Chapter 2

Minimum Power Multicast problem

In this chapter, we take into account the Minimum Power Multicast problem (MPM) in wireless Ad-Hoc networks [52]. The chapter is organized as follows: an introduction to the problem is given in section 2.1 and related works are presented in section 2.2. A formal description of the modelling aspects of the problem can be found in section 2.3, while the mathematical formulation of the MPM problem expressed in terms of a Set Covering problem is discussed in section 2.4 together with its comparison with some of the formulations that have been proposed in the literature. In section 2.5, we show some logic inequalities, whereas in 2.6, we report how to modify the graph associated with the Multicasting problem in wireless networks in order to model it as a Steiner Arborescence problem in a wired network. Section 2.7 is devoted to the description of two exact procedures for solving the problem that include the reduction technique for the Set Covering problem to reduce the huge number of the model's constraints. Finally, some computational results are illustrated in section 2.8 and some concluding remarks are summarized in 2.9.

2.1 Introduction

Ad-Hoc networks are composed of a set of mobile devices with limited resources, that communicate with each other by transmitting a radio signal without using any fixed infrastructure or centralized administration. Nowadays, this kind of networks find their applications in several fields such as exchanging messages in an area where natural disasters have destroyed the existing infrastructure or in a battlefield. They are also used, for example, to allow internet access or simply to exchange information in buildings or in trains or to enable video-conferencing, etc. (see e.g. [66], [84]). The devices of an Ad-Hoc network, called also nodes, are arbitrarily located in an area where they are able to move, but at the time of the transmission all the nodes are supposed to be stationary; all along this dissertation, we will consider only static networks. Every terminal of the network is equipped with an omnidirectional antenna in such a way that the signal is spread radially from the nodes. A device may communicate with a single–hop, i.e. directly, with any other terminal which is located within its transmission range. In order to communicate with the terminals placed out of this range a multi–hop communication has to be performed: it simply consists in making use of intermediate devices, called routers, that retransmit the received message to the directly unreachable terminals ([72], [84]). Those nodes that are not reached by any signal are called isolated nodes.

The Multicast problem consists in connecting a specified device, called "source", with a set of target terminals, called "destinations", with the possibility of using any other device of the network as router. Since the resources of the devices are limited (for example nodes are equipped with batteries) the source–destination connections should be obtained using the minimum amount of power. This objective would also have the advantage of reducing the interferences within the network and, consequently, of improving the signal quality.

The Minimum Power Multicast problem consists in assigning a transmission power to each node of the network in such a way that the source is connected to all the destinations with the minimum total transmitting power. We omit to consider interference problem in the model and we suppose that there is no constraint on the maximum transmission power of the nodes. Finally, we assume that the topology of the network and hence the exact position of all the terminals is known in advance.

2.2 Related works

The MPM problem represents a generalization of the very well known Minimum Power Broadcasting (MPB) problem. Indeed, if the set of destinations coincides with all the nodes of the network, except the source, the MPM problem reduces to the MPB problem (see e.g. Althaus *et al.* [1], Altinkemer *et al.* [3], Das *et al.* [25], Montemanni *et al.* [60], Wieselthier *et al.* [85], Yuang [88]). The MPM problem has been proved to be NP-complete (Cagalj *et al.* [13], Clementi et al. [20], [21]) and thus difficult to solve to optimality. Moreover, it is not simply a minimum Steiner Arborescence ([25], [57], [84]) connecting the source with the destinations because of the so called "broadcast property". Indeed a transmitting node reaches all the nodes of the network placed within its transmission range without any additional power, so that the amount of power in the solution of the MPM problem is not worse than the amount of power in the solution of the minimum Steiner Arborescence on the same but wired network.

While the MPB problem has attracted a wide attention in the scientific literature, the MPM problem has been scarcely studied despite its applicative importance. In fact, nowadays most of the MPM formulations available represent somehow an adaptation of the MPB models to the multicasting case. Interesting approaches to the problem are due to Wieselthier *et al.* [84] and to Das *et al.* [25]. The first authors describe an algorithm, called the Broadcast Incremental Power (BIP), and three greedy heuristics for the Multicast Power problem. The Broadcast Incremental Power (BIP) [84] is a modification of the Prim's algorithm [70]. Indeed, starting with a node $s \in V$ source of the communication and a set $L := \{s\}$, at each iteration the algorithm chooses a minimum-incremental power edge $e = (u, v) \in E$, connecting a node $u \in L$ to a node $v \in L^c$ and updates the set $L := L \cup \{v\}$. This process is repeated until $L = V$. The increment of power is the difference between the power that has to be used by a node $u \in L$ to reach a node $v \in L^c$ and the power already assigned to u.

Three different integer programming models have been proposed in [25] by Das *et al.*; these formulations for the MPM problem have been obtained as a generalization of those constructed for the MPB problem. Some specific studies for the multicast case have been considered in Guo *et al.* [36] and in Leino [53]. In particular, a linear integer formulation for the MPM problem has been presented in Leino [53] and a general scheme of a cutting plane algorithm has been used for its solution, whereas a flow-based formulation expressed in terms of a mixed integer programming has been suggested in Guo *et al.* [36].

2.3 Mathematical Models for the MPM

We shall model the MPM problem in terms of a graph, by considering the devices of the network as nodes and the transmission links as arcs like in Figure 2.1.

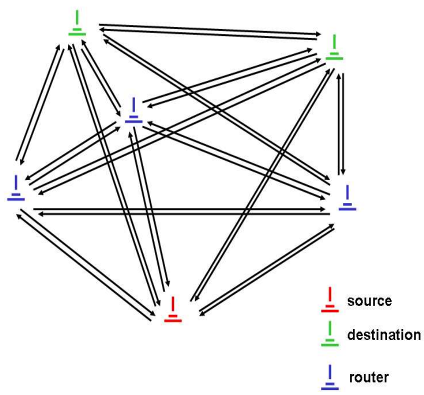

Figure 2.1: Example of a Multicast problem in a complete graph with 6 nodes

Let $G = (V, A)$ be a directed complete graph, where V represents the set of the terminals of the network and A is the set of directed arcs which connect all the possible pairs (i, j), with $i, j \in V$ and $j \neq i$. Each node $i \in V$ can receive data from any other node of the network and send data to any node in its transmission range, which is not a priori constrained to assume any fixed value. We select a particular node $s \in V$ as the source of the messages (the red antenna in Figure 2.1), and a subset of nodes $R \subset V$ whose elements are the destinations of the communication (the green antennae in Figure 2.1). Nodes belonging to $V \setminus (R \cup \{s\})$ may act either as routers, i.e., they can be involved in forwarding the messages or they may remain isolated without receiving or transmitting any signal (the blue antennae in Figure 2.1).

