Math Meth Oper Res (2011) 74:327–342
DOI 10.1007/s00186-011-0365-2

A branch and price algorithm for the minimum power multicasting problem in wireless sensor networks

R. Montemanni · V. Leggieri

Received: 7 April 2010 / Accepted: 14 July 2011 / Published online: 5 August 2011

Abstract The Minimum Power Multicast Problem arises in wireless sensor networks and consists in assigning a transmission power to each node of a network in such a way that the total power consumption over the network is minimized, while a source node is connected to a set of destination nodes, toward which a message has to be sent periodically. A new mixed integer programming model for the problem, based on paths, is presented. A practical exact algorithm based on column generation and branch and price is derived from this model. A comparison with state-of-the-art exact methods is presented, and it is shown that the new approach compares favorably to other algorithms when the number of destination nodes is moderate. Under this condition, the proposed method is able to solve previously unmanageable instances.

Keywords Minimum power topology · Wireless networks · Mixed integer linear programming · Column generation · Branch and price

1 Introduction

Wireless sensor networks are composed of a set of devices that communicate by transmitting radio signals, without using any permanently installed infrastructure.

R. Montemanni (✉)
Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Galleria 2, 6928 Manno, Switzerland
e-mail: roberto@idsia.ch

V. Leggieri
Faculty of Computer Science, Free University of Bozen-Bolzano, 39100 Bolzano, Italy
e-mail: valeria.leggieri@unibz.it

V. Leggieri
Dipartimento di Matematica, Università del Salento, 73100 Lecce, Italy

The devices, also referred to as the nodes of the network, generally use omni-directional antennae and their transmission range is determined by the power they employ in the transmission of the messages.

A device communicates directly with all the other devices located within its transmission range (single-hop communication), but it can also reach terminals located out of its range using a multi-hop communication that consists in making use of intermediate devices, acting here as routers, that relay the data packets. The devices, thus, are not only responsible for sending and receiving its own data, but they possibly forward the traffic of other terminals (see e.g. Oliveira and Pardalos 2005).

Since the very beginning of research in the area of wireless sensor networks, one of the major issues has been saving power. Such a high attention for this factor is easy to identify: the nodes of the network are usually equipped with low capacity, tiny batteries, and they have to stay alive for the longest possible time in an environment usually characterized by reduced accessibility. Wireless sensor networks are often used in commanding actuators, monitoring events or measuring values at locations difficult to be reached by people, or where a long term sensing task is required. A tight management of the power budget is imposed by all these factors. Examples of applications are habitat monitoring Mainwaring et al. (2002), civil structural monitoring Kim et al. (2007), environmental monitoring Doolin and Sitar (2005), light monitoring and control applications Li (2006) and irrigation system control in agriculture Martinez et al. (2008). Nodes (terminals) are usually characterized as low cost devices, and are expected to be deployed in a potentially inaccessible area. Recharging the sensors after the deployment might therefore not be an option, both for logistic and economical reasons. In this context, energy-efficiency becomes perhaps the most important design criterion for sensor networks, since it directly impacts on the time the network itself is kept in operation. Many sensor networks—like those we deal with—are intrinsically about dissemination of information from a well-identified *source* node. An example are those networks where a central unit has to command remote actuators, and to do this it has to rely in the wireless nodes in between. The same wireless nodes often play the double role of sensing data and routing information to the actuators, switching between the roles in a synchronous fashion, and maintaining different topologies for the two roles. Usually actuators correspond to a small subset of the wireless nodes. In this kind of applications it is critical to identify energy efficient network topologies, optimized according to the type of communications that has to be supported. In this paper, then we will concentrate on general multicasting topologies, where a piece of information has to be periodically sent from a source node to a set of target nodes of the network, called *destinations*.

The total power consumption of a network is the sum of the powers assigned to all devices and thus the Minimum Power Multicast (MPM) problem consists in minimizing this sum subject to the constraint that messages originated from the source are received by all the destinations. The MPM problem is NP-hard (see e.g. Cagalj et al. 2002; Clementi et al. 1999, 2001). It differs from the Minimum Steiner Tree problem by the so called Wireless Multicast Advantage (WMA) property (see Sect. 2 and Wieselthier et al. 2000). A particular case of MPM problem is represented by the Minimum Power Broadcast (MPB) problem, where all the nodes of the network have to receive the message transmitted from the source (all the nodes are destinations). Both the MPB and the MPM problem have attracted a wide attention in the scientific literature. Interesting approaches for the MPB and the MPM problem are discussed in Wieselthier et al. (2000, 2001), where an Incremental Power algorithm and three greedy heuristics are proposed, and in Das et al. (2003), where three different integer programming models are introduced. Contributions to the MPB problem have been given in Althaus et al. (2003), Montemanni and Gambardella (2004a,b), Montemanni et al. (2006) and Yuan (2005). Specific studies for the MPM problem have been carried out in Guo and Yang (2004), where a flow-based mixed integer program has been proposed. In Leggieri et al. (2008) (see also Leggieri 2007) the MPM problem has been expressed in terms of a set covering model and in Altinkemer et al. (2005), Bauer et al. (2008) multicommodity flow models and cut-based models have been considered. A further mixed integer programming model, the relaxation of which is used to produce lower bounds, is discussed together with some heuristic algorithms in Yuan et al. (2008).

