Mathematical Methods of Operations Research manuscript No.
(will be inserted by the editor)

Minimum Power Multicasting problem in Wireless Networks

Valeria Leggieri, Paolo Nobili and Chefi Triki

Dipartimento di Matematica, Università del Salento 73100 Lecce, Italy

Received: date / Revised version: date

Abstract In this paper we deal with the Minimum Power Multicasting problem (MPM) in wireless ad-hoc networks. By using an appropriate choice of the decision variables and by exploiting the topological properties of the problem, we are able to define an original formulation based on a Set Covering model. Moreover, we propose for its solution two exact procedures that include a preprocessing technique that reduces the huge number of the model's constraints. We also report some experimental results carried out on a set of randomly generated test problems.

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. Oliveira et al. (2005)). 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; in this paper we are considering, thus, 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 node. 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 (Rappaport (1996), Wieselthier et al. (2000)). Those nodes that are not reached by any signal are called isolated nodes.

The Multicasting 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 (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 Multicasting problem (MPM) consists, thus, 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 each node can be given power sufficient to reach all the other nodes of the network. Finally, we assume that each node knows exactly the position of the other terminals of the network.

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. (2003), Das et al. (2003), Montemanni et al. (2004), Wieselthier et al. (2001), Yuan (2005)). The MPM problem has been proved to be NP-complete (Cagalj et al. (2002), Clementi et al. (1999),(2001)) and thus difficult to solve to optimality. Moreover it is not simply a minimum Steiner Arborescence (Das et al. (2003), Magnanti et al. (1995), Padberg et al. (1983), Wieselthier et al. (2000)) connecting the source with the destinations because of the wireless multicast advantage (see section 2).

While the MPB problem has attracted a wide attention in the scientific literature, the MPM problem has been scarcely studied despite its applicative importance. Indeed, nowadays most of the MPM formulations available represent somehow an adaptation of the MPB models to the multicasting case. Interesting approaches to the MPM problem are due to Wieselthier et al. (2000) that propose the Broadcast Incremental Power (BIP) algorithm and three greedy heuristics, and to Das et al. (2003) that propose three different integer programming models which are a generalization of those constructed for the MPB problem. Some specific studies for the multicast case have been considered in Guo et al. (2004), where a flow-based formulation expressed in terms of a mixed integer program has been proposed and in Leino (2002), where a linear integer formulation for the MPM problem and a general scheme of a cutting plane algorithm for its solution have been presented.

We feel that even though most of the efforts have been done in studying the MPB problem, more attention should be given to the Minimum Power Multicasting version of the problem, which, as underlined before, is more general than the broadcasting counterpart. Indeed, in this paper we consider directly the multicasting case and we define a new formulation for the MPM problem expressed in terms of a Set Covering model and we propose two exact procedures for its solution. Our solution approaches take into account both the topology of the network and the peculiarity of the problem in order to exploit efficiently the particular structure of the constraint matrix.

The paper is organized as follows: a formal description of the modeling aspects of the problem is presented in section 2. The mathematical formulation of the MPM problem expressed in terms of a Set Covering problem is discussed in section 3 together with its comparison with some of the formulations that have been proposed in the literature. Section 4 is devoted to the description of the methods used to solve the problem and some computational results are illustrated in section 5. Finally, section 6 presents some concluding remarks and suggests some possible research developments on this topic.

2 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.

Let $G(V, A)$ be a directed complete graph, where V represents the set of terminals of the network and A 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 terminal 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, and a subset of nodes $D \subset V$ whose elements are the destinations of the communication. Nodes belonging to $V \setminus (D \cup \{s\})$ may act either as routers, i.e., they are involved in forwarding the messages or they may remain isolated without receiving or transmitting any signal.

Let n and m be two integer numbers representing respectively the cardinality of set V and that of D, with $m < n$. We note that if $m = 1$ the problem reduces to finding the minimum path from the source to the destination and if $m = n-1$ the multicasting problem reduces to a broadcasting problem.