Let n and m be two integer numbers representing respectively the cardinality of set V and that of R, with $1 \leq m < n$. We note that if $m = 1$ the problem reduces to finding the Shortest Path from the source to the destination and if $m = n - 1$ the Multicasting problem reduces to a Broadcasting problem. Despite some analogies with the Minimum Spanning Arborescence problem, the MPB problem in wireless networks has been proved to be NP-complete ([13], [20], [21]). We assume that the nodes are fixed since we are considering static networks and, thus, all the distances d_{ij} between each pair of nodes i and j in V are known in advance. This is an approximation of the real world applications, but it is not too restrictive, as one may think, especially, if we consider optimization over short time intervals and assume that the devices move slowly in the area.

For simplicity, we consider here the case in which for any distinct nodes $i, k, l \in V$, it holds: $d_{ik} \neq d_{il}$.

With each arc (i, j) it is associated a cost p_{ij} that represents the minimum amount of power required to establish a direct connection from node i to node j. As usually assumed in literature in a simple signal propagation model [72], the power p_{ij} is considered to be proportional to the power of the distance d_{ij} with an environment-dependent exponent κ whose value is typically in the interval [2,5]; therefore, $p_{ij} := (d_{ij})^{\kappa}$. Notice that the results presented in this dissertation remain valid also in case more complex signal propagation models are considered.

Most of the already defined formulations of the problem ([53], [60], [84]) use, instead of the costs p_{ij} for the arcs, an incremental cost c_{ij} defined as follows:

$$c_{ij} = p_{ij} - p_{ia^i_j} \qquad \forall (i, j) \in A,$$

where, according to the definition given in [60], the node a^i_j is the "ancestor" of j with respect to i:

$$a_j^i := \begin{cases} i & \text{if } p_{ij} = \min_{k \in V}\{p_{ik}\}, \\ \arg\max_{k \in V} \{p_{ik} | p_{ik} < p_{ij}\} & \text{otherwise.} \end{cases} \tag{2.1}$$

By introducing the so called *range assignment* function, which assigns to each node $i \in V$ its transmitting power $r(i)$:

$$r : V \to \mathbb{R}^+, \qquad i \mapsto r(i),$$

the MPM problem can be equivalently formulated defining such a function in order to minimize the quantity $\sum_{i \in V} r(i)$, while guaranteeing the connection among the source and all the destinations. Obviously, in any efficient solution, $r(i)$ must be zero or equal to p_{ij} for some j (i.e., node i does not transmit or uses exactly the amount of power necessary to reach a target node j), so we shall assume this to be the case. We want to stress here that when we talk about connection among the source and all the destinations in this chapter and in chapter 4 we do not mean necessarily a direct connection, but we do not also mean the existence of a path in the traditional sense (see Definition 1.3.2) from the source to each destination. In fact, since the nodes are equipped with omnidirectional antennae and the communication is a radio transmission, any signal forwarded by node $i \in V$ and directed to node $j \in V$ is also received by all the nodes that are not more distant than j from i, i.e., if $r(i) = p_{ij}$, then every node $k \in V$ such that $p_{ik} \leq p_{ij}$ receives the signal (see Figure 2.2). This is the so called "broadcast property" ([60], [84]) which is a peculiarity of this kind of networks. Several nodes can be, therefore, covered and reached with a single transmission and, hence, using a single transmission power.

Even though the MPM problem consists in assigning the transmission power to the nodes, as suggested before, it is convenient to consider the decision variables associated with the arcs ([25], [60]) in order to model the

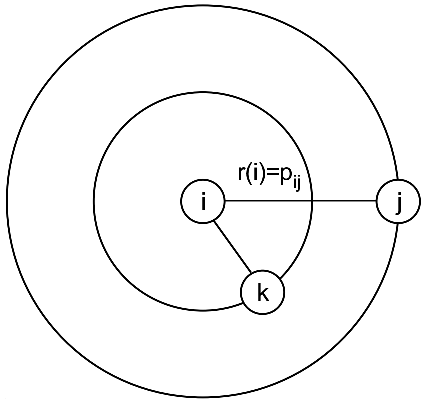

Figure 2.2: Broadcast property

link states. In particular, we want to model: (i) the event that node i is transmitting to a target node j (that is, i uses exactly an amount p_{ij} of power); (ii) the event that the transmission of node i is received by node j (that is, the power assigned to node i is not smaller than p_{ij}); and (iii) the event that arc (i, j) belongs to the underlying Steiner Arborescence which connects s with every node in R. We introduce, thus, three sets of variables, x, y and z to characterize each of the three above events.

The set of variables x describes which node transmits to whom; formally, using the *range assignment* function:

$$x_{ij} := \begin{cases} 1 & \text{if } r(i) = p_{ij}, \\ 0 & \text{otherwise.} \end{cases}$$

The set of variables y determines which nodes are in the transmission range of other nodes, i.e. for all $(i, j) \in A$, $y_{ij} = 1$, if the node i transmits and reaches node j, otherwise $y_{ij} = 0$. By expressing y variables using the definition of the function r, we can write for all $(i, j) \in A$:

$$y_{ij} := \begin{cases} 1 & \text{if } r(i) \geq p_{ij}, \\ 0 & \text{otherwise.} \end{cases}$$

Finally, the variables z define a Steiner Arborescence T, connecting s with all the destinations in R: for all $(i, j) \in A$, if $(i, j) \in T$, then $z_{ij} = 1$ (that is the node i is transmitting and the node j is reached by it), otherwise $z_{ij} = 0$.

