Prolonging Sensor Network Lifetime with Energy Provisioning and Relay Node Placement

Y. Thomas Hou[†] Yi Shi[†] Hanif D. Sherali[‡] Scott F. Midkiff[†]

[†] The Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061, USA

[‡] The Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA 24061, USA

{thou, yshi, hanfis, midkiff}@vt.edu

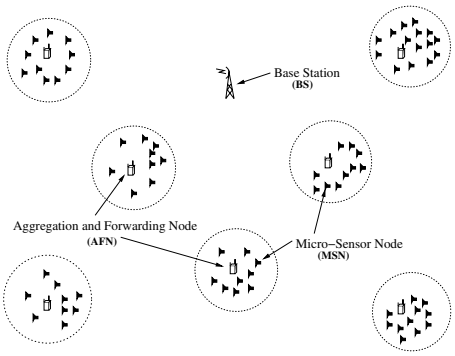
***Abstract*— Wireless sensor networks that operate on batteries have limited network lifetime. There have been extensive recent research efforts on how to design protocols and algorithms to prolong network lifetime. However, due to energy constraint, even under the most efficient protocols and algorithms, the network lifetime may still be unable to meet the mission's requirements. In this paper, we consider the energy provisioning problem for a two-tier wireless sensor network. In addition to provisioning additional energy on the existing nodes, we also consider deploying *relay nodes* (RNs) into the network to mitigate network geometric deficiency and prolong network lifetime. We formulate the joint problem of energy provisioning and relay node placement (EP-RNP) into a mixed-integer nonlinear programming (MINLP) problem. Since an MINLP problem is NP-hard in general, and even the state-of-the-art software and techniques are unable of offer satisfactory solutions, we develop a heuristic algorithm, called *SPINDS*, to address this problem. We show a number of novel algorithmic design techniques in the design of *SPINDS* that effectively transforms a complex MINLP problem into linear programming (LP) problems without losing critical points in its search space. Through numerical results, we show that *SPINDS* offers very attractive solution and some important insights to the EP-RNP problem.**

***Index Terms*— Energy provisioning, relay node placement, power control, network lifetime, flow routing, wireless sensor networks.**

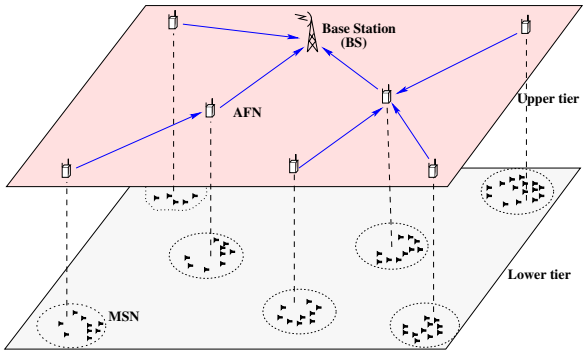
I. Introduction

Wireless sensor networks have attracted unprecedented attention in recent years. In this paper, we consider a two-tier wireless sensor network that can be used for a wide range of applications. Under the two-tier architecture, a wireless sensor network consists of a number of sensor clusters and a base-station. Each cluster is deployed around a strategic location and consists of a number of *Micro Sensor Nodes* (MSNs) and one *Aggregation and Forwarding Node* (AFN). Each MSN sends its sensing data to its local AFN, and the AFN performs in-network processing by aggregating all sensing data. The AFN then relays the composite information to the base-station via a single or multi-hop transmission (see Fig. 1).

An important performance measure for wireless sensor networks is the *network lifetime.* Recent research on maximizing network lifetime focus on devising optimal flow routing algorithms (see, *e.g.,* [6]) based on power control of each node's transmitter. However, the network lifetime, even under optimal flow routing, may still not be able to meet the mission's requirements, and consequently, other strategies are needed.



(a) Physical topology.



(b) A hierarchical view.

Fig. 1. Reference architecture for a two-tier wireless sensor network.

For the two-tier wireless sensor network, although the MSNs are not expected to be re-provisioned with additional energy due to their small size, low cost, and large number, it is plausible to consider provisioning energy to the upper-tier AFNs if we wish to extend the network's lifetime (see Section II for more details). Further, limiting energy provisioning only to the existing AFNs may not yield the most efficient solution. This is because node energy consumption behavior and network lifetime performance in a wireless sensor network are highly dependent on *network geometry.* As we shall see later in this paper, it is more efficient to deploy additional relay nodes (RNs) in the network to mitigate such network

0-7803-9012-1/05/$20.00 (C) 2005 IEEE

geometric deficiency.

In this paper, instead of studying energy provisioning (EP) (on existing AFNs only) and relay node placement (RNP) problems separately, we investigate the *joint* problem of EP and RNP for sensor networks. We also generalize the notion of EP in the sense that energy can be provisioned on either AFNs or RNs. As a result, our work can be applied to address a wide range of problems associated with EP or RNP.

Specifically, we investigate the following problem for EP-RNP: for a given network and some initial energy at each AFN, how should we allocate a total amount of additional energy E at M locations (which can be either at an existing AFN or at a new position for RN) such that the network lifetime can be maximized? We show that this EP-RNP problem can be cast into a *mixed-integer non-linear programming (MINLP)* problem. Since an MINLP problem is known to be NP-hard and even state-of-the-art techniques (*e.g.,* branch-and-bound [11]) and their software implementations (*e.g.,* BARON [2]) cannot provide a good solution, we resort to develop an efficient heuristic algorithm.

Our heuristic algorithm is called *SPINDS*, which stands for *Smart Pairing and INtelligent Disc Search.* SPINDS is an iterative algorithm that attempts to increase the network lifetime by iteratively moving an RN to a better location. Our main idea in achieving this objective is to transform the original MINLP problem into a linear programming (LP) approximation. This is achieved by two ingenious steps. In the first step, we use the so-called *smart pairing* (SP) and *intelligent disc search* (INDS) techniques to determine possible RN placements during each iteration so that network lifetime can be increased. This step transforms the original MINLP problem into a *mixed-integer linear programming* (MILP) problem. Although MILP problem appears simpler than MINLP problem, it is still NP-complete in general. In the second step, we introduce an *equivalence lemma*, which shows that if the RNs are placed wisely, then the MILP problem could be substituted by a much simpler LP problem without any compromise in network lifetime performance. Consequently, it is possible to transform the original MINLP problem into an iterative LP problem, which is polynomial.