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.

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 (Rappaport (1996)), 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 remain valid whenever more complex signal propagation models are considered.

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 minimum power multicasting 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 either does not transmit or uses exactly the amount of power necessary to reach a target node j), so we shall assume that to be the case.

Since the communication among the nodes of the network does not use any fixed infrastructure and the nodes are equipped with omnidirectional antennae, 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 Fig. 1). This is the wireless multicast advantage which is a peculiarity of this kind of networks. Even though the MPM problem consists in assigning

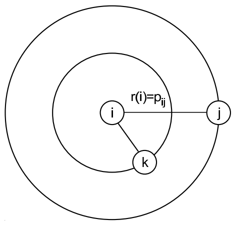

Fig. 1 Wireless multicast advantage

the transmission power to the nodes, as suggested before, it is convenient to consider the decision variables associated with the arcs in order to model the 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 D. 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 D: 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 wireless multicast advantage makes the difference between the Minimum Steiner Arborescence problem and the Minimum Multicasting problem, indeed if the objective function of the first problem can be expressed in this way: $\min \sum_{(i,j) \in A} p_{ij} z_{ij}$, the objective function for the Multicasting problem 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

Steiner Arborescence solution in the same graph. Nevertheless the Minimum Power Multicasting problem on the directed graph $G(V, A)$ can be reduced into a Minimum Steiner Arborescence problem (Liang (2002)) on a directed graph $G'(V', A')$, but in this transformation the cardinality of V' is $O(n^2)$ and the cardinality of A' is $O(n^3)$.

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, for any node $i \in V$ the following relation holds

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

Furthermore, by using inequality (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 and any other node j which is not farther than k from i also receives the transmission, therefore $y_{ij} = 1$.

Lemma 1 *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., $r(i) = p_{ik} > p_{ij}$, $x_{ij} = 0$ but $z_{ij} = 1$.

On the basis of the definition of the variables and the above observations, for all $(i,j) \in A$ the following relations bind the variables:

$$x_{ij} \leq z_{ij} \leq y_{ij}. \tag{2}$$

In the sequel we denote by S a subset of V and by S^c its complementary set in V (i.e. $V = S \dot{\cup} S^c$).

Certain formulations (see e.g. Leino (2002)) uses, 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 Montemanni et al. (2004), the node a^i_j is the *ancestor* of j with respect to i (for simplicity, we consider here the case in which $\forall\, i \in V, \nexists\, k,l \in V$ such that $p_{ik} = p_{il}$):

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

By using this notation, we describe now the formulation presented in Leino (2002)

$$\min \sum_{(i,j)\in A} c_{ij} y_{ij} \tag{3}$$

$$s.t. \quad \sum_{i \in S, j \in S^c} y_{ij} \geq 1 \qquad \forall\, S \subset V,\ s \in S,\ D \cap S^c \neq \emptyset \tag{4}$$

$$y_{ij} \leq y_{ia^i_j} \qquad \forall\, (i,j) \in A,\ a^i_j \neq i \tag{5}$$

$$y_{ij} \in \{0,1\} \qquad \forall\, (i,j) \in A. \tag{6}$$

The "connectivity constraints" (4) enforce, for each cut (S, S^c), the existence of at least one arc outgoing from a node belonging to S and incoming in a node of S^c; constraints (5) enforce the wireless multicast advantage and (6) are the domain definition constraints.

3 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 equivalent to formulation (3) – (6) and 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*.