The "broadcasting property" makes the difference between the Minimum Steiner Arborescence problem and the Minimum Power Multicast problem ([25], [84]), indeed, if the objective function of the first problem in a wired network can be expressed in this way:

$$\min \sum_{(i,j)\in A} p_{ij} z_{ij},$$

the objective function for the Multicasting problem in a wireless network is the following:

$$\min \sum_{i\in V} \max_{j\in V\setminus\{i\}} p_{ij} z_{ij}.$$

For this reason, the cost of an optimal solution of the Multicasting problem is a lower bound for the optimal Steiner Arborescence solution in the same but wired graph.

Since we assign only one power value to each node $i \in V$, there will be at most one intended target node j for i. Thus, as in [25]:

Remark 2.3.1. For any node $i \in V$ the following relation holds

$$\sum_{j\in V\setminus\{i\}} x_{ij} \leq 1.$$

Furthermore, using the inequalities of the Remark 2.3.1, it is possible to express a relation between variables y and x. Indeed, if variable $x_{ik} = 1$, it means that node i transmits with the power necessary to reach k. Any other node j which is not farther than k from i also receives the transmission, therefore, $y_{ij} = 1$. We can, thus, derive:

Remark 2.3.2. For all $(i, j) \in A$ the following relation binds the y and x variables:

$$y_{ij} = \sum_{k \in V \setminus \{i\},\ d_{ij} \leq d_{ik}} x_{ik}.$$

Moreover, we can notice that in any efficient solution, if variable $x_{ij} = 1$, then also variable $z_{ij} = 1$, since the link (i,j) belongs to the underlying Steiner Arborescence connecting the source to the destinations; on the other hand, an arc (i, j) might belong to the Steiner Arborescence even if j is not the target node of i, i.e., $r(i) = p_{ik} > p_{ij}$, with $k \in V \setminus \{i\}$ and $x_{ij} = 0$ but $z_{ij} = 1$.

On the basis of the definition of the variables and the above observations, we have:

Remark 2.3.3. For all $(i, j) \in A$ the following relations must hold

$$x_{ij} \leq z_{ij} \leq y_{ij}.$$

We describe now three formulations presented in literature. The first one is a slight modification in terms of notation of the model proposed by Leino [53]:

$$\min \sum_{(i,j) \in A} c_{ij} y_{ij} \tag{2.2}$$

$s.t.$

$$\sum_{i \in S, j \in S^c} y_{ij} \geq 1 \qquad \forall\, S \subset V,\ s \in S,\ R \cap S^c \neq \emptyset \tag{2.3}$$

$$y_{ij} \leq y_{ia^i_j} \qquad \forall\, (i, j) \in A,\ a^i_j \neq i \tag{2.4}$$

$$y_{ij} \in \{0, 1\} \qquad \forall\, (i, j) \in A. \tag{2.5}$$

The second one is an adaptation to the Multicasting problem of the MPB formulation defined in Montemanni et al [60] (by omitting the symmetric connectivity condition):

$$\min \sum_{(i,j)\in A} c_{ij} y_{ij} \tag{2.6}$$

$s.t.$

$$\sum_{i\in S, j\in S^c} z_{ij} \geq 1 \qquad \forall\, S \subset V,\ s \in S,\ R \cap S^c \neq \emptyset \tag{2.7}$$

$$y_{ij} \leq y_{ia^i_j} \qquad \forall\, (i,j) \in A,\ \ a^i_j \neq i \tag{2.8}$$

$$z_{ij} \leq y_{ij} \qquad \forall\, (i,j) \in A \tag{2.9}$$

$$y_{ij}, z_{ij} \in \{0,1\} \qquad \forall\, (i,j) \in A. \tag{2.10}$$

Observe that, since variables z_{ij} do not appear in the objective function, we can strengthen formulation (2.7) − (2.10) by substituting inequalities (2.9) with the equations $z_{ij} = y_{ij}$ without losing any optimal solution. By doing so, it is easy to see that formulation (2.7) − (2.10) is, in fact, a relaxation of formulation (2.3) − (2.5).

Finally, the last formulation is the multicasting version of the MPB formulation presented in Altinkemer et al [3]. While the first two formulations minimize the incremental cost, this model minimizes directly the power to be assigned to each arc:

$$\min \sum_{(i,j)\in A} p_{ij}x_{ij} \tag{2.11}$$

$s.t.$

$$\sum_{i\in S, j\in S^c} z_{ij} \geq 1 \qquad \forall S \subset V,\ s \in S,\ R \cap S^c \neq \emptyset \tag{2.12}$$

$$z_{ij} \leq \sum_{k\in V\setminus\{i\}, d_{ij}\leq d_{ik}} x_{ik} \qquad \forall\,(i,j) \in A \tag{2.13}$$

$$x_{ij}, z_{ij} \in \{0,1\} \qquad \forall (i,j) \in A. \tag{2.14}$$

Constraints $(2.3), (2.7)$ and (2.12) are the “connectivity constraints”, that is, for each cut (S, S^c) with $s \in S$ and $S^c \cap R \neq \emptyset$, these constraints enforce the existence of at least one arc outgoing from a node belonging to S and incoming in a node of S^c; constraints (2.4) and (2.8) are the “broadcast constraints”, enforcing the “broadcast property”; constraints (2.9) and (2.13) represent the variable relations described in Remarks 2.3.2 and 2.3.3; and constraints $(2.5), (2.10)$ and (2.14) are the domain definition constraints.

2.4 The Set Covering Formulation

In this section, we will define our Set Covering–based model for the MPM problem. We start by proposing a first formulation that we prove to be at least as strong as the formulation $(2.2) - (2.5)$. Then by exploiting the topological properties of the problem, we introduce our Set Covering model.

For convenience, we shall use the following notation: for each node $i \in V$, let v^i be an array whose components are the nodes of the network ordered with respect to an increasing distance from node i. In other words, if j and k are two indices in $\{1, \dots, n\}$ with $j \leq k$, then v^i_j and v^i_k are two nodes in V whose distances from i are related by

$$d_{iv^i_j} \leq d_{iv^i_k}.$$

We refer to v^i as a *distance array*.

