Iterated Local Optimization for Minimum Energy Broadcast

Intae Kang and Radha Poovendran
Department of Electrical Engineering, University of Washington, Seattle, WA. 98195
email: {kangit,radha}@ee.washington.edu [*]

Abstract

In our prior work, we presented a highly effective local search based heuristic algorithm called the Largest Expanding Sweep Search (LESS) to solve the minimum energy broadcast (MEB) problem over wireless ad hoc or sensor networks. In this paper, the performance is further strengthened by using iterated local optimization (ILO) techniques at the cost of additional computational complexity. To the best of our knowledge, this implementation constitutes currently the best performing algorithm among the known heuristics for MEB. We support this claim through extensive simulation study, comparing with globally optimal solutions obtained by an integer programming (IP) solver. For small network size up to 20 nodes, which is imposed by practical limitation of the IP solver, the ILO based algorithm produces globally optimal solutions with very high frequency ($> 70\%$), *and average performance is within* 1.12% *of the optimal solution.*

1. Introduction

Since wireless ad hoc and sensor networks operate over a battery energy limited environment, the design of energy-efficient network algorithms has been an active research topic. Especially, the MINIMUM ENERGY BROADCAST, first proposed and investigated by Wieselthier et al. [26, 27], and its variant MINIMUM RANGE ASSIGNMENT [4, 5] have recently attracted significant attention from research community, since broadcast is a major communication mode for information dissemination. Both the general graph and geometric graph version of this problem have been proven to be NP-complete [3, 18]. Therefore, unless $P = NP$, it is not likely there exist exact polynomial time algorithms to solve this problem.

As an *exact method* to obtain globally optimal solutions, a mixed integer programming (IP) formulation for MINIMUM ENERGY BROADCAST has been presented by Das, et al. [7]. Albeit valuable for theoretical reasons, its practical usage is limited to only small input sizes, as is the usual case for many IP problems. To deal with practical input size we still need to rely on efficient *heuristic approximate algorithms*, which usually produce satisfactory results in polynomial time bound but without the guarantee of global optimality. Over the past years, several (greedy) *construction heuristics* have been developed such that a feasible solution is built from scratch, say, using locations of nodes as input. Some of the representative algorithms in this category are: the Broadcast Incremental Power (BIP) [26], Embedded Wireless Multicast Advantage (EWMA) [3], Broadcast Average Incremental Power (BAIP) [24], Greedy Perimeter Broadcast Efficiency (GPBE) [13], Center Oriented Broadcast Routing Algorithm (COBRA) [16], Steiner tree based heuristics by Liang [18], Ant Colony System (ACS) based approach [6], and G-Remit [25], etc.

In the general field of combinatorial optimization, the *iterative improvement heuristics* have been another class of approximate algorithms, where an initial feasible solution, possibly obtained from a construction algorithm or generated randomly, is iteratively improved. Local search and other meta-heuristics (e.g., simulated annealing and tabu search) belong to the class of iterative improvement heuristics. Many variants of local search are known to be state-of-the-art heuristics for a variety of combinatorial optimization problems; for instance, for TRAVELING SALESMAN PROBLEM (TSP), while there exist many tour construction algorithms (e.g., Christofides' heuristics, see [11] for a survey), the well-known basic k-exchange and Lin–Kernighan (LK) local search heuristics [19, 20], developed over three decades ago, still remain to be the algorithms of choice in terms of efficiency and effectiveness.

On the other hand, we observe however, much less attention has been given in this direction for MINIMUM ENERGY BROADCAST. In fact, there are a few known

[*] This research was funded in part by NSF grant ANI–0093187, ONR award #: N00014-04-1-0479 and Collaborative Technology Alliance (CTA) from ARL under DAAD19-01-2-0011. All statments and opinions are that of the authors and do not represent any position of the U.S government.

algorithms that may be considered as local search algorithms in a broader sense, including Post Sweep procedure [26], Iterative Maximum Branch Minimization (IMBM) [17], r-Shrink [6], although not specifically stated as such. Interpreting these algorithms as local search facilitates analysis of advantages or disadvantages therein, and after the underlying concepts of each algorithm are distilled as local search, we can easily extend the concepts to devise better performing heuristics. Performance-wise, above heuristics (IMBM, r-Shrink, Post-Sweep) are not impressive, resulting in local optima only comparable to tree construction heuristics (e.g., EWMA) in their solution quality.

In our prior work [15], we presented a highly effective heuristic algorithm (comparable to the LK search for TSP [20]) called the Largest Expanding Sweep Search (LESS), based on local search principle. In this paper, we extend this concept and present an iterated local optimization based heuristics relying on two neighborhood structures, which are designed to be more general and to maximize the correlation with the objective function (the meaning of which will be clarified later), still maintaining computational efficiency. One shortcoming of these approaches is that they are centralized. Unless the broadcast routing tree is computed at one location and distributed to all other nodes, their practical usage as a broadcast routing protocol may be limited. However, we believe the quest to develop ever better performing algorithms has never ended (as is the case for TSP problem). Eventually, the insights gained during the course of our investigation as to which mechanism provides trees yielding smaller cost may be beneficial in other related research activities, including distributed and localized implementations.

The remainder of this paper is organized as follows. In the next section, a system model used throughout this paper and the problem formulation will be presented. In Section 3, we briefly review the concept of iterated local optimization. In Section 4, we discuss two neighborhood structures suitable for MEB problem, which are one of the most crucial ingredients in iterated local optimization. Subsequently, the details of search strategies and the choice of initial feasible solution are discussed in Section 5. Simulation results and conclusions are provided in Section 6 and 7.

2. System Model and Problem Formulation

We assume that each node (host) in a wireless ad hoc network is equipped with an omnidirectional antenna, and acquires its location information either through GPS or other localization techniques [2]. Let $\|uv\|$ be the Euclidean distance between node u and v. The required power to support a link (u, v) from node u to v is denoted by P_{uv}, which consists of RF transmit power and signal processing power for signal transmission and reception. Due to path loss in a wireless channel, the received power level is inversely proportional to distance and attenuates as $\|uv\|^{-\alpha}$, where α denotes the path loss factor which usually satisfies $2 \leq \alpha \leq 7$. Therefore, $P_{uv} = \gamma \|uv\|^{\alpha} + \lambda$ is a relatively accurate link model assuming stationary nodes and a time-invariant (or slowly varying) channel, where γ denotes the receiver sensitivity threshold and λ is a constant power consumption factor due to signal processing at a node. To avoid undue complication of notations, we tentatively assume $\gamma = 1$ and $\lambda = 0$. We note that our algorithm works independent of an individual link model. The power consumption at node u is denoted by $p(u)$, and its corresponding transmission range is $r(u)$, which are bounded by $p(u) \leq p_{\max}$ and $r(u) \leq r_{\max}$, respectively. Therefore, the required transmit power of node u to reach node v is $p(u) = P_{uv}$.

