Minimizing power consumption while ensuring connectivity in wireless networks: a new algorithm

R. Montemanni*, L.M. Gambardella

Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)
Galleria 2, CH-6928 Manno-Lugano, Switzerland

Abstract

In this paper we consider the problem of assigning transmission powers to the nodes of a wireless network in such a way that all the nodes are connected and the total power consumption is minimized. Such a topology can be used to implement minimum power broadcast trees in ad-hoc and sensor networks.

We describe a new iterative exact algorithm and a preprocessing technique able to dramatically reduce problem dimensions.

The computational results we present finally show the effectiveness of the preprocessing technique and the efficiency of the new exact algorithm. This algorithm is shown to outperform methods recently appeared in the literature.

1 Introduction

Among the most crucial issues related to ad-hoc and sensor networks is that of operation in limited energy environments, since devices are usually equipped with battery with a limited lifetime.

Since radio signals have non-linear attenuation properties, it is very energy-consuming to transmit a signal far away. Another drawback of long-distance

*Corresponding author. Tel: +41 91 610 8568 Fax: +41 91 610 8661 Email: roberto@idsia.ch.

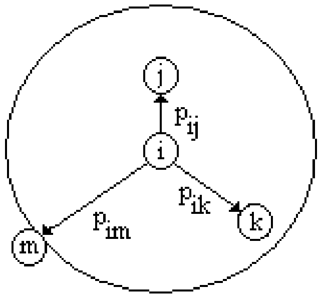

Figure 1: Communication model.

transmissions is that they tend to produce noise over the network, and for this reason they should be avoided.

The previous issues can be seen as correlated, and they can be handled together by taking advantage of the so-called *wireless multicast advantage* property (see, for example, Wieselthier et al. [11]). This property is based on the observation that, in wireless networks, devices are usually equipped with omnidirectional antennae, and for this reason multiple nodes can be reached by a single transmission. In the simple example of Figure 1, where transmission powers p are depicted, nodes j and k receive the signal originated from i and directed to m because j and k are closer to i than m, i.e. they are within the transmission range of a communication from i to m.

The property above can be used to minimize the total transmission power required to connect all the nodes. This produces a network where the sum of devices' lifetimes is maximized and, as a side effect, short distance transmissions are preferred. In particular, a well-known crucial topology problem, the *minimum power broadcast problem* (sometimes also referred to as *minimum power connectivity problem*) arises.

For a given set of nodes of a wireless network, the *minimum power broadcast* (MPB) *problem* is to assign transmission powers to the nodes in such a way that the network is connected and the total power consumption is minimized. The common assumption that no power expenditure is involved in reception/processing activities will be adopted here.

As in Althaus et al. [1], the model discussed in this paper assumes the complete knowledge of pair-wise distances between the nodes and that a communication link is established only if both nodes have a transmission range at least as big as the distance between them. Technical argumentations can be used to justify this choice. Mobility is not taken into account.

MPB has been proven to be $\mathcal{NP}$-hard in Clementi at al. [2] (see Althaus et al. [1]). Some mixed integer programming formulations which can be adapted to our problem are presented in Das et al. [4], unfortunately without any experimental result. A branch and cut algorithm based on another new integer programming formulation is proposed in Althaus et al. [1]. In Montemanni and Gambardella [8] one more integer programming formulation and some new valid reinforcing inequalities for this formulation are presented.

Heuristic approaches to the problem can be derived from those proposed in Wieselthier et al. [11], Marks II et al. [6] and Das et al. [3], where some constructing algorithms, an evolutionary approach using genetic algorithms and an ant colony system approach are respectively proposed.

In Section 2 the MPB is formally described. A new formulation for the problem is proposed in Section 3, while Section 4 is devoted to the description of a new preprocessing strategy. Section 5 presents a new exact algorithm based on the new formulation introduced in Section 3. Computational results are described in Section 6, while Section 7 contains conclusions.

