The Minimum Power Broadcast problem in Wireless Networks: a Simulated Annealing approach

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*

Arindam K. Das
Department of Electrical Engineering, University of Washington,
Box 352500, Seattle, WA 98195, U.S.A.
Email address: *arindam@ee.washington.edu*

***Abstract*— Broadcasting in wireless networks, unlike wired networks, inherently reaches several nodes with a single transmission. For omnidirectional wireless broadcast to a node, all nodes closer will also be reached. This property can be used to compute routing trees which minimize the sum of the transmitter powers.**

In this paper we present a mixed integer programming formulation and a simulated annealing algorithm for the problem. Extensive experimental results for the heuristic approach are presented. They show that the algorithm we propose is capable of improving the results of state-of-the-art algorithms for most of the problems considered. The solutions provided by the simulated annealing algorithm can be improved by applying a very fast post-optimization procedure. This leads to the best known mean results for the problems considered.

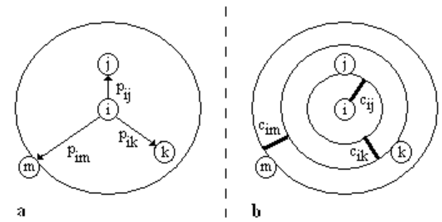

Fig. 1. **(a)** Communication model. **(b)** Costs for mathematical formulation MIP. c_{ij} is the power required to reach j from i, while c_{ik} is the additional power required to reach k when j is already reached from i. Analogously, c_{im} is the additional power required to reach node m from i while k is already reached.

I. 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-range transmissions is that they tend to produce widespread interference 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, Wieselthier et al. [1]). 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 example of Figure 1a, 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 network with an identified source node, the MPB (minimum power broadcast) 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 MPB problem in wireless networks is shown to be NP-complete in Cagalj et al. [2], and this implies that polynomial time algorithms are unlikely to exist. Some mixed integer programming formulations for the problem are described in Das et al. [3].

Wieselthier et al. [1] first observed that the so called "node based" approach is more suitable for wireless environment than the previously adopted "link-based" algorithms. They developed the *Broadcast Incremental Power (BIP)* algorithm, which is a simple sub-optimal heuristic for constructing minimum power broadcast trees in wireless networks. In this algorithm, new nodes are added to the tree on a minimum incremental cost basis, until all intended destination nodes are included. It was subsequently shown in Wan et al. [4] that the BIP algorithm has an approximation ratio between $13/3$ and 12. Other techniques that have been suggested for solving this problem include an internal nodes based broadcasting produce by Stojmenovic et al. [5], an evolutionary approach by Mark et al. [6], a localized algorithm by Cartigny et al. [7], a *swarm* based procedure by Das et al. [8] (*Ant Colony System*, ACS, see also Gambardella and Dorigo [9]). This last algorithm was hybridized within a *cluster-merge* (CM) method presented in Das et al. [10]. Some heuristic approaches for improving solutions provided by other methods was presented in Das et al. [11]. Most of these heuristic techniques are described in detail in Das et al. [12].

The rest of the paper is organized as follows. In Section II, we outline the network model. A mixed integer programming formulation, based on a novel (for the problem) incremental cost mechanism, is described in Section III. The simulated annealing paradigm and its adaptation to the MPB problem is presented in Section IV. Computational results are presented in Section V, while Section VI contains conclusions.

II. Network model

We assume a fixed N-node network with a specified source node which has to broadcast a message to all other nodes in the network. Any node can be used as a relay node to reach other nodes in the network. All nodes are assumed to have omnidirectional antennae, so that if node i transmits to node j, all nodes closer to i than j will also receive the transmission.

The signal propagation we adopt, which is the most common one in the literature (see, for example, Wieselthier et al. [1], Das et al. [12], Montemanni and Gambardella [13] and Montemanni et al. [14]), works as follows. For a node i of the network, the power required to reach another node j is given by:

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

where d_{ij} is the Euclidean distance between nodes i and j and $2 \leq \kappa \leq 4$ is the channel loss exponent.

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

We consider a centralized implementation where construction of the routing tree is done at the source node, which has complete knowledge of the locations of all nodes in the network. Finally, we assume that power expenditures due to signal reception and processing are negligible compared to signal transmission and hence the *cost* of a routing tree is equal to the sum of transmitter powers corresponding to the set of edges chosen in the tree.

III. Mixed integer linear programming formulation

In this section we present a mixed integer programming formulation for the MPB problem.

