlight (in bold), for each group of instances, the family of cuts with smallest average number of generated cuts among the five families and the instance-group presenting smallest values for the last three columns when we use jointly these cuts. Also, bold values point to the family of cuts requiring the smallest average number of cuts and smallest average CPU time to solve individually each group of instances.

Observe, in Table 3, that when using all families of cuts jointly, flow-cuts are most generated for groups 100-15, 100-20, and 100-25. For group 100-15, on average they correspond to $92,5\%$ of the total of cuts generated while for group 100-30 they correspond to $30,5\%$ of the average total of cuts. GSEC inequalities present the same average (4,8) for groups 100-15 and 100-20. They appear in the largest average number ($68,6\%$ of the total) for group 100-30. For this group, the average number of cut-set inequalities is very small (0.02%) while for group 100-15 they are the second most generated cuts. Clique cuts are the second most generated cuts for group 100-20. Odd-cycles are the third most generated cuts for

Table 3
Average B&C results using model $(M)^+$ for the 100 nodes instances grouped by sensing radius.

Instances	GSEC	Cut-set	Flow-cut	Clique	Odd-cycle	Total	$t(s)$	BC
100-15	**4.8**	44.2	925.8	18.2	8.0	1,001.0	596.4	38,485.4
Individually	20,595.0	528.6	5,248.4	**1.2**	17.0			
t(s)	1,456.5	**593.8**	1,900.8	12,386.6	1,635.5			
100-20	**4.8**	9.6	938.2	31.4	17.0	1,001.0	2,468.2	73,148.8
Individually	22,204.2	168.2	11,234.6	**3.2**	42.0			
t(s)	3,568.4	**2,498.8**	19,535.0	3,956.2	3,319.2			
100-25	320.6	3.0	605.4	**2.0**	30.4	961.4	982.4	27,521.8
Individually	1,555.6	10.6	3,163.0	**5.4**	55.2			
t(s)	**659.8**	780.4	1,778.2	778.4	825.8			
100-30	593.8	**0.2**	271.0	2.2	21.8	889.0	512.4	9,879.8
Individually	2,800.8	**1.0**	723.2	5.0	37.2			
t(s)	558.8	457.2	777.2	**418.6**	578.4			

groups 100-20, 100-25, and 100-30. All average CPU times for the four groups are larger than the corresponding values in Table 1, say 256.80, 399.60, 413.00, and 290.20 seconds, for groups 100-15, 100-20, 100-25, and 100-30, respectively.

Considering the individual use of these families of cuts in Table 3, groups 100-15 and 100-20 present the two largest average number of GSEC inequalities while group 100-30 presents the smallest average number of cut-set inequalities. The average number of clique inequalities is very small for all groups, being the maximum average value equal to 5.4 for group 100-25. The average CPU time increases drastically w.r.t. individual flow-cuts (resp. clique inequalities) when applied to group 100-20 (resp. 100-15). The use of exponential families of cuts in our B&C framework, jointly or individually, seems to drastically increase CPU times to solve our instances. We have applied distinct orders (from the one in Section 2.3) of calls to the separation procedures related to the above families of cuts, as well as generated more than one cut of a given family in every call of the separation algorithms. Nevertheless, with no success. This did not encourage us to apply these cuts to model $(F)^+$.

To close this section, for the reader interested in further details why we adopt some techniques discussed in this work to solve this problem, including the impact of the proposed inequalities for the linear relaxed and integer solutions of our models, we refer to the supplemental material in annex. Finally, we mention that the operating system aborts all runs of our algorithms when solving models (M) and (F) after reaching the available machine memory without finding a feasible solution for our instances after hours of execution time. We could not even solve the instances using the automatic Benders decomposition approach for these models. Consequently, we do not report numerical results for them.

4. Conclusions

We propose two mixed-integer programming models for designing minimum weighted clustered trees. More precisely, we refer to the hop- and flow-based models by (M) and (F), respectively. In the solution topology, master nodes determine an independent set structure and the master node of each cluster is at a distance of one hop of every other node of its cluster. Two neighboring clusters share a common bridge node. We propose a set of valid inequalities for models (M) and (F). They allow solving all our instances to optimality. The automatic CPLEX Benders' decomposition approach handles accordingly these models while reducing the CPU times required to solve our instances. Numerical experiments indicate that there is no expressive difference between solution costs and the number of nodes present in the solution of the MCT and the corresponding MDT problems. Finally, the use of the five exponential families of cuts explored in the CPLEX branch-and-cut framework to solve our instances did not give better results than solving models $(M)^+$ and $(F)^+$ directly with the CPLEX MIP solver or its Benders' decomposition approach.

