Exact algorithms for the minimum power symmetric connectivity problem in wireless networks*

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 by bidirectional links and the total power consumption is minimized.

Two mixed integer programming formulations are presented together with some new valid inequalities for the polytopes associated. A preprocessing technique and two exact algorithms based on the formulations previously introduced are also proposed.

Comprehensive computational results, which show the effectiveness of the new valid inequalities and of the preprocessing technique are presented. The experiments also show that the exact approaches we propose outperform more complex methods recently appeared in the literature.

Keywords: Telecommunications, Wireless networks, Integer programming.

1 Introduction

For a given set of nodes, the *minimum power symmetric connectivity (MPSC) problem* is to assign transmission powers to the nodes of the network in such a way that all the nodes are connected and the total power consumption over the network is minimized. It is assumed that no power expenditure is involved in reception/processing activities, and that there is no mobility.

Unlike in wired networks, where a transmission from i to m generally reaches only node m, in wireless networks with omnidirectional antennae - the case considered in this paper - it is possible to reach several nodes with a single transmission. In the example of Figure 1 nodes j and k receive the signal originated from node i and directed to node 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. This property is used to minimize the total transmission power required to connect all the nodes of the network.

Saving energy is a very important issue in wireless networks since devices are usually equipped with batteries. Another good reason to keep transmission ranges small is that this limits the interference over the network.

*The 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)

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

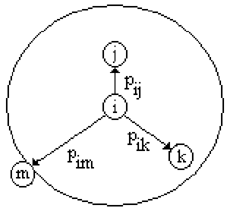

Figure 1: Communication model.

As in Althaus et al. [1], the model discussed in this paper assumes the complete knowledge of pairwise distances between the nodes and that a communication link is established only if both nodes have transmission range at least as big as the distance between them. This last assumption is justified by technical reasons.

$MPSC$ has been proven to be $\mathcal{NP}$-hard in Clementi at al. [2]. A branch and cut algorithm based on a new integer programming formulation is proposed in Althaus et al. [1]. Some mixed integer programming formulations for an optimization problem similar to $MPSC$, where a host has to broadcast a message to all the other nodes, and the bidirectionality constraint for the links is relaxed, are presented in Das et al. [4]. Different heuristic approaches for the same problem are proposed in Wieselthier et al. [11], where some constructing algorithms are described, in Marks II et al. [7], where an evolutionary approach using genetic algorithms is presented together with methods for generating initial solutions, and in Das et al. [3], where an ant colony system approach is described.

In this paper we present two new mixed integer programming formulations for the $MPSC$ problem and some new valid inequalities for the polytopes associated. We also present a new preprocessing rule and two new exact algorithms. Comprehensive computational results show the effectiveness of the new valid inequalities and of the new preprocessing techniques. Other experiments show that the new algorithms we propose outperform exact methods recently presented in the literature.

In Section 2 the problem is formally described, while in Section 3 two mixed integer programming formulations and a set of valid inequalities for the polytopes associated are described. Section 4 is devoted to the description of a new preprocessing procedure, while in Section 5 two exact algorithms, strongly based on integer programming, are proposed. Computational results are presented in Section 6, while Section 7 is devoted to conclusions.

2 Problem description

In order to represent the problem in mathematical terms, 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 a 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

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

Where β is a constant representing the energy required to set up and maintain a communication (see Heinzelman et al. [6]). This constant depends on the hardware equipping the nodes, and is independent

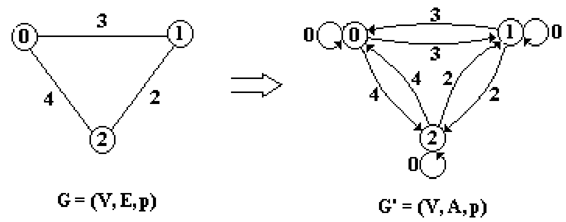

Figure 2: Example of directed graph G' derived from undirected graph G.

from the transmission distance (we will set $\beta = 0$).