We propose a new mixed integer linear programming formulation for the MPM problem (Sect. 3) and an exact algorithm based on column generation and branch and price (Sect. 4). Computational results of the new algorithm are reported in Sect. 5. Conclusions are drawn in Sect. 6. We start by formally defining the problem in Sect. 2.

2 Problem definition

A static wireless sensor network can be modeled as a graph. Let $G = (V, A)$ be a directed complete graph, where V represents the set of the devices and A the set of directed arcs which connect all the possible pairs (i, j), with $i, j \in V$ and $i \neq j$. A cost q_{ij} is associated with each arc (i, j): q_{ij} represents the minimum amount of power that has to be assigned to node i in order to establish a direct connection with node j. Following a simple signal propagation model proposed in Rappaport (1996), this value is proportional to the power of the distance d_{ij} with an environment-dependent exponent α whose value belongs to the interval [2, 5]. Therefore, in the sequel $q_{ij} := (d_{ij})^{\alpha}$. We observe that all the presented results remain valid with any other signal propagation model. A node s is selected to be the source of the communication, while a set $R \subseteq V \backslash \{s\}$ contains all the destination nodes. These nodes have to receive the messages periodically generated in s. Let $n = |V|$ and $r = |R|$. Notice that the nodes belonging to $V \setminus (R \cup \{s\})$ may act either as routers or they may remain isolated (unused). The target is to optimally allocate a transmission power to each node, in such a way that the reception of a message forwarded by the source to all the destinations is guaranteed. We define a *range assignment* function ρ, which assigns to each node $i \in V$ a transmitting power $\rho(i)$. We aim at minimizing the amount $\sum_{i \in V} \rho(i)$ while fulfilling the constraint that the topology resulting from the range assignment function ρ connects the source to all the destinations. Note that in any optimized solution, $\rho(i)$ must be either zero or equal to q_{ij} for some j (i.e., either node i does not transmit or uses exactly the amount of power necessary to reach a target node j).

An interesting property is that, with our settings, any signal transmitted by a node $i \in V$ to a node $j \in V$ is also received by all the nodes that are in the transmission range of i i.e., if $\rho(i) = q_{ij}$ then every node $l \in V$ such that $q_{il} \leq q_{ij}$ also receives the signal. This is the so-called Wireless Multicast Advantage (WMA) property Wieselthier et al. (2000).

Note that when $r = 1$ (one destination node only), the problem boils down to the Minimum Power Unicast problem, that is, to find the shortest path from the source to the destination over the graph G. In this special case the MPM is solvable in polynomial time (see Dijkstra 1959) and the WMA property is of no effect. We will take advantage of this property in the approach we will discuss in Sect. 4.

3 Problem formulation

The basic idea of the formulation we propose for the MPM problem is the following one: for each source-destination pair $s - h$, with $h \in R$, we have to identify a path (sequence of transmission links, i.e. arcs) on which messages will be routed (in a multi-hopping fashion). The selection of the paths toward every destination has to be made in such a way that the combination of these paths induces a multicasting structure with the minimum possible cost. Due to the WMA property, this structure is an augmented Steiner arborescence and it corresponds to an optimal solution to the original MPM problem.

We introduce binary variables x_{ij} associated with the arcs. In particular, for each arc $(i, j) \in A$ the variable x_{ij} has the following interpretation:

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

that is, $x_{ij} = 1$ if the node i is assigned enough power to reach exactly node j.

For each destination $h \in R$, we consider P^h as the set of all the paths connecting the source s with h. Moreover, we introduce binary variables z associated with paths connecting the source with each destination h as follows:

$$z_K^h := \begin{cases} 1 & \text{if path } K \in P^h \text{ is selected to connect } s \text{ with } h \\ 0 & \text{otherwise.} \end{cases}$$

We can now formulate the MPM problem as:

$$(M) \qquad \min \sum_{(i,j)\in A} q_{ij} x_{ij} \tag{1}$$

$$\text{s.t.} \qquad \sum_{(i,j)\in A} x_{ij} \le 1 \quad \forall i \in V \tag{2}$$

$$\sum_{K\in P^h} z_K^h = 1 \quad \forall h \in R \tag{3}$$

$$\sum_{(i,l)\in A: q_{il}\ge q_{ij}} x_{il} \ge \sum_{K\in P^h:(i,j)\in K} z_K^h \quad \forall (i,j) \in A, \quad \forall h \in R \tag{4}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{5}$$

$$z_K^h \in \{0,1\} \qquad \forall h \in R, \quad \forall K \in P^h \tag{6}$$

Constraints (2) guarantee that there is at most one non-zero x variable for each node i. Constraints (3) force exactly one path to be selected for each destination. Constraints (4) bind x variables with z variables, indeed if a path K is selected (that is if $z_K^h = 1$) then the transmission power of each node involved in the path has to be adequate. Constraints (5) and (6) define the domain of variables.