In the numerical results, we show that the proposed SPINDS can indeed place the RNs wisely and the LP substitution indeed matches the MILP formulation. We also show that SPINDS offers highly competitive performance in solving EP-RNP problem when compared to some other approaches. Furthermore, we offer some important insights on network geometric properties, RN placement, and energy provisioning. We show that deficiencies due to network geometry (or topology) have a significant impact on network lifetime. When such deficiencies exist, RN placements can be a much more efficient technique than merely provisioning additional energy on existing AFNs.

The remainder of this paper is organized as follows. In Section II, we describe the two-tier wireless sensor network architecture and give models for power consumption. We also formulate EP-RNP as an MINLP problem. Section III presents SPINDS, a polynomial-time algorithm to solve the joint EP-RNP problem. Section IV uses numerical results to demonstrate efficacy of SPINDS and offer insights on network geometry, energy provisioning, and relay node placement. Section V discusses related work and Section VI concludes this paper.

II. System Modeling and Problem Formulation

A. Reference Network Architecture

We focus on a two-tier architecture for wireless sensor networks. Figures 1 (a) and (b) respectively show the *physical* and *hierarchical* network topology for such a network. As shown in the figures, we have three types of nodes in the network: *micro-sensor nodes* (MSNs), *aggregation and forwarding nodes* (AFNs), and a *base-station* (BS). The MSNs can be application-specific sensor nodes and they constitute the lower tier of the network. They are deployed in groups (or clusters) at a strategic location for surveillance or monitoring applications. Each MSN is small and low-cost; they are densely deployed within a small geographic area. The objective of an MSN is very simple: once triggered by an event, it starts to capture live information, which it sends directly to the local AFN in one hop. It is worth pointing out that multi-hop routing among the MSNs is not necessary due to the small distance between an MSN and its AFN. Moreover, an MSN will cease to function once its battery runs out of energy. By deploying these inexpensive MSNs in clusters, and within proximity of a strategic location, it is possible to obtain a comprehensive view of the area situation by exploring the *correlation* among the scenes collected at each MSN [7]. Furthermore, the reliability of area surveillance capability can also be improved through redundancy among the MSNs in the same cluster.

For each cluster of MSNs, there is one AFN, which is different from an MSN in terms of physical properties and functions. The primary functions of an AFN are: (1) *data aggregation* (or "fusion") for data flows coming from the local cluster of MSNs, and (2) *forwarding* (or relaying) the aggregated information to the next hop AFN toward the base-station. For data fusion, an AFN analyzes the content of each data stream it receives, from which it composes a complete scene by exploiting the correlation among each individual data stream from the MSNs. An AFN can also serve as a relay node for other AFNs to carry traffic toward the base-station. Although an AFN is expected to be provisioned with much more energy than an MSN, it also consumes energy at a substantially higher rate (due to wireless communication over large distances). Consequently, an AFN has limited lifetime. Upon the depletion of energy at an AFN, we expect that the *coverage* for the particular area under surveillance will be lost, despite the fact that some of the MSNs within the cluster may still have some remaining energy.[1] Therefore, the most stringent definition for network lifetime would be the time

[1]We assume that each MSN can only forward information to its local AFN for processing.

TABLE I
NOTATION

Symbol	Definition
N	total number of AFNs in the network
T	network lifetime
E	total provisioning energy
M	total number of energy provisioning locations
e_i	initial energy at AFN i
g_i	local generated rate at AFN i
ρ	power consumption coefficient for receiving data
α	distance independent term in power consumption coefficient for receiving data
β	distance dependent term in power consumption coefficient for receiving data
m	path loss index
c_{ik} (or c_{iB})	power consumption coefficient for sending data from AFN i to AFN k (or the base-station B)
d_{ik} (or d_{iB})	distance from AFN i to AFN k (or the base-station B)
D_{ik} (or D_{iB})	distance from AFN i to AFN k (or the base-station B)
f_{ik} (or f_{iB})	rate from AFN i to AFN k (or the base-station B)
V_{ik} (or V_{iB})	total volume from AFN i to AFN k (or the base-station B)
λ_i	If we provision energy at location i, $\lambda_i = 1$; otherwise, $\lambda_i = 0$
μ_i	At location i, we provision μ_i of total energy
$L_{\max}$	maximum distance from an AFN to base-station

instance when any one of the AFNs fails. We will use this definition throughout this paper.

The last component in the two-tier architecture is the base-station. The base-station is, essentially, the *sink* node for data streams from all the AFNs in the network. A base-station may be assumed to have a sufficient battery resource provision. Therefore, its power dissipation is not a concern in our investigation.

In summary, the main function of the lower-tier MSNs is data acquisition and compression while the upper-tier AFNs are used for data fusion and relaying the information to the base-station. The routing topology can be controlled by the power level of the transmitter ([18], [20], [22]) because it directly controls the distance coverage of an AFN.

B. Power Dissipation

Table I lists the notation used in this paper. For the ease of exposition, we assume that the rate of data stream g_i generated at AFN i (after data aggregation) is of constant bit rate. For an AFN, the power consumption by data communication (*i.e.,* receiving and transmitting) is the dominant factor [1]. The power dissipation at the transmitter can be modeled as

$$p_t(i,\ k) = c_{ik} \cdot f_{ik}\ , \tag{1}$$

where $p_t(i,\ k)$ is the power dissipated at node i when it is transmitting to node k, f_{ik} is the bit rate transmitted from node i to k, and c_{ik} is the power consumption cost of radio link (i, k) and is given by

$$c_{ik} = \alpha + \beta \cdot d_{ik}^m\ , \tag{2}$$

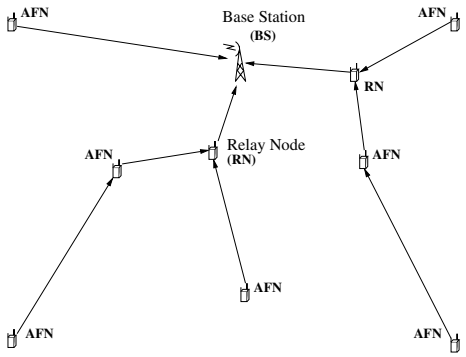


Fig. 2. Traffic routing with relay nodes.

where α is a *distance-independent* constant term, β is a coefficient term associated with the *distance-dependent* term, d_{ik} is the distance between these two nodes, and m is the path loss index, with $2 \le m \le 4$ [19]. Typical values for these parameters are $\alpha = 50$ nJ/b and $\beta = 0.0013$ pJ/b/m^4 (for $m = 4$) [12]. The power dissipation at a receiver can be modeled as [19]

$$p_r(i) = \rho \cdot \sum_{k \neq i} f_{ki}\ , \tag{3}$$

where $\sum_{k \neq i} f_{ki}$ (in b/s) is the rate of the received data stream (from other AFNs) at node i. A typical value for the parameter ρ is 50 nJ/b [12].