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

$$\min \sum_{(i,j) \in A} p_{ij} x_{ij} \tag{7}$$

$$s.t. \quad \sum_{i \in S, j \in S^c} \sum_{k \in V \setminus \{i\},\, d_{ij} \leq d_{ik}} x_{ik} \geq 1 \qquad \forall\, S \subset V,\ s \in S,\ D \cap S^c \neq \emptyset \tag{8}$$

$$\sum_{j \in V \setminus \{i\}} x_{ij} \leq 1 \qquad \forall\, i \in V \tag{9}$$

$$x_{ij} \in \{0, 1\} \qquad \forall\, (i,j) \in A. \tag{10}$$

In view of Lemma 1, it is easy to prove:

Lemma 2 *For all $i \in V$ and $j \in \{2, \ldots, n-1\}$ 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 observe that it is possible to use Lemmas 1 and 2 to augment formulation $(3)-(6)$ with variables x_{ij} and formulation $(7)-(10)$ with variables y_{ij}, so that their linear relaxations can be compared. By doing so, we can derive the following result.

Proposition 1 *The linear relaxation of formulation $(7)-(10)$ is equivalent to the linear relaxation of formulation $(3)-(6)$.*

Proof First of all, observe that, for vectors x and y related by Lemmas 1 and 2 the objective functions (3) and (7) 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

$$p_{iv^i_j} = \sum_{k=2}^{j} c_{iv^i_k}.$$

Hence, by using Lemma 2, we have

$$\sum_{j=2}^{n} p_{iv^i_j} x_{iv^i_j} = \sum_{j=2}^{n-1} \sum_{k=2}^{j} c_{iv^i_k} (y_{iv^i_j} - y_{iv^i_{j+1}}) + \sum_{k=2}^{n} c_{iv^i_k} y_{iv^i_n} =$$

$$\sum_{k=2}^{n} c_{iv^i_k} \sum_{j=k}^{n} y_{iv^i_j} - \sum_{k=2}^{n-1} c_{iv^i_k} \sum_{j=k+1}^{n} y_{iv^i_j} = \sum_{k=2}^{n} c_{iv^i_k} y_{iv^i_k}.$$

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 (7) – (10), and that $\overline{y}$ is the corresponding vector of variables obtained in Lemma 1. We have to show that $\overline{y}$ is a feasible solution for the linear relaxation of (3) – (6). Indeed, we have:

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

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

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

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

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

On the other hand, let $\overline{y}$ be a feasible solution for the linear relaxation of formulation (3) – (6) and let $\overline{x}$ be the corresponding vector of variables obtained by Lemma 2. We can show that $\overline{x}$ is a feasible solution for the linear relaxation of (7) – (10). Indeed, by using Lemma 1, constraints (8) are easily seen to be satisfied. Moreover, for any $i \in V$, by Lemma 2 we have:

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

which means that constraints (9) are also satisfied. Finally, by using (5), we have:

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

□

We can observe that in constraints (8) 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 (8).

Definition 1 *Let S be any proper subset of V. For every $i \in S$, let $v^i_{k(S)}$ be the first component in the distance array v^i which is not an element of S. $K^i(S)$ is the subset of $V \setminus \{s\}$ whose elements have distance from i greater than or equal to $d_{iv^i_{k(S)}}$.*

For a better understanding of this definition, we give an example.

Example 1 Suppose that $V := \{s, 1, 2, 3\}$, $D := \{1, 3\}$, $S := \{s, 3\}$ and that the distance arrays are the following:

$$v^s = (s, 1, 3, 2),\ v^1 = (1, s, 3, 2),\ v^2 = (2, 3, s, 1),\ v^3 = (3, 2, 1, s);$$

then $v^s_{k(S)}$ is node 1, while $v^3_{k(S)}$ is node 2 and subsets $K^s(S)$ and $K^3(S)$ are the sets $\{1, 2, 3\}$ and $\{1, 2\}$, respectively.

Proposition 2 *For all $S \subset V$ and $i \in S$, then*

$$K^i(S) \subseteq \cup_{j \in S^c} \{k \in V :\ d_{ij} \leq d_{ik}\}.$$

Proof Let S be a proper subset of V and let $i \in S$. Consider $v^i_{k(S)} \in S^c$, the first component in the distance array v^i which is not an element of S. Let

l be an element of $K^i(S)$; by definition of $K^i(S)$ it holds that $d_{il} \geq d_{iv^i_{k(S)}}$ and thus $l \in \{k \in V : \ d_{iv^i_{k(S)}} \leq d_{ik}\}$. Since $v^i_{k(S)} \in S^c$, the thesis follows.

