# Optimization of wireless sensor networks deployment with coverage and connectivity constraints

Sourour Elloumi, Olivier Hudry, Estel Marie, Agathe Martin, Agnès Plateau, Stephane Rovedakis

**Noname manuscript No.**
(will be inserted by the editor)

# Optimization of wireless sensor networks deployment with coverage and connectivity constraints

**Sourour Elloumi · Olivier Hudry · Estel Marie · Agathe Martin · Agnès Plateau · Stéphane Rovedakis**

2018 april the 18th / ?

**Abstract** Wireless sensor networks have been widely deployed in the last decades to provide various services, like environmental monitoring or object tracking. Such a network is composed of a set of sensor nodes which are used to sense and transmit collected information to a base station. To achieve this goal, two properties have to be guaranteed: *(i)* the sensor nodes must be placed such that the whole environment of interest (represented by a set of targets) is covered, and *(ii)* every sensor node can transmit its data to the base station (through other sensor nodes). In this paper, we consider the *Minimum Connected $k$-Coverage* (MC$k$C) problem, where a positive integer $k \geq 1$ defines the coverage multiplicity of the targets. We propose two mathematical programming formulations for the MC$k$C problem on square grid graphs and random graphs. We compare them to a recent model proposed by (Rebai et al 2015). We use a standard mixed integer linear programming solver to solve several instances with different formulations. In our results, we point out the quality of the LP-bound of each formulation as well as the total CPU time or the proportion of solved instances to optimality within a given CPU time.

Sourour Elloumi
ENSTA-ParisTech / UMA, 91762 Palaiseau, France
Tel.:+331 81 87 21 32
E-mail: sourour.elloumi@ensta-paristech.fr

Olivier Hudry
Télécom-ParisTech / LTCI, 46, rue Barrault, 75013 Paris, France
Tel.:+331 45 81 77 63 E-mail: olivier.hudry@telecom-paristech.fr

Estel Marie
Conservatoire National des Arts et Métiers / CEDRIC, EA 4629. 292 rue Saint-Martin, 75003 Paris, France
Tel.:+331 58808550 E-mail: estel.marie@cnam.fr

Agathe Martin
Conservatoire National des Arts et Métiers / CEDRIC, EA 4629. 292 rue Saint-Martin, 75003 Paris, France
Tel.:+331 58808546 E-mail: agathe.martin31@gmail.com

Agnès Plateau
Conservatoire National des Arts et Métiers / CEDRIC, EA 4629. 292 rue Saint-Martin, 75003 Paris, France
Tel.:+331 58808562 E-mail: aplateau@cnam.fr

Stéphane Rovedakis
Conservatoire National des Arts et Métiers / CEDRIC, EA 4629. 292 rue Saint-Martin, 75003 Paris, France
Tel.:+331 58802049 E-mail: stephane.rovedakis@cnam.fr

**Keywords** Wireless Sensor Networks · Grid Networks · Random Graphs · Sensor Deployment · Minimum Connected $k$-Coverage · Mixed Integer Linear Programming · Formulations

## 1 Introduction

In the last decades, wireless sensor networks have been widely deployed to achieve environmental monitoring and object tracking, e.g., seismic detection, fire detection or precision agriculture. A wireless sensor network is composed of a set of sensor nodes with limited memory and processing resources. Those nodes are equipped with a power supply and several kinds of sensors. Furthermore, each sensor node has a wireless interface which allows the communication with other sensors to exchange information. A sensor network is deployed in an environment where each sensor node has to periodically sense information of interest in its area. With the help of other sensors, each node has to transmit the collected data towards a base station (called *sink*). In the end, all the gathered data are used by the base station to take appropriate decisions. The sensing and the communication areas covered by a sensor are generally approximated using a disk defined by a radius, but other shapes have been considered in the literature like hexagon or oval. In the following, we denote by $R^{se}$ and $R^{co}$ respectively the sensing and communication radii of a sensor node. A major part of such networks is composed of sensor nodes with the same characteristics (homogeneous networks), however for some applications needs we can have a combination of nodes with different abilities (heterogeneous networks), that is with different communication radii, non battery and battery powered, or static and mobile nodes.

The majority of wireless sensor networks are deployed in a two-dimensional sensing area, but several works consider also a three-dimensional area to model for example an indoor deployment in a building (Chakrabarty et al 2002). The area monitored by a sensor network can be covered in part or entirely. In the latter case, all the areas have to be covered, while in the former case only a set of specific points called *targets* must be considered. In this case, the targets to cover can be positioned on the area following several patterns: a square grid, a triangular grid, a hexagon grid or randomly. In the remainder of this paper, we focus on the coverage of targets in a two-dimensional sensing field.

The constraints introduced by the deployment of wireless sensor networks imply an appropriate placement of the sensor nodes. The constraints taken into account for the location of the sensor nodes require the solution of a particular optimization problem. The coverage requirements of the field can be modelled by the classical *Minimum Dominating Set* (MDS) problem, which is NP-Hard in general graphs (Garey and Johnson 1979). Recently, Gonçalves et al. (Gonçalves et al 2011) have shown that computing the domination number of square grid graphs is a polynomial problem. Given a sensing graph $G = (V, E)$ where $V$ is a set of targets and $E$ is a set of edges representing the targets covered by the nodes (assuming sensor positions defined by $V$), a solution for the minimum dominating set problem is a set $S \subset V$ of minimum cardinality such that $\forall u \in V - S$, there is a sensor $v \in S$ which is a neighbour of $u$ in $G$. To tolerate sensor failures, some applications require a multi-coverage of the targets. This issue can be addressed by the minimum $k$-dominating set problem, where a positive integer $k$ defines the coverage multiplicity of the targets. Thus, in this case, we aim to find a minimum dominating set $S$ of $G$ such that every node not in $S$ is adjacent to at least $k$ vertices in $S$.

The *Minimum Connected Dominating Set* (MCDS) problem is a variant of the MDS problem that takes into account the connectivity constraint. This problem is also NP-Hard in general graphs (Garey and Johnson 1979). It is often used in wireless ad-hoc networks to construct and maintain a virtual backbone for message routing in the network (Das and Bharghavan 1997). A minimum connected dominating set of $G = (V, E)$ is a dominating set $S \subset V$ such that the subgraph induced by $S$ in $G$ is connected. The MCDS problem is related to the Maximum Leaf Spanning Tree (MLST) problem. Indeed, the sum of the cardinal of their respective optimal solutions is equal to $|V|$. The MLST problem consists in finding a spanning tree in $G$ with the maximum number of leaves among the spanning trees of $G$. The MLST problem has recently been proven to be APX-hard for cubic graphs (Bonsma 2012) and to be APX-hard for all $k-$regular graphs with any odd $k \geq 5$ (Reich 2016). Moreover, Guha and Khuller (Guha and Khuller 1998) showed that in general graphs the existence of an algorithm for finding the MCDS with approximation ratio $\alpha$ implies the existence of an algorithm for the MLST problem with approximation ratio $2\alpha$.

In the aforementioned problems, we assume that the sensing and communication radii are the same, i.e., $R^{se} = R^{co}$. However, there are applications in which these two radii are different, e.g., in the context of precision agriculture, humidity sensors have a small sensing radius of at most 3 to 4 meters (Roveti 2001) while the communication radius can be up to 100 meters. Rebai et al. (Rebai et al 2015) consider the same problem of sensor deployment achieving coverage and connectivity for different values of the two radii $R^{se}$ and $R^{co}$. The authors hint that the resolution complexity of this problem may depend on the links between $R^{co}$ and $R^{se}$.

In this paper, we consider the *Minimum Connected k-Coverage* (MC$k$C) problem, which is the same when $k = 1$ as the one studied by Rebai et al. The MC$k$C problem is a generalization of the MCDS problem, where we consider two distinct graphs to model the sensing and communication interactions for sensor placement. We suppose that the radii are integers with $R^{co} \geq R^{se}$. Let a wireless sensor network be defined as $\mathcal{R} = (G^{se}, G^{co})$ where $G^{se} = (X, A^{se})$ represents the sensing graph and $G^{co} = (X, A^{co})$ the communication graph. Both $G^{se}$ and $G^{co}$ are directed graphs, to model the fact that sensing and communication may not be performed in a bidirectional fashion. Moreover, we assume $G^{co}$ is a connected digraph, while it is not necessarily the case for the digraph $G^{se}$. The set of nodes $X$ includes the sink $t$ and the targets of the field (and also the locations where the sensors may be placed). We suppose as it is usual that the sink $t$ does not need to be covered and cannot send data to sensors. $A^{se}$ is a set of arcs $(i, j)$, $i \neq j$, connecting node $i$ to node $j$ (different from $t$) if the Euclidean distance $d(i, j)$ between them is no larger than the radius $R^{se}$. Similarly, $A^{co}$ is a set of arcs $(i, j)$, $i \neq j$, connecting node $i$ (different from $t$) to node $j$ if the Euclidean distance $d(i, j)$ between them is no larger than the radius $R^{co}$. We introduce the notion of $k$-*coverage* which is different from the notion of $k$-domination of $G^{se}$. The former requires that every target $v \in X$ is dominated by at least $k$ sensors in its *closed* sensing neighbourhood (i.e., the sensing neighbourhood of $v$ including $v$ itself), while the latter only imposes that every target $v \in X$, $v$ is not in the $k$-dominating set, is dominated by at least $k$ sensors in its *(open)* sensing neighbourhood. Given a sensor network $\mathcal{R} = (G^{se}, G^{co})$ and an integer $k$, a solution to the Connected $k$-Coverage problem is a set $S \subseteq X$ satisfying: *(i)* $S$ is a $k$-coverage set $S$ of $G^{se}$ such that every vertex of $S - \{t\}$ is adjacent to at least $k - 1$ other vertices of $S$ and every node in $X - S - \{t\}$ is adjacent to at least $k$ vertices in $S$, *(ii)* $S \cup \{t\}$ induces a connected subgraph of $G^{co}$. A connected $k$-coverage $S$ is minimum for the MC$k$C problem if and only if for every connected $k$-coverage $S'$ we have $|S| \leq |S'|$. For readability reasons, in the following sections when $k = 1$ the *Minimum Connected k-Coverage* (MC$k$C) problem will be denoted *Minimum Connected Coverage* (MCC) problem.