Given a network represented by a directed weighted graph $G = (V, A, c)$ with a set $V = \{1, 2, \ldots, n\}$ of nodes with cardinality $n = |V|$ and a set A of $m = |A|$ directed edges (links), and a real valued cost function $c : A \to \mathbb{R}^1$ defined on every edges $e \in A$, the MINIMUM ENERGY BROADCAST (MEB) problem seeks to find a *s-arborescence* (served as a broadcast routing tree) $T = (V, A_T)$ rooted at source node s that can reach every node $v \in V$ in a wireless network either directly or through intermediate nodes (in multiple hops) with minimum overall power cost. Then the general graph version can be formulated as:

$$\text{minimize} \sum_{u \in V} \max_{(u,v) \in A_T} \{c_{uv}\}, \; \forall T \in \mathcal{T} \subseteq G, \quad (1)$$

where $\mathcal{T}$ denotes the family of s-arborescences (directed trees) rooted at s and spanning all nodes in V. The geometric version of MEB assumes the graph be a geometric graph; That is, a directed edge $(u, v) \in V^2$ exists if and only if $\|uv\| \leq r(u)$. Note $(u, v) \in A$ does not necessarily mean $(v, u) \in A$, since $r(u) \neq r(v)$ in general.

In what follows, we will use the following notations:

- $p(u)$ = total required power to enable communications between nodes lying at range $r(u)$ apart from u.

- P_{uv} = total required power to maintain a communication link between node u and v.

- $\delta_T(u) \triangleq \{v \mid (u, v) \in A_T\}$, the *logical neighbor* of node u in a given tree T which the set of adjacent (child) nodes of node u. The cardinality of

$\delta_T(u)$ corresponds to the outdegree of node u in the tree T, i.e., $\deg(u) = |\delta_T(u)|$. The indegree of every node is always 1, except the source node s for which the indegree is 0.

- $\pi_T(u) \triangleq \{v \mid (v,u) \in A_T\}, \forall u \in V \backslash \{s\}$ and $\pi_T(s) = \phi$, the unique parent node of node u in T. We also use the notation $\pi_T(S) \triangleq \bigcup_{v \in S} \pi_T(v), \forall S \subseteq V$.
- $\mathcal{N}_u(v) \triangleq \{w \mid 0 \leq \|uw\| \leq \|uv\|,\ w \in V\}$, the *physical neighbor* of node u that is the set of all nodes within the range $r(u) = \|uv\|$.
- $\Pi_{s \to u}$ = the set of nodes lying on a directed path from the source s to node u, where $s, u \in \Pi_{s \to u}$.

3. Iterated Local Optimization (ILO)

In this section, a brief background on iterated local optimization is presented. Excellent references on basic local search can be found in [11, 22]. For more recent developments in meta-heuristics, readers are referred to [8]. Given an optimization problem with instances $(\mathcal{F}, c)$ where $\mathcal{F}$ is the domain of feasible points; $c : \mathcal{F} \to \mathbb{R}^1$ is the cost function, a *neighborhood* is a mapping $N : \mathcal{F} \longrightarrow 2^{\mathcal{F}}$ defined for every instance in $\mathcal{F}$. Then, a feasible solution $x \in \mathcal{F}$ is called *locally optimal* with respect to N (or simply locally optimal whenever N is understood by context), if $c(x) \leq c(y)$, $\forall y \in N(x)$, and is called *globally optimal*, if $c(x) \leq c(y)$, $\forall y \in \mathcal{F}$. The 'standard' *local search* algorithm for a minimization problem can be written as:

```
procedure Local Search($\mathcal{F}, c, N$)
$s^* \longleftarrow$ GenerateInitialFeasibleSolution
while $\exists s \in N(s^*)$ such that $c(s) < c(s^*)$
    $s^* \longleftarrow s$
end
return $s^*$
```

Figure 1. Pseudo-code of a standard local search procedure.

which shows that, after an initial feasible solution is generated by a predefined feasible solution generator GenerateInitialFeasibleSolution, this solution is iteratively refined until there is no further improvement within the neighborhood. The output of this procedure is a local optimum s^*, which is heavily dependent on the neighborhood N and the initial feasible solution. In general, the larger the neighborhood size, the better the quality of the local optima, i.e., if s_1^* =LocalSearch$(\mathcal{F}, c, N_1)$, s_2^* =LocalSearch$(\mathcal{F}, c, N_2)$, and $N_1 \supseteq N_2$, it is highly likely that $c(s_1^*) \leq c(s_2^*)$, although it is not always guaranteed.

```
procedure Iterated Local Optimization($\mathcal{F}, c, N_1, N_2$)
$s_0 \longleftarrow$ GenerateInitialFeasibleSolution
$s^* \longleftarrow$ LocalSearch($\mathcal{F}, c, N_1$)
repeat
    $s' \longleftarrow$ Perturbation($s^*, N_2$)
    $s^{*\prime} \longleftarrow$ LocalSearch($s', \mathcal{F}, c, N_1$)
    $s^* \longleftarrow$ AcceptanceCriterion($s^*, s^{*\prime}$)
until TerminationCondition satisfied
return $s^*$
```

Figure 2. Pseudo-code of an iterated local optimization procedure.

Most local search algorithms rely on a single neighborhood structure. As a way to improve upon local search, there have also been efforts to combine multiple neighborhoods for descent; e.g. Martin et al. [21] proposed a simulated annealing based algorithm for TSP with double-bridge 4-exhange neighborhood combined with standard local search using 3-opt neighborhood. Such a scheme is commonly called under the various names of iterated local search [11], large-step Markov chain [21], chained local search [1], multilevel local search [23], and variable neighborhood search [8]. In this paper, we refer to our approach under the more general name of *iterated local optimization.* For instance, for TSP, the chained LK (CLK) [1], Iterated LK (ILK) [11], Lin–Kernighan–Helsgaun (LKH) [9], and multilevel LKH [23] algorithms are based on the principle of iterated local optimization, and constitute the current state-of-the-art algorithms in terms of efficiency and effectiveness.