□

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

$$\min \sum_{(i,j)\in A} p_{ij}x_{ij} \tag{11}$$

$$\sum_{i\in S} \sum_{j\in K^i(S)} x_{ij} \geq 1 \qquad \forall\, S \subset V,\ s \in S,\ D \cap S^c \neq \emptyset \tag{12}$$

$$\sum_{j\in V\setminus\{i\}} x_{ij} \leq 1 \qquad \forall\, i \in V \tag{13}$$

$$x_{ij} \in \{0,1\} \qquad \forall\, (i,j) \in A. \tag{14}$$

The set of constraints (12) represents the connectivity requirements; for every cut (S, S^c) with $s \in S$ and $D \cap S^c \neq \emptyset$ there should be a node i in S that transmits with a power sufficient to reach at least one node in S^c. We notice that the "target" node j of node i (that is, the one such that $x_{ij} = 1$) needs not 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. Constraints (13) ensure that at most one power value is assigned to each node and, finally, (14) are the binary restrictions on the variables.

We now show that constraints (13) in formulation (11) – (14) are redundant for defining any optimal solution of the linear relaxation of the formulation as the objective value coefficients are non negative.

Proposition 3 *Let $\overline{x}$ be an optimal solution of* (11) *satisfying constraints* (12) *and the linear relaxation of constraints* (14)*. Then we have:*

$$\sum_{j \in V \setminus \{i\}} \overline{x}_{ij} \leq 1 \qquad \forall i \in V. \tag{15}$$

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

$$\sum_{j \in V \setminus \{h\}} \overline{x}_{hj} > 1. \tag{16}$$

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 (12) 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 (13) from the formulation. The final formulation of the problem, that we propose is a set covering formulation:

$$\min \sum_{(i,j)\in A} p_{ij}x_{ij} \tag{17}$$

$$s.t. \quad \sum_{i\in S}\sum_{j\in K^i(S)} x_{ij} \geq 1 \qquad \forall\, S \subset V,\ s \in S,\ D \cap S^c \neq \emptyset \tag{18}$$

$$x_{ij} \in \{0,1\} \qquad \forall\, (i,j) \in A. \tag{19}$$

Constraints (18) are the connectivity constraints and constraints (19) are the domain definition constraints.

Since the number of constraints (18) is $2^{n-1}-2^{n-m-1}$, the main difficulty of this formulation, besides the fact that it is an integer programming one, 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 (18). Notice, however, that in general many of the constraints (18) are redundant and can be removed from the formulation.

Proposition 4 *The value of an optimal solution of the linear relaxation of formulation* $(17)-(19)$ *is not smaller than the value of an optimal solution of the linear relaxation of formulation* $(3)-(6)$.

Proof Let $\overline{x}$ be an optimal solution of the linear relaxation of formulation $(17)-(19)$. The thesis follows proving that $\overline{x}$ is a feasible solution for the linear relaxation of formulation $(7)-(10)$ and applying Proposition 1. In view of Proposition 3 it holds that $\sum_{j\in V} \overline{x}_{ij} \leq 1$. Moreover, using Proposition

2, it is easy to show that

$$\sum_{i\in S, j\in S^c} \sum_{k\in V\setminus\{i\},\ d_{ij}\leq d_{ik}} \overline{x}_{ik} \geq \sum_{i\in S} \sum_{j\in K^i(S)} \overline{x}_{ij} \geq 1.$$

Hence, $\overline{x}$ is a feasible solution of the linear relaxation of formulation (7)-(10).