In this paper, we present two mathematical programs for the MCC and MC$k$C problems, that can also be applied to the MLST and MCDS problems. We compare them to a recent model proposed by Rebai et al. (Rebai et al 2015). We assume that the targets are deployed either via a grid with square pattern, denoted by *square grid* in the sequel, or randomly in a square area. This two kinds of deployment define planar graphs where $X$ represents the vertices (targets). In the case of grid graphs,

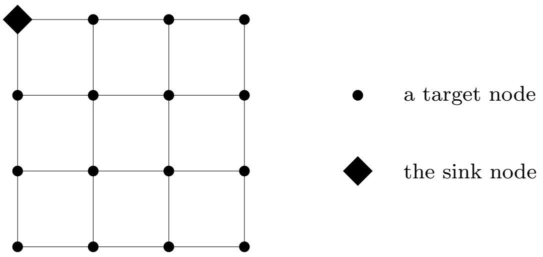

Fig. 1: Square grid with 4 rows and 4 columns.

the vertices are deployed following a square grid with unitary distances ($d = 1$) with $n$ rows and $n$ columns, i.e., $|X| = n^2$. Figure 1 shows such a grid. We choose to focus on square grid graphs, which are close to $4-$regular low density graphs, since these graphs appear to be among the most challenging instances in terms of exact resolution, and have up to now been the subject of very few specific works, as noticed in (Reich 2016). Random graphs are also considered in order to evaluate the performance of the mathematical programming formulations we propose for the MCC and MC$k$C problems on graphs where targets are not deployed following a regular pattern. In the rest of this paper, we consider that both $G^{se}$ and $G^{co}$ are connected digraphs.

The remainder of the paper is organized as follows: Section 2 presents related works. In the case where targets are located via a regular pattern, Section 3 proposes a sufficient condition on integer radii $R^{co}$ and $R^{se}$ according to which coverage of targets implies connectivity. After recalling a recent model dedicated to grid graphs, Section 4 describes two formulations that can be used for general graphs. Section 5 compares the three formulations through computational experiments on grids and random graphs. Section 5 extends our computational experiments to a generalization of the MCC problem: the Minimum Connected $k$-Coverage problem. Finally, Section 6 concludes this paper.

## 2 Related work

In (Lucena et al 2010), Lucena et al. propose enhanced formulations of the MLST problem related to a previous edge-vertex formulation and polyhedron investigations by Fujie (Fujie 2003), (Fujie 2004) and Gonçalves et al. (Gonçalves et al 2011). These two prior works did not provide experimental results of their formulations while the work by Lucena et al. shows detailed comparisons of several algorithms based on these formulations. Gendron et al. (Gendron et al 2014) later extended this study by presenting a branch-and-cut algorithm and a Benders decomposition algorithm based on the same formulation. So far they have obtained the best exact results since they succeeded in solving problems with 200 vertices and low edge density. The authors of (Reis et al 2015) propose a flow based formulation for the MLST problem. While this formulation is very simple, it gives roughly similar results to the ones exhibited by Lucena et al. (Lucena et al 2010).

Fan and Watson (Fan and Watson 2012) focus on several mixed integer linear formulations for the MCDS problem: Miller-Tucker-Zemlin (MTZ) formulation, Martin constraints, and flow formulations. They have succeeded in solving a 300 vertices instance with low density. Finally, Rebai et al. (Rebai et al 2015) propose a formulation inspired by path constraints and focus their studies on the MCC problem in square grid graphs. To our knowledge, the complexity of this problem is unknown in square grid graphs. In a recent work (Rebai et al 2016), Rebai et al. have considered another problem close to the previous one, called critical grid coverage problem, which is an

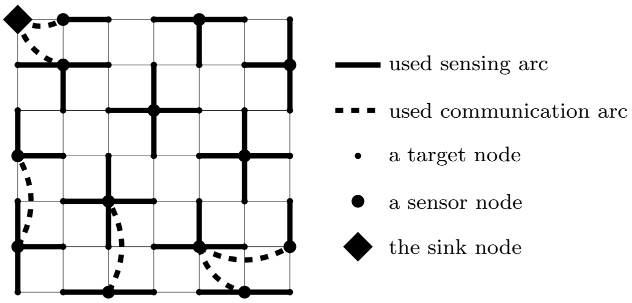

Fig. 2: A non-connected coverage solution in a 7x7 Grid with $R^{se} = 1$ and $R^{co} = 2$.

NP-Complete problem (Ke et al 2011). In this problem, only a given part of the square grid, called critical cells, have to be covered by the sensors and not all the grid. The authors proposed two mixed integer linear programming models, which are able to compute optimal solutions for square grid graphs of size up to $15 \times 15$.

## 3 When coverage implies connectivity

This section deals with a sufficient condition on integer radii $R^{co}$ and $R^{se}$ (with $R^{co} \geq R^{se}$) such that a dominating set for $G^{se}$ induces a connected subgraph in $G^{co}$ when targets are deployed according to a regular pattern. We mean by *d-regular pattern* that every target is at an Euclidean distance $d \leq R^{se}$ from all the other targets in its sensing neighbourhood.
Wang et al. (Wang et al 2003) have studied the total coverage of the area and the connectivity for the sensors placement with no restriction on $R^{se}$ and $R^{co}$. They have proven that when the communication radius $R^{co}$ is at least twice the sensing range $R^{se}$, the connectivity is automatically achieved when the total coverage of the area is reached. This result cannot be generalised to the case of a discrete set of targets coverage. Indeed, we prove in Propositions 1 and 2 that we need a larger communication radius than twice the sensing radius in order to get the connectivity ensured by the covering.

**Proposition 1** *In a wireless sensor network $\mathcal{R} = (G^{se}, G^{co})$ where targets are deployed according to a square grid, if $R^{co} = 2R^{se}$ then a dominating set for $G^{se}$ does not necessarily induce a connected subgraph in $G^{co}$.*

*Proof* We propose to build a dominating set for $G^{se}$ that does not induce a connected subgraph in $G^{co}$ when $R^{co} = 2R^{se}$.

Figure 2 presents an example of sensors placement which is a dominating set for a grid $7 \times 7$ when $R^{se} = 1$. When $R^{co} = 2R^{se} = 2$, this subset of sensors induces a subgraph of $G^{co}$ which is composed of 8 connected components and not a single one. This subgraph of $G^{co}$ is illustrated in Figure 2 where communication links between sensors are shown by dashed lines. □

In the next proposition, we propose an extension of Wang et al's result to the discrete case of target coverage when radii are integers and every pair of adjacent targets in $G^{se}$ are at distance $d$ from each other, i.e., we consider graphs where the targets are deployed following a $d$-regular pattern.

**Proposition 2** *In a wireless sensor network $\mathcal{R} = (G^{se}, G^{co})$ where targets are deployed according to a d-regular pattern, if $R^{co} \geq 2R^{se} + d$ then a dominating set of $G^{se}$ automatically induces a connected subgraph in $G^{co}$.*

*Proof* We suppose that $R^{co} = 2R^{se} + d$. Consider a sensor network $\mathcal{R} = (G^{se}, G^{co})$ and a dominating set $S$ of $G^{se}$. We suppose that the subgraph induced by $S$ is not connected in $G^{co}$. So, there exists at least one pair of adjacent targets in $G^{se}$ called $a$ and $b$ (such that $d(a, b) = d$) that are not covered by the same connected component. Consider $x$ (resp. $y$) a sensor that covers the target $a$ (resp. $b$) which belongs to a connected component named $C_a$ (resp. $C_b$).
We have $d(x, a) \leq R^{se}$ and $d(y, b) \leq R^{se}$. According to triangular inequalities respected by Euclidean distances, $d(x, y) \leq d(x, a)+d(a, b)+d(b, y)$. So, $d(x, y) \leq 2R^{se}+d$, which means that $C_a$ and $C_b$ are connected, this leads to a contradiction with our hypothesis. □

From Propositions 1 and 2, we can deduce the following result.

**Corollary 1** *In a wireless sensor network $\mathcal{R} = (G^{se}, G^{co})$ where targets are deployed according to a d-regular pattern, if $R^{co} \geq 2R^{se} + d$ then every solution for the k-coverage problem is also a solution for the connected k-coverage problem.*

*Proof* Observe first that every solution for the $k$-coverage problem in $G^{se}$ is also a solution for the 1-coverage problem (or dominating set problem). So, we can apply Proposition 2 to conclude that every solution induces a connected subgraph in $G^{co}$. □

Note that for grids, coverage implies connectivity when $R^{co} \geq 2R^{se} + 1$.

## 4 Three Mixed Integer Linear Programming formulations of MCC

We first recall the formulation in (Rebai et al 2015) which is dedicated to grids. We then present two different formulations based on single commodity flow and MTZ constraints respectively.