C. The Joint Energy Provisioning and Relay Node Placement Problem

For a network with N AFNs, where each AFN i generates data with rate g_i, suppose that the initial energy at each node is e_i $(1 \le i \le N)$. Then it is straightforward to use a linear programming (LP) approach to find an optimal flow routing schedule such that the network lifetime is maximized [6].

Now we take one step further. Suppose for a number of reasons, this network lifetime is not adequate to meet the required lifetime. Then it is necessary to take some measures to prolong the network lifetime. One straightforward measure is to provision additional energy on existing AFNs in the network. As we shall see later in this paper, there may exist intrinsic geometric deficiencies with the underlying network topology that cannot be efficiently addressed by just adding more energy on existing AFNs. Instead, a powerful technique to mitigate such geometric deficiencies would be to deploy additional *relay nodes* (RNs) at certain locations into the network (see Fig. 2). Physically, these RNs are very much similar to the AFNs, except that they do not generate any information locally as AFNs; the RNs are used solely to relay network traffic toward the base-station. We will show that deploying these RNs at certain critical positions in the network is much more efficient than just adding the same amount of energy on existing AFNs.

For a given pool of energy E and M RNs, the question to ask becomes: Where should we deploy RNs into the network

and how should we allocate the total amount of energy E into M portions such that the network lifetime can be maximized?

There is one subtle problem that needs to be clarified. Should we find that an RN happens to coincide with an AFN, what should we do with this RN? In this case, there is really no need to deploy this additional RN since we can provision the same amount of additional energy directly onto an existing AFN while achieving the same effect. Under this setting, a general interpretation for the number M might be that it represents the maximum number of possible *locations* that we can provision energy into the network.

For the joint EP-RNP problem, assume that the data rates from node i to node k and to the base-station B are f_{ik} and f_{iB}; $(x_i,\, y_i)$, $N < i \le N+M$, are variable coordinates for the placements of the M RNs; d_{ik} and d_{iB} are the distances from node i to node k and to the base-station B, c_{ik} and c_{iB} are the link costs from node i to node k and to the base-station B, respectively. For each AFN i, $1 \le i \le N$, the following flow balance equation and energy constraint must be met.

$$f_{iB} + \sum_{1\le k\le N+M}^{k\neq i} f_{ik} = \sum_{1\le m\le N+M}^{m\neq i} f_{mi} + g_i \ ,$$

$$\sum_{1\le k\le N}^{k\neq i} c_{ik} f_{ik} T + \sum_{N+1\le k\le N+M} (\alpha + \beta d_{ik}^4) f_{ik} T + \sum_{1\le m\le N+M}^{m\neq i} \rho f_{mi} T + c_{iB} f_{iB} T - E\mu_i \le e_i \ .$$

For each RN i, $N < i \le N+M$, it must also meet the flow balance equation and energy constraint

$$f_{iB} + \sum_{1\le k\le N+M}^{k\neq i} f_{ik} = \sum_{1\le m\le N+M}^{m\neq i} f_{mi} \ ,$$

$$\sum_{1\le m\le N+M}^{m\neq i} \rho f_{mi} T + \sum_{1\le k\le N+M}^{k\neq i} (\alpha + \beta d_{ik}^4) f_{ik} T + (\alpha + \beta d_{iB}^4) f_{iB} T - E\mu_i \le 0 \ ,$$

where $\sum_{i=1}^{N+M} \mu_i = 1$ and at most M of them are positive.

Denote $V_{ik} = f_{ik}T$, $V_{iB} = f_{iB}T$, $D_{ik} = d_{ik}^4$, and $D_{iB} = d_{iB}^4$, we formulate the EP-RNP problem as follows.

(EP-RNP) Maximize T
subject to

$$V_{iB} + \sum_{1\le k\le N+M}^{k\neq i} V_{ik} - \sum_{1\le m\le N+M}^{m\neq i} V_{mi} - g_i T = 0 \quad (1 \le i \le N) \tag{4}$$

$$V_{iB} + \sum_{1\le k\le N+M}^{k\neq i} V_{ik} - \sum_{1\le m\le N+M}^{m\neq i} V_{mi} = 0 \quad (N < i \le N+M) \tag{5}$$

$$\sum_{1\le k\le N}^{k\neq i} c_{ik} V_{ik} + \sum_{N+1\le k\le N+M} (\alpha V_{ik} + \beta D_{ik} V_{ik}) + \sum_{1\le m\le N+M}^{m\neq i} \rho V_{mi} + c_{iB} V_{iB} - E\mu_i \le e_i \quad (1 \le i \le N) \tag{6}$$

$$\sum_{1\le m\le N+M}^{m\neq i} \rho V_{mi} + \sum_{1\le k\le N+M}^{k\neq i} (\alpha V_{ik} + \beta D_{ik} V_{ik}) + \alpha V_{iB} + \beta D_{iB} V_{iB} - E\mu_i \le 0 \quad (N < i \le N+M) \tag{7}$$

$$\sum_{i=1}^{N+M} \lambda_i = M \tag{8}$$

$$\sum_{i=1}^{N+M} \mu_i = 1 \tag{9}$$

$$\mu_i - \lambda_i \le 0 \qquad (1 \le i \le N+M) \tag{10}$$

$$(x_i - x_k)^2 + (y_i - y_k)^2 - D_{ik}^{0.5} \le 0 \qquad (1 \le i, k \le N+M,\ i \text{ or } k > N,\ i \neq k) \tag{11}$$

$$(x_i - x_B)^2 + (y_i - y_B)^2 - D_{iB}^{0.5} \le 0 \qquad (N < i \le N+M) \tag{12}$$

$$c_{ik} = \alpha + \beta D_{ik} \qquad (1 \le i, k \le N+M,\ i \text{ or } k > N,\ i \neq k) \tag{13}$$

$$c_{iB} = \alpha + \beta D_{iB} \qquad (N < i \le N+M) \tag{14}$$

$$V_{ik}, V_{iB}, D_{ik}, D_{iB}, c_{ik}, c_{iB} \ge 0 \qquad (1 \le i, k \le N+M, i \neq k)$$

$$T, \mu_i \ge 0, \lambda_i = 0 \text{ or } 1 \qquad (1 \le i \le N+M) \tag{15}$$