Acknowledgments

The 'Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES)' (Finance Code 001) finances in part this research. Thanks for supporting this research to the Brazilian National Council for Scientific and Technological Development (CNPq grants 408084/2018-9, 303988/2021-5). Also, to the Chilean National Fund for Scientific and Technological Development (ANID/FONDECYT grant 11180107). We are very grateful to the anonymous referees for their valuable comments and suggestions.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ejor.2022.08.014.

References

Adasme, P., Andrade, R., Leung, J., & Lisser, A. (2018). Improved solution strategies for dominating trees. *Expert Systems with Applications, 100*, 30–40.
Adasme, P., Andrade, R., & Lisser, A. (2017). Minimum cost dominating tree sensor networks under probabilistic constraints. *Computer Networks, 112*, 208–222.
Akgün, I., & Tansel, B. c. (2011). New formulations of the hop-constrained minimum spanning tree problem via Miller–Tucker–Zemlin constraints. *European Journal of Operational Research, 212*(2), 263–276.
Ayaz, M., Ammad-uddin, M., Baig, I., & Aggoune, e. M. (2018). Wireless sensor's civil applications, prototypes, and future integration possibilities: A review. *IEEE Sensors Journal, 18*(1), 4–30.
Boyd, S., & Vandenberghe, L. (2013). *Convex optimization*. Cambridge University Press.
Clark, B. N., Colbourn, C. J., & Johnson, D. S. (1990). Unit disk graphs. *Discrete Mathematics, 86*(1–3), 165–177.
Desrochers, M., & Laporte, G. (1991). Improvements and extensions to the Miller—Tucker–Zemlin subtour elimination constraints. *Operations Research Letters, 10*, 27–36.
Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik, 1 (1)*, 269–271.
Ford, L. R., & Fulkerson, D. R. (1956). Maximal flow through a network. *Canadian Journal of Mathematics, 8*, 399–404.
Gavish, B., & Graves, S. C. (1978). The travelling salesman problem and related problems. *Technical report OR-078-78*. Massachusetts Institute of Technology.
Gendron, B., Lucena, A., da Cunha, A. S., & Simonetti, L. (2014). Benders decomposition, branch-and-cut, and hybrid algorithms for the minimum connected dominating set problem. *INFORMS Journal on Computing, 26*, 645–657.
Grötschel, M., Lovász, L., & Schrijver, A. (1988). *Geometric algorithms and combinatorial optimization*. Springer.
Khan, I., Belqasmi, F., Glitho, R., Crespi, N., Morrow, M., & Polakos, P. (2016). Wireless sensor network virtualization: A survey. *IEEE Communications Surveys and Tutorials, 18*(1), 553–576.
Mahjoub, A. R., & Mailfert, J. (2006). On the independent dominating set polytope. *European Journal of Combinatorics, 27*(4), 601–616.
Marzi, F., Rossi, F., & Smriglio, S. (2019). Computational study of separation algorithms for clique inequalities. *Soft Computing, 23*, 3013–3027.
Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM, 7*(4), 326–329.

Park, P., Ergen, S. C., Fischione, C., Lu, C., & Johansson, K. H. (2018). Wireless network design for control systems: A survey. *IEEE Communications Surveys and Tutorials, 20*(2), 978–1013.

Santos, A. C., Bendali, F., Mailfert, J., Duhamel, C., & Hou, K. M. (2009). Heuristics for designing energy-efficient wireless sensor network topologies. *Journal of Networks, 4*(6), 436–444.

Santos, A. C., Duhamel, C., & Belisário, L. S. (2016). Heuristics for designing multi--sink clustered WSN topologies. *Engineering Applications of Artificial Intelligence, 50*, 20–31.

Santos, A. C., Duhamel, C., Belisário, L. S., & Guedes, L. M. (2012). Strategies for designing energy-efficient clusters-based WSN topologies. *Journal of Heuristics, 18*(4), 657–675.

Vlajic, N., & Xia, D. (2006). Wireless sensor networks: To cluster or not to cluster? In *Proceedings of the international symposium on a world of wireless, mobile and multimedia networks* (pp. 258–268). Buffalo, New York: IEEE Computer Society.

de Vries, S., & Perscheid, P. (2021). A smaller extended formulation for the odd cycle inequalities of the stable set polytope. *Discrete Applied Mathematics, 303*, 14–21.