As already observed in Sect. 1, there is an immediate advantage in using paths instead of more complex structures working on wireless network problems: when dealing with a single path it is not necessary to consider directly the WMA.

4 An exact algorithm

It is impractical to generate all possible paths from the source s to any possible destination, as requested in formulation M. For this reason, formulation M as it is, is only interesting from a theoretical point of view. It is however possible to develop a practical exact algorithm, based on branch and price and column generation, using formulation M. The remainder of this section is devoted to the description of such an approach. We assume the reader is familiar with branch and price and column generation techniques (see Lúbbecke and Desrosiers 2005).

One can notice that in formulation M the integrality of z variables is induced by that on x variables. Therefore, if we relax the formulation substituting constraints (6) with constraints

$$0 \leq z_K^h \leq 1 \;\; \forall K \in P^h, \quad \forall h \in R \tag{7}$$

because of the integrality requirement for variables x, there exists an optimal solution of this relaxed problem with all the variables z assuming binary values.

Constraints (4) can be strengthened as follows:

$$\sum_{(i,l)\in A: q_{il} \geq q_{ij}} x_{il} \geq \sum_{K \in P_{ij}^h} z_K^h \;\; \forall (i,j) \in A, \quad \forall h \in R \tag{8}$$

where $P_{ij}^h = \left\{K \in P^h : \exists\, (i,l) \in K \text{ such that } q_{il} \geq q_{ij}\right\}$. The basic idea of these still valid constraints is that having more z variables allows dual variables to provide more precise information during the pricing phase (see Sect. 4.1). Moreover we notice that since in each optimal solution there is exactly one path selected for each destination, then the right hand side of constraints (8) is (as before) at most 1.

Moreover, constraints (3) can be relaxed to the following covering version of them:

$$\sum_{K \in P^h} z_K^h \geq 1 \quad \forall h \in R \tag{9}$$

This relaxation is a common practice when designing column generation algorithms. The replacement is introduced because set covering formulations are preferable to set partitioning formulations (like the original M). Indeed, when integrality requirements (5) are relaxed, dual variables have a smaller domain. Thanks to the combination of the objective function (1) with constraints (9) it is always possible to find an optimal solution using exactly one path for each destination.

It can finally be observed that constraints (2) can be removed from the formulation, since they are implicitly implied by the objective function (1). All these changes are implemented in the remainder of the paper.

Since the set covering model has a number of variables which is exponential in the size of the instances, to compute a valid lower bound, we may use a column generation technique. In particular, integrality conditions are relaxed and the following restricted set covering master problem (RM) is considered:

$$(RM) \qquad \min \sum_{(i,j)\in A} q_{ij} x_{ij} \tag{10}$$

$$\text{s.t.} \qquad \sum_{K\in \overline{P^h}} z_K^h \geq 1 \quad \forall h \in R \tag{11}$$

$$\sum_{(i,l)\in A: q_{il}\geq q_{ij}} x_{il} \geq \sum_{K\in \overline{P_{ij}^h}} z_K^h \; \forall (i,j) \in A, \quad \forall h \in R \tag{12}$$

$$0 \leq x_{ij} \leq 1 \quad \forall (i,j) \in A \tag{13}$$

$$0 \leq z_K^h \leq 1 \quad \forall h \in R, \quad \forall K \in \overline{P^h} \tag{14}$$

where $\overline{P^h}$ is a subset of P^h, and $\overline{P_{ij}^h} = \{K \in \overline{P^h} : \exists\, (i,l) \in K \text{ such that } q_{il} \geq q_{ij}\}$.

Initially RM includes only $|R|$ columns from z variables, one for each destination, with a coefficient 1 in the corresponding covering constraint, and 0 elsewhere. These columns correspond to the s-h paths of a heuristic solution of the problem (see Sect. 4.3). Such a solution can be obtained by any algorithm. Notice that in the most trivial case, it can be represented by the direct connection of the source with each destination in a single-hop fashion. The presence of these columns ensures that a feasible solution always exists for RM independently of the columns generated dynamically.