A high level description of iterated local optimization is presented in Fig. 2, which utilizes two neighborhood structures N_1 and N_2. First, starting from an initial feasible solution s_0, after applying LocalSearch$(\mathcal{F}, c, N_1)$ using the neighborhood N_1, we arrive at a local optimum s^*. Next we apply perturbation or 'kick' to s^* using Perturbation(s^*, N_2) by the neighborhood N_2 to escape the local optimum (by allowing temporary hill climbing), whose outcome is an intermediate stage s'. LocalSearch$(\mathcal{F}, c, N_1)$ is once again applied but now using s' as an initial feasible solution, and accept or reject the output depending on the AcceptanceCriterion. This step is repeated until a termination condition is satisfied. We will discuss individual component in the context of MINIMUM ENERGY BROADCAST in Section 5.

4. Two Neighborhood Structures for Iterated Local Optimization

The local optimization for MINIMUM ENERGY BROADCAST (MEB) requires an arborescence as an initial feasible solution and the intermediate improved solution at each iteration. To preserve tree structure, the basic premise is that every node should be connected from the source, and there should be no cycle. Given a tree T, we can uniquely and easily identify the corresponding range assignment vector $\overline{r} = (r(1), r(2), \ldots, r(n)) \in \mathcal{R} \subset \mathbb{R}^n$ in $\Theta(n)$ time, where $\mathcal{R}$ is the domain of feasible solutions for MINIMUM RANGE ASSIGNMENT (MRA). But the converse is not true, because a group of trees are mapped into the same range assignment vector $\overline{r}$. See Fig. 3 for example. In effect, the domain of feasible solutions $\mathcal{R}$ for MRA is a partition of the domain of the feasible solutions $\mathcal{T}$ for MEB. Thus if we deal with range assignments $\overline{r}$ instead of trees T, the search space of local search [11,22] for MEB problem may be significantly simplified.

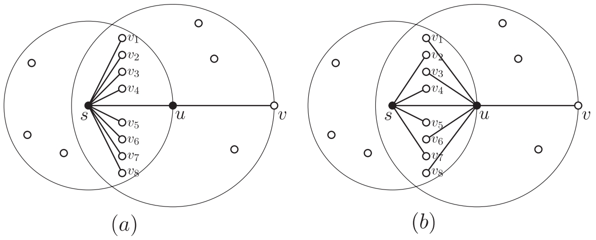


Figure 3. Range assignment vs. tree. (a) A tree and its range assignment. (b) For the same range assignment, multiple trees are allowable. For this specific topology, $2^8 = 256$ trees are mapped to the same range assignment $r(s) = \|su\|$, $r(u) = \|uv\|$. Unaffected edges are not displayed.

In general, it is suggested that neighborhoods be designed in order to induce maximum correlation between objective function values of adjacent points [11]. In order to maximize correlation between objective function and neighborhood function, it is better to directly deal with range assignments instead of trees. As observed by Wieselthier et al. [26], this also matches well with the general philosophy that node-based approach (range) is preferred to link-based approach (tree) in wireless environment. This observation is the basis of our new approach and leads to an algorithm that converges quickly to a high quality local optimum. In the following, if at most k ranges of nodes are allowed to increase from a given range assignment $\overline{r}$, we call it an order-k (range-based) neighborhood and denote by $NH^{(k)}(\overline{r})$, i.e.,

$$\begin{aligned} & NH^{(k)}(\overline{r}) \\ & = \{\overline{r}' \mid r'(v) \geq r(v), \forall v \in S \subset V \wedge |S| \leq k\}. \end{aligned}$$

4.1. Expanding Sweep Search Neighborhood

First we present an order-1 range-based neighborhood structure called the *Expanding Sweep Search* (ESS) neighborhood which also preserves tree connectivity from source s. The idea of ESS is simple: in exchange for an increase in a node's power (or equivalently range), how much reduction in other nodes' power can be achieved is tested. Before we proceed further, we note that all operations in our algorithm are based on the following trivial inequality: for any finite sets of real numbers S_1 and S_2 such that $S_1 \subseteq S_2$,

$$\max\{S_1\} \leq \max\{S_2\}. \tag{2}$$

Thus it works on general graphs with arbitrary edge costs, even in the case of asymmetric edge costs. However, if a geometric graph model is used, we can leverage its geometric properties to find more efficient algorithms.

The ESS transformation may be best explained with the illustrations presented in Fig. 4. Fig. 4(a) shows the original tree T, where paths from source s to nodes c, e, f are omitted. The direction of each edge is always assumed away from s. The transmission range of each node is also drawn. Let us keep focus on node u, which originally has the range $r(u) = \|ud\|$. Suppose now node u decides to expand its range to reach a destination node v such that $\|uv\| > \|ud\|$, at the increase in total cost by

$$\Delta^+(u) = P_{uv} - P_{ud} \geq 0 \tag{3}$$

which is called the *incremental power*. The expanded range is illustrated as a solid line circle in Fig. 4(b). Then every node within the range, i.e., the physical neighbor $\mathcal{N}_u(v) = \{a, b, d, g, h, i, j, u, v\}$, can hear the broadcast message from s through node u. Thus, regardless of previous parent node, each node in $\mathcal{N}_u(v)$ may update its parent node to be node u, since they can get the message from node u. However, cautions should be given since if any of the path nodes from the source to node u, denoted by $\Pi_{s\to u} = \{s, \ldots, c, b, a, u\}$, would update its parent to node u, the path from s to u would be broken and the tree might become disconnected. Therefore, only the nodes $\{d, g, h, i, j, v\}$, the physical neighbors other than the nodes in $\Pi_{s\to u}$, are allowed to become the children of node u. That is, every node in $\mathcal{M}_u(v) \triangleq \mathcal{N}_u(v) \setminus \Pi_{s\to u}$ (marked with an

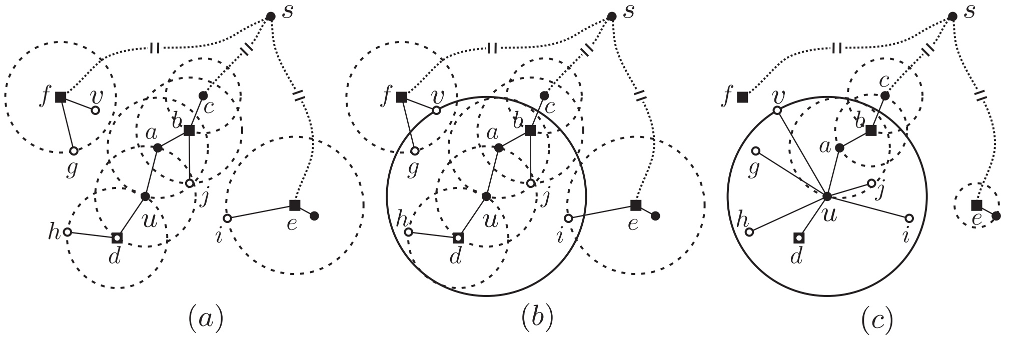


Figure 4. Illustration for neighborhood structures (a) Original tree T. (b) Node u expands its range to reach node v. (c) A neighbor tree T' of T in Expanding Sweep Search (ESS) neighborhood.

empty circle in Fig. 4) updates its parent to node u. The resulting transformed tree T' by this parent updates is shown in Fig. 4(c).