Differently from the mathematical formulations previously appeared (see, for example, Das et al. [3]), formulation MIP is based on an incremental mechanism over the variables representing transmission powers. This mechanism has been already used to formulate the *Minimum power symmetric connectivity problem* (see Montemanni and Gambardella [13], [15] and Montemanni et al. [14]), and its use guarantees a better formulation in terms of linear relaxation, which means shorter computation times to solve the original mixed integer program.

A dummy arc (i,i) with $p_{ii} = 0$ is inserted into A for each $i \in V$. p_{ij} is defined by equation (1) when $i \neq j$.

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

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

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

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

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

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(k,i)\in A} x_{ki} = \begin{cases} |V|-1 & \text{if } i = s \\ -1 & \text{otherwise} \end{cases} \quad \forall i \in V \tag{6}$$

$$x_{ij} \in \mathcal{R} \qquad \forall (i,j) \in A \tag{7}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{8}$$

Fig. 2. The mixed integer programming formulation MIP.

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 example of Figure 1a, 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 mixed integer programming formulation MIP, presented in Figure 2, is based on a network flow model (see Magnanti and Wolsey [16]). Variable x_{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.

The costs associated with these variable in the objective function (3) are connected with the incremental mechanism described above, and are given by the following formula:

$$c_{ij} = p_{ij} - p_{ia_j^i} \quad \forall (i,j) \in A \tag{9}$$

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_j^i ($p_{ia_j^i}$). In Figure 1b the costs arising from the example of Figure 1a 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) connect the flow variables x to y variables. Equations (6) define the flow problem, while (7)s and (8)s are domain definition constraints. We refer the interested reader to Magnanti and Wolsey [16] for a more detailed description of the spanning tree formulation behind the formulation presented above.

A family of facet defining valid inequalities can be added to formulation MIP. They contribute to make the linear relaxation of the formulation tighter, and consequently help to speed up solvers while solving MIP.

$$\sum_{(j,i)\in A} y_{ji} \geq 1 \quad \forall i \in V \backslash \{s\} \tag{10}$$

Inequalities (10) specify that, $\forall\ i \in V \backslash \{s\}$ there must be at least one node transmitting at a power which allows it to reach i. Every feasible solution to the MPB must present this characteristic.

IV. Simulated annealing algorithm

A. General description

Simulated annealing is a metaheuristic algorithm derived from thermodynamic principles. It has been applied originally to combinatorial optimization in Kirkpatrick et al. [17]. It can be used to find (near) minimum cost solutions[1] of difficult problems characterized by vast search spaces, on which it is impossible to obtain the optimal solution by running exact algorithms.

The search proceeds with the cost function reducing most of the time, but it is allowed to increase sometimes to permit escape from local minima which are not global minima. The analogy with thermodynamics, and specifically with the way that liquids freeze and crystalize, or metals cool and anneal, is in the strategy adopted to accept or not accept cost-increasing solutions. At high temperatures, the molecules of a liquid move freely with respect to one another. If a liquid metal is cooled quickly (i.e. quenched), it does not reach a minimum energy state but a somewhat higher energy state corresponding, in the mathematical sense, to a suboptimal solution. On the other hand, if the liquid is cooled slowly, thermal mobility is restricted. The atoms are often able to line themselves up and form a pure crystal that is completely regular. The crystal is the state of minimum energy for the system, which corresponds to the optimal solution in a mathematical optimization problem. The algorithm is based on the connection of the physical concept of temperature with the mathematical concept of the probability of accepting a cost-increasing solution. The probability will be high initially and will decrease slowly, like the temperature in the annealing process which produces the regular crystal.

In the next section a mapping of the paradigm to the MPB problem will be described.

B. *A simulated annealing algorithm for the* MPB *problem*

Each solution for the MPB problem is represented by the set of transmission powers assigned to the nodes of the network, while the fitness value of each solution (analogous to the energy of the system in the thermodynamic case) is represented by the sum of the transmission powers of all the nodes.

In our algorithm, only solutions which provide a connected broadcasting tree ($i.e.$, feasible solutions) are considered. This means that the algorithm moves from a connected broadcasting tree to another. The goal is to find a solution with minimum cost.