The physical interpretation of the above formulation is as follows. The set of constraints in (4) are bit volume balance equations for those AFNs that can generate their own traffic. The set of constraints in (5) are bit volume balance equations for those potential RNs (that do not generate their own traffic). The set of inequalities in (6) are the energy constraints for the N AFNs. The set of inequalities in (7) are the energy constraints for the M potential RNs. The first term in these inequalities represents the energy spent on receiving data streams from other nodes, and the second term represents the energy spent on transmitting data streams to other nodes. The sets of constraints in (8), (9), (10), and (15) ensure that we can provision energy to at most M locations (including stand alone RNs and those RNs that coincide with AFNs). In particular, the set of constraints in (10) and (15) assert that $0 \le \mu_i \le 1$, λ_i can be only 0 or 1, and $\mu_i > 0$ only if $\lambda_i = 1$. The constraints in (11) and (12) represent the inter-nodal distances, whereas the constraints in (13) and (14) represent the link costs among the nodes.[2]

III. SPINDS: A Competitive Heuristic Solution

The problem formulation for EP-RNP is in the form of a *mixed-integer nonlinear programming (MINLP)* problem, which is NP-hard in general [9]. The state-of-the-art techniques for solving MINLP problem include *Generalized Benders Decomposition* [10], *Outer Approximation* [8], and *Branch-and-Bound* [11] methods. Since our problem is non-convex, the Generalized Benders Decomposition and Outer Approximation methods would not work well. The current state-of-the-art software for solving this type of problem is *BARON* [2], which was developed by Prof. Nick Sahinidis' group at the University of Illinois and is based on *branch-and-bound/reduce* techniques [21]. For EP-RNP, we find that BARON can only give a reasonably good solution when N

[2]Note that we use inequalities in (11) and (12) instead of equalities. This is because these inequalities have the convexity property and thus are easier to solve. After we obtain an optimal solution, we can change these inequalities back to equalities by re-assigning exact values to D_{ik}'s and D_{iB}'s. Consequently, we will obtain an optimal solution [3].

and M are very small (*e.g.,* N less than 5) and it fails to provide reasonable lower and upper bounds for network of moderate size.

In this section, we present a competitive heuristic algorithm to the EP-RNP problem. Before we describe this algorithm, we present some basic results on the relationship among the search space, MILP/LP formulation, and optimality, which will be the basis for some of the simplifications we will make in the algorithmic design.

A. Some Basic Results

Suppose that the locations for the M potential RNs are *fixed* (although they may not be optimally placed). Then c_{ik} and c_{iB} (in Eqs. (13) and (14)) are now all constants and we can solve the energy provisioning problem by tackling a mixed-integer linear programming (MILP) problem, which we call EP(AFN+RN). EP(AFN+RN) attempts to allocate a total amount of additional energy E to M points, where these M points are an optimal set of M nodes drawn from the collection of N AFNs and M RNs. Unfortunately, an MILP problem is NP-complete in general [9]. Although there exist software (*e.g.,* CPLEX and LINDO) for solving MILP problems, the computational time with such software is only acceptable for a *one-time* computation. In other words, such software are not suitable for a large number of *repetitive* routine calls as required in our heuristic algorithm. To ensure that the heuristic algorithm is computationally efficient, we must find an alternative approach other than solving the MILP problem directly.

Let us examine the following simplified problem. Instead of drawing an optimal set of M points out of the N AFNs and M RNs, we consider provisioning energy *only* to the M RNs and denote this problem as EP(RN). The problem formulation for EP(RN) is similar to the EP(AFN+RN) problem except that the set of constraints $\sum_{i=1}^{N+M} \mu_i = 1$, $\sum_{i=1}^{N+M} \lambda_i = M$ ($\lambda_i = 0$ or 1), and $\mu_i - \lambda_i \leq 0$ are replaced by $\sum_{i=N+1}^{N+M} \mu_i = 1$ and $\mu_i = 0$ ($1 \leq i \leq N$); and when an RN coincides with an AFN, there is no energy consumption for receiving and transmitting data stream between them. Clearly, this EP(RN) problem is an LP and can be solved efficiently [15].

It is not hard to see that for the same fixed network topology and initial energy on each AFN, an optimal solution for the EP(RN) problem is not better than an optimal solution for the EP(AFN+RN) problem. This is intuitive and can be easily proved by noting that the solution for the EP(RN) only consider one special case for the EP(AFN+RN), *i.e.,* provisioning energy only to the M RNs.

Now, let us consider the following situation. Instead of comparing the solutions to EP(AFN+RN) and EP(RN) for one topology instance, how about that we try out *all* possible locations for placing the M RNs and compare the *best* solution under EP(AFN+RN) and EP(RN) among all possible placement topologies? The answer to this question is a key to our algorithmic design and is given in the following lemma.

Lemma 1: **(Optimal Equivalence)** *Suppose that the M RNs can be arbitrarily placed over some region (including those locations for the N AFNs). Then, the best placement solution (among all possible solutions) for problem EP(AFN+RN) yields the same network lifetime performance as the best placement solution for problem EP(RN).*

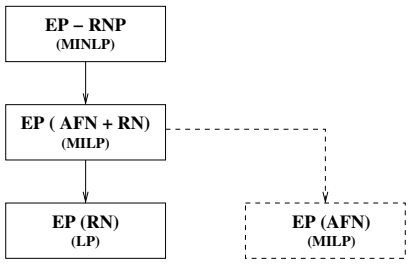


Fig. 3. Relationship among the problems and their complexity in our investigation.

The proof is omitted here. Interested readers are referred to [13].

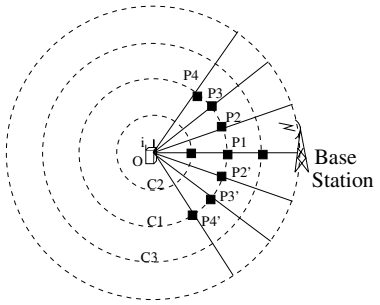
Lemma 1 suggests that, if we choose the M-node placement points wisely, then the solution to the simpler EP(RN) problem will yield a similar result as that to the EP(AFN+RN) problem. We will exploit this result in the design of our heuristic algorithm. Figure 3 shows the relationship among all the problems we have explored so far in this paper. EP-RNP is an MINLP problem, which is NP-hard and most difficult to solve. If we assume that the locations for the M RNs are fixed, then the EP-RNP problem becomes EP(AFN+RN), which is an MILP problem and is NP-complete in general. However, for a one-time computation, software packages such as LINDO can solve it in acceptable time for the sizes of network under our investigation. On the other hand, if we consider provisioning energy only onto the RNs, then the EP(AFN+RN) problem becomes EP(RN), which is LP and can be solved efficiently. In Section IV, we will also consider the case that energy is only added onto the existing AFNs (without relay nodes). This makes EP(AFN+RN) problem become EP(AFN), which is an LP problem.