Using the model described above, power requirements are symmetric, i.e. $p_{ij} = p_{ji}$. It is important to notice that the theoretical results presented in this paper remain valid also in case different signal propagation models are adopted.

$MPSC$ can be formally described as follows. Given the set V of the nodes of the network, a *range assignment* is a function $r : V \rightarrow \mathcal{R}^+$. 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. $MPSC$ 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 $MPSC$ can be given as follows. Let $G = (V, E, p)$ be a weighted, undirected complete 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 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, 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 a spanning tree is contained in any connected graph, $MPSC$ can be described as the problem of finding the spanning tree T with minimum power cost $c(T)$. This observation will be used in Section 3 for the mixed integer programming formulations presented there.

3 Mixed integer programming formulations

A weighted, directed, complete graph $G' = (V, A, p)$ is derived from G by defining $A = \{(i, j) | i, j \in V\}$, i.e. for each edge in E there are the respective two 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 the example of Figure 2, where for sack of simplicity power is not proportional to distance, a directed graph is derived from an undirected one.

In order to describe the new mathematical formulations, we need the following definition.

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

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

According to this definition, (i, a_j^i) is the arc originated in node i with the highest cost such that

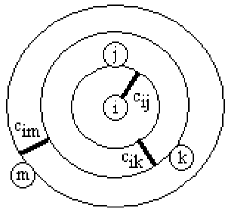

Figure 3: Costs for the mathematical formulations. c_{ij} is the power required to reach j from i, while c_{ik} is the additional power required to reach k when j is already reached from i. Analogously, c_{im} is the additional power required to reach node m from i while k is already reached.

$p_{ia^i_j} < p_{ij}$[1]. 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 the example of Figure 1, arc (i, k) is the ancestor of arc (i, m), (i, j) is the ancestor of (i, k) and the dummy arc (i, i) is returned as the ancestor of (i, j).

The two formulations are based on an incremental mechanism over the variables representing transmission powers. The costs associated with these variable in the objective functions (4) and (11) will be given by the following formula:

$$c_{ij} = p_{ij} - p_{ia^i_j} \quad \forall (i, j) \in A \tag{3}$$

c_{ij} is equal to the power required to establish a transmission from nodes i to node j (p_{ij}) minus the power required by nodes i to reach node a^i_j ($p_{ia^i_j}$). In Figure 3 the costs arising from the example of Figure 1 are depicted. As far as we are aware, this incremental mechanism has never been used before within mathematical models for the $MPSC$ problem.

3.1 Formulation $MPSC1$

The mixed integer programming formulation described in this section is inspired by those presented in Das et al. [4]. It is based based on a network flow model (see Magnanti and Wolsey [8]).

In formulation $MPSC1$ the node s from which to broadcast is the root of the spanning tree, and one unit of flow is sent from s to every other node. Variable x_{ij} (with $i \neq j$) represents the flow on arc (i, j). Variable y_{ij} (with $i \neq j$) is 1 when node i has a transmission power which allows it to reach node j, $y_{ij} = 0$ otherwise.

[1] For sack of simplicity, we have considered the (usual) case where $\forall i \in V \nexists k, l \in V$ s.t. $p_{ik} = p_{il}$. In case this is not true, the following formula, which breaks ties, has to be used in place of (2):

$$a^i_j = \begin{cases} i & \text{if } p_{ij} = \min_{k \in V}\{p_{ik}\} \\ arg\max_{k \in V} \left\{ p_{ik} \middle| \begin{pmatrix} (p_{ik} < p_{ij} \wedge (\nexists l \in V \text{ s.t. } p_{ik} = p_{il} \wedge l > k)) \\ \vee\ (p_{ij} = p_{ij} \wedge (\nexists l \in V \text{ s.t. } p_{ik} = p_{il} \wedge j > l > k)) \end{pmatrix} \right\} & \text{otherwise} \end{cases}$$