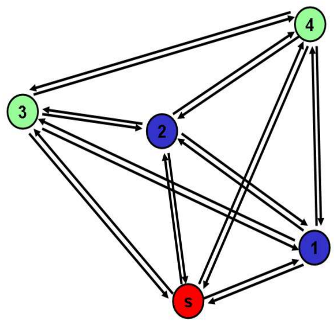

Figure 2.3: Example for the distance arrays

Example 2.4.1. For the network in Figure 2.3 the distance arrays are the following: $v^s = (s, 1, 2, 3, 4), v^1 = (1, s, 2, 4, 3), v^3 = (3, 2, s, 4, 1)$,
$v^4 = (4, 2, 1, 3, s)$.

By Remark 2.3.2, we have:

Remark 2.4.1. For all $i \in V$ and $j \in \{2, \dots, n-1\}$ the following relations must hold

$$x_{iv^i_j} = y_{iv^i_j} - y_{iv^i_{j+1}}$$

and for $j = n$:

$$x_{iv^i_n} = y_{iv^i_n}.$$

We propose now a first formulation which uses only the variables x:

$$\min \sum_{(i,j)\in A} p_{ij}x_{ij} \tag{2.15}$$

$s.t.$

$$\sum_{i\in S, j\in S^c} \sum_{k\in V\setminus\{i\},\ d_{ij}\leq d_{ik}} x_{ik} \geq 1 \qquad \forall\, S\subset V,\ s\in S,\ R\cap S^c\neq\emptyset \tag{2.16}$$

$$\sum_{j\in V\setminus\{i\}} x_{ij} \leq 1 \qquad \forall\, i\in V \tag{2.17}$$

$$x_{ij}\in\{0,1\} \qquad \forall\,(i,j)\in A. \tag{2.18}$$

We notice that it is possible to use Remarks 2.3.2 and 2.4.1 to augment formulation (2.2) – (2.5) with variables x_{ij} and formulation (2.15) – (2.18) with variables y_{ij}, so that their linear relaxations can be compared. By doing so, we can derive the following result.

Proposition 2.4.1. *The linear relaxation of formulation* (2.15) – (2.18) *is equivalent to the linear relaxation of formulation* (2.2) – (2.5).

Proof. First of all, observe that, since vectors x and y are related as in Remarks 2.3.2 and 2.4.1 the objective functions (2.2) and (2.15) express the same quantity. In fact, by the definition of incremental costs, for any $i\in V$ and $j\in\{2,\ldots,n\}$ we have

$$p_{iv_j^i} = \sum_{k=2}^{j} c_{iv_k^i}.$$

Hence, by using Remark 2.4.1, we have

$$\sum_{j=2}^{n} p_{iv_j^i}x_{iv_j^i} = \sum_{j=2}^{n-1}\sum_{k=2}^{j} c_{iv_k^i}(y_{iv_j^i}-y_{iv_{j+1}^i}) + \sum_{k=2}^{n} c_{iv_k^i}y_{iv_n^i} =$$

$$\sum_{k=2}^{n} c_{iv_k^i} \sum_{j=k}^{n} y_{iv_j^i} - \sum_{k=2}^{n-1} c_{iv_k^i} \sum_{j=k+1}^{n} y_{iv_j^i} = \sum_{k=2}^{n} c_{iv_k^i} y_{iv_k^i}.$$

Consequently, we have

$$\sum_{(i,j)\in A} p_{ij}x_{ij} = \sum_{i\in V}\sum_{j=2}^{n} p_{iv_j^i}x_{iv_j^i} = \sum_{i\in V}\sum_{k=2}^{n} c_{iv_k^i}y_{iv_k^i} = \sum_{(i,j)\in A} c_{ij}y_{ij}.$$

Assume now that $\overline{x}$ is a feasible solution of the relaxation of (2.15) – (2.18), and that $\overline{y}$ is the corresponding vector of variables obtained in Remark 2.3.2. We have to show that $\overline{y}$ is a feasible solution for the linear relaxation of (2.2) – (2.5). Indeed, we have:

$$\sum_{i\in S, j\in S^c} \overline{y}_{ij} = \sum_{i\in S, j\in S^c} \sum_{k\in V\setminus\{i\},\, d_{ij}\le d_{ik}} \overline{x}_{ik} \geq 1.$$

Moreover, for any $(i,j) \in A$ such that $a_j^i \neq i$, since variables $\overline{x}_{ij}$ are not negative, we have:

$$\overline{y}_{ij} = \sum_{k\in V\setminus\{i\},\, d_{ij}\le d_{ik}} \overline{x}_{ik} \leq \overline{x}_{ia_j^i} + \sum_{k\in V\setminus\{i\},\, d_{ij}\le d_{ik}} \overline{x}_{ik} = \sum_{k\in V\setminus\{i\},\, d_{ia_j^i}\le d_{ik}} \overline{x}_{ik} = \overline{y}_{ia_j^i}$$

and, for any $(i,j) \in A$,

$$0 \leq \overline{y}_{ij} = \sum_{k\in V\setminus\{i\},\, d_{ij}\le d_{ik}} \overline{x}_{ik} \leq \sum_{j\in V\setminus\{i\}} \overline{x}_{ij} \leq 1.$$

On the other hand, let $\overline{y}$ be a feasible solution for the linear relaxation of formulation (2.2) – (2.5) and let $\overline{x}$ be the corresponding vector of variables obtained by Remark 2.4.1. We can show that $\overline{x}$ is a feasible solution for the linear relaxation of (2.15) – (2.18). Indeed, by using Remark 2.3.2, constraints (2.16) are easily seen to be satisfied. Moreover, for any $i \in V$, by Remark 2.4.1 we have:

$$\sum_{j \in V \setminus \{i\}} x_{ij} = \sum_{j=2}^{n} x_{iv_j^i} = \sum_{j=2}^{n-1} (y_{iv_j^i} - y_{iv_{j+1}^i}) + y_{iv_n^i} = y_{iv_2^i} \leq 1,$$

which means that constraints (2.17) are also satisfied. Finally, by using (2.4), we have:

$$0 \leq \overline{y}_{ia_j^i} - \overline{y}_{ij} = \overline{x}_{ia_j^i} \leq 1.$$

□

By using similar arguments as those in the proof of Proposition 2.4.1 and letting variables x and y be related according to Remarks 2.3.2 and 2.4.1, it is easy to prove the following:

Remark 2.4.2. Any feasible solution to the linear relaxation of formulation (2.6) – (2.5) is also feasible for the linear relaxation of formulation (2.15) – (2.18).