2 Problem description

In order to formalize the problem, a model for signal propagation has to be selected. We adopt the model presented in Rappaport [10]. Signal power falls as $\frac{1}{d^\kappa}$, where d is the distance from the transmitter to the receiver and κ is an environment-dependent coefficient, typically between 2 and 4 (we will set $\kappa = 4$). Under this model, and adopting the usual convention (see, for example, Althaus et al. [1]) that every node has the same transmission efficiency and the same detection sensitivity threshold, the power requirement for supporting a link from node i to node j, separated by a distance d_{ij}, is then given by

$$p_{ij} = (d_{ij})^\kappa \tag{1}$$

Using the model described above, power requirements are symmetric, i.e. $p_{ij} = p_{ji}$.

MPB can be formally described as follows:
Given the set V of the nodes of the network, a *range assignment* is a func-

tion $r : V \rightarrow \mathcal{R}^+$. A *bidirectional link* between nodes i and j is said to be established under the range assignment r if $r(i) \geq p_{ij}$ and $r(j) \geq p_{ij}$. Let now $B(r)$ denote the set of all bidirectional links established under the range assignment r. MPB is the problem of finding a range assignment r minimizing $\sum_{i \in V} r(i)$, subject to the constraint that the graph $(V, B(r))$ is connected.

As suggested in Althaus et al. [1], a graph theoretical description of MPB can be given as follows:
Let $G = (V, E, p)$ be an edge-weighted graph, where V is the set of vertices corresponding to the set of nodes of the network and E is the set of edges containing all the possible (unsorted) pairs $\{i, j\}$, with $i, j \in V$, $i \neq j$. A cost p_{ij} is associated with each edge $\{i, j\}$. It corresponds to the power requirement defined by equation (1).

For a node i and a spanning tree T of G (see, for example, Kruskal [7]), let $\{i, i_T\}$ be the maximum cost edge incident to i in T, i.e. $\{i, i_T\} \in T$ and $p_{ii_T} \geq p_{ij}$ $\forall \{i, j\} \in T$. The *power cost* of a spanning tree T is then $c(T) = \sum_{i \in V} p_{ii_T}$. Since any connected graph contains a spanning tree, and a broadcast tree must be connected, MPB can be described as the problem of finding the spanning tree T with minimum power cost $c(T)$. This observation is at the basis of the integer programming formulation which will be presented in Section 3.

3 A new integer programming formulation

A weighted, directed graph $G' = (V, A, p)$ is derived from G by defining $A = \{(i, j), (j, i) | \{i, j\} \in E\} \cup \{(i, i) | i \in V\}$, i.e. for each edge in E there are the respective two (oriented) arcs in A, and a dummy arc (i, i) with $p_{ii} = 0$ is inserted for each $i \in V$. p_{ij} is defined by equation (1) when $i \neq j$. In order to describe the new integer programming formulation for MPB, we also need the following definition.

Definition 1. *Given* $(i, j) \in A$*, we define the* ancestor *of* (i, j) *as*

$$a^i_j = \begin{cases} i & \textit{if } p_{ij} = \min_{\{i,k\} \in E} \{p_{ik}\} \\ \arg\max_{k \in V} \{p_{ik} | p_{ik} < p_{ij}\} & \textit{otherwise} \end{cases} \tag{2}$$

According to this definition, (i, a^i_j) is the arc originated in node i with the highest cost such that $p_{ia^i_j} < p_{ij}$. In case an *ancestor* does not exist for arc (i,j), vertex i is returned, i.e. the dummy arc (i,i) is addressed.

In formulation IP^{MPB} a spanning tree (eventually augmented) is defined by z variables: $z_{ij}=1$ if edge $\{i,j\}$ is on the spanning tree, $z_{ij}=0$ otherwise. Variable y_{ij} is 1 when node i has a transmission power which allows it to reach node j, $y_{ij}=0$ otherwise.

$$
\begin{aligned}
(IP^{MPB}) \quad \text{Min} & \sum_{(i,j)\in A} c_{ij}y_{ij} && && (3)\\
\text{s.t.}\ & y_{ij} \le y_{ia^i_j} && \forall (i,j)\in A, a^i_j \neq i && (4)\\
& z_{ij} \le y_{ij} && \forall \{i,j\}\in E && (5)\\
& z_{ij} \le y_{ji} && \forall \{i,j\}\in E && (6)\\
& \sum_{\substack{i\in S, j\in V\setminus S,\\ \{i,j\}\in E}} z_{ij} \ge 1 && \forall S \subset V && (7)\\
& z_{ij} \in \{0,1\} && \forall \{i,j\}\in E && (8)\\
& y_{ij} \in \{0,1\} && \forall (i,j)\in A && (9)
\end{aligned}
$$