$$
\begin{aligned}
(MPSC1) \quad & \text{Min} \sum_{(i,j)\in A} c_{ij} y_{ij} && && (4)\\
& \text{s.t. } y_{ij} \le y_{ia_j^i} && \forall (i,j) \in A, a_j^i \neq i && (5)\\
& x_{ij} \le (|V|-1) y_{ij} && \forall (i,j) \in A && (6)\\
& x_{ij} \le (|V|-1) y_{ji} && \forall (i,j) \in A && (7)\\
& \sum_{(i,j)\in A} x_{ij} - \sum_{(k,i)\in A} x_{ki} = \begin{cases} |V|-1 & \text{if } i = s \\ -1 & \text{otherwise} \end{cases} && \forall i \in V && (8)\\
& x_{ij} \in \mathcal{R} && \forall (i,j) \in A && (9)\\
& y_{ij} \in \{0,1\} && \forall (i,j) \in A && (10)
\end{aligned}
$$

Constraints (5) realize the incremental mechanism by forcing the variables associated with arc (i, a_j^i) 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 (6) and (7) connect the flow variables x to y variables. Equations (8) define the flow problem, while (9)s and (10)s are domain definition constraints. We refer the interested reader to Magnanti and Wolsey [8] for a more detailed description of the spanning tree formulation behind the formulation presented above.

3.2 Formulation $MPSC2$

In the novel formulation $MPSC2$ a spanning tree is defined by z variables. Variable z_{ij} is 1 if edge $\{i,j\}$ is on the spanning tree, $z_{ij} = 0$ otherwise. Variable y_{ij} (with $i \neq j$) is 1 when node i has a transmission power which allows it to reach node j, $y_{ij} = 0$ otherwise.

$$
\begin{aligned}
(MPSC2) \quad & \text{Min} \sum_{(i,j)\in A} c_{ij} y_{ij} && && (11)\\
& \text{s.t. } y_{ij} \le y_{ia_j^i} && \forall (i,j) \in A, a_j^i \neq i && (12)\\
& z_{ij} \le y_{ij} && \forall \{i,j\} \in E && (13)\\
& z_{ij} \le y_{ji} && \forall \{i,j\} \in E && (14)\\
& \sum_{\substack{\{i,j\}\in E,\\ i\in S, j\in V\setminus S}} z_{ij} \ge 1 && \forall S \subset V && (15)\\
& z_{ij} \in \{0,1\} && \forall \{i,j\} \in E && (16)\\
& y_{ij} \in \{0,1\} && \forall (i,j) \in A && (17)
\end{aligned}
$$

Constraints (12) realize the incremental mechanism by forcing the variables associated with arc (i, a_j^i) 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 (13) and (14) connect the spanning tree variables z to y variables. Equations (15) state that all the vertices have to be mutually connected in the subgraph induced by z variables, while (16)s and (17)s are domain definition constraints.

3.3 New valid inequalities

The meaning of y variables is the same in formulations $MPSC1$ and $MPSC2$. For this reason it is possible to define structural inequalities, based on y variables, which are valid for both the formulations.

The new constraints will be used to strengthen formulations $MPSC1$ and $MPB2$. In Section 6 we will present an experimental study which shows that the computational times required to solve the two integer programs are drastically reduced when these new constraints are added to them.

In the remainder of this section 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 $A_y = \{(i,j) \in A | y_{ij} = 1 \text{ in the solution of } MPSC\}$.

Theorem 1 (Connectivity inequalities). *The set of inequalities*

$$y_{ij} = 1 \quad \forall (i,j) \in A \ s.t. \ a_j^i = i \tag{18}$$

is valid for formulations $MPSC1$ and $MPSC2$.

Proof. In order to have the graph G_y connected, each node must be able to communicate with at least one other node. Then its transmission power must be sufficient to reach at least the node which is closest to it, i.e. $y_{ia_j^i} = 1$. □

Theorem 2 (Bidirectional inequalities 1). *The set of inequalities*

$$y_{a_j^i i} \geq y_{ia_j^i} - y_{ij} \quad \forall (i,j) \in A \ s.t. \ a_j^i \neq i \tag{19}$$

is valid for formulations $MPSC1$ and $MPSC2$.