### 4.1 Model of Rebai et al. (Rebai et al 2015) (MIP1)

The model proposed in (Rebai et al 2015) is only defined for grid graphs. It relies on the distinction between two types of paths connecting a sensor to the sink: *direct* and *non-direct* connection paths. For any node $i$, consider the smallest rectangle on the grid that contains this node and the sink $t$. Let $V_i$ denotes the set of nodes located inside this rectangle. Let $G^{co}_d = (X, A^{co}_d)$ denotes the partial subgraph of $G^{co}$ where, for every $j' \in X \backslash \{t\}$, only the arcs $(j', j)$ where $j \in V_{j'}$ are kept. Figure 3 gives an example of $G^{co}_d$ for a $4 \times 4$ grid. A path from a node $i$ to the sink is said to be a *direct connection path* if it only uses arcs in $A^{co}_d$. Otherwise, if the path includes at least one arc that does not belong to $A^{co}_d$, it is said to be an *non-direct connection path.* Examples of direct and non-direct connection paths are illustrated by Figure 4. It was shown in (Rebai et al 2015) that the length (number of arcs) of a non-direct connection path in an optimal solution can be upper bounded by $P = \lceil \frac{2(n-1)}{R^{co}} \rceil$. When $R^{co} = 1$, $P$ can be viewed as the maximal Manhattan distance in $A^{co}$, i.e., the sum of the number of rows and the number of columns of the grid.

The model proposed in (Rebai et al 2015) is given in Formulation MIP1. For $i \in X$, the binary variable $z_i$ is equal to 1 if a sensor is placed on node $i$ and there exists a direct connection path in $G^{co}_d$ from this sensor to the sink, and 0 otherwise. For $i \in X$ and $p \in \{0, ..., P\}$, the binary variable $z^p_i$ is equal to 1 if a sensor placed on node $i$ is located at $p$ sensors from a direct connection path to the sink, and 0 otherwise. It can be noted that $z^0_i = 1$ means that there exists $(i, j) \in A^{co}$ such that $z_j = 1$. On the

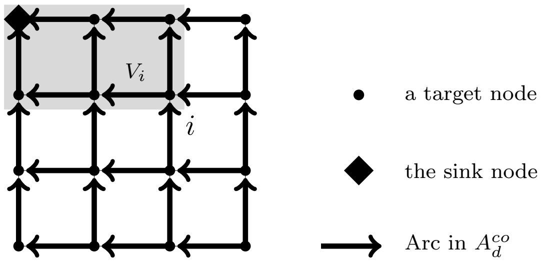

Fig. 3: Graph $G_d^{co}$ for a $4 \times 4$ grid with $R^{co} = 1$.

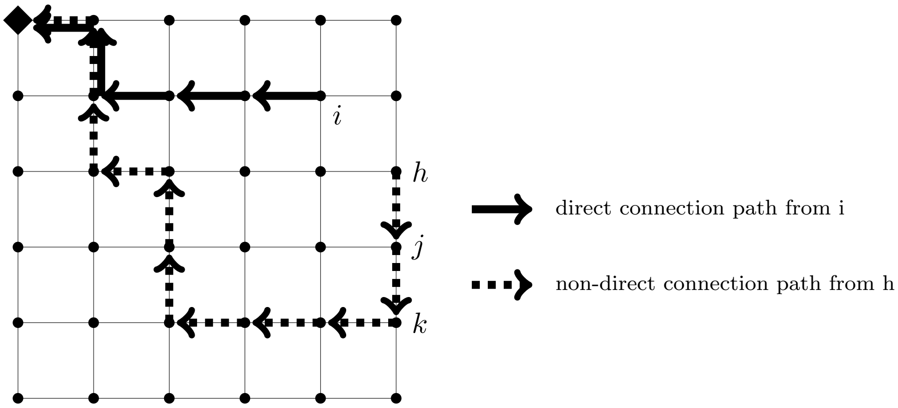
$i$
$h$
$j$
$k$
direct connection path from i
non-direct connection path from h

Fig. 4: Direct and non-direct connection paths with $R^{co} = 1$.

example given by Figure 4, $z_i = z_k = 1$, $z_j^0 = 1$ and $z_h^1 = 1$.

**Formulation MIP1**

$$\min \sum_{i \in X} \left( z_i + \sum_{p=0}^{P} z_i^p \right)$$

s.t.

$$\sum_{i:(i,j) \in A^{se}} \left( z_i + \sum_{p=0}^{P} z_i^p \right) \geq 1 \qquad j \in X \qquad (1)$$

$$\sum_{(j,i) \in A_d^{co}} z_i \geq z_j \qquad j \in X : (j,t) \notin A^{co} \qquad (2)$$

$$\sum_{i:(j,i) \in A^{co}} z_i \geq z_j^0 \qquad j \in X \setminus \{t\} \qquad (3)$$

$$\sum_{i:(j,i) \in A^{co}} z_i^{p-1} \geq z_j^p \qquad j \in X \setminus \{t\}, p = 1, ..., P \qquad (4)$$

$$z_i^p = 0 \qquad i : (i,t) \in A^{co}, p = 1, ..., P \qquad (5)$$

$$z_i \in \{0,1\},\ z_i^p \in \{0,1\} \qquad i \in X,\ p = 0, ..., P$$

The objective function measures the total number of sensors placed on the grid. Indeed, the expression $\left(z_i + \sum\limits_{p=0}^{P} z_i^p\right)$ is equal to 1 if a sensor is placed on node $i$ and 0 otherwise. Constraints (1) are covering constraints. They mean that each node of the grid must belong to the neighbourhood of at least one sensor. With Constraints (2) one can check that a sensor placed on node $j$ has a direct connection path to the sink if there exists a successor $i$ of $j$ in $G_d^{co}$ that has a direct connection path to the sink. Similarly, Constraints (3) are related to non-direct connection paths. A sensor placed on node $j$ has an non-direct connection path to the sink if a sensor placed on node $i$ belongs to its communication neighbourhood and has a direct connection path to the sink. Constraints (4) implement the induction condition on non-direct connection paths. A sensor placed on node $j$ is located at a distance of $p$ nodes from a direct connection path to the sink if at least one sensor in its communication neighbourhood is located at a distance of $(p-1)$ nodes from a direct connection path. Finally, Constraints (5) ensure that if a sensor is placed on a node $i$ which has the sink in its communication neighbourhood then this sensor is connected to the sink via the arc $(i,t)$.

For a $n \times n$ grid, the number of variables and the number of constraints are bounded by $O(|X|\sqrt{|X|})$.

Below, we present two models that can be used for general graphs. Let us first recall that a solution of the MCC problem can be viewed as a subset $S$ of $X$. Set $S$ represents the nodes where sensors will be placed. Each sensor in $S$ communicates with the sink $t$ through a path in $A^{co}$ joining nodes from $S$. Moreover, each node $i$ in $X \setminus \{t\}$ is covered by a sensor located either on $i$ or on one of its neighbours in $G^{se}$. This solution $S$ can be represented by a directed spanning tree of the grid rooted at $t$. In the in-tree, a node with no sensor is connected to a sensor by an arc in $A^{se}$ and then this sensor is connected to the sink $t$ through a path of sensors connected by communication arcs in $A^{co}$.

### 4.2 Single Commodity Flow Model (MIP2)

The Single Commodity Flow (SCF) model that we describe hereafter was introduced by Gavish (Gavish 1982) for solving the directed minimal spanning tree problem and recently used by Reis et al. for the MLST problem (Reis et al 2015) and Fan and Watson (Fan and Watson 2012) for the MCDS problem. It is based on the idea that every node of $X$ will send one unit of flow towards the sink node $t$. The flow is conveyed by arcs from $A^{se}$ and from $A^{co}$. A node with no sensor will send a unit of flow through an arc in $A^{se}$. A node with a sensor will gather incoming flows from $A^{se}$ and $A^{co}$ and send all through a unique outgoing arc in $A^{co}$. For this model (given in Formulation MIP2), we need to define the following variables:

- $x_i$, $i \in X$, is equal to 1 if a sensor is placed on node $i$ and 0 otherwise,
- $f_{ij}^{co}$, $(i,j) \in A^{co}$, is the amount of communication flow on arc $(i,j)$ from node $i$ to node $j$, if a sensor on $i$ communicates with a sensor on $j$. Variable $f_{ij}^{co}$ is a non-negative real number, upper-bounded by the number of nodes in $X \setminus \{t\}$, i.e., $n^2 - 1$,
- $f_{ij}^{se}$, $(i,j) \in A^{se}$, is the amount of sensing flow from node $i$ to node $j$, if no sensor is placed on $i$ and the sensing flow from $i$ is sent to the sensor on node $j$. It is equal to 0 otherwise. It follows from the definition that $0 \leq f_{ij}^{se} \leq 1$.

We also need to define the following notations. $M_i$ is any large enough integer that we fix to $(n^2 - 1)$ in our experiments. $\delta_i^-$ denotes the in-degree of $i$ in $A^{co}$.