We can notice that in constraints (2.16) the coefficients of some variables x_{ij} could be greater than one. This suggests to strengthen the formulation by reducing to one all the left-hand-side coefficients of constraints (2.16). In order to describe the resulting constraints, we introduce the following notation.

Let S be any proper subset of V. For every $i \in S$, we label with $v_{k(S)}^i$ the first component in the distance array v^i which is not an element of S. Furthermore, we denote by $K^i(S)$ the subset of $V \setminus \{s\}$ whose elements are all the nodes of the network different from the source and having distance from i greater than or equal to $d_{iv_{k(S)}^i}$. For a better understanding of this notation, we give an example.

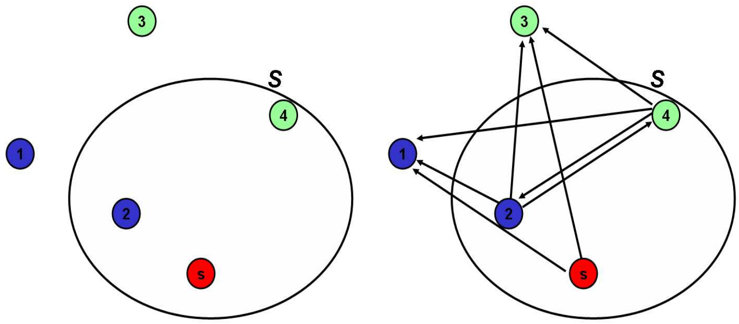

Figure 2.4: Example for constraints (2.20)

Example 2.4.2. Looking at Figure 2.4, $V := \{s, 1, 2, 3, 4\}$, $R := \{3, 4\}$ and $S := \{s, 2, 4\}$. The distance arrays are: $v^s = (s, 2, 4, 1, 3)$, $v^1 = (1, 2, 3, s, 4)$, $v^2 = (2, s, 1, 4, 3)$, $v^3 = (3, 4, 1, 2, s)$, $v^4 = (4, 3, s, 2, 1)$; thus $v^s_{k(S)}$ and $v^2_{k(S)}$ are node 1, while $v^4_{(k(S))}$ is node 3 and $K^s(S) := \{1, 3\}$, $K^2(S) := \{1, 3, 4\}$ and $K^4(S) := \{1, 2, 3\}$.

Now we are able to present the strengthened formulation of the MPM problem:

$$\min \sum_{(i,j)\in A} p_{ij} x_{ij} \tag{2.19}$$

$s.t.$

$$\sum_{i\in S} \sum_{j\in K^i(S)} x_{ij} \geq 1 \qquad \forall\, S \subset V,\ s \in S,\ R \cap S^c \neq \emptyset \tag{2.20}$$

$$\sum_{j\in V\setminus\{i\}} x_{ij} \leq 1 \qquad \forall\, i \in V \tag{2.21}$$

$$x_{ij} \in \{0, 1\} \qquad \forall\, (i, j) \in A. \tag{2.22}$$

The set of constraints (2.20) represents the connectivity requirements; for every cut (S, S^c) with $s \in S$ and $R \cap S^c \neq \emptyset$ there should be a node

i in S that transmits with a power sufficient to reach at least one node in S^c. We remark that the "target" node j of node i (that is, the one such that $x_{ij} = 1$) does not need to be in S^c, indeed, j can belong to S, but the distance between i and j must be greater than the distance from i to a node in S^c. For example, the presence of one of the arcs in Figure 2.4 would satisfy the constraint (2.20) relative to the choice of $S = \{s, 1, 4\}$. Constraints (2.21) ensure that at most one power value is assigned to each node and, finally, (2.22) are the binary restrictions on the variables.

We now show that constraints (2.21) in the last formulation are redundant for defining any optimal solution of the linear relaxation of the formulation as the objective value coefficients are non negative.

Proposition 2.4.2. *Let $\overline{x}$ be an optimal solution of* (2.19) *satisfying constraints* (2.20) *and the linear relaxation of constraints* (2.22)*. Then we have:*

$$\sum_{j \in V \setminus \{i\}} \overline{x}_{ij} \leq 1 \qquad \forall i \in V.$$

Proof. Assume that there exists $h \in V$ such that

$$\sum_{j \in V \setminus \{h\}} \overline{x}_{hj} > 1. \tag{2.23}$$

Let $l \in \{1, ..., n\}$ be the smallest index such that:

$$\sum_{j=l+1}^{n} \bar{x}_{hv_j^h} \leq 1,$$

let R denote the set $\{v_l^h, v_{l+1}^h, \ldots, v_n^h\}$ and $r = v_l^h$. By setting, for all $j \in V \setminus \{h\}$,

$$x^*_{hj} = \begin{cases} \overline{x}_{hj} & \text{if } j \in R \setminus \{r\}, \\ 1 - \sum_{j \in R \setminus \{r\}} \overline{x}_{hj} & \text{if } j = r, \\ 0 & \text{otherwise,} \end{cases}$$

we have that: $x^*_{hr} = 1 - \sum_{j \in R \setminus \{r\}} \overline{x}_{hj} < \overline{x}_{hr}$ and, thus,

$$\sum_{j \in V \setminus \{h\}} p_{hj} x^*_{hj} < \sum_{j \in V \setminus \{h\}} p_{hj} \overline{x}_{hj}.$$

Let, for any node $i \in V \setminus \{h\}$ and for any node $j \in V \setminus \{i\}$, $x^*_{ij} = \overline{x}_{ij}$. Then, the new solution x^* is feasible, since constraints (19) are still satisfied. Moreover, we have that:

$$\sum_{(i,j) \in A} p_{ij} x^*_{ij} < \sum_{(i,j) \in A} p_{ij} \overline{x}_{ij}.$$

This leads to a contradiction, since $\overline{x}$ is by assumption an optimal solution.