Proof. If $y_{ij} = 1$ then $y_{ia_j^i} = 1$ because of inequalities (5) and consequently in this case the constraint does not give any new contribution.

If $y_{ij} = 0$ and $y_{ia_j^i} = 0$ then again the constraint does not give any new contribution.

If $y_{ij} = 0$ and $y_{ia_j^i} = 1$ then the transmission power of node i is set to reach node a_j^i and nothing more. The only reason for node i to reach node a_j^i and nothing more is the existence of a bidirectional link on edge $\{i, a_j^i\}$ in G_y. Consequently $y_{a_j^i i}$ must be equal to 1, as stated by the constraint. □

Theorem 3 (Bidirectional inequalities 2). *The set of inequalities*

$$y_{ji} \geq y_{ij} \ \forall (i,j) \in A \ s.t. \ \nexists (i,k) \in A, a_k^i = j \tag{20}$$

is valid for formulations $MPSC1$ and $MPSC2$.

Proof. If $y_{ij} = 0$ the constraint does not give any new contribution.

If $y_{ij} = 1$ then the transmission power of node i is set in such a way to reach node j, which is the farthest node from i in G. 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 constraint. □

Theorem 4 (Tree inequality). *The inequality*

$$\sum_{(i,j) \in A} y_{ij} \geq 2(|V| - 1) \tag{21}$$

is valid for formulations $MPSC1$ and $MPSC2$.

Proof. In order to be strongly connected, the directed graph G_y must have at least $2(|V|-1)$ arcs, as stated by constraint (21). □

Theorem 5 (Strong connectivity inequalities). *The set of inequalities*

$$\sum_{j\in V \ s.t. \ (j,i)\in A} y_{ji} \geq 1 \quad \forall i \in V \tag{22}$$

is valid for formulations $MPSC1$ and $MPSC2$.

Proof. Each node must be in the transmission range of at least one other node in order to have the graph G_y strongly connected. □

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

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

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

Theorem 6 (Reachability inequalities 1). *The set of inequalities*

$$\sum_{(k,l)\in A \ s.t. \ k\in\mathcal{R}_i, l\in V\setminus\mathcal{R}_i} y_{kl} \geq 1 \quad \forall i \in V \tag{23}$$

is valid for formulations $MPSC1$ and $MPSC2$.

Proof. 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. $V\setminus\mathcal{R}_i$). □

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

Theorem 7 (Reachability inequalities 2). *The set of inequalities*

$$\sum_{(k,l)\in A \ s.t. \ k\in\mathcal{Q}_i, l\in V\setminus\mathcal{Q}_i} y_{kl} \geq 1 \quad \forall i \in V \tag{24}$$

is valid for formulations $MPSC1$ and $MPSC2$.

Proof. 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. $V\setminus\mathcal{Q}_i$) and the other nodes of the graph (i.e. $\mathcal{Q}_i$). □

3.3.1 Dominance rules

The following theorem states that when inequalities (18) are used, a simplified version of inequality (21), with a smaller number of non-zero elements, can be adopted.

Theorem 8. *The inequality*

$$\sum_{(i,j)\in A \ s.t. \ a_j^i \neq i} y_{ij} \geq |V| - 2 \tag{25}$$

is valid for formulations $MPSC1$ and $MPSC2$ and, if used together with inequalities (18), is equivalent to inequality (21).

Proof. Since inequalities (18) force exactly one y variable to be equal to 1 for each $i \in V$, the y variables set to 1 by inequalities (18) are $|V|$ in total. This observation, used within inequality (21), leads to inequality (25), which is consequently valid and equivalent to constraint (21) when used together with inequalities (18). □

A dominance of inequalities (18), (19) and (20) used together, on inequalities (18) and (22) used together is defined in the following theorem.

