An exact algorithm for the min-power symmetric connectivity problem in wireless networks

Roberto Montemanni, Luca Maria Gambardella
Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)
Galleria 2, CH-6928 Manno-Lugano, Switzerland
Email addresses: {*roberto,luca*}*@idsia.ch*

Technical Report IDSIA-23-03

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 of the network are connected and the total power consumption is minimized. The problem, which in the literature is referred to as the minimum-power symmetric connectivity problem, can be used to implement the broadcasting structure with minimum power consumption.

A mixed integer programming formulation is presented together with some new valid reinforcing inequalities.

Computational results show the effectiveness of the novel solving approach we propose. More complex methods recently appeared in the literature are outperformed by our technique.

I. Introduction

Some of the most crucial issues in ad-hoc and sensor networks are related with the limited energy available, since devices are usually equipped with battery with a limited lifetime.

Radio signals have non-linear attenuation properties, and for this reason it is very energy-consuming to transmit a signal far away. Long-distance transmissions also tend to produce widespread interference over the network, and for this reason they should be avoided.

It is possible to see the previous issues as correlated, and to handle them together by exploiting the so-called *wireless multicast advantage* property (see, Wieselthier et al. [1]). This property is based on the principle that in wireless networks, where devices equipped with omnidirectional antennae, multiple nodes can be reached by a single transmission. In the (simplified) example of Figure 1(a) nodes j and k are closer to node i than node m, then the signal originating in node i, and directed to node m, will be received also by nodes j and k, since they are within the transmission range of a communication from node i to node m.

For a given set of nodes, the *min-power symmetric connectivity problem* ($MPSCP$) is to assign transmission powers to the nodes of the network in such a way that all the nodes are connected by bidirectional links and the total power consumption over the network is minimized. Having bidirectional links simplifies one-hop transmission protocols by allowing acknowledgement messages to be sent back for every packet (see Althaus et al. [2]). A solution of this problem can be used to implement a minimum power broadcast tree, and for this reason we will refer to the problem as the *minimum power broadcast problem.* It is assumed that no power expenditure is involved in reception/processing activities, that a complete knowledge of pairwise distances between nodes is available, and that there is no mobility. It is important to observe that, notwithstanding these assumptions, the algorithm can be used also in a distributed/dynamic framework. In this case, at each node the algorithm can be run as a local strategy to update the transmission power on the basis of information concerning the position of (some of) the other nodes (distributed fashion). These information have to be dynamically updated (dynamic network). Following this strategy each node has constantly a (local) estimation of the network situation.

As stated in Althaus et al. [2], $MPSCP$ has been proven to be $\mathcal{NP}$-hard in Clementi at al. [3]. Some mixed integer programming formulations for a very similar 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. [2]. Many heuristic approaches have been presented. See, for example, Wieselthier et al. [1].

In this paper we present a mixed integer programming formulation and some new valid inequalities for the polytope associated. We will show that the results obtained by the branch and cut algorithm described in Althaus et al. [2] can be strongly improved by directly solving the formulation we present reinforced with our new inequalities. We will also compare the results obtained by our novel solving approach with those achieved by some methods recently presented.

The authors were 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).

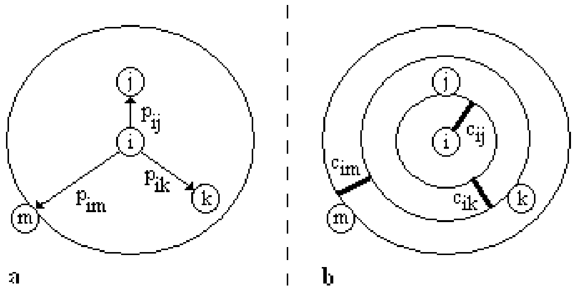

Fig. 1. Communication model (a) and costs for the mathematical formulation (b).

II. 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 [5], and used in most of the papers appeared in the literature (see, for example, Wieselthier et al. [1], Das et al. [4] and Althaus et al. [2]). According to this model, 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. [2]) 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. $p_{ij} = p_{ji}$.

We assume that there is no constraint on maximum transmission powers of nodes. However, the algorithm we discuss in this paper can be extended straightforwardly to the case when this assumption does not hold. If, for example, node i cannot reach node j even when it is transmitting to its maximum power (i.e. $d_{ij}^\kappa >$ maximum power of node i), then p_{ij} can be redefined as $+\infty$.

$MPSCP$ can be formally described as follows. Given the set V of the nodes of the network, a *range assignment* is a function $r : V \to \mathcal{R}^+$. 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. $MPSCP$ 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. [2], a graph theoretical description of $MPSCP$ can be given as follows. Let $G = (V, E, p)$ be an edge-weighted 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, $MPSCP$ 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 III for the mixed integer programming formulation presented there.