In formulation IP^{MPB} an incremental mechanism is established over y variables (i.e. transmission powers). The costs associated with y variables in the objective function (3) are given by the following formula:

$$c_{ij} = p_{ij} - p_{ia^i_j} \quad \forall (i,j)\in A \qquad (10)$$

c_{ij} is equal to the power required to establish a transmission from node i to node j (p_{ij}) minus the power required by node i to reach node a^i_j ($p_{ia^i_j}$). In Figure 2 a pictorial representation of the costs arising from the example of Figure 1 is given.

Constraints (4) realize the incremental mechanism by forcing the variable associated with arc (i,a^i_j) to assume value 1 when the variable associated with arc (i,j) has value 1, i.e. the arcs originated in the same node are activated in increasing order of p. Inequalities (5) and (6) connect the spanning tree variables z to transmission power variables y. Basically, given edge $\{i,j\}\in E$, z_{ij} can assume value 1 if and only if both y_{ij} and y_{ji} have value 1. Equations (7) state that all the vertices have to be mutually connected in the subgraph induced by z variables, i.e. the (eventually augmented) spanning tree. Constraints (8) and (9) define variable domains.

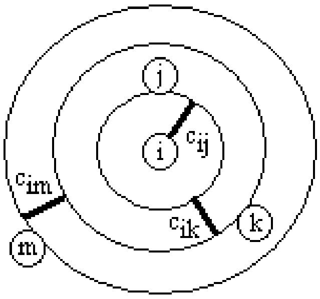

Figure 2: Costs for the mathematical formulation IP^{MPB}.

3.1 Valid inequalities

A set of valid inequalities is proposed in Montemanni and Gambardella [8] for a formulation described in the same paper. Most of these inequalities can be easily adapted to formulation IP^{MPB}, and the remainder of this section is devoted to their description in terms of formulation IP^{MPB}.

In order to describe these valid inequalities, we will refer to the subgraph of G' defined by the y variables with value 1 as G_y. Formally, $G_y = (V, A_y)$, where $A_y = \{(i,j) \in A | y_{ij} = 1$ in the current solution of $IP^{MPB}\}$.

Connectivity inequalities

Since graph G_y must be connected by definition, each node i must be able to communicate with at least another node. Its transmission power must then be sufficient to reach at least the node j which is closest to it. This can be expressed through the following set of inequalities:

$$y_{ia^i_j} = 1 \quad \forall (i,j) \in A \text{ s.t. } a^i_j = i \tag{11}$$

Bidirectional inequalities 1

For each arc $(i,j) \in A$, if $y_{ij} = 0$ and $y_{ia^i_j} = 1$ then the transmission power of node i is set to reach node a^i_j and nothing more. The only reason for node i to reach node a^i_j and nothing more is the existence of a bidirectional link on edge $\{i, a^i_j\}$ in G_y. Consequently $y_{a^i_j i}$ must be equal to 1. This is what the following set of constraints states.

$$y_{a^i_j i} \geq y_{ia^i_j} - y_{ij} \quad \forall (i,j) \in A \text{ s.t. } a^i_j \neq i \tag{12}$$

Notice that if $y_{ij} = 1$ then $y_{ia^i_j} = 1$ because of inequalities (4) and consequently in this case the constraint does not give any new contribution. If $y_{ij} = 0$ and $y_{ia^i_j} = 0$ then again the constraint does not give any new contribution.

Bidirectional inequalities 2
Consider arc $(i,j) \in A$, where j is the farthest node from i (i.e. $\nexists (i,k) \in A, a^i_k = j$) and suppose $y_{ij} = 1$. The only reason for node i to reach node j is the existence of a bidirectional link on edge $\{i,j\}$ in G_y. Consequently y_{ji} must be equal to 1, as stated by the following set of constraints.

$$y_{ji} \geq y_{ij} \ \forall (i,j) \in A \text{ s.t. } \nexists (i,k) \in A, a^i_k = j \tag{13}$$

Notice that if $y_{ij} = 0$ the constraint does not give any contribution to formulation IP^{MPB}.