Theorem 9. *If inequalities (18) are in use, inequalities (22) are dominated by inequalities (19) and (20).*

Proof. Inequalities (18) imply that for each $i \in V$ there exists at least one $k \in V$ such that $y_{ik} = 1$, while inequalities (19) force the first variable $y_{a^i_j i}$ such that $y_{ij} = 0$ and $y_{ia^i_j} = 1$ to assume value 1. This forces constraints (22) to be satisfied for each $i \in V$ where $\exists k \in V$ for which $y_{ik} = 0$, since $y_{a^i_k i}$ will be 1 because of inequalities (19). If $y_{ik} = 1 \; \forall k \in V$ then inequalities (20) guarantee that constraint (22) is satisfied also for i. □

Theorem 10 states that inequalities (22) can be left out when inequalities (18), (19) and (20) are used together.

4 Preprocessing procedure

The results described in this section are used to delete some arcs of graph G' and consequently to reduce the number of variables of formulations $MPSC1$ and $MPSC2$.

We suppose an heuristic solution for the problem, heu, is available, and its cost is $cost(heu)$. All the variables that, if active, would induce a cost higher than $cost(heu)$ can be deleted from the problem.

Theorem 10. *If the following inequality holds*

$$p_{ij} + p_{ji} + \sum_{\substack{k \in V \setminus \{i,j\}, \\ a^k_l = k}} p_{kl} > cost(heu) \tag{26}$$

then arc (i, j) can be deleted from A.

Proof. Using the same intuition at the basis of the proofs of Theorems 2 and 3, we have that 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} (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 left hand side of inequality (26) represents then a lower bound for the total power required in order to maintain the network connected in case node i transmits to a power which allows it to reach node j and nothing farther. For this reason, if inequality (26) holds, arc (i, j) can be deleted from A. □

It is important to notice that once arc (i, j) is deleted from A, the value of the ancestor of node k, with $a^i_k = j$, has to be updated to a^i_j.

5 Exact algorithms

5.1 Exact algorithm $EX1$

$EX1$ solves directly $MPSC1_R$, which is formulation $MPSC1$ (see Section 3.1) reinforced with the inequalities (18), (19), (20), (23), (24) and (25). These inequalities have been chosen on the basis of the experimental results which will be presented in Section 6.1.

5.2 Exact algorithm $EX2$

$EX2$ is based on formulation $MPSC2_R$, which is formulation $MPSC2$ (see Section 3.2) reinforced with the inequalities (18), (19), (20), (23), (24) and (25). These inequalities have been chosen on the basis of the experimental results which will be presented in Section 6.1.

The idea at the basis of the method is that it is very difficult to deal directly with constraints (15) of formulation $MPSC2_R$ in case of large problems. For this reason some techniques which leave some of these constraints out have to be considered. In this section we present an iterative approach which in the beginning does not consider any constraint (15), and adds them step by step in case they are violated.

In order to speed up the approach, the following inequality should also be added to the initial integer program.

$$\sum_{\{i,j\}\in E} z_{ij} \geq |V| - 1 \tag{27}$$

Inequality (27) 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, defined as $MPSC2_R$ without constraints (15) but with inequality (27), is solved and the values of the z variables in the solution are examined. If the edges corresponding to variables with value 1 form a spanning tree then the problem has been solved to optimality, otherwise constraints (28), described below, are added to the integer program and the process is repeated.

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

$$\sum_{\substack{\{i,j\}\in E,\\ i\in C, j\in V\setminus C}} z_{ij} \geq 1 \quad \forall C \in \mathcal{CC} \tag{28}$$

Inequalities (28) force z variables to connect the (elsewhere disjoint) connected components of $\mathcal{CC}$.

Algorithm $EX2$ is summarized in Figure 4, where a pseudo-code is presented.

Build integer program IP;
sol := optimal solution of IP;
$\mathcal{CC}$:= connected components defined by z variables of sol;
While ($|\mathcal{CC}| > 1$)
    Add inequalities (28) to IP;
    sol := optimal solution of IP;
    $\mathcal{CC}$:= connected components defined by z variables of sol;
Return sol;

Figure 4: A pseudo-code for the exact algorithm $EX2$.