□

Table 1 Average gap

n	m	**(3)-(6)** *gap*	**(17)-(19)** *gap*	n	m	**(3)-(6)** *gap*	**(17)-(19)** *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

Naturally, there are cases in which the value of an optimal solution of the linear relaxation of $(3) - (6)$ is strictly smaller than the value of an optimal solution of the linear relaxation of $(17) - (19)$. We have done several experiments in order to compare from a practical point of view the two formulations. In Table 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 number of nodes/destinations. We indicate with *gap* the value $(OPT - LB)/LB$. From the results reported in Table 1, it is highlighted first the fact that the lower bound of the set covering formulation is usually much better than the lower bound of formulation (3) − (6), second, for problems with few nodes and few destinations the optimal solution of the linear relaxation of our proposed formulation is often already an integer solution. Further additional computational results of this type are summarized in section 5.

4 Solution Methods

As discussed before, the main difficulty of the set covering formulation is represented by the huge number of constraints (18), but a considerable help may be given by the structure of the formulation. Here we propose two exact solution methods to deal with the problem.

In the first procedure, we generate the whole constraint matrix, but we take into account only a subset of its rows and we add iteratively the violated constraints. 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 (18) 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 no more violated constraints can be 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 the rows corresponding to the subsets S such that $|S^c| = 1$. Moreover, we look first for rows violated by the current solution among those that have not been already considered and eliminated by the preprocessing procedure in a previous iteration. Only at the end, we add also the currently violated rows that were dominated in one of the submatrices during the procedure.

In our second method, violated constraints are generated iteratively on the basis of the current solution looking at its support. We start with a formulation with only the inequalities (18) generated by the sets $S = \{s\}$ and $S = \{s, v_2^s\}$ and we solve the resulting linear relaxation. Given $\overline{x}$ the current solution, we define variables y using the equality in Lemma 1 and the graph $G'' = (V, A'')$ where A'' is the set of the arcs (i, j) of A such that y_{ij} is not zero. Initially, in order to find violated constraints without solving maximum flow problems, we define S as the set of the nodes of V belonging to one of the directed paths starting from the source in the graph G'' and as long as there are some destinations that are not in S, we add to the formulation the constraint (18) corresponding to S and we solve again the linear relaxation of the problem.

When all the destinations are in S and the current solution is not integer, for each source-destination pair a maximum flow problem with the current y values as capacities is solved and if a maximum flow value is less than one, we generate the constraint (18) corresponding to the minimum capacity cut.

The iterative procedure can be summarized as follows:

Step 0: Let F be the set covering formulation with only constraints (18) corresponding to $S = \{s\}$ and $S = \{s, v_2^s\}$; solve the linear relaxation of F.

Step 1: Let $\overline{x}$ be optimal solution of F, define variable y from $\overline{x}$ (using Lemma 1), define $G'' = (V, A'')$ and S as described above.

Step 2: If there is at least one destination not in S, add to F the constraint (18) relative to the set S, perform the preprocessing of the constraint matrix, solve the linear relaxation of F and go to Step 1.

Step 3: If the current solution is fractional then go to Step 4 else Stop.

Step 4: For each source-destination pair, solve the maximum flow problem with the current y as capacities.

Step 5: If all the values of the maximum flow problems are greater than or equal to 1, solve the integer problem and go to Step 1.

Step 6: For all source-destination pair such that the maximum flow value is lower than 1, add to F all the constraint (18) relative to S corresponding to the minimum capacity cut and then perform the preprocessing of the constraint matrix, solve the linear relaxation of F and go to Step 1.

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: 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 (18) is, albeit huge, finite.

5 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.

The experiments have been performed on a set of test problems with increasing number of nodes and of possible destinations; for each problem 20 different instances have been 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.

In the sequel we indicate with *Cplex* the results obtained by solving the entire problem (including all the constraints), and with *method I* and *method II* the results obtained by using the two methods described in section 4, respectively. All the methods use Cplex 9.1 to solve the resulting PL or PLI

Table 2 Average computational times (in sec.)

		Cplex		**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

problems.