**Formulation MIP2**

$$\min \quad \sum_{i \in X} x_i$$

s.t.

$$\sum_{i \in X:(i,t) \in A^{co}} x_i \geq 1 \qquad (6)$$

$$x_i + \sum_{j \in X:(i,j) \in A^{se}} x_j \geq 1 \qquad i \in X \setminus \{t\} \qquad (7)$$

$$\sum_{j \in X:(j,t) \in A^{co}} f_{jt}^{co} = n^2 - 1 \qquad (8)$$

$$x_i + \sum_{j \in X:(i,j) \in A^{se}} f_{ij}^{se} = 1 \qquad i \in X \setminus \{t\} \qquad (9)$$

$$\sum_{j \in X:(j,i) \in A^{co}} f_{ji}^{co} + \sum_{j \in X:(j,i) \in A^{se}} f_{ji}^{se} + x_i = \sum_{j \in X:(i,j) \in A^{co}} f_{ij}^{co} \qquad i \in X \setminus \{t\} \qquad (10)$$

$$\sum_{j \in X:(i,j) \in A^{co}} f_{ij}^{co} \leq M_i x_i \qquad i \in X \setminus \{t\} \qquad (11)$$

$$\sum_{j \in X:(j,i) \in A^{se}} f_{ji}^{se} \leq (\delta_i^- - 1) x_i \qquad i \in X \setminus \{t\} \qquad (12)$$

$$x_i \in \{0,1\} \qquad i \in X$$

$$0 \leq f_{ij}^{co} \leq n^2 - 1, f_{ij}^{co} \in \mathbb{R} \qquad (i,j) \in A^{co}$$

$$0 \leq f_{ij}^{se} \leq 1, f_{ij}^{se} \in \mathbb{R} \qquad (i,j) \in A^{se}$$

With Constraints (6), the coverage of the sink node $t$ by one of its predecessor nodes in the communication graph is satisfied. The coverage of the other nodes is satisfied by Constraints (7). Constraint (8) expresses that the incoming communication flow at the sink node is the aggregation of flows sent by each node in $X \setminus \{t\}$. Constraints (9) ensure that if a sensor is placed on node $i$ ($x_i = 1$), the sensing outflow from $i$ in $A^{se}$ is 0. But, if no sensor is placed on $i$ ($x_i = 0$), node $i$ sends one unit of flow on a unique sensing arc in $A^{se}$. Constraints (10) ensure the connectivity of the solution by flow conservation: if no sensor is placed on node $i$, the incoming – communication and sensing – flow is equal to outgoing communication flow. However, if a sensor is located on node $i$, the outgoing communication flow is equal to the incoming flow plus one. Constraints (11) ensure that if no sensor is placed on node $i$ then its outgoing communication flows are equal to 0. Finally, Constraints (12) mean that if no sensor is placed on node $i$ then its incoming sensing flow is equal to 0. On the contrary, if a sensor is placed on node $i$ then the incoming sensing flow is the number of non-sensor nodes that use $i$ and this number is limited by $(\delta_i^- - 1)$.

The number of variables is bounded by $O(|A^{co}|)$ (since we make the assumption $R^{se} \leq R^{co}$) and the number of constraints by $O(|X|)$.

## 4.3 Formulation based on Miller-Tucker-Zemlin Model (MIP3)

Here, we change the way of handling the connectivity requirement. The model of this subsection is inspired from the Miller-Tucker-Zemlin (MTZ) formulations of spanning trees and other graph applications. MTZ constraints have been initially used to solve the travelling salesman problem (Miller et al 1960).

For this model (given in Formulation MIP3), we need to define the following variables:

- $x_i$, $i \in X$, is the same variable as above,
- $y^{co}_{ij}$, $(i,j) \in A^{co}$, is also a binary variable, equal to 1 if and only if a communication link from $i$ to $j$ is used from a sensor placed on node $i$ to a sensor placed on node $j$.
- $y^{se}_{ij}$, $(i,j) \in A^{se}$, is a binary variable, equal to 1 if and only if a node $i$, with no sensor, is covered by a sensor placed on node $j$ through the arc $(i,j)$. With these definitions of the $y^{co}$ and $y^{se}$ variables, the set of arcs such that one of these variables is equal to 1 should build a spanning oriented tree of $X$, rooted at $t$.
- $L_i$, $i \in X$, counts the number of sensors in the path of sensors from $i$ to the sink node if a sensor is placed on node $i$. If $L_t = 0$, $L_i$ can be defined as a non-negative real number.

**Formulation MIP3**

$$\min \quad \sum_{i \in X} x_i$$

s.t.

Constraints (6), (7)

$$\sum_{j \in X:(i,j) \in A^{co}} y^{co}_{ij} = x_i \qquad i \in X \setminus \{t\} \qquad (13)$$

$$x_i + \sum_{j \in X:(i,j) \in A^{se}} y^{se}_{ij} = 1 \qquad i \in X \setminus \{t\} \qquad (14)$$

$$\sum_{(j,i) \in A^{se}} y^{se}_{ji} + \sum_{(j,i) \in A^{co}} y^{co}_{ji} \leq (\delta^-_i - 1) x_i \qquad i \in X \setminus \{t\} \qquad (15)$$

$$L_t = 0 \qquad (16)$$

$$L_i \geq L_j + 1 - (n^2 - 1)(1 - y^{co}_{ij}) \qquad (i,j) \in A^{co} \qquad (17)$$

$$L_i \geq 0 \qquad i \in X$$

$$x_i \in \{0,1\} \qquad i \in X$$

$$y^{co}_{ij} \in \{0,1\} \qquad (i,j) \in A^{co}$$

$$y^{se}_{ij} \in \{0,1\} \qquad (i,j) \in A^{se}$$

Constraints (13) say that the number of outgoing communication arcs from a non-sink node $i$ is equal to 1 if $i$ receives a sensor and 0 otherwise. Constraints (14) say that the number of outgoing sensing arcs from a non-sink node $i$ is equal to 0 if $i$ receives a sensor and 1 otherwise. Observe that Constraints (14) and (9), as well as variables $y^{se}_{ij}$ and $f^{se}_{ij}$, are actually the same. Constraints (15) ensure that a non-sink node $i$ has incoming arcs only if a sensor is placed on $i$ and that the number of incoming arcs of $i$ is limited by its in-degree minus one. Constraints (17) are the original Miller-Tucker-Zemlin constraints to guarantee that the solutions are directed subtrees rooted at $t$ which span selected sensors in $G^{co}$.

The number of variables and the number of constraints are bounded by $O(|A^{co}|)$.

## 5 Comparison of the three MIP formulations

The objective of our computational experiments is to compare the formulations on two kinds of instances : grid sensor networks and randomly generated graphs. We coded the MIP formulations by use of the AMPL mathematical programming modeller (Fourer et al 1993) and solve the mathematical programming problems by use of `Cplex12.6.2` (IBM-ILOG 2014) with a time limit of 1 hour. All our instances have been tested on an Intel(R) Xeon(R) CPU E5-2680 v3 2.50GHz with 48 CPU and with 64 GB of RAM.

### 5.1 Results for the MCC problem

#### *5.1.1 Grid sensor networks*

All the considered instances in this section are grids with $n$ rows and $n$ columns, denoted $G_n$, with different values of $R^{co}$ and $R^{se}$. The sink node $t$ is located at the left high corner.

We performed computational experiments on several instances of $G_n$ with $n = 6$ to 15, $R^{se}$ varying from 1 to 3 and $R^{se} \leq R^{co} \leq 2R^{se}$ (see Proposition 2 in Section 3). Table 1 summarizes the number of decision variables and constraints for each formulation MIP1, MIP2, and MIP3, as described in the previous section. The number of variables of MIP2 and MIP3 depends on $|A^{co}|$, in other words it depends on the value of $R^{co}$. We can also observe that the number of constraints of MIP3 actually depends on $|A^{co}|$. Moreover, the out-degree of a node $x$ in $G^{co}$, denoted by $\delta^+_x$, is upper bounded by $(2R^{co} + 1)^2$ (the number of nodes in the square of side length $2R^{co}$ which contains the circle of radius $R^{co}$). Thus, $|A^{co}|$ is bounded by $O(|X| \times (R^{co})^2)$.

| Number of | MIP1 | MIP2 | MIP3 |
|---|---|---|---|
| Variables | $O(\|X\|\sqrt{\|X\|})$ | $O(\|X\|(R^{co})^2)$ | $O(\|X\|(R^{co})^2)$ |
| Constraints | $O(\|X\|\sqrt{\|X\|})$ | $O(\|X\|)$ | $O(\|X\|(R^{co})^2)$ |

Table 1: Number of variables and constraints for each formulation

Our results are detailed in Table 2 as well as in Table 3 where the first column *Instance* gives the characteristics of the instance in the format $n\_R^{se}\_R^{co}$. The best results are emphasized in bold.

In Table 2, the next columns give, for each formulation MIP1, MIP2, and MIP3, $(i)$ the *bound* computed as the optimal value of the continuous relaxation of the formulation, obtained by solving specifically the LP-relaxation of the model and $(ii)$ the initial gap, denoted by $gap_i$, between the best known solution value, denoted *bkn*, computed by one of the three formulations and the *bound* as a percentage, i.e., precisely, $\frac{bkn-bound}{bkn} * 100$.

For instances, when $R^{se} = R^{co} = 1$, we can observe that MIP3 always provides the best bound by continuous relaxation. The average gap on these 10 instances is equal to about 37.2% with MIP1, 26.5% with MIP2, and 8.6% with MIP3. When $R^{se} = 1$ and $R^{co} = 2$, MIP1, MIP2 and MIP3 provide similar bounds with an average gap of 22.6%. Again for $R^{se} = R^{co} = 2$, MIP3 provides globally better bounds. However, for the remaining instances with $(R^{se}, R^{co}) = (2, 3)$, $(3, 3)$, and $(3, 4)$, the difference between the three bounds is not enough significant while we observe that the bounds become stronger.