Now let us analyze how much reduction in total tree cost we can achieve by this transformation. Note that updating parents of node in $\mathcal{M}_u(v)$ in turn gives a chance for the original parents $w \in \mathcal{Q}_u(v) \triangleq \pi(\mathcal{M}_u(v))\backslash\{u\} = \{b, d, e, f\}$ marked with square in Fig. 4 to reduce their ranges resulting in power reduction by

$$\begin{aligned}\Delta^{-}(w) &= p(w) - \max_{k\in\delta_T(w)\backslash\mathcal{M}_u(v)}\{P_{wk}\} \qquad (4)\\ &= \max_{k\in\delta_T(w)}\{P_{wk}\} - \max_{k\in\delta_T(w)\backslash\mathcal{M}_u(v)}\{P_{wk}\} \geq 0,\end{aligned}$$

since $\delta_T(w)\backslash\mathcal{M}_u(v) \subseteq \delta_T(w)$ and using (2). Accounting for every affected node $w \in \mathcal{Q}_u(v)$, the pure *gain* $\mathcal{G}(u,v)$ in total cost is

$$\mathcal{G}(u,v) = \sum_{w\in\mathcal{Q}_u(v)} \Delta^{-}(w) - \Delta^{+}(u), \qquad (5)$$

where the first term on the right hand side is called *sweeping gain*. We denote the ESS transformation by the postfix form $T' = T \,||\, \mathcal{ESS}(u,v)$. Then the expanding sweep search neighborhood denoted by NH_{ess} is defined as

$$NH_{ess}(T) = \{T' \mid T' = T \,||\, \mathcal{ESS}(u,v),\ \forall u,v \in V\}.$$

Since $|NH_{ess}(T)| \leq \binom{n}{2}$, the size of neighborhood is $O(n^2)$. After the expanding sweep search neighborhood NH_{ess} is clearly defined, the LESS algorithm [15] is nothing more than LocalSearch$(\mathcal{T}, c, NH_{ess})$ with the steepest descent search strategy. That is, given an initial feasible solution, the best neighbor among NH_{ess} is chosen at each iteration, until there is no more gain in cost.

```
procedure Expanding Sweep Search ($ESS$) $(u,v)$
Input: a $s$-arborescence $T \in \mathcal{T}$
$T' \leftarrow T$
expand range of node $u$ to $v$, i.e., $r(u) \leftarrow \|uv\|$,
    $\forall v \in V$ s.t. $\|uv\| \geq r(u), v \notin \Pi_{s\to u}$
$\mathcal{M}_u(v) \leftarrow \mathcal{N}_u(v)\backslash\Pi_{s\to u}$
$\pi_{T'}(\mathcal{M}_u(v)) \leftarrow u$
return $T'$
```

Figure 5. Pseudo-code for Expanding Sweep Search (ESS) transformation.

4.2. Edge Exchange Neighborhood

An alternative (and traditional) approach is to consider the tree structure itself instead of range assignments. The most natural neighborhood structure conceivable when the tree structure is a feasible solution is the *elementary tree transformation* for undirected graph as described in Papadimitriou and Steiglitz (see Chapter 19 in [22]); that is, add a non-tree edge to a tree and remove an edge from the unique cycle formed by previous edge addition. Because we are dealing with a rooted directed tree (arborescence) for MINIMUM ENERGY BROADCAST, we can only apply elementary tree transformation on the underlying graph of the directed graph. Therefore we call it an *Edge Exchange Neighborhood* (NH_{ee}) that is an adaptation of the elementary tree transformation to directed graphs.

An equivalent interpretation of edge exchange neighborhood is as follows: Remove an edge (u,v) from T which partitions the tree into two subtrees $T_1 = (S_{uv}, A_{T_1})$ and $T_2 = (V\backslash S_{uv}, A_{T_2})$ where S_{uv} denotes the connected component containing s and u. Adding an edge $(u',v') \in S_{uv} \times V\backslash S_{uv}$ to $T_1 \cup T_2$ and updating the direction of edges in T_2 forms a neighbor of T:

$$NH_{ee}(T) = \{T' \mid T' = (V, A_T \backslash (u,v) \cup (u',v')), \\ \forall (u,v) \in A_T,\ \forall (u',v') \in S_{uv} \times V \backslash S_{uv}, \\ \text{reverse edge direction } \Pi_{v \to v'}\}.$$

The size of neighborhood is $n|S_{uv}|(|V| - |S_{uv}|) = O(n^3)$. Checking the cost of a tree $c(T)$, using a simple array as the data structure for a tree, requires $O(n)$ running time. Hence if we use the algorithm LocalSearch$(\mathcal{T}, c, NH_{ee})$ with steepest descent strategy, each iteration may take $O(n^4)$, which is quite expensive. In the course of updating the direction of edges in T_2, it is possible that $O(n)$ nodes may be required to increase their ranges. Therefore, this neighborhood is complementary to NH_{ess}. Other possible extension to NH_{ee} neighborhood is to combine NH_{ee} and NH_{ess}: that is, for every expanded range in NH_{ee}, we can apply the ESS transformation. Although the combined neighborhood may give better performance, running time per iteration is costly. Trade-offs in neighborhood size, time complexity and the quality of local optima is required. Therefore, we restrict the usage of NH_{ee} as a perturbation method for iterated local optimization. Since LocalSearch$(\mathcal{T}, c, NH_{ess}) \neq$LocalSearch$(\mathcal{T}, c, NH_{ee})$ in general, they are hence mutually beneficial in enhancing the quality of local optimum obtained by the other.

Theorem 1 *The following properties of these two neighborhoods satisfy:*
(a) Expanding Sweep Search neighborhood encompasses neighborhood for Post Sweep algorithm.
(b) Edge Exchange neighborhood encompasses neighborhoods for IMBM and 1-Shrink.