III. Mixed integer programming formulation

The mixed integer programming formulation described in this section can be seen as an evolution of the adaptation of one of those proposed in Das et al. [4]. It is based on the representation of the spanning tree problem through a network flow model (see Magnanti and Wolsey [6]).

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 order to describe the formulation, we also need the following definition.

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

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

According to this definition, (i, a_j^i) is the arc originated in node i with the highest cost such that $p_{ia_j^i} < p_{ij}$. 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 the formulation $MPSCP$, an arbitrary node s is elected the root of the spanning tree, and one unit of flow is sent from s to every other node. Variable z_{ij} represents the flow on arc (i, j). Variable y_{ij} is 1 when node i has a transmission power which allows it to reach node j; $y_{ij} = 0$ otherwise.

$$(MPSCP) \quad \text{Min} \sum_{(i,j)\in A} c_{ij} y_{ij} \tag{3}$$

$$\text{s.t. } y_{ij} \leq y_{ia_j^i} \qquad \forall (i,j) \in A, a_j^i \neq i \tag{4}$$

$$z_{ij} \leq (|V| - 1) y_{ij} \qquad \forall (i,j) \in A \tag{5}$$

$$z_{ij} \leq (|V| - 1) y_{ji} \qquad \forall (i,j) \in A \tag{6}$$

$$\sum_{(s,j)\in A} z_{sj} - \sum_{(k,s)\in A} z_{ks} = |V| - 1 \tag{7}$$

$$\sum_{(i,j)\in A} z_{ij} - \sum_{(k,i)\in A} z_{ki} = -1 \qquad \forall i \in V \backslash \{s\} \tag{8}$$

$$z_{ij} \in \mathcal{R} \qquad \forall (i,j) \in A \tag{9}$$

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

In formulation $MPSCP$ an incremental mechanism is established over y variables (i.e. transmission powers). The costs associated with y variable in the objective function (3) are then given by the following formula:

$$c_{ij} = p_{ij} - p_{ia_j^i} \quad \forall (i,j) \in A \tag{11}$$

c_{ij} is then 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_j^i ($p_{ia_j^i}$). In Figure 1(b) the costs arising from the example of Figure 1(a) are depicted.

Constraints (4) 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 (5) and (6) connect the flow variables z to y variables. Equations (7) and (8) define the flow problem, while (9)s and (10)s are domain definition constraints. The interested reader can find a more detailed description of the flow problem behind the formulation above in Magnanti and Wolsey [6].

IV. New valid inequalities for formulation $MPSCP$

In this section we will define some new valid inequalities for formulation $MPSCP$ together with some theoretical results which define dominance rules between groups of inequalities. The inequalities, which are based on structural characteristics of the problem, will be used to strengthen formulation $MPSCP$. In Section V we will present a study which shows that the computational times required to solve $MPSCP$ are drastically reduced when these new constraints are added to the formulation.

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 $A_y = \{(i, j) \in A | y_{ij} = 1 \text{ in the solution of } MPSCP\}$.

A. *Connectivity inequalities*

In order to have the graph G_y connected, 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:

$$y_{ia_j^i} = 1 \quad \forall (i,j) \in A \text{ s.t. } a_j^i = i \tag{12}$$

B. *Bidirectional inequalities 1*

For each arc $(i, j) \in A$, 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. This is what the following set of constraints states.

$$y_{a_j^i i} \geq y_{ia_j^i} - y_{ij} \quad \forall (i,j) \in A \text{ s.t. } a_j^i \neq i \tag{13}$$

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

C. Bidirectional inequalities 2

This set of inequalities is based on the same logic of those presented in Section IV-B. 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.

$$y_{ji} \geq y_{ij} \ \forall (i,j) \in A \text{ s.t. } \nexists (i,k) \in A, a^i_k = j \tag{14}$$

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

D. 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.

$$\sum_{(i,j)\in A} y_{ij} \geq 2(|V|-1) \tag{15}$$

E. Strong connectivity inequalities

Each node must be in the transmission range of at least one other node in order to have the graph G_y strongly connected. This is stated by the following set of inequalities.

$$\sum_{j \in V \text{ s.t. } (j,i)\in A} y_{ji} \geq 1 \quad \forall i \in V \tag{16}$$

F. 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: $\mathcal{R}_i = \{j \in V |\ j \text{ can be reached from } i \text{ in } G_a\}$.

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:

$$\sum_{(k,l)\in A \text{ s.t. } k \in \mathcal{R}_i, l \in V\backslash \mathcal{R}_i} y_{kl} \geq 1 \quad \forall i \in V \tag{17}$$

G. Reachability inequalities 2

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

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

These inequalities are very similar to those presented in Section IV-F. They 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:

$$\sum_{(l,k)\in A \text{ s.t. } l \in \mathcal{Q}_i, k \in V\backslash \mathcal{Q}_i} y_{lk} \geq 1 \quad \forall i \in V \tag{18}$$

H. Dominance rules

Two theoretical results on the new valid inequalities are described in this section. They define dominance rules over these inequalities.