In Table 3, we provide the results of the branch-and-bound phases of the three formulations. For each formulation MIP1, MIP2, and MIP3, we give $(i)$ the CPU time

|  | MIP1 |  | MIP2 |  | MIP3 |  |
|---|---|---|---|---|---|---|
| Instance | bound | $gap_i$ | bound | $gap_i$ | bound | $gap_i$ |
| 6_1_1 | 9 | 35.7 | 11 | 21.4 | 13 | **7.1** |
| 7_1_1 | 12 | 40 | 14 | 30 | 18 | **10** |
| 8_1_1 | 15 | 42.3 | 18 | 30.7 | 23 | **11.5** |
| 9_1_1 | 19 | 36.6 | 22 | 26.6 | 28 | **6.7** |
| 10_1_1 | 23 | 41.0 | 27 | 30.7 | 35 | **10.3** |
| 11_1_1 | 27 | 42.5 | 33 | 29.7 | 42 | **10.6** |
| 12 _1_1 | 32 | 38.4 | 39 | 25 | 49 | **5.7** |
| 13_1_1 | 37 | 42.1 | 45 | 29.6 | 58 | **9.4** |
| 14_1_1 | 43 | 41.8 | 51 | 31.1 | 67 | **9.5** |
| 15_1_1 | 49 | 39.5 | 59 | 27.2 | 76 | **6.2** |
| 6_1_2 | 9 | **18.1** | 9 | **18.1** | 9 | **18.1** |
| 7_1_2 | 12 | **20** | 12 | **20** | 12 | **20** |
| 8_1_2 | 15 | **21.0** | 15 | **21.0** | 15 | **21.0** |
| 9_1_2 | 19 | **20.8** | 19 | **20.8** | 19 | **20.8** |
| 10_1_2 | 23 | **20.6** | 23 | **20.6** | 23 | **20.6** |
| 11_1_2 | 27 | **22.9** | 27 | **22.9** | 27 | **22.9** |
| 12_1_2 | 32 | **23.8** | 32 | **23.8** | 32 | **23.8** |
| 13_1_2 | 37 | **24.4** | 37 | **24.4** | 37 | **24.4** |
| 14_1_2 | 43 | **23.2** | 43 | **23.2** | 43 | **23.2** |
| 15_1_2 | 49 | **24.6** | 49 | **24.6** | 49 | **24.6** |
| 6_2_2 | 4 | 42.8 | 5 | **28.5** | 5 | **28.5** |
| 7_2_2 | 6 | **25** | 6 | **25** | 6 | **25** |
| 8_2_2 | 7 | 36.3 | 7 | 36.3 | 8 | **27.2** |
| 9_2_2 | 9 | **30.7** | 9 | **30.7** | 9 | **30.7** |
| 10_2_2 | 10 | 41.1 | 11 | **35.2** | 11 | **35.2** |
| 11_2_2 | 12 | 33.3 | 13 | **27.7** | 13 | **27.7** |
| 12_2_2 | 14 | 39.1 | 15 | **34.7** | 15 | **34.7** |
| 13_2_2 | 16 | 38.4 | 17 | 34.6 | 18 | **30.7** |
| 14_2_2 | 19 | 38.7 | 19 | 38.7 | 20 | **35.4** |
| 15_2_2 | 21 | 43.2 | 22 | 40.5 | 23 | **37.8** |
| 6_2_3 | 4 | **0** | 4 | **0** | 4 | **0** |
| 7_2_3 | 6 | **0** | 6 | **0** | 6 | **0** |
| 8_2_3 | 7 | **12.5** | 7 | **12.5** | 7 | **12.5** |
| 9_2_3 | 9 | **0** | 9 | **0** | 9 | **0** |
| 10_2_3 | 10 | **16.6** | 10 | **16.6** | 10 | **16.6** |
| 11_2_3 | 12 | **14.3** | 12 | **14.3** | 12 | **14.3** |
| 12_2_3 | 14 | **12.5** | 14 | **12.5** | 14 | **12.5** |
| 13_2_3 | 16 | **15.7** | 16 | **15.7** | 16 | **15.7** |
| 14_2_3 | 19 | **13.6** | 19 | **13.6** | 19 | **13.6** |
| 15_2_3 | 21 | **19.2** | 21 | **19.2** | 21 | **19.2** |
| 6_3_3 | 3 | **0** | 3 | **0** | 3 | **0** |
| 7_3_3 | 4 | **0** | 4 | **0** | 4 | **0** |
| 8_3_3 | 4 | **0** | 4 | **0** | 4 | **0** |
| 9_3_3 | 4 | 20 | 5 | **0** | 4 | 20 |
| 10_3_3 | 5 | **28.5** | 5 | **28.5** | 5 | **28.5** |
| 11_3_3 | 6 | 25 | 7 | **12.5** | 7 | **12.5** |
| 12_3_3 | 8 | **20** | 8 | **20** | 8 | **20** |
| 13_3_3 | 9 | **25** | 9 | **25** | 9 | **25** |
| 14_3_3 | 9 | 30.7 | 10 | **23.0** | 9 | 30.7 |
| 15_3_3 | 10 | **37.5** | 10 | **37.5** | 10 | **37.5** |
| 6_3_4 | 3 | **0** | 3 | **0** | 3 | **0** |
| 7_3_4 | 4 | **0** | 4 | **0** | 4 | **0** |
| 8_3_4 | 4 | **0** | 4 | **0** | 4 | **0** |
| 9_3_4 | 4 | **0** | 4 | **0** | 4 | **0** |
| 10_3_4 | 5 | **16.6** | 5 | **16.6** | 5 | **16.6** |
| 11_3_4 | 6 | **14.2** | 6 | **14.2** | 6 | **14.2** |
| 12_3_4 | 8 | **0** | 8 | **0** | 8 | **0** |
| 13_3_4 | 9 | **0** | 9 | **0** | 9 | **0** |
| 14_3_4 | 9 | **10** | 9 | **10** | 9 | **10** |
| 15_3_4 | 10 | **16.6** | 10 | **16.6** | 10 | **16.6** |

Table 2: Continuous relaxation bounds and gaps for the three formulations on grid sensor networks

| | MIP1 | | MIP2 | | MIP3 | |
|---|---|---|---|---|---|---|
| Instance | CPU / $gap_f$ | nb nodes | CPU / $gap_f$ | nb nodes | CPU / $gap_f$ | nb nodes |
| 6_1_1 | 7.2s | 8969 | 0.41s | 1274 | **0.35s** | 0 |
| 7_1_1 | 839s | 2M | 28.5s | 420K | **3.6s** | 17K |
| 8_1_1 | 25.8% | 4M | 12.0% | 13M | **89.7s** | 447K |
| 9_1_1 | 29% | 1M | 748.8s | 975K | **26.2s** | 54K |
| 10_1_1 | 39% | 283K | 14.7% | 854K | **4.1%** | 3M |
| 11_1_1 | 38% | 513K | 18.5% | 854K | **8.1%** | 1M |
| 12_1_1 | 38% | 160K | 17.0% | 885K | **3500s** | 2M |
| 13_1_1 | 42% | 44K | 22.1% | 651K | **8.2%** | 1M |
| 14_1_1 | 42% | 36K | 24.0% | 407K | **8.4%** | 1M |
| 15_1_1 | 40% | 17K | 23.4% | 283K | **4.3%** | 529K |
| 6_1_2 | 30s | 121K | **0.34s** | 0 | 0.53s | 0 |
| 7_1_2 | 11.5% | 10M | 7.9s | 23K | **4.3s** | 4K |
| 8_1_2 | 14.8% | 5M | 42.1s | 305K | **33.5s** | 86K |
| 9_1_2 | 18.2% | 3M | **1956.9s** | 2M | 2127.5s | 2M |
| 10_1_2 | 22.4% | 2M | 9.1% | 3M | **8.7%** | 2M |
| 11_1_2 | 26.8% | 932K | **12.3%** | 1M | 13.9% | 2M |
| 12_1_2 | 31.6% | 22K | **16.7%** | 1M | **16.7%** | 1M |
| 13_1_2 | 36.9% | 7729 | 18.1% | 1M | **17.8%** | 1M |
| 14_1_2 | 42.4% | 1432 | 18.6% | 1M | **18.1%** | 1M |
| 15_1_2 | 48.5% | 238 | **20.9%** | 1M | 21.4% | 1M |
| 6_2_2 | 3.7s | 9522 | **0.52s** | 0 | 0.9s | 0 |
| 7_2_2 | 15.3s | 21K | **5.3s** | 2602 | 6.9s | 14K |
| 8_2_2 | 704s | 1M | 37.4s | 106K | **26.2s** | 45K |
| 9_2_2 | 15.3% | 9M | 364.9s | 208K | **274.2s** | 317K |
| 10_2_2 | 32.6% | 2M | 17.2% | 797K | **16.3%** | 2M |
| 11_2_2 | 29.5% | 1341K | **13.2%** | 491K | 18.8% | 1M |
| 12_2_2 | 39% | 952K | 23.1% | 523K | **21.1%** | 1M |
| 13_2_2 | 38.5% | 406K | 24.7% | 311K | **24.6%** | 595K |
| 14_2_2 | 42.8% | 155K | 31.5% | 168K | **30.3%** | 477K |
| 15_2_2 | 44% | 67K | 36.1% | 119K | **34.8%** | 397K |
| 6_2_3 | **0.1s** | 0 | 0.36s | 0 | 0.34s | 0 |
| 7_2_3 | 3.2s | 560 | **0.15s** | 0 | 1.9s | 290 |
| 8_2_3 | 8.5s | 2879 | **0.84s** | 0 | 1.4s | 0 |
| 9_2_3 | 11.6s | 2670 | **2.6s** | 102 | 3.8s | 1234 |
| 10_2_3 | 1909s | 265K | **12.6s** | 1022 | 22.5s | 13K |
| 11_2_3 | 13% | 690K | **28.0s** | 1600 | 207.5s | 83K |
| 12_2_3 | 9.4% | 384K | **342.2s** | 60K | 851.9s | 325K |
| 13_2_3 | 18.4% | 1095K | 5.3% | 475K | **2472.5s** | 599k |
| 14_2_3 | 20.7% | 108K | **8.6%** | 360K | 16.7% | 336K |
| 15_2_3 | 31% | 8153 | 14.6% | 273K | **14.4%** | 618K |
| 6_3_3 | **0.03s** | 0 | 0.16s | 0 | 0.54s | 0 |
| 7_3_3 | **0.18s** | 0 | 0.26s | 0 | 1.3s | 122 |
| 8_3_3 | 0.57s | 0 | **0.28s** | 0 | 1.7s | 219 |
| 9_3_3 | 3.4s | 173 | 3.9s | 0 | **1.8s** | 2334 |
| 10_3_3 | 47s | 31K | 14.3s | 2117 | **8.9s** | 985 |
| 11_3_3 | 174s | 52K | **18.63s** | 278 | 175.4s | 191K |
| 12_3_3 | 20% | 1655K | 797.6s | 41K | **690.5s** | 137K |
| 13_3_3 | 29.4% | 486K | 17.8% | 361K | **16.6%** | 532K |
| 14_3_3 | 30.8% | 185K | **17.6%** | 189K | 21.3% | 772K |
| 15_3_3 | 31.2% | 396K | 28.4% | 168K | **24.7%** | 328K |
| 6_3_4 | 0.24s | 0 | **0.04s** | 0 | 0.34s | 0 |
| 7_3_4 | 0.89s | 94 | **0.06s** | 0 | 0.60s | 0 |
| 8_3_4 | 1.8s | 17 | **0.07s** | 0 | 1.12s | 0 |
| 9_3_4 | 0.27s | 0 | **0.15s** | 0 | 0.43s | 0 |
| 10_3_4 | 5.8s | 78 | **1.94s** | 0 | 7.9s | 119 |
| 11_3_4 | 1233s | 277K | **1.9s** | 0 | 8.3s | 1K |
| 12_3_4 | 27.2s | 1548 | **1.7s** | 0 | 50.2s | 16K |
| 13_3_4 | 216s | 42K | **0.91s** | 0 | 55.1s | 17K |
| 14_3_4 | 1204s | 5666 | **57.5s** | 312 | 178.3s | 47K |
| 15_3_4 | 23.5% | 351K | **323.4s** | 11K | 13.9% | 227k |