6 Computational results

Computational tests have been carried out on problems randomly generated as described in Althaus et al. [1]. For each problem of size $|V|$ generated, $|V|$ points (nodes) have been chosen uniformly at random from a grid of size 10000 × 10000. A SUNW Ultra-30 machine has been used for the tests, and ILOG CPLEX[2] 6.0 has been adopted to solve integer programs.

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

Table 1: Average improvements to the linear program $LR(MPSC1)$.

Extra inequalities considered	(Cost LR / Cost IP) $\lvert V\rvert = 10$	$\lvert V\rvert = 20$
none	0.26	0.21
(18)	0.60	0.48
(19)+(20)	0.26	0.21
(21)	0.35	0.32
(23)	0.33	0.31
(24)	0.34	0.31
(23)+(24)	0.38	0.38
(18)+(19)+(20)	0.82	0.63
(18)+(25)	0.63	0.52
(18)+(23)+(24)	0.71	0.63
(18)+(23)+(24)+(25)	0.71	0.64
(18)+(19)+(20)+(25)	0.84	0.65
(18)+(19)+(20)+(23)+(24)	0.91	0.78
(18)+(19)+(20)+(23)+(24)+(25)	0.91	0.78

6.1 New valid inequalities

Table 1 shows how the lower bounds for the optimal solution costs of $MPSC1$ provided by $LR(MPSC1)$, the linear relaxation of $MPSC1$, improves very much when the new valid inequalities presented in Section 3.3 are added to the linear relaxation itself. Each row of the table shows the ratios between the solution cost of the reinforced linear relaxation and the solution cost of the integer program $MPSC1$.

In Table 1 we consider problems with $|V| = 10$ and $|V| = 20$. Ten instances are generated and solved for each of these values and average results are presented.

The results in Table 1 are very promising. The valid inequalities presented in Section 3.3 are capable of an average improvement in the lower bound provided by $LR(MPSC1)$ from 0.26 to 0.91 for problems with $|V| = 10$ and from 0.21 to 0.78 for problems with $|V| = 20$.

Another interesting information which emerges from Table 1 is that the quality of the estimates incrementally increases when new valid inequalities are added. This suggests that the constraints described in Section 3.3 describe different structural characteristics of the polytope of $MPSC1$.

In Table 2 we present the computation times necessary to solve the integer program $MPSC1$ when some of the valid inequalities presented in Section 3.3 are added to it. Averages over 10 runs for problems with $|V| = 10$ and $|V| = 20$ are presented. In the column $|V| = 20$ only some significant entries are reported.

The results presented in Table 2 are interesting because different combination of families of inequalities lead to very different computation times. The introduction of some inequalities (e.g. (21)) leads to average computation times which are longer than those obtained by solving the original integer program without reinforcements (row *none*), but on the other hand, the best results are achieved when all the new inequalities are added to $MPSC1$ (last row). This confirms the indication already given by Table 1 about the mutual complementarity of the new inequalities we propose. Using all of these constraints the average computation times are reduced by a factor of 140 for problems with $|V| = 10$ and by a factor of 1919 for problems with $|V| = 20$.

Since the role of y variables - the only ones involved in the new valid inequalities - is the same in both formulations $MPSC1$ and $MPSC2$, it is reasonable to expect that the results presented in this section are valid also for $MPSC2$.

Table 2: Average solving times (sec) for some reinforced versions of formulation $MPSC1$.