We report also the number of nodes of the network n, the number of destinations m, the average execution time T in seconds, its standard deviation σ and the number of iterations It required to solve the problem. Moreover in Tables 4 and 5 we report the percentage $NS\%$ of the instances which are not solved within the time limit.

The best average solution time among the solving procedures is highlighted with a bold character. The results in Table 2 are related to networks with 5, 10 and 15 nodes combined with all the possible number 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

Table 3 Average computational times (in sec.)

		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 4 Average computational times (in sec.)

		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

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 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.

Table 5 Average gap and computational times (in sec.)

		(3)-(6)				(17)-(19)			
n	m	gap	T_{LP}	T	$NS\%$	gap	T_{LP}	T	$NS\%$
20	1	0.759	0.119	0.197		0	0.057	**0.057**	
20	5	0.581	0.243	1.235		0.015	0.268	**0.306**	
20	10	0.553	0.344	3.417		0.048	0.482	**0.694**	
20	15	0.551	0.381	8.371		0.044	0.587	**0.779**	
20	19	0.587	0.404	8.939		0.040	0.637	**0.904**	
30	1	0.678	1.506	1.753		0	1.288	**1.288**	
30	5	0.576	1.748	128.57	10	0.043	3.085	**6.404**	
30	10	0.621	2.103	388.87	15	0.038	4.736	**8.930**	
30	15	0.632	2.306	573.33	20	0.047	5.228	**7.789**	
30	29	0.661	2.701	685.98	40	0.051	6.300	**9.077**	

In Table 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 4. For the MPM problems on networks with more than 30 nodes, the first method fails to solve any of the problems 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 40 nodes and for networks with up to 50 nodes only few instances are not solved within the time limit of an hour. Instances with 100 nodes have been solved by now for just a limited number of destinations.

Table 6 Average computation times (sec)

Algorithms	n					
	10	15	20	30	35	40
AL	0.250	2.125	8.308	75.228	266.245	1768.421
EX1	0.022	0.086	1.003	25.895	145.766	1156.266
method II	0.024	0.143	0.904	9.077	48.720	109.460
Preprocessing + EX2	0.006	0.023	0.070	1.576	3.294	9.303

We also carried out some experiments to compare our formulation and method with the results available in the literature.

The first results complete those presented in Table 1 by extending the comparison to the computational times. Specifically, we report in Table 5 the average gap as defined in section 3, the time T_{LP} and T for obtaining the optimal solution of the linear relaxation and of the integer problem, respectively, for both formulations $(3)-(6)$ and $(17)-(19)$ by using method II. The results show that solving the linear relaxation of formulation $(3)-(6)$ is in general faster than solving the linear relaxation of formulation $(17)-(19)$. However, this situation is inverted whenever the gap and T are considered. Indeed, formulation $(17)-(19)$ outperforms $(3)-(6)$ both from the point of view of the quality of the optimal solution of the linear relaxation and of the computational times of the integer problem. Moreover, formulation $(3)-(6)$ fails to solve some instances with 30 nodes within the time limit.

The last set of experiments has the objective of comparing the performance of method II with respect to three exact methods reported in literature for solving the broadcasting problem. More specifically, in Table 6 we report, for an increasing number of nodes and for $m = n - 1$, the average computational times of **method II** on randomly generated test problems together with the results of the algorithm **AL** proposed in Althaus et al. (2003) and the algorithms **EX1** and **Preprocessing + EX2** both developed by Montemanni et al. (2004). We should note that in the case of these three algorithms the execution times have been normalized (as suggested in Dongarra (2004)) in order to take into account the different characteristics of the computing systems. While the superiority of our method with respect to methods AL and EX1 appears to be clear for networks having more than 20 nodes, the state-of-the-art algorithm Preprocessing + EX2 continues to have better performances. This advantage, clearly due to the incorporation of a preprocessing procedure into algorithm EX2 (see Montemanni et al. (2004)), suggests a promising way to speed up method II by developing a specialized preprocessing approach. This issue may be considered in a future work.