Proof. (a) Without range expansion in NH_{ess}, inspecting gain for current range assignment is no different from the post sweep algorithm [26]. (b) The neighborhoods used in Iterative Maximum Branch Minimization (IMBM) [17] and r-shrink algorithms [6] are obtained by updating a node's parent to a node which is not one of its descendants. Clearly, this is a special case of NH_{ee} when $v = v'$. ■

Therefore, every local optimum by NH_{ess} (or NH_{ee}) is also a local optimum by Post Sweep (or IMBM and 1-Shrink), respectively. Furthermore, note that although T' in Fig. 4(c) is a one-hop neighbor of T in NH_{ess}, the corresponding tree structure is large number of hops away in terms of edge exchange neighborhood NH_{ee}. This is the true advantage of using range assignments in our algorithms, because the number of required iteration can be significantly reduced, compared to using tree structure, without much added complexity.

5. Search Strategies and Algorithm Details

Our current implementation of iterated local optimization in this work strictly follows the reference high level description in Fig. 2. It is based on deterministic local search LocalSearch$(\mathcal{T}, c, NH_{ess})$, and a random perturbation Perturbation(s^*, NH_{ee}). We will discuss each component in Fig. 2 in the order of appearance. Before we further proceed, let us discuss some of the underlying philosophies of our algorithm.

The underlying philosophy of our current work is exactly the opposite of the Iterative Maximum Branch Minimization (IMBM) [17] and r-shrink [6] algorithms. As suggested by the names of these algorithms, "range reduction" is emphasized, meaning that the maximum branch (edge) and the corresponding range of each node is progressively broken into smaller pieces (i.e., into multihop edges or ranges). For instance in IMBM, as an initial feasible solution, a single maximum range from the source s covering the rest of the nodes, i.e., $r(s) = \max_{v \in V}\{\|sv\|\}$ and the corresponding star topology rooted at s, is used. It is iteratively reduced and transformed to different multihop tree structure [17]. The r-shrink algorithm resembles the IMBM, but relies on a deterministic maximum branch reduction method.

On the other hand, our LESS algorithm [15] progressively expands the ranges of nodes. While range expansion and range reduction are like the flip side of the coin (one can be achieved at the cost of the other), the implication in local search is radically different. That is, to reduce the range of a node, we need to specifically search for the nodes whose increase in range can reduce the total cost. However, as observed in expanding sweep search, expansion in a node's range naturally determines which nodes can reduce their ranges and do not require further computations, which is one advantage of our approach. Furthermore, there are other reasons range expansion may be generally preferable to range reduction for MEB.

Let us consider the topology illustrated in Fig. 6, where two groups of m nodes lie at the inner and outer circle of radius r_1 and r_2, respectively, where $|r_2 - r_1|$ is assumed very small. Suppose the configuration in Fig. 6(a) is given as the initial feasible solution, as is the case for IMBM. While the net difference between two tree cost may be small (r_2^α vs. $r_1^\alpha + m(r_2 - r_1)^\alpha$), either deterministically or probabilistically searching for the occasions which enable reduction in range $r(s)$ is

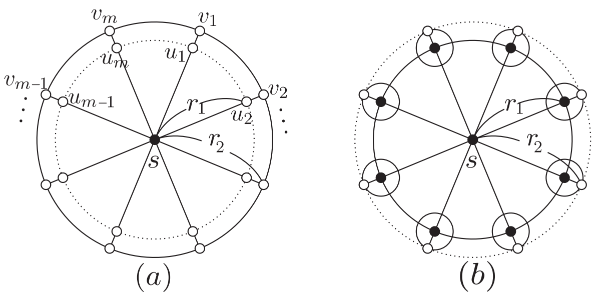


Figure 6. Illustration of a stable constellation demonstrating that reduction in range is difficult in general.

extremely difficult. That is, only if all m interior nodes simultaneously increase their ranges, we can jump out of the local optimum to convert to the better configuration in Fig. 6(b). In terms of the number of hops in neighborhood, these two solutions are far apart and hence we may assume there exists a high potential barrier in between these configurations. In the course of range reduction, similar situation occurs very frequently and hence it is easy to fall into a poor local optimum.

GenerateInitialFeasibleSolution: Now it becomes quite clear which initial feasible solution is a suitable choice. Note that the stable configuration discussed above may have been avoided from the outset (although not completely), if we have chosen MST as the initial solution. In addition, since MST consists of short length edges [12], it provides richer search space for range expansion algorithms as ours. Also approximation ratio of MST is known to be 12 [24]. By improving upon MST, the same worst case bound is guaranteed. For the same reason, both BIP and SPT are equally viable choices but not considered in this paper. Certainly, repeatedly restarting from a randomly generated tree as a starting point can only make our results stronger.

LocalSearch: As the LocalSearch algorithm in Fig. 2, we could use our LESS algorithm [15]. However, we can speed up convergence significantly by selecting the first k largest gain at each iteration as long as the ranges are disjoint, for which we call the variable-order disjoint (VOD) steepest descent search strategy. This strategy in some sense resembles the variable-depth search by Lin and Kernighan [20] in that indefinite orders are considered as long as the gain is kept positive. We will call this algorithm as LESS$_{vod}$. The details are listed in Fig. 7.

Perturbation: As discussed in Section 4.2, we adopt edge exchange neighborhood NH_{ee} as our perturbation

```
procedure Variable-Order Disjoint LESS (LESS$_{\text{vod}}$)
Input: an $s$-arborescence $t$
/* requires searching whole neighborhood */
1. $\mu(u) \longleftarrow \underset{\{v:\|uv\|\geq r(u), v\notin\Pi_{s\rightarrow u}\}}{\arg\max} \{\mathcal{G}(u,v)\}, \forall u \in V$
2.  sort $u_1, u_2, \ldots, u_k$, $k \leq n$ so that
        $\mathcal{G}(u_i, \mu(u_i)) \geq \mathcal{G}(u_j, \mu(u_j)) > 0,$
        $\forall 1 \leq i < j \leq k$
5.  $R \longleftarrow \{u_1\}$
6.  for $i = 2, ..., k$
/* if ranges are disjoint */
7.        if $\|u_i v\| \geq \|u_i \mu(u_i)\| + \|v \mu(v)\|, \forall v \in R$
8.              $R \longleftarrow R \cup \{u_i\}$
9.  end
/* iteratively apply $\mathcal{ESS}$ $|R|$ times */
10.  $t' \longleftarrow t \parallel \mathcal{ESS}(v, \mu(v)), \forall v \in R$
11. if $c(t') < c(t)$
/* if there is gain, repeat iteration */
12.        $t \longleftarrow t'$; goto step 1
13. return $t'$
```

Figure 7. Pseudo-code for LESS$_{\text{vod}}$ algorithm.

method. Due to its $O(n^3)$ size neighborhood, steepest descent type search strategy can be very expensive. Therefore, random perturbation in NH_{ee} is used in this paper, which requires only $O(1)$ running time. This allows most of the time in iterated local optimization will be spent by LocalSearch algorithms.