For each destination $h \in R$, given a current solution of the restricted master problem RM, the reduced cost of each column $K \in P^h$ is the following one:

$$c_K^h := \sum_{(i,j)\in A_K^h} -\nu_{ij}^h - \lambda^h \tag{15}$$

where ν_{ij}^h is the non-positive dual variable associated with the constraint (12) defined for destination h and arc (i,j), λ^h is the non-negative dual variable associated with the covering constraint (11) defined for destination h and $A_K^h = \{(i,j) \in A : \exists\, (i,j) \in K \text{ such that } q_{il} \geq q_{ij}\}$. At each column generation iteration the linear relaxation of RM is solved, and we search for new columns with negative reduced costs (15). Instead of explicitly computing the reduced cost of all the variables in the problem at each iteration, we solve each time several pricing problems, one for each destination $h \in R$, in order to identify one or more columns with negative reduced costs. If columns with negative reduced cost are found, they are inserted into RM and the process is iterated; otherwise, the optimal fractional solution of the linear relaxation of the RM is also an optimal solution of the linear relaxation of the original formulation M.

The main elements of the algorithm are analyzed in details in the remainder of this section.

4.1 The pricing subproblem

Each pricing problem is a special case of shortest path problem with resource constraints (SPPRC) formulated on a modification of the original graph G, with nonnegative costs $-\nu_{ij}^h$ and resources q_{ij} (the original costs) on the arcs. For a destination $h \in R$ the problem is to find the shortest path(s) from s to h such that the sum of the resources on the arcs belonging to the path is less than or equal to UB, where UB is the best upper bound available so far for the original multicasting problem. The rationale is that we are only interested in paths that might produce an improving global solution. The algorithm discussed in Irnich and Desaulniers (2006) is adopted to solve the SPPRC.

4.2 The column generation subproblem

In this section we present the column generation algorithm and its components. This algorithm is executed at each node of branch and price search-tree (see Sect. 4.3).

4.2.1 Initialization

At each non-root node of the search-tree, our algorithm initializes RM with the set of columns considered so far during the execution of the branch and price algorithm, except for those that have become infeasible because of the branching.

4.2.2 Columns management

At each iteration of the column generation algorithm we solve a pricing problem for each destination $h \in R$, and all columns with negative reduced cost that have been found (see Sect. 4.1) are inserted into the RM. Notice that following this strategy, up to $|R|$ new columns are added at each iteration. Preliminary tests suggested that this strategy is a good tradeoff between solving the pricing problem for just a few destinations at each iteration (less new columns), and adding more than one column for each destination at each iteration (more new columns).

4.2.3 Lower bounding and termination

It is well known that one of the drawbacks of column generation is the so-called *tailing-off* effect. In our case it corresponds to the situation when a lot of iterations that do not significantly modify the optimal value of RM are necessary to prove optimality. Alternative techniques that allow an earlier termination of column generation by providing enhanced valid lower bounds have been proposed. By linear programming duality, it is possible to obtain the following classic lower bound:

$$lb = \tilde{\gamma} + \sum_{h \in R} \tilde{c}^h \leq \gamma^* \tag{16}$$

where $\tilde{\gamma}$ is the cost of the last solution of RM, $\tilde{c}^h$ is the reduced cost obtained by solving the pricing subproblem for destination h [see (15)] and γ^* is the optimal solution of the subproblem associated with the current node of the branch and bound tree. When lb is greater than or equal to the best incumbent feasible solution, the current node is fathomed. In our implementation we also consider an early termination technique that stops the computation when there is no improvement for a given number of consecutive iterations, regulated by parameter *UI* (in this case the node is not fathomed, but eventually reconsidered later). This technique, coupled with the branching strategy discussed in Sect. 4.3, helps to speed up the branch and price algorithm, by eliminating the tailing-effects locally present at the nodes of the tree-search, even if it might postpone the fathoming of some nodes. If column generation is terminated because the limit on the number of consecutive non-improving iterations has been reached, the best lower bound encountered during the column generation process [according to (16)] is kept as the final lower bound of the node. For our experiments, according to some preliminary tests, we set $UI = 300$.

4.2.4 Upper bounding

At each iteration of the column generation procedure, with probability $(|V| - set)/|V|$ (where set is the number of variables set to 1 at the current search tree node) we produce a feasible (but typically overestimated) multicasting structure, by assigning to each node i the maximum transmission power q_{ij} such that the corresponding x_{ij} is greater than 0 in the last solution of RM. Then we run the *Sweep* local search (see Wieselthier et al. (2001) for a detailed description of the method) from this starting solution. In this way we obtain a heuristic solution which is often of good quality. The rationale behind the probability is to run the upper bounding procedure not too often, and only when convenient. If a new best solution has been retrieved (new UB), we also insert the corresponding paths into RM as new columns (if not already there). Notice that the use of such a heuristic upper bound strongly contribute to the performance of the overall algorithm we propose, which would otherwise converge more slowly. On the other hand, it might appear that such a strategy introduces a certain computational overhead, being the Sweep method run many times. However, experiments suggest that—in the economy of the whole algorithm— it is convenient to run Sweep often, because of its extremely short running times.