6 Concluding Remarks

In this paper we dealt with the Minimum Power Multicasting problem in ad-hoc networks. We proposed a Set Covering–based formulation for the problem, 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 must be done. Besides the preprocessing approach, the theoretical and polyhedral properties of the model may be investigated together with a better way of generating violated constraints. These could be future lines of investigation.

Acknowledgment

We want to thank D.Yuan for the fruitful discussion we had on an earlier version of this paper.

References

1. Althaus E, Călinescu G, Măndoiu II, Prasad S, Tchervenski N, Zelikovsky A (2003) Power Efficient Range Assignment in Ad-hoc Wireless Networks. Proceedings of the IEEE WCNC'03: 1889-1894.
2. Altinkemer K, Salman FS, Bellur P (2005) Solving the Minimum Energy Broadcasting Problem in Ad Hoc Wireless Networks by Integer Programming. Proceedings of the INOC 2005, B2: 635-642.
3. Cagalj M, Hubaux JP, Enz C (2002) Minimum-Energy Broadcast in All-Wireless Networks: NP-Completeness and Distribution Issues. Proceedings of the MOBICOM 2002.

4. Clementi A, Penna P, Silvestri R (1999) Hardness results for the power range assignment problem in packet radio networks. Lectures Notes on Computer Science. 1671: 195-208.

5. Clementi A, Crescenzi P, Penna P, Rossi G, Vocca P (2001) On the complexity of computing minimum energy consumption broadcast subgraphs. Symposium on Theoretical Aspects of Computer Science: 121-131.

6. Das AK, Marks RJ, El-Sharkawi M, Arabshani P, Gray A (2003) Minimum power broadcast trees for wireless networks: integer programming formulations. Proceedings of the IEEE INFOCOM Conference.

7. Dongarra JJ (2004) Performance of Various Computers Using Standard Linear Equations Software. Technical report CS-89-85.

8. Guo S, Yang O (2004) Optimal Tree Construction for Minimum–Energy Multicast in Wireless Ad Hoc Networks. Proceedings of the ICNP Conference.

9. Leino J (2002) Optimal Multicast Routing in Ad Hoc Networks. Technical report.

10. Leggieri V, Montemanni R, Triki C (2006) Mixed Integer Formulations for the Probabilistic Minimum Energy Broadcast Problem in Wireless Networks. Submitted.

11. Liang W (2002) Constructing Minimum-Energy Broadcast Trees in Wireless Ad Hoc Networks. Proceedings of the MOBIHOC'02. 112-122.

12. Magnanti TL, Wolsey L (1995) Optimal trees. In Ball MO et al (eds) Network Models. North Holland, Amsterdam, pp. 503-615.

13. Montemanni R, Gambardella LM (2004) Exact algorithms for the minimum power symmetric connectivity problem in wireless networks. Computers and Operations Research. 31: 1667-1680.

14. Oliveira CAS, Pardalosand PM (2005) A survey of combinatorial optimization problems in multicast routing. Computers & Operations Research 32.

15. Padberg M, Wolsey L (1983) Trees and cuts. Annals of Discrete Mathematics, 17: 511-517.

16. Rappaport T (1996) Wireless Communications: Principles and Practices. Prentice Hall.

17. Wan PJ, Călinescu G, Li XY, Frieder O (2001) Minimum energy broadcast routing in static ad hoc wireless networks. Proceedings of the IEEE Infocom: 1162-1171.

18. Wieselthier JE, Nguyen G, Ephremides A (2001) Algorithms for energy-efficient multicasting in static ad hoc networks. Mobile Networks and Application. 6: 251-263.

19. Wieselthier JE, Nguyen G, Ephremides A (2000) On the construction of energy-efficient broadcast and multicast trees in wireless networks. Proceedings of the IEEE INFOCOM 2000 Conference.

20. Yuan D (2005) An Integer Programming Approach for the Minimum-Energy Broadcast Problem in Wireless Networks. Proceedings of the INOC 2005, B2: 643-650.