AcceptanceCriterion: As in simulated annealing or large-step Markov chain [21], the following acceptance criterion is popularly used:

$$\text{AcceptanceCriterion}(s^*, s^{*\prime}) = \begin{cases} s^{*\prime} & \text{if } c(s^{*\prime}) < c(s^*) \\ s^{*\prime} & \text{if } c(s^{*\prime}) > c(s^*), \text{ w.p. } e^{\frac{c(s^*)-c(s^{*\prime})}{T}} \\ s^* & \text{otherwise} \end{cases}$$

where deteriorating perturbed intermediate solution $s^{*\prime}$ is accepted with probability (w. p.) $\exp\{\frac{c(s^*)-c(s^{*\prime})}{T}\}$, and T is a tunable parameter called temperature. If $T = 0$, only the better solution is accepted. If T is very high, random walk type behavior can be accomplished. In this paper, we adopt this acceptance criterion with $T = 0$. Simulations using $T > 0$ and changing temperature over time (cooling schedule) will be a direct but interesting extension to current work.

TerminationCondition: We can terminate the iteration: (i) if s^* has reached the global optimum, (ii) if maximum pre-specified number of iteration has reached, or (iii) if pre-specified time has elapsed. In case the globally optimal solution is known (say, using an IP solver), we use (i); otherwise, we use (iii) with maximum time limit set to 60 seconds using a Pentium 1.6GHz equipped machine. Now that all components in iterated local optimization is established, we proceed to simulations results.

6. Simulation Results

Over 1000×1000 m^2 deploy region, network configurations are generated by uniformly distributing n nodes, one of the node is selected as the source s, and broadcast routing trees rooted at s are constructed. By varying the number of nodes within the deploy region, we have effectively changed the node density per unit area. The path loss factors $\alpha = 2$ and 4 are used in our simulations. For each network instance, globally optimal solutions are obtained following the IP formulation in [7]. As an IP solver, CPLEX [10], one of the most sophisticated commercial IP solver, was used. From our experience, a network size of up to 20 nodes is the practical maximum limit solvable in a reasonable time (<3 hours per instance). However, CPLEX being a general purpose IP solver, it does not rule out the possibility that one may develop customized solver by generating problem-specific cuts for branch-and-cut methods [22] either to speed up the calculation or to allow for larger network sizes.

For small network sizes $4 \leq n \leq 20$, we are able to compare our results with optimal solutions by CPLEX, and they are summarized in Table 1 and 2. The expressions to obtain the values in each column is summarized underneath the tables. Each entry in the tables is an average of 1000 randomly generated network instances. We note that obtaining the whole globally optimal solutions took several weeks using a Pentium 2.8GHz processor equipped machine, which demonstrates impracticableness of the IP approach in real situations. All other algorithms are implemented in MATLAB environment. We allowed up to 60 seconds for the iterated local optimization (ILO) using $NH_{ess} + NH_{ee}$ to run before interrupted. As the performance comparison measure, we presented both the average total transmit power and the average percentage excess ($\overline{P.E.}$) over global optima defined as

$$\overline{P.E.} = \frac{1}{k}\sum_{j=1}^{k} \left[c_j(T_{\text{algo}})/c_j(T_{\text{OPT}}) - 1\right] \times 100\%,$$

for the algorithm *algo*. Let p_{opt} denote the probability (frequency) of local optima by ILO being actually global optima (OPT). The values in second parenthesis in column (c) roughly represents $(1 - p_{opt}) = \Pr\left[c(T_{\text{ILO}}) > c(T_{\text{OPT}})\right]$ in percentage.

We can observe that ILO consistently produces very high quality local optima, among which globally optimal solutions are obtained with high frequency. As noted earlier, NH_{ee} is complementary to NH_{ess}, and hence using NH_{ee} converts a large fraction of suboptimal cases into global optima. For instance, for $\alpha = 2$, $n \leq 20$, we get $\overline{P.E.} \leq 1.117\%$ and $(1 - p_{opt}) \leq 30.8\%$. In case of $\alpha = 4$, we can get global optima for almost all instances, with $p_{opt} \geq 96.5\%$ and $\overline{P.E.} \leq 0.039\%$ for $n \leq 20$. Out of interest we also tested local search using NH_{ee} alone, and its results are reported in column (d). We can see that the result is not impressive enough, although still better than IMBM or 1-Shrink. However, we should note that this level of performance is achieved without random restarts as in IMBM. The reason we get better performance for $\alpha = 4$ is quite clear. As observed in [3, 14, 26], the larger the path loss factor α, the larger the penalty for increasing the transmit power. As the wireless broadcast advantage becomes negligible, more link-based characteristics is emphasized. In such case even local search using NH_{ee} alone performs reasonably well with $\overline{P.E.} \leq 1.887\%$, since NH_{ee} is a link-based neighborhood. Readers may also find interest in the performance of BIP or MST relative to the global optimum.

For a larger network size $n > 20$, comparison with optimal solutions by the IP solver was not possible. Hence as a reference, performance comparison of LESS, MST, BIP [26], and EWMA [3] are presented in Fig. 8. Extrapolating the performance level of LESS demonstrated in Table 1 and 2 to larger network sizes, Fig. 8 provides a valuable measure how other algorithms perform relative to the optimal. Average total transmit power is compared in Fig. 8(a) and (c), and the ratio of them is compared in Fig. 8(b) and (d) for $\alpha = 2$ and 4. In all figures, it is easy to notice that LESS consistently outperforms others by a wide margin, even EWMA, which used to be state-of-the-art. For instance, LESS performs approximately 25∼31% (7.5∼8.5%) better than MST, over the entire range from $n = 20$ to 1000 for $\alpha = 2$ (resp. $\alpha = 4$). For $\alpha = 4$, due to high dynamic range, it is hard to appreciate the performance gain in Fig. 8(c) but it is clearly visible in Fig. 8(d). Clearly, this suggest we should densely deploy network nodes, which has more favorable effect than efficient algorithms for large α. For many practical situations when a high quality local optimum usually suffices, our work can be used for benchmarking other algorithms for comparison.