Table 3: Results of the branch-and-bound phase of the three formulations on grid sensor networks

| | MIP1 | | | | MIP2 (flow) | | | | MIP3 (MTZ) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $gap_i$ | $gap_f$ | CPU | *#opt* | $gap_i$ | $gap_f$ | CPU | *#opt* | $gap_i$ | $gap_f$ | CPU | *#opt* |
| G_1_1 | 40.0 | 40.2 | 423.1s | 2/10 | 28.2 | 20.5 | 14.4s | 3/10 | 8.7 | 6.6 | 2.0s | 5/10 |
| G_1_2 | 21.9 | 34.8 | 30s | 1/10 | 21.9 | 15.9 | 0.34s | 4/10 | 21.9 | 16.1 | 0.53s | 4/10 |
| G_2_2 | 36.9 | 37.7 | 241s | 3/10 | 33.2 | 24.3 | 13.4s | 4/10 | 31.3 | 24.3 | 10.4s | 4/10 |
| G_2_3 | 10.4 | 25.8 | 386.5s | 5/10 | 10.4 | 11.6 | 3.3s | 7/10 | 10.4 | 15.6 | 5.8s | 8/10 |
| G_3_3 | 18.7 | 30.1 | 37.5s | 6/10 | 17.6 | 21.2 | 6.27s | 7/10 | 17.4 | 20.9 | 31.6s | 7/10 |
| G_3_4 | 5.7 | - | 298.8s | 9/10 | 5.7 | - | 7.2s | 10/10 | 5.7 | - | 33.6s | 9/10 |
| average | 21.7 | 33.73 | 236.1s | | 19.1 | 18.7 | **10.2s** | | **15.9** | **16.7** | 16.0s | |

Table 4: Synthesis of numerical results for MIP1, MIP2 and MIP3

(in seconds) for the whole branch-and-bound phase when it stops before reaching the time limit, or, alternatively, the final gap denoted by $gap_f$ when the time limit is reached, and $(ii)$ the number of generated nodes.

Here again, we can observe different behaviours of the models depending on $R^{se}$ and $R^{co}$ but the most striking observation is that with MIP1, we can solve 26 instances over the 60 considered within the time limit of one hour, while MIP2 can solve 34 instances and MIP3, two more instances than MIP2. Focusing on the instances where $R^{se} = R^{co} = 1$, we can observe that MIP3 either solves the instances faster, or can solve instances that neither MIP1 nor MIP2 can solve within the time limit, or reaches the time limit with a better final gap. Concerning the other pairs $(R^{se}, R^{co})$, the performances of MIP2 and MIP3 seem rather similar. Our final observation is that MIP1 is always outperformed either by MIP2 or MIP3 on 55 instances. Table 4 summarizes our numerical results. It presents, for each pair $(R^{se}, R^{co})$, the average initial and final gaps (in percentage). The average $gap_f$ is computed over the subset of instances not solved by both formulations, as a consequence $gap_f$ is not always lower than $gap_i$. Also, the average CPU time (in seconds) for the whole branch-and-bound phase is computed only for instances solved by the three formulations. If we focus on MIP2 and MIP3, Table 4 shows that MIP3 slightly outperforms MIP2 over 3 criteria: MIP3 solves three more instances to optimality and provides better average initial and final gaps. However, if the average CPU time for the whole branch-and-bound phase is computed only for instances solved by MIP2 and MIP3, it is divided by 1.6 in favour of MIP2.

In the following, we do not take into account MIP1 model since it is dedicated to grid sensor networks.

### *5.1.2 Random sensor networks*

In this section, we propose to examine the performance of our two models for randomly generated graphs.

Our instances are generated as follows. We first defined a square area of side $s$ in the euclidean plane with the origin of the euclidean plane as the left low corner of the square. $n$ targets were then randomly generated with a uniform distribution in this square. Given $s$ and $n$, we generated sensing and communication graphs relatively to $R^{se}$ varying from 1 to 3 and $R^{se} \leq R^{co} \leq 2R^{se}$. We repeated the generation until the graphs become connected. For each pair $(s,n)$, we selected five random targets placement. The sink node $t$ is always located at the origin of the euclidean plan. An illustration of the generation of 50 targets in a square of side 4 is given in Figure 5. In this example, an optimal solution of sensors placement is represented by twelve rhombi.

Table 5 compares the average number of arcs of our testbed instances for $R^{co} = R^{se} = 1$ versus grid instances, and shows the large density of our instances.

Our results are detailed in Table 6 where the first column *Instances* gives the characteristics of the instance in the format $n\_s\_R^{se}\_R^{co}$. The column *CPU* lists the

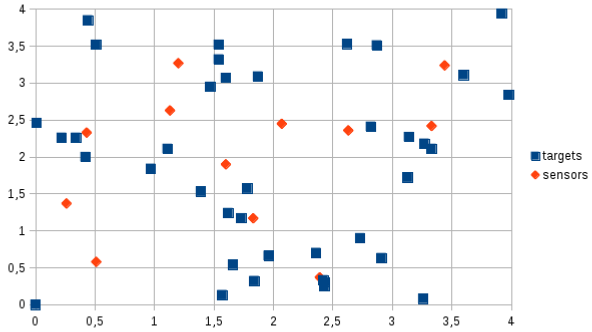

Fig. 5: n=50 targets generated in a square of side $s = 4$.

| Grid graphs | | Random graphs | |
|---|---|---|---|
| n | $\lvert A^{co} \rvert$ | n | $\lvert A^{co} \rvert$ |
| 49 | 84 | 51 | 370,6 |
| 81 | 144 | 81 | 671,6 |
| 100 | 180 | 101 | 747,8 |
| 121 | 220 | 121 | 778,0 |
| 144 | 264 | 151 | 1259,4 |

Table 5: Comparison of $|A^{co}|$ between grid and random graphs when $R^{co} = R^{se} = 1$

average CPU time for instances solved to optimality within one hour. The best results are emphasized in bold.

| | Flow model MIP2 | | | | MTZ model MIP3 | | | |
|---|---|---|---|---|---|---|---|---|
| *Instances* | $gap_i$% | $gap_f$% | CPU | *#solved* | $gap_i$% | $gap_f$% | CPU | *#solved* |
| G_50_4_1_1 | **28.8** | - | **1.01s** | 5/5 | 36.4 | - | 17.4s | 5/5 |
| G_50_4_1_2 | 0.0 | - | **0.04s** | 5/5 | 0.0 | - | 0.76s | 5/5 |
| G_80_5_1_1 | **33.3** | - | **13.1s** | 5/5 | 37.9 | - | 747,6s | 5/5 |
| G_80_5_1_2 | 0.00 | - | **0.17s** | 5/5 | 0.00 | - | 1.43s | 5/5 |
| G_100_6_1_1 | **32.9** | - | **42.7s** | **5/5** | 37.1 | 11.05 | 1999.9s | 1/5 |
| G_100_6_1_2 | 0.00 | - | **0.23s** | 5/5 | 0.00 | - | 1.95s | 5/5 |
| G_120_7_1_1 | **40.8** | **8.3** | **258.1s** | **4/5** | 41.3 | 19.9 | 3600s | 0/5 |
| G_120_7_1_2 | 0.0 | - | **0.27s** | 5/5 | 0.0 | - | 3.0s | 5/5 |
| G_120_7_2_2 | **21.1** | - | **63.5s** | 5/5 | 28.3 | - | 341.0s | 5/5 |
| G_120_7_2_3 | 8.6 | - | **3.00s** | 5/5 | 8.6 | - | 5.2s | 5/5 |
| G_150_7_1_1 | **35.6** | **7.9** | **866.4s** | **2/5** | 39.9 | 20.4 | 3600s | 0/5 |
| G_150_7_1_2 | 0.0 | - | **0.24s** | 5/5 | 0.0 | - | 2.7s | 5/5 |
| G_150_7_2_2 | **21.9** | - | **134.1s** | **5/5** | 26.7 | 17.6 | 203.6s | 3/5 |
| G_150_7_2_3 | **2.9** | - | **2.4s** | 5/5 | 5.7 | - | 7.3s | 5/5 |