□

By the above Proposition, we can remove constraints (2.21) from the formulation. Moreover, since all the powers are positive values, we notice that, in any optimal solution, no node is assigned the power to reach exactly the source, so that all the incoming arcs of A in the source s can be eliminated from the graph:

$$A := A \setminus \{(i,j) \in A : \ i \in V, j = s\}.$$

The final formulation of the problem, that we propose is a Set Covering formulation:

$$\min \sum_{(i,j) \in A} p_{ij} x_{ij} \tag{2.24}$$

$s.t.$

$$\sum_{i \in S} \sum_{j \in K^i(S)} x_{ij} \geq 1 \qquad \forall\, S \subset V, \ s \in S, \ R \cap S^c \neq \emptyset \tag{2.25}$$

$$x_{ij} \in \{0,1\} \qquad \forall\, (i,j) \in A. \tag{2.26}$$

Table 2.1: Average gap for (2.3)-(2.5) and for (2.25)-(2.26)

		(2.3)-(2.5)	*(2.25)-(2.26)*			*(2.3)-(2.5)*	*(2.25)-(2.26)*
n	m	*gap*	*gap*	n	m	*gap*	*gap*
5	1	0.21183	0	15	1	0.48164	0
5	2	0.27884	0	15	2	0.49797	0
5	3	0.19820	0	15	3	0.44208	0
5	4	0.17085	0	15	4	0.40148	0.00002
				15	5	0.38226	0.00002
10	1	0.36262	0	15	6	0.35043	0.00708
10	2	0.41995	0	15	7	0.33496	0.00952
10	3	0.34237	0	15	8	0.28470	0.01015
10	4	0.35768	0.00009	15	9	0.29569	0.01280
10	5	0.32836	0.00028	15	10	0.28654	0.01123
10	6	0.32093	0.00390	15	11	0.27004	0.01793
10	7	0.30090	0.00626	15	12	0.26053	0.01835
10	8	0.29403	0.00971	15	13	0.24193	0.01835
10	9	0.24807	0.00666	15	14	0.23624	0.02104

Constraints (2.25) are the connectivity constraints and constraints (2.26) are the domain definition constraints.

Since the number of constraints (2.25) is $2^{n-1} - 2^{n-m-1}$, the main difficulty of this problem, beyond the fact that it is an integer problem, is caused by the huge number of such constraints. Moreover, it is evident that the broadcasting version of this problem has the maximum number of constraints of type (2.25). Notice, however, that in general many of the constraints (2.25) are redundant and can be removed from the formulation because they are dominated by other constraints in (2.25).

Remark 2.4.3. The optimal solution of the linear relaxation of the Set Covering formulation provides a lower bound that is more effective than the lower bound produced by the optimal solution of the linear relaxation of the formulation (2.2) – (2.5).

In order to compare the two formulations we have done several experiments. In Table 2.1 each column reports the average value of the gap between the optimal value OPT of the integer problem and the optimal value LB of the linear relaxation of the two formulations for 20 randomly generated problems for each combination of the number of nodes/destinations. We indicate with *gap* the value $(OPT - LB)/LB$. From the results reported in Table 2.1, it is highlighted firstly that the lower bound of the Set Covering formulation is much better than the lower bound of the formulation $(2.2) - (2.5)$, secondly that for problems with few nodes and few destinations the optimal solution of the linear relaxation of our proposed formulation is already an integer solution.

2.5 Logic inequalities

We present some inequalities that can be added to the problem and that can be found just considering logic properties of the MPM problem.

Remark 2.5.1. The following inequalities:

$$\text{(i)} \qquad x_{ij} + x_{ji} \leq 1 \qquad \forall\, i \in V,\ j \in \delta^+(i);$$

$$\text{(ii)} \qquad \sum_{i \in V \setminus \{j\}} x_{ij} \leq 1 \qquad \forall\, j \in V;$$

are inequalities that reduce the feasible region of the MPM problem but they do not cut off any fractional optimal solution of the linear relaxation of (2.24) – (2.26).

Remark 2.5.2. The number of the arcs of an optimal integer solution of the MPM problem (that is the number of the transmissions in an optimal solution) should be at most the number of arcs in an acyclic graph spanning all the nodes of the network and hence $\sum_{(i,j) \in A} x_{ij} \leq n - 1$. We can notice that if the power assigned to the source is exactly the power necessary to reach its most distant destination, placed in the k^{th} position of the array v^s, then all the destinations are reached by the signal generated by the source and no other transmission must be performed in order to create the connection. This remark can be expressed with the constraint:

$$\sum_{(i,j)\in A\setminus\{(s,v_k^s)\}} x_{ij} \leq (n-1)(1-x_{sv_k^s}). \tag{2.27}$$

In an optimal solution, if the source s transmit to the node v_k^s then the right hand side of (2.27) is zero and this force all the other variable x_{ij} to be zero otherwise it holds: $\sum_{(i,j)\in A\setminus\{(s,v_k^s)\}} x_{ij} \leq \sum_{(i,j)\in A} x_{ij} \leq n-1$ and the constraint (2.27) is fulfilled.

Remark 2.5.3. The inequalities

$$\sum_{j\in\delta^-(i)} x_{ji} \leq \sum_{j\in\delta^+(i)} x_{ij} \qquad \forall\, i \in V \setminus (R \cup \{s\}) \tag{2.28}$$

are the flow-balance constraints (see e.g. [47]). If i is a router and i is directly reached by a communication originated by a node j in the network, constraint (2.28) forces node i to transmit. In no optimal integer solution a router i is a leaf of the arborescence, indeed, if it exists $j \in \delta^-(i)$ such that $x_{ji} = 1$ and for each $k \in \delta^+(i)$ the variables x_{ik} are all equal to zero, the cost p_{ji} paid for this type of solution can be reduced making j transmit to a node h closer to j than i without disconnecting any destination.

2.6 Multicasting problem and Minimum Steiner Arborescence

Minimum Power Multicast problem on the directed graph $G = (V, A)$ can be reduced into a Minimum Steiner Arborescence problem ([14], [55]) on a directed graph $G' = (V', A')$. The graph $G' = (V', A')$ can be constructed as follows: for each node $i \in V$, consider the set of the outgoing arcs from i (see Definition 1.3.5), $\delta^+(i)$. For each arc $(i,j) \in \delta^+(i) \setminus \{(i, v_2^i)\}$ a node u should be inserted into the graph and the arc (i,j) should be split into the arcs (i,u) and (u,j). The cost of the arc (i,j) is assigned to the arc (i,u), whereas a zero cost is assigned to (u,j). Furthermore, all the arcs (u,k) with $p_{ik} \leq p_{ij}$ should be added to the graph with a zero cost.