4.2.5 Stabilization

In the current implementation of the algorithm we do not embed any stabilization technique (see Rousseau et al. 2007). The algorithm might benefit from the implementation of such an approach. Namely, the tailing effects might be further reduced. However we reputed that the (marginal) speed-up would not change the ranking of the algorithms clearly emerging from the computational experiments reported in Sect. 5.2.

4.3 The branch and price algorithm

The column generation algorithm described in Sect. 4.2 is executed at every node of the search-tree in a branch and price framework. In this section we describe the search strategy, the upper bounding technique, and the branching strategies we employed within the branch and price framework.

4.3.1 Initial heuristic solution

An initial heuristic solution (providing an upper bound) is calculated as follows:

– A shortest path from source s to each destination node in R is calculated (see Dijkstra 1959), where costs are given by power requirements.
– An initial solution corresponds to the topology composed by all the arcs involved in the shortest paths, with the related transmission powers.
– The Sweep local search (see Wieselthier et al. 2001) is run to improve the topology.

The whole process runs in polynomial time and is extremely fast. Notice that the procedure is intuitively effective especially for problems with a few destinations, where the union of the shortest paths is a good approximation of the topology. As we will observe in Sect. 5.2, our branch and price approach is naturally suitable for these problems, therefore also the initial heuristic method is designed along the same direction.

4.3.2 Search strategy

We explore the search-tree according to a best-first policy, where subproblems to be expanded are ranked in non-decreasing values of the lower bounds generated during the column generation phase (see Sect. 4.2). Some preliminary experiments clearly highlighted that this strategy dominates other heuristic branching strategies (e.g. a classic depth-first search), both in terms of computation time and number of visited nodes.

4.3.3 Upper bounding

The generation of heuristic solutions is delegated to the inner column generation solver (see Sect. 4.2). Such an approach is probably not the usual one, but it is shown to be particularly efficient in our case.

4.3.4 Branching

The branching strategy we implemented in the final version of the algorithm is the most intuitive one. When the column generation problem has been solved at a node, the fractional variable x_{ij} with the highest power requirement q_{ij} is identified, and two new search-tree nodes are created. In the first one the fractional variable x_{ij} is forced to 1, in the second to 0. Usually such a strategy is regarded as not very fruitful in a branch and price framework, since intuitively tend to produce unbalanced search trees: setting a variable to 1 means to set the power of a node (strong decision), while by setting a variable to 0 we simply reduce by one the feasible transmission levels of a node (weak decision). However, after having experimented more complex branching strategy, our conclusion was that this simple strategy was the most effective for the problem under investigation. An explanation for this might be that, in our problem, setting a variable to 0 is not such a weak decision: good heuristic solutions tend to be available since the very first levels of the search tree (thanks to the upper bounding strategy implemented), and forbidding a power level for a node often leads to suboptimal Steiner arborescences with high lower bounds, that can be therefore pruned.

5 Computational results

The branch and price algorithm we propose has been coded in ANSI C, CPLEX 12.1 has been used to solve linear programs. All the experiments have been carried out on a computer equipped with an Intel Core 2 Duo 2.4GHz processor and 4GB of memory.

Section 5.1 is devoted to the description of the benchmark instances. In Sect. 5.2 we compare the branch and price algorithm we propose with a strong flow-based compact model presented in Altinkemer et al. (2005) (it is the second compact model among those discussed in Altinkemer et al. (2005), and can be regarded as a strong model in general) and with the set covering approach described in Leggieri et al. (2008) (see also Leggieri 2007), which can be regarded as a state-of-the-art method for the problem considered in this paper. The aim is to identify those situations in which the proposed algorithm is promising with respect to other methods previously discussed in the literature.

5.1 Benchmark instances

The experiments have been performed on a set of random test instances, derived according to Leggieri et al. (2008). The n nodes of the network are generated at random over a $10,000 \times 10,000$ grid. The source node s and the destination nodes in R are selected at random among the nodes generated. Parameter α—regulating signal propagation, see Sect. 2—is finally set to 2 (this choice is the most common in the literature).

5.2 Experimental results

We have run experiments with an increasing total number of nodes and, given a total number of nodes, with an increasing number of destination nodes. The impact of these two factors on the performance of the method we propose is estimated, and compared with the impact on the approach discussed in Leggieri et al. (2008) and on the adaptation to multicasting of the compact version of the second model discussed in Altinkemer et al. (2005), here used as references.

In Tables 1 and 2 we present the results of the proposed branch and price approach and of the methods described in Leggieri et al. (2008) and Altinkemer et al. (2005) for some significant combinations of $|V|$ and $|R|$ (first column). For each statistic indicator considered we present average and standard deviation over 10 runs, where a maximum computation time of 1 h is allowed for each run. Namely, the indicators are the computation time in seconds (for all the methods) and, for the branch and price, the total number of linear programming solved, the total number of nodes of the search-tree visited and the ratio between the lower bound produced at the search-tree root and the optimal solution cost. The first column of each group of results contains the number of instances that were not solved to optimality in the given maximum computation time (over the 10 instances considered).