Table 6: Synthesis of Numerical results for random sensor networks

For the whole of instances, MIP2 (flow model) outperforms MIP3 (MTZ model) over all criteria. Over the 70 tested instances, 66 (resp. 54) are solved to optimality within one hour by MIP2 (resp. MIP3) and the MIP solving for instances solved by both models needs only 12,6 seconds in average for MIP2 compared to 121,1 seconds for MIP3. Thus, MIP2 is ten times faster than MIP3. This can be partly explained by the fact that the average initial gap $gap_i$ provided by MIP2 is better than for MIP3: 16% versus 19%. Finally, the average final gap $gap_f$ computed for the instances, not solved within one hour by both models, is 2,6 times larger for MIP3 than for MIP2 (8.1% vs 21.5%). Important density of random graphs can also explain the superiority of MIP2 over MIP3, since the number of constraints of MIP3 depends on $|A^{co}|$.

## 5.2 Results for the MC$k$C problem with $k = 2$ or 3

In this section, we consider a generalization of the MCC problem: the Minimum Connected $k$-Coverage (MC$k$C) problem where a positive integer $k$ defines the coverage multiplicity of the targets.
In order to satisfy the $k$-coverage constraint, our two models MIP2 and MIP3, dedicated to the MCC problem, can easily be modified by replacing the 1-coverage constraint

$$x_i + \sum_{j \in X:(i,j) \in A^{se}} x_j \geq 1 \quad \forall i \in X \setminus \{t\}$$

by the $k$-coverage constraint

$$x_i + \sum_{j \in X:(i,j) \in A^{se}} x_j \geq k \quad \forall i \in X \setminus \{t\}$$

### *5.2.1 Grid sensor networks*

For square grids and a given $R^{se}$, it exists $k_{max}$ such that, for all $k > k_{max}$, the MC$k$C problem does not admit a solution. For example, for $R^{se} = 1$, a target located in a corner of the grid has a sensing neighbourhood of cardinality 3. So, this target cannot be covered more than three times. We conclude to $k_{max} = 3$ when $R^{se} = 1$. Figure 6 illustrates the case $k_{max} = 6$ when $R^{se} = 2$. Table 7 lists $k_{max}$ for $R^{se}$ varying from 1 to 4.

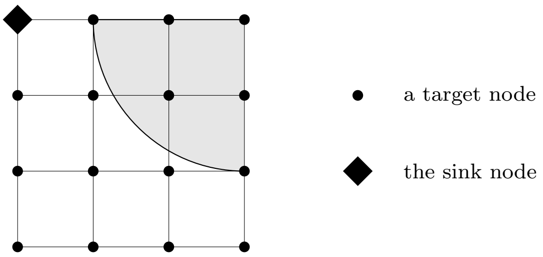

Fig. 6: Sensing neighbourhood for a target in a corner of the grid when $R^{se} = 2$.

| $R^{se}$ | $k_{max}$ |
|---|---|
| 1 | 3 |
| 2 | 6 |
| 3 | 11 |
| 4 | 17 |

Table 7: Maximal values for $k$

Numerical results for Grid sensor networks are summarized in Table 8 for MIP2 and in Table 9 for MIP3, in which the best results are emphasized in bold. Note that the values presented for each pair ($R^{se}$,$R^{co}$) in these two tables are average values over the ten grid instances. MIP2 (based on single flow commodity) solves three more instances than MIP3 (based on MTZ constraints) for $k = 2$ and two more instances for $k = 3$. For both models, two behaviours are observed. First, when $k = 2$, instances with $R^{se} = R^{co}$ are more difficult to solve. Indeed, initial gaps are larger: 15.3% (resp.

14.1%) in average for MIP2 (resp. MIP3), whereas, for instances with $R^{co} = R^{se} + 1$, the average gap is reduced to 4.5%. Secondly, the Minimum Connected $k$-Coverage Problem is easier to solve when $k$ is larger. Indeed, for $k = 3$, MIP2 (resp. MIP3) solves 59 (resp. 57) of the 60 instances within the time limit of one hour. Those observations can be explained by the fact that less extra sensors are necessary to ensure connectivity when $k$ and $R^{co}$ increase.

| | 2-coverage | | | | 3-coverage | | | |
|---|---|---|---|---|---|---|---|---|
| *Instance* | $gap_i$% | $gap_f$% | CPU | *#solved* | $gap_i$% | $gap_f$% | CPU | *#solved* |
| G_1_1 | 18.1 | 12.7 | 1034s | 3/10 | 3.5 | - | 21s | 10/10 |
| G_1_2 | 4.9 | - | 280s | 10/10 | 3.3 | - | 22s | 10/10 |
| G_2_2 | 18.5 | 11.4 | 212s | 5/10 | 3.5 | 2.4 | 140s | 9/10 |
| G_2_3 | 3.7 | - | 15s | 10/10 | 2.5 | - | 44s | 10/10 |
| G_3_3 | 9.4 | - | 568s | 10/10 | 3.7 | - | 16s | 10/10 |
| G_3_4 | 5.4 | - | 8s | 10/10 | 2.9 | - | 3s | 10/10 |

Table 8: Numerical results for minimum connected $k$-coverage with adapted flow model (MIP2).

| | 2-coverage | | | | 3-coverage | | | |
|---|---|---|---|---|---|---|---|---|
| *Instance* | $gap_i$% | $gap_f$% | CPU | *#solved* | $gap_i$% | $gap_f$% | CPU | *#solved* |
| G_1_1 | 15.6 | 14.2 | 169s | 3/10 | 3.5 | 1.0 | 144s | 9/10 |
| G_1_2 | 4.8 | 1.53 | 449s | 9/10 | 3.3 | - | 59s | 10/10 |
| G_2_2 | 16.5 | 11.4 | 46s | 4/10 | 4.0 | 4.6 | 411s | 8/10 |
| G_2_3 | 3.6 | - | 266s | 10/10 | 2.5 | - | 211s | 10/10 |
| G_3_3 | 10.2 | 11.3 | 422s | 9/10 | 3.9 | - | 96s | 10/10 |
| G_3_4 | 5.4 | - | 12s | 10/10 | 2.9 | - | 11s | 10/10 |

Table 9: Numerical results for minimum connected $k$-coverage with adapted MTZ model (MIP3).

### *5.2.2 Random sensor networks*

Numerical results for Random sensor networks are summarized in Table 10 for the minimum connected 2-coverage problem. Table 11 sums up the results obtained for the set of graph instances with 150 nodes. The best results are emphasized in bold.

We have the same observation than for grid square instances. Indeed, when $k = 2$, instances with $R^{se} = R^{co}$ are more difficult to solve. Initial gaps are larger: 8% (resp. 10.5%) in average for MIP2 (resp. MIP3), whereas, for instances with $R^{co} > R^{se}$, the average gap is reduced to 0% for both. The Minimum Connected $k$-Coverage problem is easier to solve when $k$ is larger. Futhermore, MIP2 outperforms MIP3 when considering multiple coverage of targets. The superiority of MIP2 is particularly high for random graphs as for the MCC problem. Concerning the MC2C problem, MIP2 solves the 70 instances to optimality within one hour and the MIP resolution needs only 26.8 seconds in average. MIP3 solves 11 less instances than MIP2 and takes 78 times longer to solve the same instance pool.

