

# Maximum Lifetime Routing and Data Aggregation for Wireless Sensor Networks

Cunqing Hua and Tak-Shing Peter Yum

Department of Information Engineering  
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong  
{chua0, yum}@ie.cuhk.edu.hk

**Abstract.** In this paper we solve the maximum lifetime routing (MLR) problem for a sensor network by joint optimizing routing and data aggregation. We present a smoothing method to overcome the nondifferentiability of the objective function. By exploiting the special structure of the network, we derive the necessary and sufficient conditions to achieve the optimality. Based on these conditions, a gradient descent algorithm is designed for its solution. The proposed algorithm is shown to converge to the optimal value efficiently under all network configurations. The incorporation of optimal routing and data aggregation is shown via many examples to provide significant improvement of the network lifetime.

## 1 Introduction

Energy-efficient routing [1–3] has long been studied in the context of wireless ad-hoc networks and sensor networks. Its basic idea is to route the packets through the minimum energy pathes so as to reduce the end-to-end energy consumption. But this tends to overload the minimum energy path, causing the nodes on this path quickly run out of battery energy and disconnecting a vital link. This is undesirable in particular for sensor network where sensor nodes are collaborating for common work.

To cope with this problem, many researchers have proposed to study the maximum lifetime problem based on the linear programming formulation [4–7]. Here, instead of trying to minimize the path energy consumption, the objective is to select the route and the corresponding power levels to maximize the network lifetime. According to the criticality of a specific mission, the network lifetime may have different definitions, such as given by [8]. The key problem that these schemes try to address is how to find the route and the corresponding flow rates without centralized computations.

The above schemes are applicable for both wireless ad-hoc networks and sensor networks. But for sensor networks, an important feature was not considered in these schemes. It is well-known that data collected in the sensor network is spatially correlated, that is, there exists redundancy in the data collected by the neighboring nodes. It is therefore possible to reduce transmission overhead by aggregating the data at the intermediate nodes. Some research efforts have been made to exploit the data correlation feature to improve the performance of various communication protocols [9–14]. These work illustrated that data aggregation can greatly improve the performance of various communication protocols(channel coding, routing, MAC, etc.).

In this paper, we present a model to incorporate the routing and data aggregation into the maximum lifetime problem. By taking the data correlation into the consideration, the network lifetime can be extended from two dimensions. One is to aggregate the data at the intermediate nodes so as to reduce the transmission overhead of the nodes near the sink node, the other is to do maximum lifetime routing as done by [5–7]. However, these two should be considered simultaneously. In our model, we adopt the geometric routing [15] whereby the routing is determined solely according to the node positions, and each node is associated with a set of routing variables denoted as the fraction of traffic towards its downstream neighbors. We formulate the maximum lifetime routing(MLR) problem as an optimal routing problem where the objective is to find the optimal set of variables for each node to maximize the network lifetime. The contribution of this paper includes: (i) Our model allows different data correlation models such as that proposed in [11] to be incorporated without

intervening the underlying routing scheme. (ii) We propose a smoothing method to overcome the non-differentiability of the MLR problem, with which we can design a localized algorithm for each node to compute its routing variables without significant message exchanges. The first feature is desirable because data correlation model depends highly on the specific application, so our model allows different correlation models to work with the underlying routing seamlessly, while the second feature is a must for practical implementation of the algorithm in a sensor network with large number of nodes.

In the following, we first present the system models and define the maximum lifetime routing problem in Section 2. In Section 3, we propose a smoothing function for the maximum lifetime routing problem and provide the analytic results on the optimality conditions. Simulation results are presented in Section 4 and finally we conclude this work in Section 5.

## 2 System Model

We can model the topology of a wireless sensor network as a undirected graph  $G(N, A)$ , where  $N$  is the set of nodes,  $A$  is the set of undirected links. A special node  $t \in N$  is called the sink node who is the destination of all other nodes. To capture the characteristic of the network, we need to specify, in addition, the routing model, the data correlation and aggregation model and the power consumption model.

### 2.1 Geometric Routing Model

The routing algorithm suitable for use belongs to the class of *geometric routing* algorithms [16]. Every sensor node is assumed to know its own position as well as that of its neighbors. Each node can forward packets to its neighbor nodes within its transmission range that are closer to the sink node than itself. In essence, using *geometric routing*, node makes routing decisions with only the position information of the involved nodes. It is therefore a localized algorithm and particularly suitable for large sensor networks.

Let  $N_i$  denote the set of neighbors of a node  $i$  and  $N_i = \{j | d_{ij} \leq R, j \in N\}$ , where  $d_{ij}$  is the Euclidean distance of node  $i$  and node  $j$ , and  $R$  is the radius of the transmission range. According to the *geometric routing* rule, only those neighbors that are closer to the sink node  $t$  can serve as the downstream nodes. Let us denote this set of downstream neighbors as  $S_i = \{k | d_{kt} < d_{it}, k \in N_i\}$ . Symmetrically, the set of upstream neighbors is denoted as  $A_i$ . Each link between node  $i$  and its downstream neighbor  $k \in S_i$  has a routing variable  $\phi_{ik}$  to denote the fraction of the aggregated traffic of node  $i$  that will be routed through node  $k$ . Clearly, the *flow conservation law* requires  $\sum_{k \in S_i} \phi_{ik} = 1$ .