Table 1. Comparison of local search using NH_{ess} and NH_{ee} with global optima ($\alpha = 2$)

(a) n	(b) OPT (P.E.)	(c) ILO (P.E.) $(1 - p_{opt})$	(d) NH_{ee} (P.E.)	(e) BIP (P.E.)	(f) MST (P.E.)
4	4.3143e+5	4.3143e+5 (0.0000%) (0.0%)	4.3455e+5 (0.7232%)	4.4192e+5 (2.5994%)	4.7631e+5 (10.4685%)
5	4.3102e+5	4.3102e+5 (0.0000%) (0.0%)	4.3787e+5 (1.5893%)	4.5459e+5 (5.9326%)	4.9352e+5 (15.5838%)
6	4.2652e+5	4.2655e+5 (0.0059%) (0.2%)	4.3691e+5 (2.4360%)	4.5998e+5 (8.1004%)	5.0399e+5 (18.8144%)
7	4.2720e+5	4.2734e+5 (0.0297%) (0.7%)	4.4146e+5 (3.3380%)	4.7062e+5 (10.5279%)	5.1693e+5 (22.3715%)
8	4.2408e+5	4.2444e+5 (0.0854%) (2.0%)	4.4421e+5 (4.7467%)	4.7761e+5 (13.3438%)	5.2574e+5 (25.2631%)
9	4.1824e+5	4.1850e+5 (0.0562%) (2.0%)	4.4001e+5 (5.2051%)	4.7966e+5 (15.8574%)	5.2605e+5 (27.6818%)
10	4.1977e+5	4.2045e+5 (0.1762%) (3.9%)	4.4510e+5 (6.0343%)	4.8863e+5 (17.2536%)	5.3655e+5 (29.3122%)
11	4.1404e+5	4.1513e+5 (0.2542%) (6.1%)	4.4061e+5 (6.4173%)	4.8709e+5 (18.2900%)	5.3244e+5 (29.6363%)
12	4.1079e+5	4.1195e+5 (0.2723%) (6.8%)	4.4078e+5 (7.3006%)	4.8602e+5 (19.1320%)	5.3351e+5 (31.2965%)
13	4.0735e+5	4.0853e+5 (0.2748%) (7.3%)	4.4168e+5 (8.4276%)	4.8940e+5 (20.8748%)	5.3479e+5 (32.5564%)
14	4.0776e+5	4.1012e+5 (0.5679%) (13.8%)	4.4337e+5 (8.7331%)	4.9478e+5 (22.1620%)	5.3809e+5 (33.2930%)
15	4.0603e+5	4.0853e+5 (0.6181%) (15.5%)	4.4256e+5 (8.9969%)	4.8921e+5 (21.1684%)	5.3572e+5 (33.1597%)
20	3.9952e+5	4.0390e+5 (1.1170%) (30.8%)	4.4340e+5 (10.9832%)	4.9831e+5 (25.4930%)	5.3865e+5 (36.0863%)

Table 2. Comparison of local search using NH_{ess} and NH_{ee} with global optima ($\alpha = 4$)

(a) n	(b) OPT (P.E.)	(c) ILO (P.E.) $(1 - p_{opt})$	(d) NH_{ee} (P.E.)	(e) BIP (P.E.)	(f) MST (P.E.)
5	1.3383e+11	1.3383e+11 (0.0000%) (0.0%)	1.3448e+11 (0.4205%)	1.3686e+11 (2.8603%)	1.4025e+11 (5.3330%)
6	1.1127e+11	1.1127e+11 (0.0000%) (0.0%)	1.1187e+11 (0.5850%)	1.1504e+11 (3.5228%)	1.1854e+11 (6.4568%)
7	1.0075e+11	1.0074e+11 (0.0009%) (0.1%)	1.0176e+11 (0.8120%)	1.0476e+11 (4.2778%)	1.0713e+11 (7.0146%)
8	0.8822e+11	0.8825e+11 (0.0397%) (0.2%)	0.8932e+11 (1.1653%)	0.9249e+11 (4.9752%)	0.9527e+11 (8.3369%)
9	0.7811e+11	0.7813e+11 (0.0207%) (0.4%)	0.7895e+11 (0.9693%)	0.8223e+11 (5.8491%)	0.8418e+11 (8.7395%)
10	0.7295e+11	0.7297e+11 (0.0412%) (0.8%)	0.7367e+11 (1.0714%)	0.7713e+11 (6.0221%)	0.7916e+11 (9.1413%)
11	0.6441e+11	0.6444e+11 (0.0438%) (0.9%)	0.6522e+11 (1.2194%)	0.6837e+11 (6.2592%)	0.7015e+11 (9.0619%)
12	0.5806e+11	0.5808e+11 (0.0384%) (0.6%)	0.5891e+11 (1.3915%)	0.6174e+11 (6.3946%)	0.6336e+11 (9.3804%)
13	0.5431e+11	0.5432e+11 (0.0418%) (1.5%)	0.5534e+11 (1.7298%)	0.5796e+11 (6.8126%)	0.5912e+11 (9.1565%)
14	0.4971e+11	0.4972e+11 (0.0468%) (1.4%)	0.5045e+11 (1.4731%)	0.5318e+11 (7.2157%)	0.5434e+11 (9.9587%)
15	0.4571e+11	0.4573e+11 (0.0645%) (2.1%)	0.4647e+11 (1.7533%)	0.4875e+11 (6.8394%)	0.4987e+11 (9.5634%)
20	0.3303e+11	0.3305e+11 (0.0391%) (3.5%)	0.3365e+11 (1.8868%)	0.3529e+11 (7.2012%)	0.3663e+11 (11.2691%)

(a) $n =$ network size (b) $\frac{1}{k}\sum_{j=1}^{k} c_j(T_{\mathrm{OPT}})$ (c) $\frac{1}{k}\sum_{j=1}^{k} c_j(T_{NH_{ess}+NH_{ee}})$ (d) $\frac{1}{k}\sum_{j=1}^{k} c_j(T_{NH_{ee}})$ (e) $\frac{1}{k}\sum_{j=1}^{k} c_j(T_{BIP})$ (f) $\frac{1}{k}\sum_{j=1}^{k} c_j(T_{MST})$

7. Conclusions

We investigated efficient heuristics for the minimum energy broadcast (MEB) problem. While there have been much efforts in developing efficient tree construction algorithms beginning from the BIP algorithm, we provided different algorithms based on an iterative improvement approach. Specifically, through the use of iterated local optimization techniques, it is demonstrated that the performance of our former algorithm, Largest Expanding Sweep Search (LESS), can be further enhanced. Extensive comparison study with globally optimal solutions obtained by an integer programming solver confirmed that the produced solutions are very close to global optima in most cases, and the average percentage excess over the optimum is limited to merely within 1.12 percents up to the network size of 20 nodes. Moreover, unlike the exact method by integer programming approach, our algorithms can handle very large network sizes and hence can serve as the basis for benchmarking other developed algorithms. We believe the gained insights from this work may spur the development of better performing localized algorithms.