B. SPINDS: Procedural Description

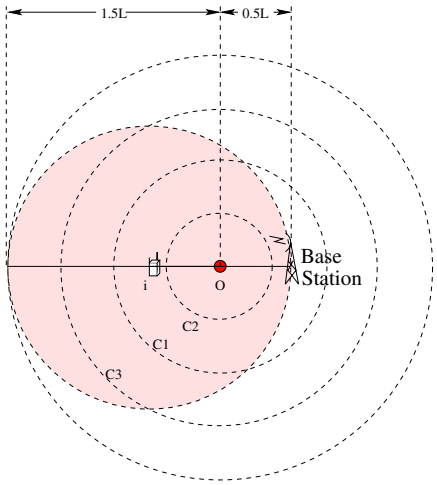
We are now ready to present our heuristic algorithm. The heuristic algorithm that we developed is called *SPINDS*, which stands for *Smart Pairing* and *INtelligent Disc Search.* The main idea of SPINDS is as follows. Suppose we start with some initial locations for the M RNs. If these locations for M RNs are not optimal, then it is possible to relocate some RN to a better location so that the network lifetime can be further extended. Now we repeat this process iteratively. Eventually, when movement of any RN cannot further increase network lifetime, we declare that the M RNs are placed at optimal locations and the algorithm terminates.

The proposed SPINDS algorithm consists of two phases: (1) Smart Pairing (SP), and (2) INtelligent Disc Search (INDS) and works as follows. Initially, we put all RNs at the base-station B.[3] At the beginning of each iteration, we first use an

[3]If we have better position that we know a priori, we can start by putting RNs at these locations. This will help speed up the algorithm's running time.



(a) Node i is an AFN.



(b) Node i is an RN.

Fig. 4. INDS step in the SPINDS algorithm.

LP to obtain a best flow routing under these RNs' locations. Based on current incoming and outgoing flows at each node, we we estimate the lifetime of each node (include AFNs and RNs) and order the nodes in increasing order of node lifetime. We identify the node with the smallest node lifetime through this process and denote it as node i. Note that node i can be either an AFN or an RN.

Suppose that node i is an AFN. In this case, we consider node i as the center point relative to other nodes and denote it as O. We then make a list of all RNs in the order of increasing distance to point O. Then we pair node i with an RN, say r_j, that is farthest away from point O in the list of the RNs. This is the *smart pairing* step in SPINDS.

Once an RN r_j is paired with node O, we attempt to move this RN to a better location within the disc region where the disc is centered at point O and has a radius (say L) equal to the distance between point O and the base-station (see Fig. 4(a)). Note that it is sufficient to search this disc area (with radius L) for RN since AFN i would reach the base-station with a shorter distance if an RN is outside this disc. It is also necessary to search the entire area of the disc (instead of only the segment between O and B). This is because we are not interested in the increase of any individual node's lifetime, but rather the lifetime of the entire network. An increase of network lifetime will need the *collaboration* of re-arranging flow routing topology among *all* N AFNs and M potential RNs, which means that any point on the disc could be a potential candidate to place this RN and make an improvement in network lifetime.

Since AFN i has finite energy, the closer the RN r_j moves to AFN i, the longer the lifetime of AFN i can be prolonged. The closest position to AFN i, in the extreme case, is point O itself. Therefore, we first try to put the RN r_j to coincide with point O, which corresponds to the situation that energy will be provisioned on AFN i directly. With this placement, if the network lifetime is increased, we are done. Otherwise, the possible distance from r_j to i is in $(R_l, R_u) = (0, L)$. We search a circle C_1 having radius $R = (R_l + R_u)/2 = L/2$. In particular, we start from point P_1 (see Fig. 4(a)) and move along the circle C_1 with equal phase angle θ. That is, we try points $(P_2, P_2^{'})$, $(P_3, P_3^{'})$, $(P_4, P_4^{'})$, and so forth on the circle C_1 over 360 degrees. If the network lifetime increases when the RN is placed at any of these new points on the circle C_1, we update R_u by R and move to circle C_2; otherwise (no network lifetime improvement), we update R_l by R and move to circle C_3. Again, the radius of the new circle is $R = (R_l + R_u)/2$. Then we repeat the search process for the points on the new circle as we have done for circle C_1. Clearly, the radius of each circle involved in the search process resembles a binary search. Eventually, the search terminates if $(R_u - R_l)$ is less than a threshold δ_L. This is the so-called *intelligent disc search* (INDS) step.

The case when node i is an RN is similar to that for the case when i is an AFN, except that the center of the disc, O, is now defined as the mid-point between RN i and the base-station (see Fig. 4(b)). The reason why we choose this mid-point as the disc center is as follows. Since node i is an RN, its energy is therefore also adjustable. Thus, the lifetime of RN i can be prolonged by adding more energy. A good starting point to place an RN (from the viewpoint of RN i) would be the mid-point between i and B, which we choose as the disc center O. Since point O is the center point, we organize the RNs (excluding RN i) in a list in increasing distance toward point O, and designate the RN having a largest distance toward point O as RN r_j. We pair $(i,\ r_j)$ together and start INDS. Although the search region is still a disc centered at i with radius L (shaded area in Fig. 4(b)), where L is the distance between i and B, the center of the search circle is now O. Hence, the largest circle that covers the shaded disc has a radius of $1.5L$. However, when we search points on circles C_1, C_2, C_3, and so forth for a better RN placement point, it is only necessary to search the portion of the circle that lies within the shaded disc area. The radius of each circle involved in the search process also resembles a binary search.

At the end of each iteration, we will either have moved the RN to a new location and obtained an increased network lifetime (as well as an energy provisioning strategy) or have no improvement and thus this RN r_j will not be moved. In

Notation for the SPINDS algorithm.

NS: A stack of nodes, including N AFNs and M RNs.
In NS, nodes closer to the top have smaller lifetimes.
O: Center of search disc. If current pairing node i is an AFN,
O is i, else O is the mid-point between i and B;
RS_i: A stack of RNs that is dynamically formed according to the current pairing node.
In RS, nodes closer to the top are farther away from O.
$[R_l, R_u]$: The search space for radius.
δ_L: The threshold for radius change.
θ: The phase (or angle) increment during a search on a circle.