A first conclusion from the results presented in Table 1 can be drawn about the scalability of the method we propose when the number of destination nodes is increased, while the total number of nodes is kept constant. All the numbers of destination nodes between 5 and 29 are considered when $|V| = 30$ (in steps of 2). It emerges that the branch and price method is the fastest one for problems with a small/medium number of destination nodes. The situation is different for larger destination sets. After a certain threshold (11 in our case) the set covering method discussed in Leggieri et al. (2008) becomes the fastest method. The compact model presented in Altinkemer et al. (2005) is always dominated by the other approaches, and it fails to conclude the computation for many problems in the given time. It is however interesting to comment about the computation times of the compact model discussed in Altinkemer et al. (2005). It appears that for large values of $|R|$ the method consistently either fails or is able to compute the optimal solution in a rather short time. This indicates that the performance of the method becomes significantly instance-dependent.

A second study about the scalability of the method when the number of destination nodes is kept constant while the total number of nodes of the network is increased is presented in Table 2. The number of destination nodes ($|R|$) is kept constant at 5, 9 and 13 respectively, while the total number of nodes is increased from 20 to 100 (in steps of 10). Taking into account the failures of the different approaches, it is possible to have a clear ranking on the approaches on these problems with a moderate number of destination nodes: the compact model discussed in Altinkemer et al. (2005) is dominated by the set covering method discussed in Leggieri et al. (2008), which in turn is dominated by the brach and price approach we propose.

From Tables 1 and 2 it can be observed that the lower bounds produced at the root of the search-tree are always extremely tight (notice that the linear relaxation of the strongest model discussed in Bauer et al. (2008) is able to provide equally strong lower bounds). Moreover, it can be observed that the number of linear programs solved and the number of nodes visited by the branch and price method are both directly proportional to the computation times. On the other hand, the standard deviation is always high for all the indicators and methods considered: the difficulty of each instance seems to be extremely related to the topology of the instance itself. This deserves a deeper analysis in a future study.

6 Conclusion

A new mixed integer linear model based on paths has been proposed for the minimum power multicast problem in wireless sensor networks. The model has an impractical

Table 1 Comparison with the set covering approaches described in Leggieri et al. (2008) and the compact model presented in Altinkemer et al. (2005)

$\lvert V\rvert, \lvert R\rvert$	Branch and price									Set covering (Leggieri et al. 2008)			Compact model (Altinkemer et al. 2005)		
	Fail	LPs solved		B&B nodes visited		Root lb/opt		Seconds		Fail	Seconds		Fail	Seconds	
		Avg	SD	Avg	SD	Avg	SD	Avg	SD		Avg	SD		Avg	SD
30, 5	–	25.40	29.50	2.80	3.46	0.99	0.02	1.02	1.26	–	9.04	15.65	–	48.61	63.18
30, 7	–	95.50	233.22	19.80	59.45	0.99	0.04	8.64	24.21	–	25.40	72.29	–	180.72	390.25
30, 9	–	164.00	384.16	35.40	102.56	0.99	0.04	18.81	51.09	–	25.65	68.77	–	156.69	312.83
30, 11	–	192.70	453.60	38.90	114.31	0.98	0.04	29.87	82.86	–	37.75	104.49	–	108.30	157.01
30, 13	–	226.40	485.31	42.80	115.95	0.98	0.04	54.33	149.05	–	42.17	100.27	–	72.68	51.26
30, 15	–	540.80	769.34	101.60	198.74	0.98	0.05	112.41	205.27	–	39.54	98.76	–	102.56	111.57
30, 17	–	634.70	1,505.18	197.00	582.67	0.97	0.05	230.42	651.21	–	84.58	236.73	–	55.29	31.50
30, 19	–	625.00	1,387.63	185.00	544.78	0.98	0.04	205.51	542.63	–	70.08	198.82	3	63.01	32.02
30, 21	–	1,230.00	2,399.75	419.70	879.87	0.98	0.05	656.06	1,403.12	–	32.99	80.46	2	49.70	31.32
30, 23	–	1,413.10	3,164.02	457.90	1,262.15	0.97	0.05	1,145.92	3,145.73	–	9.98	9.51	4	59.67	33.55
30, 25	–	870.20	1,526.33	169.40	364.38	0.97	0.04	625.67	1,297.13	–	11.86	11.21	3	45.89	37.29
30, 27	–	749.80	1,348.74	153.00	334.75	0.97	0.04	511.10	1,173.24	–	11.13	10.71	–	34.15	25.69
30, 29	–	1,094.80	1,774.91	225.00	419.21	0.96	0.05	1,022.58	2,087.87	–	19.52	39.20	1	22.58	12.68

Statistics over ten runs. (part 1)