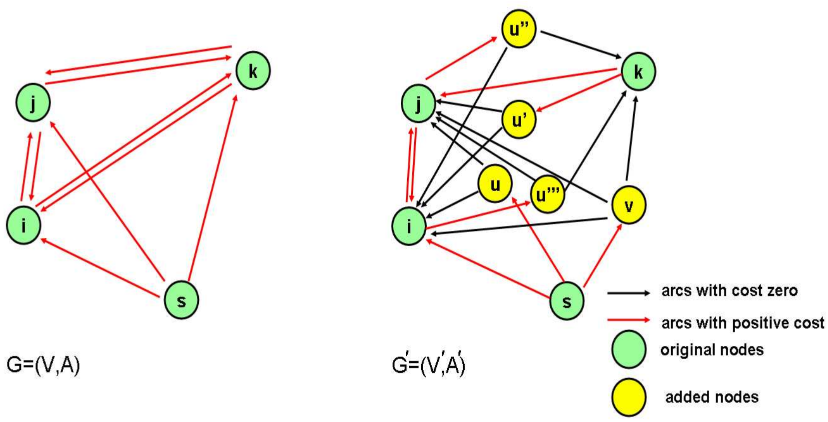

Figure 2.5: The graph for a Multicast problem in wireless network and the graph for the equivalent Steiner Arborescence problem

With this transformation $(n-2)+(n-1)(n-3)$ new nodes are added to the original graph so that in total $|V'| = (n-1)^2$, whereas the $(n-1)^2$ arcs of G are substituted by $(n^3-n^2-2n)/2$, i.e. $|A'| = (n^3-n^2-2n)/2$. The cardinality of V' is $O(n^2)$ and the cardinality of A' is $O(n^3)$; the size of the problem, thus, grows very rapidly as the size of the original problem increases.

Example 2.6.1. Figure 2.5 is a little example of a graph $G = (V,A)$ for the Multicasting problem with 4 nodes and of the graph $G' = (V', A')$ on which the Steiner Arborescence problem has the same optimal solution value as the optimal solution value of the Multicast problem. All the arcs in red are arcs with strictly positive costs, while the arcs in black have costs zero.

2.7 Solution Methods

As discussed before, the main difficulty for the solution of the Set Covering formulation is represented by the set of constraints (2.25), but a considerable help may be given by the structure of the formulation. Here, we propose two solution methods that exploit such structure.

In the first procedure, we generate the whole constraint matrix, but we take into account only a subset of its rows. Indeed, initially, we create a submatrix by selecting $n-1$ rows and we perform a preprocessing on this submatrix in order to erase dominated rows and columns, then we solve the integer problem and finally, we check whether violated constraints exist. If all the constraints (2.25) of the problem are satisfied, the procedure is interrupted since the optimal solution has been found, otherwise, we add at most n^2 violated rows at a time and we repeat the iterative process for the new submatrix until an optimal solution is found.

We specify that among the first $n-1$ rows of the initial submatrix, we select the row corresponding to the inequality relative to the subset $S = \{s\}$ and all the rows corresponding to the inequalities relative the subsets S such that $|S^c| = 1$. Moreover, whenever we find a row which is dominated in the current submatrix, we label it and we do not admit the possibility of reintroducing it in any subsequent matrix; only at the end of the procedure, before electing the optimal solution we check whether all the erased constraints are satisfied, otherwise we add the violated ones and the whole process is repeated.

In our second method, violated constraints are generated iteratively on the basis of the current solution looking at its support (see Definition 1.1.9). We start with the inequalities (2.25) generated by the sets $S := \{s\}$ and $S := \{s, v_2^s\}$ and we solve the resulting linear relaxation of the problem. On the basis of the optimal solution, we define the related variables y using the equality in the Remark 2.3.2 and we construct the connected component of the network starting with the source. The connected component of the source is the set of the nodes of the graph such that there exists a directed path from the source to these nodes using the arcs in which the values of the variables y are not zero. While at least one destination is not connected to the source, the cut (2.25), generated by the set S of the nodes belonging to the connected component of the source, is added to the formulation and the linear relaxation of the problem is solved again until all the destinations are in the connected component of the source. At this point, if the current solution is integer, then the procedure is interrupted, otherwise a maximum flow problem from the source to each destination with the current y values as capacities is solved (see Definition 1.4.3). If all the maximum flow values are at least one and the current optimal solution is fractional, then the current integer problem is solved and if all the destinations are connected to the source the procedure is interrupted, otherwise the cut (2.25) generated by the set S of the nodes connected to the source is generated and the integer problem is solved again. If at least one maximum flow value is less than one, then we define the set S corresponding to the cuts with minimum capacity (see Proposition 1.4.1), we add these constraints to the current formulation and we solve again the linear relaxation of the current problem. Every time a set of rows is added to the current submatrix, we perform the preprocessing (see Proposition 1.2.1). The procedure sketched above can be formalized by means of the following procedure:

Step 0: Let F be a formulation for problem MPM with only the constraints generated by $S = \{s\}$ and $S = \{s, v_2^s\}$ among the constraints (2.25);

Step 1: Solve the linear relaxation of F, and let $\overline{x}$ be the optimal solution;

Step 2: Define variable y as in Remark 2.3.2 and find the connected component of the source;

Step 3: If there is at least one destination that is not connected to the source, define S, the set of the nodes connected to the source, add the constraint (2.25) relative to S to the current formulation, perform the preprocessing of the constraint matrix and go to Step 1;

Step 4: If all the destinations are connected to the source and the current solution is integer; Stop.