The starting solution for the SA algorithm is obtained from that provided by the BIP algorithm (see Wieselthier et al. [1]), a fast polynomial time constructive heuristic method. In order to provide the SA algorithm a richer search space, this solution is perturbed in such a way that it remains in the attraction-basin of the solution provided by BIP algorithm, but less deep inside the corresponding local minimum valley. This helps the algorithm to move to different local optima easily. The perturbation phase works as follows. Each node i is considered and if it is not already transmitting at its maximum possible power (i.e. to reach the node farthest away from it, subject to its eventual maximum power constraint) then, with probability p_p, its power is increased in such a way that node i can reach one more node. It is important to observe that with the given perturbation strategy, each initial solution is feasible since transmission powers can only be augmented (i.e. solution cost can only increase), starting from the values provided by BIP algorithm, which is feasible by definition.

Preliminary tests showed that the starting solution obtained as described above usually provides a better final solution than those obtained by computing (and eventually perturbing) the *Minimum Spanning Tree* (i.e. ignoring the wireless multicast advantage) using Prim's algorithm [18], or by running the *Stochastic tree generation* algorithm presented in Marks et al [6].

The SA algorithm then enters an iterative state, where the simulation of the annealing process is carried out. In this phase the broadcasting tree is repeatedly disconnected and repaired. The disconnect and repair strategies we adopt can be seen as a probabilistic version of the r-*shrink* tree-improvement algorithm described in Das et al. [11] and are explained in detail below.

In the remainder of this section, we will refer to the current solution as S_O. During the first iteration, S_O is initialized to the solution obtained by perturbing the heuristic solution provided by the BIP algorithm.

At each iteration of the simulated annealing algorithm, the following steps are carried out:

- A random node i is selected among the ones transmitting in the current solution S_O.
- The power of node i is decreased in such a way that it can reach one less node than in solution S_O (notice that this could cause node i to stop transmitting). We will refer to the node which is not reached anymore by node i as j.
- If solution S_N is still providing a feasible broadcasting tree - this happens if i was using more power than necessary in solution S_O - no operation is carried out on solution S_N.
- If solution S_N does not provide a feasible broadcasting tree anymore, the broadcasting subtree not containing node j - we will refer to as $SubT$ - is identified and one of its nodes, k, is selected, according to the following mechanism. With probability p_r, node k is selected at random among those with $p_{kj} < +\infty$, while with probability $(1 - p_r)$ it is selected as the node of $SubT$ which reconnects the broadcasting tree with the minimum increment in power.
- The new solution S_N is accepted with probability given

[1] Here and in the following we suppose the methods to be applied to minimization problems. It is trivial to adapt the descriptions for the maximization case.

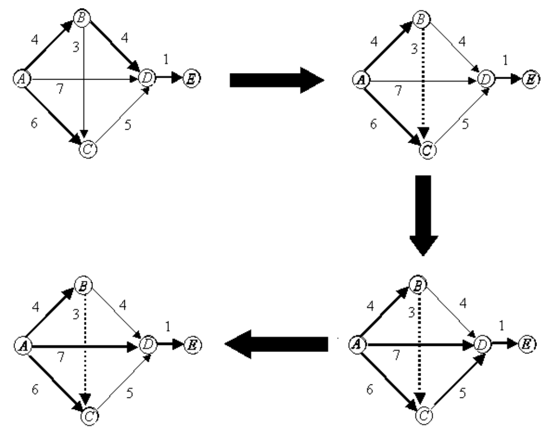

Fig. 3. Example of iteration of the SA algorithm.

by:

$$\min\left\{1, e^{-\frac{Cost(S_N)-Cost(S_O)}{t}}\right\} \tag{11}$$

where $Cost(S)$ is a function returning the total transmission power (cost) of solution S.

A (simplified) example of iteration of the SA algorithm is presented in Figure 3. In the example, for simplicity, only some arcs are considered (we suppose the others have cost $+\infty$) and transmission powers required to connect nodes (associated to arcs) are not proportional to distances. Starting from the top-left corner and moving clockwise, a transmitting node (B) of the current solution is randomly selected and its power is reduced in such a way that it can reach one node less (the power of B is reduced from 4 to 3). The subtree containing nodes D and E is now disconnected from the other nodes, and there are two ways to reconnect it (we do not take into account the reinsertion of the arc just removed): to increase the power of node A from 6 to 7 (arc (A, D)) or to increase the power of node C from 0 to 5 (arc (C, E)). In the example, node A is selected. The new connected structure so obtained has the same cost (11) of the original solution, and according to formula (11), it is accepted to become the new current solution.