Table 2 Comparison with the set covering approaches described in Leggieri et al. (2008) and the compact model presented in Altinkemer et al. (2005)

$\|V\|, \|R\|$	Branch and price									Set covering (Leggieri et al. 2008)			Compact model (Altinkemer et al. 2005)		
	Fail	LPs solved		B&B nodes visited		Root lb/opt		Seconds		Fail	Seconds		Fail	Seconds	
		Avg	SD	Avg	SD	Avg	SD	Avg	SD		Avg	SD		Avg	SD
20, 5	–	13.20	9.50	1.80	1.93	0.99	0.03	0.23	0.16	–	0.20	0.11	–	2.07	2.66
30, 5	–	25.40	29.50	2.80	3.46	0.99	0.02	1.02	1.26	–	9.04	15.65	–	48.61	63.18
40, 5	–	22.80	31.47	2.50	4.74	1.00	0.00	1.73	2.39	–	11.30	11.52	1	148.60	149.19
50, 5	–	28.50	35.43	1.80	2.53	1.00	0.00	4.16	5.72	–	54.08	61.16	2	957.93	943.36
60, 5	–	76.10	169.95	6.50	16.04	0.99	0.02	19.85	46.88	–	81.58	133.65	4	646.52	727.52
70, 5	–	156.40	285.43	33.20	74.60	1.00	0.01	64.96	126.01	3	679.24	1,140.99	8	730.47	831.61
80, 5	–	99.00	156.26	6.90	15.47	1.00	0.00	54.17	91.84	5	616.43	916.34	10	–	–
90, 5	–	254.30	377.00	40.60	81.55	0.99	0.01	277.86	502.87	5	886.01	731.48	10	–	–
100, 5	–	208.50	377.68	22.70	45.75	1.00	0.00	138.36	267.30	7	1,195.23	868.34	10	–	–
20, 9	–	34.56	31.48	4.00	5.52	0.99	0.02	0.88	0.81	–	0.27	0.14	–	2.71	2.54
30, 9	–	164.00	384.16	35.40	102.56	0.99	0.04	18.81	51.09	–	25.65	68.77	–	156.69	312.83
40, 9	–	115.50	130.77	9.80	14.97	0.99	0.02	15.68	18.75	–	29.37	29.64	2	628.53	532.78
50, 9	–	830.90	1508.29	164.60	319.77	0.99	0.02	440.06	1074.70	2	657.33	1,015.54	6	2,498.92	369.47
60, 9	–	195.80	311.62	22.90	42.45	1.00	0.00	72.78	124.64	1	386.43	312.39	7	1,724.71	1,000.04
70, 9	–	1,640.10	1,925.16	224.20	317.75	0.98	0.03	2,316.00	3,098.61	7	1,598.27	1,558.41	10	–	–
80, 9	–	3,203.90	7,575.14	226.60	510.58	0.94	0.08	2,075.83	2,627.56	8	1,265.79	836.06	10	–	–
90, 9	–	1,438.30	2,349.84	72.80	102.80	0.93	0.09	1,869.76	1,938.65	10	–	–	10	–	–
100, 9	10	–	–	–	–	–	–	–	–	10	–	–	10	–	–

Table 2 Continued

$\|V\|, \|R\|$	Branch and price									Set covering (Leggieri et al. 2008)			Compact model (Altinkemer et al. 2005)		
	Fail	LPs solved		B&B nodes visited		Root lb/opt		Seconds		Fail	Seconds		Fail	Seconds	
		Avg	SD	Avg	SD	Avg	SD	Avg	SD		Avg	SD		Avg	SD
20, 13	–	56.90	50.89	7.20	9.86	0.98	0.03	1.84	1.75	–	0.35	0.12	–	2.59	2.24
30, 13	–	226.40	485.31	42.80	115.95	0.98	0.04	54.33	149.05	–	42.17	100.27	–	72.68	51.26
40, 13	–	896.20	694.42	352.20	402.15	0.98	0.05	876.59	985.39	–	308.18	512.74	–	684.37	507.35
50, 13	–	2,638.40	3,605.82	635.40	1,218.06	0.97	0.06	1,990.13	2,397.38	2	785.85	1,076.32	10	–	–
60, 13	–	2,845.30	5,269.15	365.20	556.76	0.96	0.06	2,427.08	1,906.26	6	1,769.72	1,237.99	10	–	–
70, 13	–	5,411.60	5,827.00	98.10	80.61	0.93	0.07	2,218.33	1,705.11	7	1,939.56	976.31	10	–	–
80, 13	1	191.90	479.71	64.40	92.17	0.63	0.13	42.48	106.14	10	–	–	10	–	–
90, 13	10	–	–	–	–	–	–	–	–	10	–	–	10	–	–
100, 13	10	–	–	–	–	–	–	–	–	10	–	–	10	–	–

Statistics over ten runs. (part 2)

number of variables, but an efficient exact algorithm, based on column generation and branch and price, can be derived. Computational results show that this new algorithm compares favorably with state-of-the-art methods for a moderate number of destination nodes, therefore, under these settings, it becomes a reference approach for the considered problem.