Smart Pairing (SP) in the SPINDS algorithm.

```
Initialization:
    Put all M RNs at the base-station B and let increase=1;
    Solve an EP(RN) problem to obtain the optimal T, energy
    allocation, and flow routing;
while (increase==1) {
    Reset increase=0;
    for (i = 1; i ≤ N + M; i++)
        Estimate node i's lifetime;
    Sort nodes in non-increasing order of its estimated node
    lifetime and arrange the sorted list with a stack NS;
    while ((NS!=NULL) and (increase==0)) {
        i = pop (NS);
        Sort all RNs in non-decreasing order of distance to O
        and arrange the sorted list with a stack RS_i;
        (if i is an RN, i will not be included in RN_i)
        while (RS_i!=NULL) {
            j = pop(RS_i);
            increase=INDS(i, j);
            if (increase==1)
                break; }}}
```

INtelligent Disc Search (INDS) in the SPINDS algorithm.

```
int INDS(int i, int j) {
    relocate=0;
    if (i is an AFN)
        R_u = d_iB;
    else // i.e., i is an RN
        R_u = d_iB * 1.5;
    Try to put RN r_j at point O, if network lifetime increases {
        Relocate r_j to point O;
        return 1; }
    R_l=0;
    while (R_u − R_l > δ_L) {
        R=(R_l+R_u)/2;
        Try to put RN r_j on the circle centered at O and with
        radius R for every θ degree;
        //When i is an RN, if d_ij > d_iB, no need to try this point
        If network lifetime increases {
            Relocate RN r_j to the new position;
            relocate=1;
            R_u=R; }
        else R_l=R; }
    return relocate; }
```

Fig. 5. An implementation of SPINDS algorithm.

the former case, we move on to the next iteration (start with a new set of pairing). In the latter case, we drop RN r_j from the current pairing and choose the next RN in the list of RNs, which is the RN that is second farthest away from point O. Should all the RNs have been considered for pairing, we move on to the next node on the list with the second smallest node lifetime and perform the same pairing and search process. The algorithm terminates when the network lifetime cannot be further improved after pairing all the nodes with all the RNs during an iteration. The pseudo-code for the SPINDS algorithm is given in Fig. 5.

Substituting MILP with LP. Note that for each placement trial for RN r_j, we need to solve an EP(AFN+RN) problem, which is an MILP. Due to repetitive routine calls by SPINDS, the computational burden would be prohibitively high. Fortunately, Lemma 1 shows that, if the heuristic algorithm is designed wisely, then solving a simpler EP(RN) (which is an LP) would yield the same result. Thus, we will use the simpler EP(RN) computation for each RN r_j placement decision during each iteration of SPINDS. It is not hard to show that SPINDS algorithm terminates in polynomial time.

TABLE II

LOCATIONS, DATA GENERATING RATE, AND INITIAL ENERGY OF EACH AFN FOR A 10-AFN NETWORK.

AFN	(x_i, y_i) (m)	g_i (kb/s)	e_i (kJ)
1	(45,450)	2	18.63
2	(-50,90)	10	49.68
3	(-455,475)	7	83.60
4	(460,-85)	9	62.20
5	(0,-480)	2	15.76
6	(-500,-485)	5	87.58
7	(-485,40)	1	45.51
8	(100,-490)	8	52.27
9	(470,-495)	1	16.38
10	(95,-40)	3	95.69

IV. Simulation Results

In this section, we present simulation results. We demonstrate the performance of SPINDS for general network configurations and compare it to some other approaches. We will also provide important insights on the EP-RNP problem.

We will use the 10-AFN and 50-AFN network topologies for our numerical investigation. Without loss of generality, in all network topologies, we assume that the base-station is at the origin point $(0, 0)$ (in meters). Tables II and III give each AFN's location (x_i, y_i) (in meters), local data generating rate g_i (in kb/s), and initial energy e_i (in kJ) for each topology, respectively, all of which are generated randomly. The amount of available provisioning energy for the 10-AFN and 50-AFN networks are 1000 kJ and 500 kJ, respectively, which are also set randomly. We set $\theta = 30$ degrees and $\delta_L = 50$ m for our EP-RNP algorithm.

To demonstrate the performance of SPINDS, we compare it with two other approaches to the EP-RNP problem. The first approach is the *greedy incremental* (GI) algorithm and is based on the following simple idea. Although it is not computational feasible to perform exhaustive search for placing M RNs *simultaneously*, it is possible to choose an optimal position to place *one* RN at a time. The best location for placing one node can be found by exhaustively searching *all* tiny grids that are drawn within the feasible region. Once the location for this RN is fixed, we can place the next RN following the same process. Under this approach, the RNs are placed one by one until all M potential RNs are placed. We choose grid size to be 10×10 m for the 1000×1000 network dimension, which corresponds to 10^4 grids.

Another approach that we use in comparison is to provision the available energy E only on the existing N AFNs without deploying additional RNs, which is the so-called EP(AFN) problem in Fig. 3. In this approach, since M provisioning points can only be chosen from the existing N AFNs, we

TABLE III

LOCATIONS (M), DATA GENERATING RATE (KB/S), AND INITIAL ENERGY (KJ) FOR EACH AFN FOR THE 50-AFN NETWORK.