It is important to observe that in the final state of the example of Figure 3, node B is still transmitting with power 3, while the solution would remain feasible even without the contribution of node B (i.e. B not transmitting) and the total cost would be smaller (8 instead of 11). This characteristic of considering also solutions which are not fully optimized is very important for the SA algorithm we propose, since this strategy permits to extensively explore the search space, looking for interesting attraction basins, without being trapped into deep local minima. A confirmation to this hypothesis will be given by some computational experiments presented in Section V.

Note that by using formula (11), not only improving solutions are accepted, but also solutions that do not improve the current one are sometimes accepted. Their acceptance probability is regulated by variable t, which is analogous to temperature in the real annealing process. Accepting non-improving solutions helps the algorithm to escape from local minima.

Temperature t, which initially assumes the value given by parameter t_{init}, is decreased every time C_T consecutive iterations are carried out without improvements to the best solution retrieved so far. This simulates the annealing process. The following rule is adopted to regulate parameter t:

$$t := \alpha\, t \tag{12}$$

where $0 < \alpha < 1$ is a user defined parameter regulating (together with parameter C_T) the speed of the annealing process.

In the beginning, the temperature t is high and most of the new configurations are accepted. As the algorithm proceeds, t is reduced until it reaches a value where non-improving configurations are all rejected. When t goes below a given threshold, T_t, the SA algorithm is stopped.

The post-optimization algorithm *Sweep* (see Wieselthier et al. [1]), which aims to reduce the power of nodes transmitting at unnecessary high power, is run after the SA algorithm terminates. It is important to observe that the computation time required by the sweep algorithm is negligible for the problems considered.

A simplified pseudo-code of the simulated annealing algorithm is presented in Figure 4.

V. Computational results

The simulated annealing algorithm was tested on 25, 50, 75, 100, 150 and 200-node networks in a 5×5 grid. In each case, 50 networks were randomly generated and the tree powers were averaged to obtain the mean tree power.

Parameter κ of equation (1) was chosen to be equal to 2 for all cases. Tests for the SA algorithm were carried out on a computer equipped with an Intel Celeron 1.3 GHz processor and 256 MB of memory.

The parameter settings adopted for the simulations are summarized in Table I. It is important to observe that these parameter values guarantee quick solving times (no more than a few seconds for the biggest problems). This is very encouraging also in the light of the result of some preliminary tests. They suggest that solving formulation MIP (reinforced with inequalities (10)) takes on average one hour for each instance of the smallest problems (the commercial solver CPLEX[2] has been used), making this last approach impossible to use for real problems. Tests not reported in this paper also suggest that the simulated annealing based algorithm is not very sensitive to the changes in parameter values, which are almost independent from the problem dimensions.

Table II aims to show the benefit of our implementation of the SA algorithm. Only networks with up to 100 nodes are considered here. We compare its results (column *Section*

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

```
S_O := BIP();
For i := 1 to N;
    If (rand(0,1) < p_p)
        increase the power of i in solution S_O;
BestS := S_O;
t := t_init;
C := 0;
While (t < T_t)
    If(C > C_T)
        t := α t;
    S_N := S_O;
    C := C + 1;
    i := random transmitting node of S_N;
    decrease the power of i in S_N
        (node j is not reached anymore by i);
    If (S_N is a feasible solution)
        S_O := S_N;
    Else
        SubT := subtree of S_N not containing j;
        If (rand(0,1) < p_r)
            k := random node in SubT;
        Else
            k := node in SubT which reconnects
                solution S_N with the minimum
                increase in power;
        If(rand(0,1) < e^(-(Cost(S_N) - Cost(S_O))/t))
            S_O := S_N;
    If ( Cost(S_N) < Cost(BestS) )
        BestS := S_N;
        C := 0;
Return BestS;
```

Fig. 4. Simulated annealing algorithm for the MPB problem.

TABLE I

PARAMETER SETTING FOR THE SIMULATED ANNEALING ALGORITHM.

Parameter	Meaning	Value
p_p	Perturbation probability (initial solution)	0.3
p_r	Random selection probability for reconnection	0.2
t_{init}	Initial temperature	0.2
C_T	Iteration interval for temperature update	30000
α	Annealing parameter	0.9
T_t	Stopping criterion (temperature threshold)	0.1

IV-B of the table) with those of a modified version of the algorithm where after each change in the solution (i.e. during each iteration), the sweep algorithm (see Section IV-B, near the end) is run, and the solution is brought to a more deep local optimum. The solution so obtained is used for the comparison with the old one in formula (11) (column *Sweep at each iteration* of the table). The superiority of the implementation of the SA algorithm we described in Section IV-B (i.e. without local optimization at each iteration) is clear. A justification for this could be that the other version of the SA algorithm tends to be trapped into local minimum, from which it does not manage to come out with a single movement. This does not happen to the SA algorithm described in Section IV-B, which also accept redundant solutions, i.e. with unnecessary transmission power at some nodes. This allows the method to float among different attraction basins, which leads to a better exploration of the search-space.