Tree inequality
In order to be strongly connected, the directed graph G_y must have at least $2(|V|-1)$ arcs, as stated by the following constraint.

$$\sum_{(i,j) \in A} y_{ij} \geq 2(|V|-1) \tag{14}$$

Reachability inequalities 1
In order to define this set of valid inequalities, we need the following definitions.

Definition 2. *$G_a = (V, A_a)$ is the subgraph of the complete graph G' such that $A_a = \{(i,j)|\ a^i_j = i\}$.*

Notice that $|A_a| = |V|$ by definition.

Definition 3. *$\mathcal{R}_i = \{j \in V|\ j$ can be reached from i in $G_a\}$.*

The inequalities are based on the consideration that, since graph G_y must be strongly connected, it must be possible to reach every node j starting from each node i. This implies that at least one arc must exist between the nodes which is possible to reach from i in G_a (i.e. $\mathcal{R}_i$) and the other nodes of the graph (i.e. $V \backslash \mathcal{R}_i$). The following set of inequalities arises:

$$\sum_{\substack{(k,l)\in A,\\ k\in\mathcal{R}_i, l\in V\backslash\mathcal{R}_i}} y_{kl} \geq 1 \quad \forall i \in V \tag{15}$$

Reachability inequalities 2
In order to define this set of valid inequalities, we need the following definition.

Definition 4. $\mathcal{Q}_i = \{j \in V |\ i$ *can be reached from* j *in* $G_a\}$.

These inequalities are based on the idea that, since graph G_y must be strongly connected, it must be possible to reach every node i from every other node j of the graph. This means that at least one arc must exist between the nodes which cannot reach i in G_a (i.e. $V \backslash \mathcal{Q}_i$) and the other nodes of the graph (i.e. $\mathcal{Q}_i$). The following set of constraints arises:

$$\sum_{\substack{(l,k)\in A,\\ l\in\mathcal{Q}_i, k\in V\backslash\mathcal{Q}_i}} y_{lk} \geq 1 \quad \forall i \in V \tag{16}$$

In the remainder of this paper we will refer to formulation IP^{MPB} reinforced with inequalities (11)-(16) as IP_R^{MPB}. We will use this last, reinforced formulation because the extra inequalities strictly constrain y variables to assume quasi-feasible values (in terms of IP^{PMB}) only and this leads to much shorter solving times (see Montemanni and Gambardella [8]).

4 Preprocessing procedure

The theoretical result described in this section is used to reduce the number of edges of a problem (and consequently the number of variables of formulation IP^{MPB}).

Given a problem, we suppose we have an heuristic solution, *heu*, with cost $cost(heu)$ for it. Given a node i, all its transmission power levels that, if implemented, would induce a cost higher than $cost(heu)$ can be ignored. More formally:

Theorem 1. *If the following inequality holds*

$$2p_{ij} + \sum_{\substack{k \in V \setminus \{\{i\} \cup \{j\}\}, \\ a_l^k = k}} p_{kl} \geq cost(heu) \tag{17}$$

then edge $\{i, j\}$ can be deleted from E.

Proof. If p_{ij} is the power of node i in a solution, this means that the power of node j must be greater than or equal to $p_{ji}(= p_{ij})$, i.e. arc (j, i) must be in the solution, because otherwise there would be no reason for node i to reach node j. The sum in the left hand side of the inequality represents a lower bound for the power required by nodes different from i and j to maintain the network connected. The left hand side of inequality (17) represents then a lower bound for the total power required in case node i transmits to a power which allows it to reach node j and nothing farther. For this reason, if inequality (17) holds, edge $\{i, j\}$ can be deleted from E. □

It is important to notice that once edge $\{i, j\}$ is deleted from E, the value of the ancestor of arc (i, k) $((j, l))$ with $a_k^i = j$ $(a_l^j = i)$ has to be updated to a_j^i (a_i^j).

5 Iterative exact algorithm

It is very difficult to deal with constraints (7) of formulation IP_R^{MPB} in case of large problems. For this reason some techniques which leave some of them out have to be considered. In this section we present an iterative approach which in the beginning does not consider constraints (7) at all, and then adds them step by step only in case they are violated.