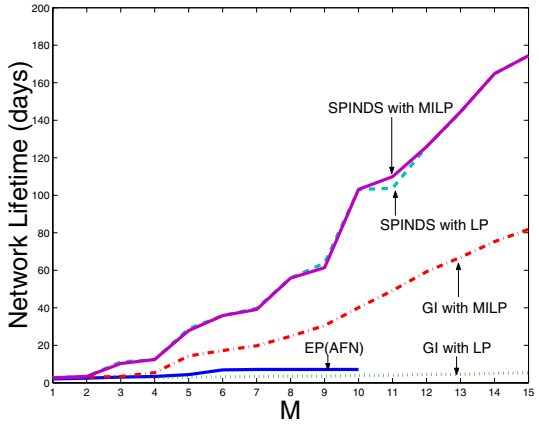
AFN	(x_i, y_i)	g_i	e_i	AFN	(x_i, y_i)	g_i	e_i	AFN	(x_i, y_i)	g_i	e_i
1	(-300,320)	2	83.2	18	(-390,330)	8	67.2	35	(60,500)	7	80.6
2	(340,-420)	8	69.9	19	(290,430)	2	61.0	36	(-40,350)	6	100.0
3	(-120,290)	4	91.2	20	(-160,-380)	9	56.9	37	(-80,130)	6	92.8
4	(250,470)	5	100.0	21	(-10,-350)	4	60.8	38	(-200,490)	10	56.9
5	(-270,-220)	5	96.5	22	(-450,-250)	4	93.9	39	(420,-440)	6	89.9
6	(-10,430)	7	58.3	23	(-340,-410)	9	69.7	40	(-70,-120)	3	62.5
7	(0,300)	7	97.6	24	(-190,-390)	3	80.7	41	(-70,-100)	5	52.3
8	(-470,-340)	9	84.4	25	(-320,0)	9	60.7	42	(420,330)	8	95.2
9	(500,-90)	3	85.6	26	(-400,-180)	4	93.9	43	(-60,-400)	9	64.7
10	(190,500)	8	99.9	27	(60,-140)	10	84.5	44	(-340,300)	3	92.4
11	(-320,-440)	3	61.7	28	(-300,300)	3	54.6	45	(-490,180)	2	79.0
12	(-200,-390)	9	73.5	29	(-200,-260)	6	96.8	46	(240,140)	9	76.6
13	(430,220)	2	63.0	30	(-160,-420)	7	69.5	47	(110,-250)	2	76.2
14	(-80,-350)	5	73.3	31	(320,-360)	7	79.7	48	(390,490)	2	93.2
15	(330,460)	7	86.3	32	(-380,-310)	9	83.5	49	(-60,400)	4	89.1
16	(60,30)	7	100.0	33	(-370,-110)	5	80.4	50	(410,-400)	6	69.0
17	(180,120)	3	92.3	34	(-220,190)	8	64.2	—			

must have $M \leq N$. Note that the constraint $M \leq N$ does not apply to SPINDS or GI, where M can exceed N.

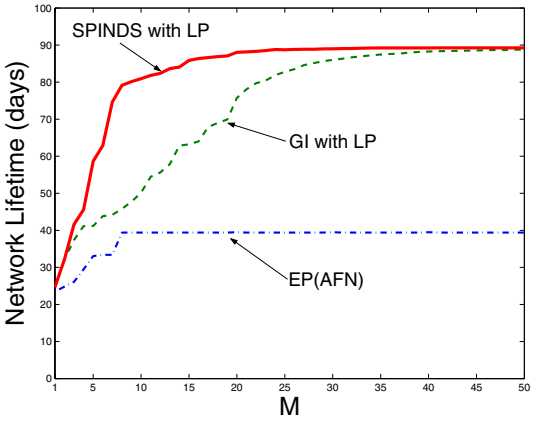
Figure 6(a) shows that, given total available energy $E = 1000$ kJ, the maximum network lifetime obtained under different energy provisioning approaches for the 10-AFN network. For the SPINDS and GI approaches, we also used both LP and MILP in the solution process. There are several important observations from this figure. First, we note that for the SPINDS algorithm, the numerical results using the MILP and LP match closely with each other. Recall that in Lemma 1, if we can choose the M-node placement points *wisely,* without losing critical points in the search space, then the solution obtained by solving the simple EP(RN) problem will yield the same result as that obtained by solving the EP(AFN+RN) problem. Therefore, we see that the SPINDS algorithm indeed explores search space wisely, thereby justifying the use of the LP (and thus polynomial) instead of the MILP in our algorithmic design.

Second, we examine the *greedy incremental* (GI) approach using both the MILP and LP techniques. Clearly, the LP technique for the GI approach is considerably worse than the MILP approach. This is mainly due to the fact that, under greedy algorithm, the locations for RNs deployed during earlier iterations cannot be changed in future iterations. As a result, although the location for each individual RN is best chosen during each *incremental placement,* the locations for the M RNs, *when they are considered jointly,* are poorly chosen. Consequently, Lemma 1 would not be applicable here and we conclude that the GI approach cannot offer good solutions for EP-RNP with LP techniques.

Third, under the EP(AFN) approach where there is no RN and additional energy can only be added on the existing AFNs, the network lifetime performance is very poor comparing to SPINDS. Even when M increases, the increase in network lifetime is still very small. This phenomena conclusively demon-



(a) A 10-AFN network.



(b) A 50-AFN network.

Fig. 6. Network lifetime results from various energy provisioning approaches and computational techniques.

strates that there indeed exists deficiencies in this network topology and energy provisioning on existing AFNs alone cannot mitigate this problem and bring much improvement in network lifetime performance. In this case, RN placement is the only viable approach to fundamentally mitigate network geometric deficiency and prolong network lifetime.

Finally, we find that under the same total amount of available energy, the number of RNs can have a significant impact on the overall network lifetime performance. For example, in Fig. 6(a), under SPINDS, the network lifetime can increase 65 folds as the number of energy provisioning points (M) increases from 1 to 15 under the same total provisioning energy of 1000 kJ.

To explore the performance limits of RNP, in Fig. 6(b), we plot the network lifetime performance for a $N = 50$ node network under three different approaches.[4] For $N = 50$ node network, the geometric deficiency problem is less of an issue comparing to the $N = 10$ network discussed earlier. As a result, we suspect that the improvement of SPINDS over other approaches may not be very significant. In Fig. 6(b), we find that SPINDS is still noticeably better than GI when the total number of nodes in the network ($M + N$) is less than 70 (or $M \leq 20$). For $M > 20$, or the total number of nodes in the network exceeds 70 ($N + M > 70$), the difference in network lifetime performance between SPINDS and GI diminishes. Specifically, both SPINDS and GI tends to reach a saturation point as the number of RNs increases. The interpretation for this phenomena is that when the network density becomes sufficiently high, all of its geometric deficiency will be effectively mitigated (even under GI approach). As a result, once above a density threshold, the network lifetime will reach a saturation point over which RNP can no longer further increase this lifetime limit. For the network under consideration, the network lifetime limit is approximately 89 days. Even under this scenario, there is still advantage of using SPINDS over GI. This is because SPINDS tends to approach this limit much faster than GI. In particular, with only $M = 15$ RNs, SPINDS can almost reach this limit, while under GI, it will take at least $M = 30$ RNs.

V. Related Work

Due to energy constraint, the operational lifetime for a wireless sensor network is limited. As a result, there is a flourish of research activities on how to maximize network lifetime in recent years. Most of these efforts (see, e.g., [4], [5], [6], [14], [25]) studied lifetime problem under given node energy. The possibility of provisioning additional energy was not considered.