TABLE II

MEAN TREE POWERS OBTAINED BY DIFFERENT IMPLEMENTATIONS OF THE SIMULATED ANNEALING ALGORITHM.

N	Sweep at each iteration	Section IV-B
25	11.08	9.98
50	11.10	9.67
75	11.14	9.84
100	11.19	9.94

TABLE III

MEAN TREE POWERS OBTAINED BY DIFFERENT ALGORITHMS.

N	BIP	BIP + sweep	BIP + 1-shrink	ACS	CM + 1-shrink	SA	SA + sweep
25	12.46	12.14	11.25	10.21	10.23	9.98	**9.95**
50	11.67	11.45	10.68	10.04	9.90	9.67	**9.65**
75	11.63	11.37	10.67	-	9.88	9.84	**9.74**
100	11.60	11.36	10.55	-	9.87	9.94	**9.82**
150	11.31	11.07	-	-	-	10.45	**10.35**
200	11.27	11.04	-	-	-	11.01	**10.25**

It is worth to observe that running the sweep algorithm at each iteration also slows down quite considerably the whole algorithm.

A comparison of our SA approach with some state-of-the-art algorithms recently appeared is presented in Table III. In the first column the different networks considered are listed. In the remaining columns the mean tree powers for different algorithms are presented. In particular BIP, BIP followed by the sweep algorithm (see Wieselthier et al. [1]), BIP followed by the 1-shrink algorithm (see Das et al. [11]), ACS (see Das et al. [8]) and CM (see Das et al. [10]) followed by the 1-shrink algorithm (see Das et al. [11]) are considered together with the SA algorithm, which is the one discussed in this paper. The last column contains the results obtained by SA followed by the sweep algorithm. Percentage improvements in the mean tree powers over the BIP solutions are shown in Table IV. Entries of the tables marked with "-" correspond to combination for which no result is available.

From Table III and Table IV it can be seen that the SA algorithm is able to substantially improve the results achieved by the other algorithms for all the problems apart from those

TABLE IV

PERCENTAGE IMPROVEMENTS (%) IN MEAN TREE POWER OVER BIP ALGORITHM.

N	BIP + sweep	BIP + 1-shrink	ACS	CM + 1-shrink	SA	SA + sweep
25	2.57	9.71	18.06	17.90	19.90	**20.14**
50	1.89	8.48	13.93	15.17	17.14	**17.31**
75	2.24	8.25	-	15.05	15.39	**16.25**
100	2.07	9.05	-	14.91	14.31	**15.34**
150	2.12	-	-	-	7.60	**8.49**
200	2.04	-	-	-	2.31	**9.05**

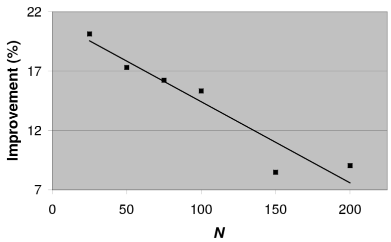

Fig. 5. Percentage improvements (%) in mean tree power of SA + sweep over BIP.

with 100 nodes. It works particularly well on small/medium size problems. On the other hand, it is comparable to the CM + 1-shrink algorithm for problems with 100 nodes. No comparison with the other algorithms (apart from BIP and BIP + sweep) is possible for problems with more than 100 nodes.

A small further improvement in the solutions provided by SA algorithm can be obtained by running the sweep algorithm, which - as observed in Section IV-B - has negligible computation times, on them. This improvement confirms the hypothesis we formulated in Section IV-B, i.e. that SA tends to produce solutions which are not fully optimized to local minima. As explained before, we believe that this property plays an important role in the performance of the algorithm we propose: SA is able to investigate the search-space searching for good attraction basins, without concentrating too much on local minima, then the sweep algorithm is able to integrate the behavior of SA, bringing these solutions down to their local minima. In fact, running sweep after SA leads to the best mean results for all of the problems, also for those with $|V| = 100$.