The following theorem states that when inequalities (12) are used, a simplified version of inequality (15), with a smaller number of non-zero elements, can be adopted.

Theorem 1: The inequality

$$\sum_{(i,j)\in A \text{ s.t. } a^i_j \neq i} y_{ij} \geq |V| - 2 \tag{19}$$

is valid for the polytope $X(MPSCP)$ and, if used together with inequalities (12), is equivalent to inequality (15).

Proof: Since inequalities (12) force exactly one y variable to be equal to 1 for each $i \in V$, the y variables set to 1 by inequalities (12) are $|V|$ in total. This observation, used within inequality (15), leads to inequality (19), which is consequently valid and equivalent to constraint (15) when used together with inequalities (12). ■

TABLE I

IMPROVEMENTS TO THE LINEAR RELAXATION OF $MPSCP$. AVERAGES OVER 10 RUNS.

Linear Program	$\left(\frac{Cost(MPSCP_{LR})}{Cost(MPSCP)}\right)$ $\|V\| = 10$	$\|V\| = 20$
$MPSCP_{LR}$	0.26	0.21
$MPSCP_{LR}$+(12)	0.60	0.48
$MPSCP_{LR}$+(13)+(14)	0.26	0.21
$MPSCP_{LR}$+(15)	0.35	0.32
$MPSCP_{LR}$+(17)	0.33	0.31
$MPSCP_{LR}$+(18)	0.34	0.31
$MPSCP_{LR}$+(17)+(18)	0.38	0.38
$MPSCP_{LR}$+(12)+(13)+(14)	0.82	0.63
$MPSCP_{LR}$+(12)+(19)	0.63	0.52
$MPSCP_{LR}$+(12)+(17)+(18)	0.71	0.63
$MPSCP_{LR}$+(12)+(17)+(18)+(19)	0.71	0.64
$MPSCP_{LR}$+(12)+(13)+(14)+(19)	0.84	0.65
$MPSCP_{LR}$ + (12) + (13)+ (14) + (17) + (18)	0.91	0.78
$MPSCP_{LR}$ + (12) + (13)+ (14) + (17) + (18) + (19)	0.91	0.78

A dominance of inequalities (12), (13) and (14) used together, on inequalities (12) and (16) used together is defined in the following theorem.

Theorem 2: If inequalities (12) are in use, inequalities (16) are dominated by inequalities (13) and (14).

Proof: Inequalities (12) imply that for each $i \in V$ there exists at least one $k \in V$ such that $y_{ik} = 1$, while inequalities (13) force the first variable $y_{a_j^i i}$ such that $y_{ij} = 0$ and $y_{ia_j^i} = 1$ to assume value 1. This forces constraints (16) to be satisfied for each $i \in V$ where $\exists k \in V$ for which $y_{ik} = 0$, since $y_{a_k^i i}$ will be 1 because of inequalities (13). If $y_{ik} = 1$ $\forall k \in V$ then inequalities (14) guarantee that constraint (16) is satisfied also for i. ■

Theorem 10 states that inequalities (16) can be left out when inequalities (12), (13) and (14) are used together.

The results presented in this section will be taken into account in the experiments presented in Section V.

V. Computational results

The tests presented in this section are on problems generated as described in Althaus et al. [2]. For a given problem-size $|V|$, $|V|$ points (nodes) are chosen uniformly at random from a grid of size 10000 × 10000.

Tests have been carried out on a SUNW Ultra-30 machine, and ILOG CPLEX 6.0 (www.cplex.com) has been used to solve integer programs.

In this paper we consider networks with up to 40 nodes, but it is important to point out that when the algorithm is used within a distributed/dynamic environment (see Section I), the typical local vision of a node can be estimated in a few tens of nodes, which may reflect in a much larger network.

A. Improvements to the linear relaxation of $MPSCP$

In this section $MPSCP_{LR}$, the linear relaxation of formulation $MPSCP$, is considered. It is obtained by substituting constrains (10) with the following ones:

$$0 \leq y_{ij} \leq 1 \quad \forall (i,j) \in A \tag{20}$$

Table I shows how the lower bounds for the optimal solution costs of $MPSCP$ provided by $MPSCP_{LR}$ improves very much when the new valid inequalities presented in Section IV are added to the linear relaxation. Each row of the table shows the ratios between the solution cost of the reinforced linear relaxation ($Cost(MPN_{LR})$) and the solution cost of the integer program $MPSCP$ ($Cost(MPN)$).