Step 5: If all the destinations are connected to the source and the current solution is fractional go to Step 6;

Step 6: For each source-destination pair, solve the maximum flow problem with the current y as capacities;

Step 7: If all the values of the maximum flow problems are greater than or equal to 1, solve the integer problem, $\overline{x}$ is the optimal solution and go to Step 2;

Step 8: If at least one value of the maximum flow problems is lower than 1; define S corresponding to the minimum capacity cut; add the constraints (2.25) relative to S to the current formulation, perform the preprocessing of the constraint matrix, solve the linear relaxation of the problem and go to Step 6.

The preprocessing of the matrix, used in both methods, consists in finding and erasing the dominated columns and rows. We take advantage of the fact that the matrix is composed by only ones and zeros and we use the common preprocessing techniques for the Set Covering problem (see Proposition 1.2.1). A dominated column is either a null column or a column whose cost (power) is not smaller than that of another column which is, component-wise, not greater, while a row is dominated if there exists another row of the matrix which is, component-wise, not greater. The convergence of both the procedures is guaranteed because the number of inequalities (2.25) is, albeit huge, finite.

2.8 Experimental Results

We have implemented the solution algorithms in C and we have run the codes on a Dual Intel Xeon 3.2GHz machine with 4 GB RAM memory using the version 9.1 of Cplex as solver.

The experiments have been performed on a set of test problems with increasing number of nodes and of possible destinations; for each problem size, 20 different instances are generated. The nodes of the networks have been uniformly generated on a grid of size 10000×10000 and the source and the destinations have been randomly selected among the generated nodes as well. To obtain the power values from the distances we have set the coefficient κ to 2, while we have set to 3600 seconds the maximum resolution time, after which the solution process is interrupted.

Our computational results have been summarized in Tables 2.2, 2.3 and 2.4 in which we indicate with *Cplex* 9.1 the solution by the integer cplex solver of the entire problem (including all the constraints), with *method I* the method of choosing violated inequalities among all the generated constraints and with *method II* the method in which we generate violated constraints

Table 2.2: Average computational times for randomly generated problems with up to 15 nodes

		Cplex 9.1		*method I*			*method II*		
n	m	T	σ	T	σ	It	T	σ	It
5	1	**0.0000**	0.000	0.0005	0.000	2.1	0.001	0.002	2.8
5	2	**0.0000**	0.000	0.0002	0.000	2.2	0.002	0.004	3.6
5	3	**0.0000**	0.000	0.0002	0.000	2.4	0.001	0.003	4.1
5	4	**0.0000**	0.000	0.0002	0.000	2.6	0.002	0.004	4.5
10	1	0.010	0.005	**0.000**	0.000	2.7	0.003	0.006	5.5
10	2	0.016	0.005	**0.003**	0.004	2.8	0.008	0.009	8.0
10	5	0.025	0.004	**0.002**	0.012	2.9	0.015	0.718	12.3
10	9	0.022	0.004	**0.004**	0.005	3.0	0.024	0.014	15.3
15	1	1.207	0.171	0.073	0.047	3.4	**0.015**	0.022	10.1
15	5	3.849	0.522	0.127	0.046	4.1	**0.079**	0.054	28.5
15	10	4.859	2.217	0.134	0.077	3.6	**0.127**	0.054	36.7
15	14	5.171	2.615	**0.115**	0.061	5.7	0.143	0.058	38.5

on the basis of the nodes reachable by the signal spread by the source. All the methods use Cplex to solve the resulting LP or IP problems.

In the Tables 2.2, 2.3 and 2.4, we report the number of nodes of the network n, the number of destinations m, the average execution time T, its standard deviation σ and the average number of iterations It required to solve the problem. Moreover, in Table 2.4 we report the percentage $NS\%$ of the not solved instances within the time limit.

The best solution average time among the solving procedures is highlighted with a bold character. The results in Table 2.2 are related to networks with 5, 10 and 15 nodes combined with all the possible numbers of destinations. It is clear that for networks with 5 and 10 nodes, all the procedures solve the MPM problem quite quickly; Cplex seems to be more efficient only when $n = 5$, whereas the first method works better when $n = 10$. When we increase the value of n the second method has the best

Table 2.3: Average computational times for randomly generated problems with 20 nodes

		method I			*method II*		
n	m	T	σ	It	T	σ	It
20	1	2.628	1.606	5.8	**0.057**	0.059	19.1
20	5	4.923	2.030	6.4	**0.306**	0.228	45.4
20	10	4.828	2.086	5.4	**0.694**	0.392	62.0
20	15	4.207	1.684	4.9	**0.779**	0.412	65.0
20	19	4.034	1.328	4.1	**0.904**	0.678	66.6

Table 2.4: Average computational times for randomly generated problems with 30, 50 and 100 nodes

		method II			
n	m	T	σ	It	$NS\%$
30	1	1.288	1.315	61.4	
30	10	8.930	6.086	111.7	
30	15	7.789	4.609	108.4	
30	29	9.077	5.325	106.4	
50	1	6.647	7.588	74.7	
50	10	512.223	401.593	294.2	10
50	25	640.236	889.187	248.0	30
50	49	712.714	646.270	214.5	10
100	1	348.916	375.378	143.0	
100	5	927.537	606.565	212.8	60

performance. For networks with 15 nodes, the first method is the most efficient when the number of destination is greater than 10 and so for the broadcasting version of the problem.

In Table 2.3, we present the results for the MPM problem on networks with 20 nodes; while it is not possible to solve any of these problems generating the whole matrix of constraints, the second method outperforms the first method even when $m = n - 1$.

A different situation is shown in Table 2.4. For the MPM problems on networks with more than 30 nodes, the first method fails to solve the problem because of the memory required to generate the whole constraint matrix. On the contrary, the second method is still able to solve the MPM problem on networks with up to 50 nodes, but presently there are still some instances not solved within the time limit of an hour. Instances with 100 nodes have been solved, by now, for just a limited number of destination.

2.9 Concluding Remarks

We have proposed a Set Covering–based formulation for the Minimum Power Multicasting problem in Ad-Hoc networks, and we presented two possible algorithms for its solution. We carried out an experimental study by using a set of test problems randomly generated having a number of nodes ranging from 5 to 100. While we think that the presented formulation represents an original and effective approach to the problem, we are conscious that some improvements should be done. The theoretical and polyhedral properties of the model may be investigated together with a better way of generating violated constraints. In this direction goes the following chapter.