References

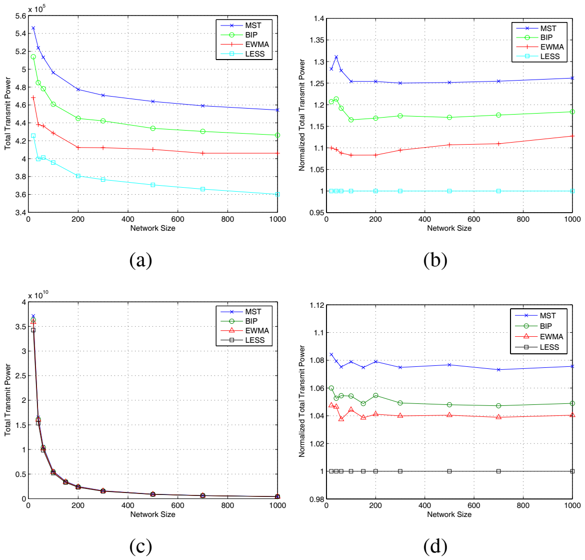
[1] D. Applegate, W. Cook, and A. Rohe. Chained linkernighan for large traveling salesman problems. *INFORMS Journal on Computing*, 15(1):82–92, 2003.

[2] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34, Oct. 2000.

[3] M. Cagalj, J. P. Hubaux, and C. Enz. Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues. In *Proc. ACM/IEEE MOBICOM '02*, Atlanta, Georgia, 2002.

[4] A. E. F. Clementi, P. Crescenzi, P. Penna, G. Rossi, P. Vocca, A. Ferreira, and H. Reichel. On the complexity of computing minimum energy consumption broadcast subgraphs. In *STACS 2001. 18th Annual Symposium on Theoretical Aspects of Computer Science. Proceedings Lecture Notes in Computer Science Vol.2010*, pages 121–31, 2001.

[5] A. E. F. Clementi, P. Penna, R. Silvestri, H. Reichel, and S. Tison. The power range assignment problem



(a) (b)

(c) (d)

Figure 8. Performance comparison of MST, BIP, EWMA, and LESS in terms of (a) mean total transmit power ($\alpha = 2$), (b) mean ratio of total transmit power ($\alpha = 2$), (c) mean total transmit power ($\alpha = 4$), and (d) mean ratio of total transmit power ($\alpha = 4$).

in radio networks on the plane. In *STACS 2000. 17th Annual Symposium on Theoretical Aspects of Computer Science. Proceedings Lecture Notes in Computer Science Vol.1770*, pages 651–60, 2000.

[6] A. Das, R. Marks, M. El-Sharkawi, P. Arabshahi, and A. Gray. r-shrink: a heuristic for improving minimum power broadcast trees in wireless networks. In *IEEE GLOBECOM 2003*, volume 1, San Francisco, CA, USA, 1-5 Dec. 2003.

[7] A. Das, R. Marks, M. El-Sharkawi, P. Arabshahi, and A. Gray. Minimum power broadcast trees for wireless networks: integer programming formulations. In *IEEE INFOCOM 2003*, volume 2, pages 1001–10, San Francisco, CA, USA, 2003.

[8] F. Glover and G. A. Kochenberger. *Handbook of metaheuristics*. Kluwer Academic Publishers, Boston, 2003.

[9] K. Helsgaun. An effective implementation of the linkernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–30, 2000.

[10] ILOG CPLEX. In *http://www.ilog.com/products/cplex/*, 2004.

[11] D. S. Johnson and L. McGeoch. The traveling salesman problem: a case study. In *Local search in combinatorial optimization*, edited by Emile Aarts and Jan Karel Lenstra, Chichester [England], New York, 1997. Wiley.

[12] I. Kang and R. Poovendran. A comparison of power-efficient broadcast routing algorithms. In *IEEE GLOBECOM 2003*, San Francisco, CA, 2003.

[13] I. Kang and R. Poovendran. A novel power-efficient broadcast routing algorithm exploiting broadcast efficiency. In *IEEE Vehicular Technology Conference (VTC)*, Orlando, FL, 2003.

[14] I. Kang and R. Poovendran. Power-efficient broadcast routing in adhoc networks using directional antennas: technology dependence and convergence issues. Technical Report UWEETR-2003-0015, University of Washington, July 2003.

[15] I. Kang and R. Poovendran. Broadcast with heterogeneous node capability. In *IEEE GLOBECOM 2004*, Dallas, TX, USA, 2004.

[16] I. Kang and R. Poovendran. COBRA: Center-oriented broadcast routing algorithms for wireless adhoc networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, Atlanta, GA, 2004.

[17] F. Li and I. Nikolaidis. On minimum-energy broadcasting in all-wireless networks. In *IEEE Conference on Local Computer Networks (LCN 2001)*, pages 193–202, Tampa, FL, Nov. 2001.

[18] W. Liang. Constructing minimum-energy broadcast trees in wireless ad hoc networks. In *ACM MobiHoc 2002*, Lausanne, Switzerland, 2002.

[19] S. Lin and B. Kernighan. An effective heuristic procedure for solving large travelling salesman problems. *Bulletin of the Operations Research Society of America Operations Research Society of America 41st annual meeting. Abstracts only, 26-28 April 1972*, 20(suppl.1):B/89, 1972.

[20] S. Lin and B. Kernighan. An effective heuristic algorithm for the travelling-salesman problem. *Operations Research*, 21(2):498–516, 1973.

[21] O. Martin and S. Otto. Combining simulated annealing with local search heuristics. *Annals of Operations Research*, 63:57–75, 1996.

[22] C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Prentice Hall, Englewood Cliffs, N.J., 1982.

[23] C. Walshaw. A multilevel approach to the travelling salesman problem. *Operations Research*, 50(5):862–77, 2002.

[24] P. J. Wan, G. Calinescu, X. Y. Li, and O. Frieder. Minimum-energy broadcast routing in static ad hoc wireless networks. In *Proc. IEEE INFOCOM '01*, pages 1162–71 vol.2, 2001.

[25] B. Wang and S. K. Gupta. G-remit: An algorithm for building energy efficient multicast trees in wireless ad hoc networks. In *IEEE International Symposium on Network Computing and Applications*, pages 265–272, 16-18 April 2003.

[26] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *Proc. IEEE INFOCOM 2000*, volume 2, pages 585–94, 2000.

[27] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. Energy-efficient broadcast and multicast trees in wireless networks. *Mobile Networks and Applications*, 7(6):481–92, 2002.