In order to speed up the approach, the following inequality should also be added to the initial integer problem IP_R^{MPB}:

$$\sum_{\{i,j\} \in E} z_{ij} \geq |V| - 1 \tag{18}$$

Inequality (18) forces the number of active z variables to be at least $|V| - 1$ - this condition is necessary in order to have a spanning tree - already at the very first iterations of the algorithm.

The integer program IP, which is defined as IP_R^{MPB} without constraints (7) but with inequality (18), is solved and the values of the z variables in the solution are examined. If the edges corresponding to z variables with value 1 form a spanning tree then the problem has been solved to optimality, otherwise constraints (19), described below, are added to the integer program and the process is repeated.

At the end of each iteration, the last available solution is examined and, if edges corresponding to z variables with value 1 generate a set $\mathcal{CC}$ of connected components with $|\mathcal{CC}| > 1$, then the following inequalities are added to the formulation:

$$\sum_{\substack{i \in C, j \in V \setminus C, \\ \{i,j\} \in E}} z_{ij} \geq 1 \quad \forall C \in \mathcal{CC} \tag{19}$$

Inequalities (19) force z variables with value 1 to connect the (elsewhere disjoint) connected components in $\mathcal{CC}$ to each other.

The algorithm we propose is summarized in Figure 3, where a pseudo-code is presented.

```
Build integer program IP;
sol := optimal solution of IP;
CC := connected components defined by z variables of sol;
While (|CC| > 1)
      Add inequalities (19) to IP;
      sol := optimal solution of IP;
      CC := connected components defined by z variables of sol;
return sol;
```

Figure 3: A pseudo-code for the exact algorithm.

6 Computational results

In this section we present some experiments we have carried out on problems randomly generated as described in Althaus et al. [1]. For each problem of size $|V|$ generated, $|V|$ points - they are the nodes of the network - have been chosen uniformly at random from a grid of size 10000 $\times$ 10000.

The preprocessing technique and the exact algorithm we propose have been implemented in ANSI C, and the callable library of ILOG CPLEX[1] 6.0 has been used to solve the integer programs encountered during the execution of the exact algorithm. Tests have been carried out on a SUNW Ultra-30 machine.

In Section 6.1 we test the performance of the preprocessing technique described in Section 4, while in Section 6.2 the exact algorithm presented in Section 5 is compared with other exact methods recently appeared in the literature.

6.1 Preprocessing procedure

In order to apply the preprocessing procedure described in Section 4, a heuristic solution to the problem must be available. For this purpose we use one of the simplest algorithms available, MST, which works by calculating the *Minimum Spanning Tree* T (see Prim [9]) on the weighted graph with costs defined by equation (1), and by assigning the power of each transmitter i to p_{ii_T}, as described near the end of Section 2. More complex algorithms, which guarantee better performance, have been proposed (see, for example, Althaus et al. [1]). It is worth to observe that if these algorithms have had been adopted, also the preprocessing technique would have produced better results than those reported in the remainder of this section.

In Table 1 we present, for different values of $|V|$, the average percentage of arcs deleted by the preprocessing procedure over fifty runs. Table 1 suggests that the preprocessing technique we propose dramatically simplifies problems. It is also interesting to observe that the percentage of arcs deleted considerably increases when the number of nodes ($|V|$) increases. This means that, when dimensions increase, the extra complexity induced by extra nodes is partially mitigated by the increased efficiency of the preprocessing technique. This should help to contain the complexity explosion faced when the number of nodes goes up.

Notwithstanding this does not appear from the tests presented, it is also important to observe that the computation times required by the preprocessing technique are always negligible (i.e. in the order of a few seconds for the biggest problems considered).

[1]http://www.cplex.com.

Table 1: Preprocessing procedure. Average performance.

$\lvert V\rvert$	Arcs deleted (%)
10	57.556
15	63.781
20	66.526
25	70.393
30	72.464
35	74.647
40	76.106
45	77.568
50	78.688

6.2 Iterative exact algorithm

In Table 2 we present the average computation times required by different exact algorithms to solve problems for different values of $|V|$. Fifty instances have been considered for each value of $|V|$.