For $k = 3$, Table 11 focuses on graph instances with 150 instances. It shows the same observations as for other random graph instances, i.e., MIP2 outperforms MIP3 for all performance criteria: initial gaps, CPU and number of instances solved. In detail, for instances with $R^{co} > R^{se}$, initial gap in average is reduced to 0.005% (resp. 0.01%) for MIP2 (resp. MIP3), CPU time resolution is divided by 12 for MIP2, and MIP2 solves three more instances.

| | Flow model MIP2 | | | | MTZ model MIP3 | | | |
|---|---|---|---|---|---|---|---|---|
| *Instance* | $gap_i$% | $gap_f$% | CPU | *#solved* | $gap_i$% | $gap_f$% | CPU | *#solved* |
| G_50_1_1 | 8.3 | - | **0.31s** | 5/5 | 12.9 | - | 64.6s | 5/5 |
| G_50_1_2 | 0.0 | - | **0.05s** | 5/5 | 0.0 | - | 0.72s | 5/5 |
| G_80_1_1 | 7.5 | - | **1.6s** | 5/5 | 11.7 | - | 410,4s | 5/5 |
| G_80_1_2 | 0.00 | - | **0.15s** | 5/5 | 0.00 | - | 1.9s | 5/5 |
| G_100_1_1 | 11.9 | - | **3.5s** | 5/5 | 12.5 | 5.5 | 341.7s | 4/5 |
| G_100_1_2 | 0.00 | - | **0.19s** | 5/5 | 0.00 | - | 1.6s | 5/5 |
| G_120_1_1 | 15.3 | - | **344.9s** | 5/5 | 17.8 | 7.2 | 3600s | 0/5 |
| G_120_1_2 | 0.0 | - | **0.18s** | 5/5 | 0.0 | - | 1.9s | 5/5 |
| G_120_2_2 | 3.1 | - | **2.3s** | 5/5 | 4.7 | - | 19.4s | 5/5 |
| G_120_2_3 | 0.0 | - | **0.30s** | 5/5 | 0.0 | - | 3.5s | 5/5 |
| G_150_1_1 | 9.6 | - | **19.2s** | 5/5 | 11.9 | 6.5 | 3600s | 0/5 |
| G_150_1_2 | 0.0 | - | **0.24s** | 5/5 | 0.0 | - | 3.5s | 5/5 |
| G_150_2_2 | 1.7 | - | **1.9s** | 5/5 | 1.7 | - | 18.2s | 5/5 |
| G_150_2_3 | 0.0 | - | **0.33s** | 5/5 | 0.0 | - | 6.4s | 5/5 |

Table 10: Numerical results for minimum connected 2-coverage with adapted MIP2 and MIP3.

| | Flow model MIP2 | | | | MTZ model MIP3 | | | |
|---|---|---|---|---|---|---|---|---|
| *Instance* | $gap_i$% | Bd | CPU/$gap_f$% | *#nodes* | $gap_i$% | Bd | CPU/$gap_f$% | *#nodes* |
| G_1_1_1 | 0.00 | 54 | **0.5s** | 0 | 0.00 | 54 | 6.9s | 25K |
| G_1_1_2 | 0.02 | 62 | **0.5s** | 0 | 0.02 | 62 | 965.7s | 4M |
| G_1_1_3 | 0.02 | 60 | **4.2s** | 0 | 0.02 | 60 | 3.2% | 3M |
| G_1_1_4 | 0.00 | 59 | **0.3s** | 0 | 0.02 | 58 | 1.7% | 20M |
| G_1_1_5 | 0.02 | 64 | **1.1s** | 159 | 0.02 | 64 | 1.5% | 20M |
| G_1_2_1 | 0.0 | 54 | **0.38s** | 0 | 0.0 | 54 | 1.8s | 0 |
| G_1_2_2 | 0.0 | 62 | **0.15s** | 0 | 0.0 | 62 | 1.5s | 0 |
| G_1_2_3 | 0.0 | 60 | **0.2s** | 0 | 0.0 | 60 | 1.5s | 0 |
| G_1_2_4 | 0.0 | 58 | **0.14s** | 0 | 0.0 | 58 | 1.9s | 0 |
| G_1_2_5 | 0.0 | 64 | **0.16s** | 0 | 0.0 | 64 | 1.0s | 0 |
| G_2_2_1 | 0.0 | 18 | **0.44s** | 0 | 0.0 | 18 | 20.6s | 20K |
| G_2_2_2 | 0.0 | 19 | **0.33s** | 0 | 0.0 | 19 | 6.4s | 2K |
| G_2_2_3 | 0.0 | 17 | **1.1s** | 0 | 0.0 | 17 | 5.9s | 7K |
| G_2_2_4 | 0.0 | 19 | **0.38s** | 0 | 0.0 | 19 | 20.5s | 10K |
| G__2_5 | 0.0 | 18 | **0.32s** | 0 | 0.0 | 18 | 6.4s | 5K |
| G_2_3_1 | 0.0 | 18 | **0.36s** | 0 | 0.0 | 18 | 5.9s | 162 |
| G_2_3_2 | 0.0 | 19 | **0.61s** | 0 | 0.0 | 19 | 5.6s | 655 |
| G_2_3_3 | 0.0 | 17 | **0.29s** | 0 | 0.0 | 17 | 4.3s | 155 |
| G_2_3_4 | 0.0 | 19 | **0.58s** | 0 | 0.0 | 19 | 7.8s | 1K |
| G_2_3_5 | 0.0 | 18 | **0.47s** | 0 | 0.0 | 18 | 9.6s | 3K |

Table 11: Numerical results for minimum connected 3-coverage for graphs with 150 nodes.

## 6 Conclusion

Concerning the MCC problem, for all instances of grid sensor networks, either the MIP2 or MIP3 model yields a better LP-bound at the root of the branch-and-bound process than the MIP1 formulation of Rebai et al. Furthermore, those two formulations outperform MIP1 with a higher proportion of solved instances, a reduced CPU time and a lower number of explored nodes in the tree search. MIP3 (based on MTZ constraints) provides the best average results. Concerning random sensor networks, MIP2 has much better performances than MIP3 over all criteria. This observation can be partly explained by the larger density of random graph instances that significantly increases the number of constraints of MIP3. Our computational experiments confirm the difficulty of solving the MCC problem with classical mathematical mixed integer linear formulations inspired by the literature for this kind of placement problems. The solving difficulty is especially true for small values of $R^{se}$ and $R^{co}$. We note that the quality of the LP-bound and thus the efficiency of our two models are sensitive to the values of $R^{se}$ and $R^{co}$. This remark remains valid for the MC$k$C problem. On the other

hand, MIP2 (based on single flow commodity) outperforms MIP3 when considering multiple coverage of targets regardless of testbed instances.
Future works will consist in *(i)* improving our two general models by valid inequalities that take into account particular structures of the grid, such as symmetry, and *(ii)* testing them on general graphs.

## References

Bonsma P (2012) Max-leaves spanning tree is APX-hard for cubic graphs. Journal of Discrete Algorithms 12:14–23

Chakrabarty K, Iyengar SS, Qi H, Cho E (2002) Grid coverage for surveillance and target location in distributed sensor networks. IEEE Transactions on Computers 51(12):1448–1453

Das B, Bharghavan V (1997) Routing in ad-hoc networks using minimum connected dominating sets. In: Communications, ICC '97 Montreal, Towards the Knowledge Millennium, IEEE, vol 1, pp 376–380

Fan N, Watson J (2012) Solving the connected dominating set problem and power dominating set problem by integer programming. In: Combinatorial Optimization and Applications: 6th International Conference (COCOA), Springer, Lecture Notes in Computer Science, vol 7402, pp 371–383, DOI 10.1007/978-3-642-31770-5_33

Fourer R, Gay DM, Kernighan BW (1993) AMPL: A Modeling Language for Mathematical Programming. The Scientific Press (now an imprint of Boyd & Fraser Publishing Co.), Danvers, MA, USA

Fujie T (2003) An exact algorithm for the maximum leaf spanning tree problem. Comput Oper Res 30(13):1931–1944

Fujie T (2004) The maximum-leaf spanning tree problem: Formulations and facets. Networks 43(4):212–223

Garey MR, Johnson DS (1979) Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, USA

Gavish B (1982) Topological design of centralized computer networks–formulations and algorithms. Networks 12(4):355–377

Gendron B, Lucena A, da Cunha AS, Simonetti L (2014) Benders decomposition, branch-and-cut, and hybrid algorithms for the minimum connected dominating set problem. INFORMS Journal on Computing 26(4):645–657

Gonçalves D, Pinlou A, Rao M, Thomassé S (2011) The domination number of grids. SIAM Journal on Discrete Mathematics 25(3):1443–1453

Guha S, Khuller S (1998) Approximation algorithms for connected dominating sets. Algorithmica 20(4):374–387

IBM-ILOG (2014) IBM ILOG CPLEX 12.6 Reference Manual. URL `http://www-01.ibm.com/support/knowledgecenter/SSSA5P_12.6.0/ilog.odms.studio.help/Optimization_Studio/topics/COS_home.html`

Ke W, Liu B, Tsai M (2011) The critical-square-grid coverage problem in wireless sensor networks is NP-Complete. Computer Networks 55(9):2209–2220, DOI 10.1016/j.comnet.2011.03.004, URL `https://doi.org/10.1016/j.comnet.2011.03.004`

Lucena A, Maculan N, Simonetti L (2010) Reformulations and solution algorithms for the maximum leaf spanning tree problem. Computational Management Science 7(3):289–311

Miller CE, Tucker AW, Zemlin RA (1960) Integer programming formulation of traveling salesman problems. J ACM 7(4):326–329

Rebai M, Le Berre M, Snoussi H, Hnaien F, Khoukhi L (2015) Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks. Comput Oper Res 59:11–21

Rebai M, Afsar HM, Snoussi H (2016) Exact methods for sensor deployment problem with connectivity constraint in wireless sensor networks. International Journal of Sensor Networks (IJSNET) 21(3):157–168, DOI 10.1504/IJSNET.2016.078324, URL `https://doi.org/10.1504/IJSNET.2016.078324`

Reich A (2016) Complexity of the maximum leaf spanning tree problem on planar and regular graphs. Theoretical Computer Science 626(C):134–143

Reis M, Lee O, Usberti F (2015) Flow-based formulation for the maximum leaf spanning tree problem. Electronic Notes in Discrete Mathematics 50:205 – 210

Roveti DK (2001) Choosing a humidity sensor: a review of three technologies. URL `http://www.sensorsmag.com/sensors/humidity-moisture/choosing-a-humidity-sensor-a-review-three-technologies-840`

Wang X, Xing G, Zhang Y, Lu C, Pless R, Gill C (2003) Integrated coverage and connectivity configuration in wireless sensor networks. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, ACM, SenSys'03, pp 28–39