A final, important, observation is about the improvements over the BIP algorithm, reported in Table IV. From the table it clearly appears that the improvements over BIP algorithm guaranteed by the other algorithms decreases when the number of nodes considered increases. This phenomenon is also depicted in the chart of Figure 5, where the improvements guaranteed by SA + sweep over BIP are reported. A linear trend-line has been also added to better read the chart. A justification for the phenomenon is that the wireless advantage property is gradually lost as the density increases, since many nodes are packed in the (small) grid. This causes long-distance transmissions to be not convenient, and solutions similar to minimum spanning trees - like those provided by the BIP algorithm - tend to be close to optimality. This vanishes the effort of more complex algorithms to heavily improve the average results of BIP.

VI. Conclusion

In this paper, we have presented a new mixed integer programming formulation and a new heuristic approach for the minimum power broadcast problem in wireless networks. The heuristic algorithm is based on the simulated annealing paradigm and which proves to be very suitable for the problem under consideration.

Experimental simulations show that the simulated annealing algorithm is able to provide high quality solutions, which are significantly better than those generated by the BIP algorithm. The solutions are also better than the best results reported thus far in the literature.

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 the IEEE Infocom 2000 Conference, Tel-Aviv, Israel, March 26 - 30*, 2000, pp. 585–594.

[2] M. Cagalj, J. Hubaux, and C. Enz, "Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues," in *Proceedings of the Mobicom 2002 Conference, Atlanta, GA, September 23-28*, 2002.

[3] A. Das, R. Marks, M. El-Sharkawi, P. Arabshahi, and A. Gray, "Minimum power broadcast trees for wireless networks: integer programming formulations," in *Proceedings of the IEEE Infocom 2003 Conference, San Francisco, CA, March 30 - April 3*, 2003.

[4] P.-J. Wan, G. Călinescu, X.-Y. Li, and O. Frieder, "Minimum-energy broadcast routing in static ad hoc wireless networks," in *Proceedings of the IEEE Infocom 2001 Conference, Anchorage, AL, April 22 - 26*, 2001.

[5] I. Stojmenovic, M. Seddigh, and J. Zunic, "Internal nodes based broadcasting in wireless networks," in *Proceedings of the HICCS-34 Conference, Island of Maui, HA, January 3-6*, 2001.

[6] R. M. II, A. Das, M. El-Sharkawi, P. Arabshahi, and A. Gray, "Minimum power broadcast trees for wireless networks: optimizing using the viability lemma," in *Proceedings of the IEEE International Symphosium on Circuits & Systems, Scottsdale, AZ, May 26-29*, 2002.

[7] J. Cargigny, D. Simplot, and I. Stojmenović, "Localized minimum-energy broadcasting in ad-hoc networks," in *Proceedings of the IEEE Infocom 2003 Conference, San Francisco, CA, March 30 - April 3*, 2003.

[8] A. Das, R. Marks, 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 CAS Workshop on Wireless Communications and Networking, Pasadena, CA, September 5-6*, 2002.

[9] L. Gambardella and M. Dorigo, "Solving symmetric and asymmetric TSPs by ant colonies," in *Proceedings of the IEEE ICEC 1996 Conference, Nagoya, Japan, May 20-22*, 1996, pp. 622–627.

[10] A. Das, R. Marks, M. El-Sharkawi, P. Arabshahi, and A. Gray, "A cluster-merge algorithm for solving the minimum power broadcast problem in large scale wireless networks," in *Proceedings of the Milcom 2003 Conference, Boston, MA, October 13-16*, 2003.

[11] ——, "r-shrink: A heuristic for improving minimum power broadcast trees in wireless networks," in *Proceedings of the IEEE Globecom 2003 Conference, San Francisco, CA, December 1-5*, 2003.

[12] A. Das, "Minimum power broadcast and maximization of time-to-first-failure in broadcast wireless networks," Ph.D. dissertation, University of Washington, May 2003.

[13] R. Montemanni and L. Gambardella, "Minimum power symmetric connectivity problem in wireless networks: a new approach," in *Proceedings of the IEEE MWCN 2004 Conference, Paris, France, October 25-27*, 2004, to appear.

[14] R. Montemanni, L. Gambardella, and A. Das, "Mathematical models and exact algorithms for the min-power symmetric connectivity problem: an overview," 2004, to appear, in *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, Jie Wu ed., CRC Press.

[15] R. Montemanni and L. Gambardella, "Exact algorithms for the minimum power symmetric connectivity problem in wireless networks," *Computers and Operations Research*, to appear.

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

[17] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[18] R. Prim, "Shortest connection networks and some generalizations," *Bell System Technical Journal*, vol. 36, pp. 1389–1401, 1957.