The results in column Althaus et al. [1] are those presented in [1] (obtained on an AMD Duron 600MHz PC) multiplied by a factor of 3.2 (as suggested in Dongarra [5]). This makes them comparable with the other results of the table. From Table 2 it is possible to estimate the benefit derived from the use of the preprocessing technique described in Section 4. The computational times of the method described in Montemanni and Gambardella [8] are improved up to 17 times (for $|V| = 40$) when this technique is used. From the same columns of Table 2 we can also notice that the better performance of the preprocessing technique when the number of nodes is high (see Section 6.1) is reflected into the measure of the improvements in computation times guaranteed by the technique itself.

Table 2 also shows that the new exact algorithm we propose outperforms the other exact methods. In particular it is important to observe that the gap between the computational times of this algorithm and those of the other methods tends to increase when the number of nodes considered increases.

Table 2: Exact algorithms. Average computation times (sec).

$\|V\|$	Althaus et al. [1]	Montemanni and Gambardella [8]	Preprocessing + Montemanni and Gambardella [8]	Preprocessing + Iterative exact algorithm
10	2.144	0.192	0.078	0.052
15	18.176	0.736	0.289	0.196
20	71.040	8.576	0.715	0.601
25	188.480	33.152	4.924	2.181
30	643.200	221.408	28.908	13.481
35	2278.400	1246.304	87.357	28.172
40	15120.000	9886.080	583.541	79.544

7 Conclusion

In this paper we have considered the problem of assigning transmission powers to the nodes of a wireless network in such a way that all the nodes are connected and the total power consumption is minimized.

We have presented a new integer programming formulation for the problem and a new exact algorithm based on this formulation. A preprocessing technique, able to dramatically reduce problem dimensions, is also proposed.

Computational results have been finally presented. They prove the effectiveness of the preprocessing technique and the efficiency of the new exact algorithm. This last algorithm has been shown to outperform methods recently appeared in the literature.

Acknowledgements

This work was partially supported by the Future & Emerging Technologies unit of the European Commission through Project "BISON: Biology-Inspired techniques for Self Organization in dynamic Networks"(IST-2001-38923).

References

[1] E. Althaus, G. Călinescu, I.I. Măndoiu, S. Prasad, N. Tchervenski, and A. Zelikovsky. Power efficient range assignment in ad-hoc wireless net-

works. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'03)*, pages 1889–1894, 2003.

[2] A. Clementi, P. Penna, and R. Silvestri. On the power assignment problem in radio networks. Technical Report TR00-054, Electronic Colloquium on Computational Complexity (ECCC), 2000.

[3] A. Das, R.J. Marks II, M.A. El-Sharkawi, P. Arabshahi, and A. Gray. The minimum power broadcast problem in wireless networks: an ant colony system approach. In *Proceedings of the IEEE Workshop on Wireless Communications and Networking*, 2002.

[4] A.E. Das, R.J. Marks, M. El-Sharkawi, P. Arabshani, and A. Gray. Minimum power broadcast trees for wireless networks: integer programming formulations. In *Proceedings of the IEEE INFOCOM 2003 Conference*, 2003.

[5] J.J. Dongarra. Performance of various computers using standard linear algebra software in a fortran environment. Technical Report CS-89-85, University of Tennessee, October 2003.

[6] R.J. Marks II, A.K. Das, M. El-Sharkawi, P. Arabshani, and A. Gray. Minimum power broadcast trees for wireless networks: optimizing using the viability lemma. In *Proceedings of the IEEE Interational Symposium on Circuits and Systems (ISCAS 2002)*, 2002.

[7] J.B. Kruskal. On the shortest spanning subtree of a graph and the travelling salesman problem. *Preceedings of the American Mathematical Society*, 7:48–50, 1956.

[8] R. Montemanni and L.M. Gambardella. A new approach for the minimum power broadcast problem in wireless networks. Technical report, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, November 2003.

[9] R.C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.

[10] T. Rappaport. *Wireless Communications: Principles and Practices.* Prentice Hall, 1996.

[11] J. Wieselthier, G. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *Proceedings of the IEEE INFOCOM 2000 Conference*, pages 585–594, 2000.