Acknowledgments The authors would like to thank Janos Barta for the useful discussions related to the work presented in this paper, and the anonymous reviewers for the precious suggestions provided.

References

Althaus E, Călinescu G, Măndoiu II, Prasad S, Tchervenski N, Zelikovsky A (2003) Power efficient range assignment in ad-hoc wireless networks. In: Proceedings of the IEEE wireless communications and networking conference (WCNC'03), pp 1889–1894

Altinkemer K, Salman FS, Bellur P (2005) Solving the minimum energy broadcasting problem in ad hoc wireless networks by integer programming. In: Proceedings of the INOC 2005 conference, pp B2.635–B2.642

Bauer J, Haugland D, Yuan D (2008) Analysis and computational study of several integer programming formulations for minimum-energy multicasting in wireless ad hoc networks. Networks 52(2):57–68

Cagalj M, Hubaux J-P, Enz C (2002) Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues. In: Proceedings of the MOBICOM 2002

Clementi A, Crescenzi P, Penna P, Rossi G, Vocca P (2001) On the complexity of computing minimum energy consumption broadcast subgraphs. In: Symposium on theoretical aspects of computer science, pp 121–131

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, pp 195–208

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

Dijkstra EW (1959) Note on two problems in connection with graphs. Numerische Mathematik 1:269–271

Doolin DM, Sitar N (2005) Wireless sensors for wildfire monitoring. In: SPIE proceedings, pp 477–484

Guo S, Yang O (2004) Optimal tree construction for minimum-energy multicast in wireless ad hoc networks. In: Proceedings of the ICNP conference

Irnich S, Desaulniers G (2006) Shortest path problems with resource constraints. In: Desaulniers G et al (ed) Column generation. Springer, USA pp 33–65

Kim S, Pakzad S, Culler DE, Demmel J, Fenves G, Glaser S, Turon M (2007) Health monitoring of civil infrastructures using wireless sensor networks. In: IPSN proceedings, pp 254–263

Leggieri V (2007) Multicast problems in telecommunication networks. Ph.D. thesis, Università del Salento

Leggieri V, Nobili P, Triki C (2008) Minimum power multicasting problem in wireless networks. Math Methods Oper Res 68:295–311

Li S-F (2006) Wireless sensor actuator network for light monitoring and control application. In: Proceedings of the IEEE CCNC 2006 conference, pp 974–978

Lúbbecke ME, Desrosiers J (2005) Selected topics in column generation. Oper Res 53(6):1007–1023

Mainwaring A, Culler D, Polastre J, Szewczyk R, Anderson J (2002) Wireless sensor networks for habitat monitoring. In: ACM WSNA proceedings, pp 88–97

Martinez D, Blanes F, Simo J, Crespo A (2008) Wireless sensor and actuator networks: characterization and case study for confined spaces healthcare applications. In: Proceedings of the IMCSIT 2008 conference, pp 687–693

Montemanni R, Gambardella LM (2004) Exact algorithms for the minimum power symmetric connectivity problem in wireless networks. Comput Oper Res 31(10):1667–1680

Montemanni R, Gambardella LM (2004) Minimum power symmetric connectivity problem in wireless networks: a new approach. In: IEEE MWCN proceedings, pp 496–508

Montemanni R, Gambardella LM, Das AK (2006) Models and algorithms for the MPSCP: an overview. In: Wu J (ed) Handbook on theoretical and algorithmic aspects of sensor, ad hoc wireless, and peer-to-peer networks. Auerbach Publications, Boston, pp 133–146

Oliveira CAS, Pardalos PM (2005) A survey of combinatorial optimization problems in multicast routing. Comput Oper Res 32:1953–1981

Rappaport T (1996) Wireless communications: principles and practices. Prentice Hall, Englewood Cliffs

Rousseau LM, Gendreau M, Feillet D (2007) Interior point stabilization for column generation. Oper Res Lett 35(5):660–668

Wieselthier J, Nguyen G, Ephremides A (2000) On the construction of energy-efficient broadcast and multicast trees in wireless networks. In: Proceedings of the IEEE INFOCOM 2000 conference, pp 585–594

Wieselthier J, Nguyen G, Ephremides A (2001) Algorithms for energy-efficient multicasting in static ad hoc networks. Mobile Netw Appl 6:251–263

Yuan D (2005) An integer programming approach for the minimum-energy broadcast problem in wireless networks. In: Proceedings of the INOC 2005, pp B2.643–B2.650

Yuan D, Bauer J, Haugland D (2008) Minimum-energy broadcast and multicast in wireless networks: An integer programming approach and improved heuristic algorithms. Ad Hoc Netw 6(5):696–717