At the time of this work, there is very limited work that directly addresses energy provisioning problem. Although there are some efforts devoted to how to exploit possible renewable energy in wireless sensor networks (see, e.g., [16], [17]), these efforts have important difference with the problem considered in this paper. Specifically, these efforts assume sensor nodes are capable of generating renewable energy over an extended period of time. As a result, a sensor network could have indefinite operational lifetime. In contrast, this paper does not explicitly assume each sensor node has energy-generation capability. Instead, we only consider a one-time energy provisioning for the network with the objective of maximizing network lifetime.

Relay node deployment has been been explored in [23], [24]. In [24], Xu et al. proposed three random deployment strategies for relay nodes, namely, connectivity-oriented, lifetime-oriented, and hybrid deployment. However, there was no explicit network lifetime optimizing formulation, as we did in this paper. In [23], the authors assume relay nodes are mobile and can move around in the network to relay information. This assumption is valid in the case when energy is not a major concern on these mobile relay nodes. However, when relay nodes are also energy-constrained (as we have considered in this paper), the energy consumption associated with mobility could become a serious concern.

VI. Conclusions

In this paper, we investigated the important problem of energy provisioning for wireless sensor networks. We considered a two-tier wireless sensor network and studied the joint problem of energy provisioning and relay node placement (EP-RNP) for the upper tier aggregation and forwarding nodes (AFNs) to increase network lifetime. Since the EP-RNP problem formulation is NP-hard, we developed an efficient polynomial-time heuristic algorithm, SPINDS, that solves the EP-RNP problem. SPINDS is an iterative algorithm that attempts to increase the network lifetime by iteratively moving an RN to a better location. The polynomial running time property of SPINDS was achieved by transforming the original MINLP problem into an iterative LP problem. Through numerical results, we showed that the proposed SPINDS is highly competitive in solving EP-RNP problem when compared with some other approaches. We also offered some important insights on network geometric properties, RN placement, and energy provisioning.

VII. Acknowledgements

The work by Y.T. Hou, Y. Shi and S.F. Midkiff has been supported in part by National Science Foundation (NSF) under Grants ANI-0312655 and CNS-0347390 and by Office of Naval Research (ONR) under Grants N00014-03-1-0521 and N00014-05-1-0179. The work of H.D. Sherali was supported in part by the NSF under Grant DMI-0094462.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks (Elsevier),* vol. 38, no. 4, pp. 393-422, 2002.

[2] BARON Global Optimization Software, `http://archimedes.scs.uiuc.edu/baron/baron.html`.

[3] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms,* second edition, John Wiley & Sons, Inc., New York, NY, 1993.

[4] Note that the complexity of the SPINDS with MILP approach and the GI with MILP approach are both too high for numerical computation. Therefore, their results are not shown in Fig. 6(b).

[4] M. Bhardwaj and A.P. Chandrakasan, "Bounding the lifetime of sensor networks via optimal role assignments," in *Proc. IEEE Infocom,* pp. 1587–1596, New York, NY, June 23-27, 2002.

[5] T.X. Brown, H.N. Gabow, and Q. Zhang, "Maximum flow-life curve for a wireless ad hoc network," in *Proc. ACM MobiHoc,* pp. 128–136, Long Beach, CA, Oct. 4–5, 2001.

[6] J.-H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *Proc. IEEE Infocom,* pp. 22–31, Tel Aviv, Israel, March 26–30, 2000.

[7] J. Chou, D. Petrovis, and K. Ramchandran, "A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks," in *Proc. IEEE Infocom,* pp. 1054–1062, San Francisco, CA, March 30–April 3, 2003.

[8] M.A. Duran and I.E. Grossmann, "An outer-approximation algorithm for a class of mixed-integer nonlinear programs," *Mathematical Programming,* vol. 36, pp. 307–339, 1986.

[9] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness,* W. H. Freeman and Company, New York, NY, 1979.

[10] A.M. Geoffrion, "Generalized benders decomposition," *Optimization Theory and Applications,* vol. 10, no. 4, pp. 237–260, Oct. 1972.

[11] O.K. Gupta and A. Ravindran, "Branch and bound experiments in convex nonlinear integer programming," *Management Science,* vol. 31, pp. 1533–1546, 1985.

[12] W. Heinzelman, *Application-specific Protocol Architectures for Wireless Networks,* Ph.D. thesis, Massachusetts Institute of Technology, June 2000.

[13] Y.T. Hou, Yi Shi, H.D. Sherali, and S.F. Midkiff, "On energy provisioning and relay node placement for wireless sensor networks," *Technical Report,* the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, June 2005. Available at `http://www.ece.vt.edu/~thou/Publications.html`.

[14] K. Kalpakis, K. Dasgupta, and P. Namjoshi, "Maximum lifetime data gathering and aggregation in wireless sensor networks," in *Proc. IEEE International Conference on Networking (ICN),* pp. 685–696, Atlanta, GA, Aug. 2002.

[15] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proc. ACM Symposium on Theory of Computing,* pp. 302–311, 1984.

[16] L. Lin, N.B. Shroff, and R. Srikant, "Asymptotically optimal power-aware routing for multihop wireless networks with renewable energy sources," in *Proc. IEEE Infocom,* Miami, FL, March 13-17, 2005.

[17] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proc. ACM/IEEE International Symposium on Information Processing in Sensor Networks (IPSN),* pp. 457–462, Los Angeles, CA, April 25–27, 2005.

[18] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proc. IEEE Infocom,* pp. 404–413, Tel Aviv, Israel, March 26–30, 2000.

[19] T.S. Rappaport, *Wireless Communications: Principles and Practice,* Prentice Hall, NJ, 1996.

[20] V. Rodoplu and T.H. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications,* vol. 17, no. 8, pp. 1333–1344, Aug. 1999.

[21] M. Tawarmalani, *Mixed Integer Nonlinear Programs: Theory, Algorithms and Applications,* Ph.D. thesis, Univ. of Illinois at Urbana-Champaign, Aug. 2001.

[22] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *Proc. IEEE Infocom,* pp. 1388–1397, Anchorage, AK, April 22–26, 2001.

[23] W. Wang, V. Srinivasan, and K. Chua, "Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks," to appear in *Proc. IEEE Mobicom,* Cologne, Germany, Aug. 28–Sept. 2, 2005.

[24] K. Xu, H. Hassanein, and G. Takahara, "Relay node deployment strategies in heterogeneous wireless sensor networks: multiple-hop communication case," in *Proc. IEEE SECON,* Santa Clara, CA, Sept. 2005.

[25] H. Zhang and J. Hou, "On deriving the upper bound of α-lifetime for large sensor networks," in *Proc. ACM MobiHoc,* pp. 121–132, Tokyo, Japan, May 24–26, 2004.