Extra inequalities considered	Comp. time (sec) $\|V\| = 10$	Comp. time (sec) $\|V\| = 20$
none	20.97	8615.32
(18)	5.12	-
(19)+(20)	4.99	1050.37
(21)	34.06	-
(23)	20.92	-
(24)	18.04	-
(23)+(24)	16.26	7918.93
(18)+(19)+(20)	0.36	-
(18)+(25)	7.19	-
(18)+(23)+(24)	4.14	-
(18)+(23)+(24)+(25)	6.36	411.15
(18)+(19)+(20)+(25)	0.37	78.59
(18)+(19)+(20)+(23)+(24)	0.17	-
(18)+(19)+(20)+(23)+(24)+(25)	0.15	4.49

Table 3: Average percentage of arcs deleted.

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

6.2 Preprocessing procedure

In order to apply the preprocessing procedure described in Section 4, a heuristic solution to the problem has to be available. For this purpose we use one of the simplest algorithms available, MST, which works by calculating the *Minimum Spanning Tree* (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. It is worth to notice that if better algorithms (see, for example, Althaus et al. [1]) have had been adopted, also the preprocessing technique would have produced better results than those reported in the remainder of this section.

In Table 3 we present, for different values of $|V|$, the average percentage of arcs deleted by the preprocessing procedure over 50 runs.

Table 3 suggests that the preprocessing technique we propose dramatically simplifies problems. In particular it is interesting to observe how the percentage of deleted arcs considerably increases when the number of nodes increases. This means that, when dimensions increase, the extra complexity induced by extra nodes is partially mitigated by the efficiency increase of the preprocessing technique.

The computation time required by the preprocessing technique was always negligible (i.e. in the order of a few seconds for the biggest problems).

6.3 Exact algorithms

In Table 4 we present the average computation times required by the exact algorithms for different values of V. Fifty instances are considered for each value of $|V|$.

The results in the second column are those presented in Althaus et al. [1], obtained on an AMD Duron 600MHz PC multiplied by a factor of 3.2 (as suggested in Dongarra [5]) in order to make them comparable with the other results of the table.

Table 4: Average computation times (sec).

Algorithms	$\|V\|$						
	10	15	20	25	30	35	40
Althaus et al. [1]	2.144	18.176	71.040	188.480	643.200	2278.400	15120.000
$EX1$	0.192	0.736	8.576	33.152	221.408	1246.304	9886.080
Preprocessing + $EX1$	0.078	0.289	0.715	4.924	28.908	87.357	583.541
Preprocessing + $EX2$	0.052	0.196	0.601	2.181	13.481	28.172	79.544

Table 4 shows that the new exact algorithms we propose outperform the other methods. In particular it is important to observe that the gap between the computational times of these algorithms and those of the other methods tends to increase when the number of nodes considered increases.

The comparison of the third and fourth rows of Table 4 also highlights the benefit derived from the use of the preprocessing technique described in Section 4. The computational times of the algorithm $EX1$ are improved up to 17 times (for $|V| = 40$) when this technique is used.

7 Conclusion

The minimum power symmetric connectivity problem in wireless network has been studied in this paper. Two new mixed integer programming formulations for the problem has been proposed together with some new valid inequalities for the corresponding polytopes. Two exact algorithm based on the new formulations were also presented together with a new preprocessing technique.

Experimental results have been finally presented. They show the effectiveness of the new valid inequalities and of the new preprocessing technique. A validation for the new exact algorithms is also given. They are proven to outperform methods recently appeared in the literature.

References

[1] E. Althaus, G. Călinescu, I.I. Măndoiu, S. Prasad, N. Tchervenski, and A. Zelikovsky. Power efficient range assignment in ad-hoc wireless networks. 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.K. Das, R.J. Marks II, M. 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.K. 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] W.R. Heinzelman, A. Sinha, A. Wang, and A.P. Chandrakasan. Energy-scalable algorithms and protocols for wireless microsensor networks. In *Proceedings of ICASSP*, 2000.

[7] 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 Symphosium on Circuits and Systems (ISCAS 2002)*, 2002.

[8] T.L. Magnanti and L. Wolsey. Optimal trees. In *Network Models, Handbook in Operations Research and Management Science* (M.O. Ball et al. eds.), volume 7, pages 503–615. North-Holland, 1995.

[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.