In Table I 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 I are very promising. The valid inequalities presented in Section IV are capable of an average improvement in the value of the ratio $Cost(MPSCP_{LR})/Cost(MPSCP)$ from 0.26 to 0.91 for problems with $|V| = 10$ and from 0.21 to 0.78 for problems with $|V| = 20$.

TABLE II
IMPROVEMENTS TO THE COMPUTATION TIMES OF $MPSCP$. AVERAGES OVER 10 RUNS.

Integer Program	Comp. time (sec)	
	$\|V\| = 10$	$\|V\| = 20$
$MPSCP$	20.97	8615.32
$MPSCP$+(12)	5.12	-
$MPSCP$+(13)+(14)	4.99	1050.37
$MPSCP$+(15)	34.06	-
$MPSCP$+(17)	20.92	-
$MPSCP$+(18)	18.04	-
$MPSCP$+(17)+(18)	16.26	7918.93
$MPSCP$+(12)+(13)+(14)	0.36	-
$MPSCP$+(12)+(19)	7.19	-
$MPSCP$+(12)+(17)+(18)	4.14	-
$MPSCP$+(12)+(17)+(18)+(19)	6.36	411.15
$MPSCP$+(12)+(13)+(14)+(19)	0.37	78.59
$MPSCP$ + (12) + (13)+ (14) + (17) + (18)	0.17	-
$MPSCP$ + (12) + (13)+ (14) + (17) + (18) + (19)	0.15	4.49

A comparison of the different rows of Table I also suggests that inequalities (12) alone already guarantee a great improvement in the quality of the linear programming relaxation. Estimates improve even more when inequalities (12) are combined with constraints (13) and (14).

Another interesting information which emerges from Table I is that the quality of the estimates incrementally increases when new valid inequalities are added. This suggests that the constraints described in Section IV describe different structural characteristics of the problem. The only exception is represented by inequality (19), which in the experiments of Table I do not generate any new improvement when added to $MPSCP_{LP}$+(12)+(13)+(14)+(17)+(18). It is however convenient to include it anyway because it is possible to see that it is not dominated by the other constraints from a theoretical point of view, and on the other hand its contribution to the complexity of the problem is negligible, being it just one inequality with few non-zero elements.

B. Improvements to the computation times of $MPSCP$

In Table II we present the computation times necessary to solve the integer program $MPSCP$ when some of the valid inequalities presented in Section IV 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 II are interesting because different combination of families of inequalities lead to very different computation times. The introduction of some inequalities (e.g. (15)s) leads to average computation times which are longer than those obtained by solving the original integer program without reinforcements (i.e. $MPSCP$), but on the other hand, the best results are achieved when all the new inequalities are added to $MPSCP$ (last row). This confirms the indication already given by Table I 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 $|V| = 20$.

C. Comparison with other methods

In Table III we present the average computation times (over 50 runs) for problems with different values of $|V|$. We report the times (in seconds) required by the approach described in Althaus et al. [2] and to solve the integer program described in Section III reinforced with the inequalities presented in Section IV (column $MPSCP^R$).

Since we carried out our tests on a SUNW Ultra-30 machine, and we want to compare our results with those reported in Althaus et al. [2], which are obtained on an AMD Duron 600MHz PC, we divided our computation times by a factor of 3.2, as suggested in Dongarra [7]. This makes the computation times comparable.

Computation times reported in Table III indicate that the exact algorithm we suggest, i.e. solving the integer programming formulation $MPSCP$ reinforced with the inequalities presented in Section IV, clearly outperforms the (more complex) method described in Althaus et al. [2], especially for small and medium size problems.

TABLE III

Exact methods comparison. Averages over 50 runs.

$\lvert V \rvert$	Computation time (sec)	
	Althaus et al. [2]	$MPSCP^R$
10	0.67	0.06
15	5.68	0.23
20	22.20	2.68
25	58.90	10.36
30	201.00	69.19
35	712.00	389.47
40	4725.00	3089.40

VI. Conclusion

The minimum symmetric connectivity problem in wireless network has been studied in this paper. A mixed integer programming formulation has been proposed together with some new valid inequalities for the corresponding polytope.

Experimental results show that the formulation is much simpler to solve when the new valid inequalities are added to it. The resulting approach is finally proven to be faster than more complex methods recently presented.

We are currently researching on a framework where the algorithm we propose is used locally at each node of a distributed/dynamic network. The objective is to evaluate the behavior of our novel approach in this context.

References

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

[2] 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 WCNC'03*, pages 1889–1894, 2003.

[3] A. Clementi, P. Penna, and R. Silvestri. On the power assignment problem in radio networks. Technical Report TR00-054, ECCC, 2000.

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

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

[6] 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.

[7] 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.