### 2.2 Data Correlation and Aggregation Model

The aggregated traffic  $\lambda_i$  of a node  $i$  is a superposition of two parts: local traffic generated by the node itself when sensing the surrounding environments, and the transit traffic from upstream nodes. In other words,

$$\lambda_i = r_i + \sum_{j \in A_i} \lambda_j \phi_{ji}, \quad i = 1, 2, \dots, N. \quad (1)$$

In sensor networks, data collected at neighboring nodes are often spatially correlated. It is benefit to remove the redundant information collected at upstream nodes to reduce traffic overhead at the downstream nodes. To capture this feature, we adopt a data correlation model similar to that studied in [11]. Specially, if no side information is available from other nodes, the raw data  $X_j$  at a node  $j$  is originally entropy coded with  $H(X_j) = Y$  bits. However, it can be reduced to  $H(X_j | X_{i1}, \dots, X_{ik}) = y \leq Y$  bits at a downstream node  $i$ , where  $\{X_{i1}, \dots, X_{ik}\}$  is the set of side information available at node  $i$ . We define the correlation coefficient  $q_{ji}$  for a node  $j$  and node  $i$  as

$q_{ji} = 1 - H(X_j|X_{i1}, \dots, X_{ik})/H(X_j)$ , and obviously  $0 \leq q_{ji} \leq 1$ . Then the aggregated traffic after considering the correlation is

$$\lambda_i = r_i + \sum_{j \in A_i} \lambda_j \phi_{ji} (1 - q_{ji}), \quad i = 1, 2, \dots, N. \quad (2)$$

### 2.3 Power Consumption Model

A sensor node consumes power when it is sensing and generating data, receiving, transmitting, or even simply standby. The power  $e_g$  for generating one bit of data is assumed to be the same for all nodes. The standby power consumed by a node, again assumed to be the same for all nodes and independent of traffic, is denoted by  $e_s$ . For power used in receiving and transmitting, we adopt the *first order radio model* described in [1]. Specially, a node needs  $\epsilon_{elec} = 50nJ$  to run the circuitry and  $\epsilon_{amp} = 100pJ/bit/m^2$  for the transmitting amplifier. So the power consumed by a node in receiving a unit of data is given by

$$e_r = \epsilon_{elec} \quad (3)$$

and the power consumed in transmitting a unit of data packet to neighbor node  $j$  is given by

$$e_{ij} = \epsilon_{elec} + \epsilon_{amp} \cdot d_{ij}^n \quad (4)$$

Here we consider the path loss of exponent  $n$ , which is usually  $2 \leq n \leq 4$  for the free-space and short-to-medium-range radio communication. The mean power consumption of node  $i$ , denoted as  $w_i$ , is therefore

$$w_i = e_s + e_g r_i + e_r \sum_{j \in A_i} \lambda_j \phi_{ji} + \lambda_i \sum_{k \in S_i} e_{ik} \phi_{ik} \quad (5)$$

Where the first term is the standby power consumption, the second term is the power for sensing, the third term is the power for receiving and the fourth term is the power for transmitting.

### 2.4 Maximum Lifetime Problem

Assume each node  $i$  has an initial battery energy  $E_i$ , the lifetime  $T_i$  of node  $i$  is defined as the expected time for the battery energy  $E_i$  to be exhausted, that is,  $T_i = E_i/w_i$  where  $w_i$  is given by (5). Similar to [5, 6], we define the network lifetime  $T_{net}$  as the time that the first node that runs out of energy, that is

$$T_{net} = \min_{i \in N} T_i \quad (6)$$

The power consumption  $w_i$  is a function of  $\mathbf{r}$ ,  $\boldsymbol{\lambda}$  and  $\boldsymbol{\phi}$ . However, the set of aggregated traffic  $\boldsymbol{\lambda}$  can be obtained from  $\mathbf{r}$  and  $\boldsymbol{\phi}$  from (2). Therefore,  $T_{net}$  depends only on  $\mathbf{r}$ ,  $\boldsymbol{\phi}$  and the initial battery energy  $\mathbf{E}$ . If  $\mathbf{r}$  and  $\mathbf{E}$  are given, the network lifetime is solely determined by the set of routing variables  $\boldsymbol{\phi}$ . We therefore state the maximum lifetime routing(MLR) problem as follows:

**MLR:** *Given the traffic generating rate  $\mathbf{r} = \{r_i\}$ , the initial battery energy  $\mathbf{E} = \{E_i\}$  and the data correlation coefficient  $\mathbf{q} = \{q_{ij}\}$ , finding a set of routing variable  $\boldsymbol{\phi} = \{\phi_{ij}\}$  for a sensor network  $G(N, A')$  such that the network lifetime  $T_{net}$  is maximized.*

Let  $\tilde{w}_i$  denote the normalized power consumption, that is,  $\tilde{w}_i = w_i/E_i$ . It is obvious that maximizing the network lifetime  $T_{net}$  is equivalent to minimize the maximum normalized power consumption  $\tilde{w}_i$  for all  $i \in N$ . We therefore rewrite the MLR problem formally as

$$\begin{aligned} & \text{minimize } \max_{i \in N} \tilde{w}_i \\ & \text{subject to } \phi_{ij} \geq 0, \sum_{j \in S_i} \phi_{ij} = 1, \forall i. \end{aligned} \quad (7)$$

### 3 Distributed Solution for the MLR Problem

The max function (7) in the MLR problem is nondifferentiable, so some simple solutions based on the gradient descent methods are not directly applicable. There are many different approaches that have been studied to overcome this difficulty. One is to transform the min – max problem to an equivalent optimization problem (e.g. [7]), such that subgradient algorithms can be used to solve the transformed problem. There is also a family of regularization approaches to obtain the smooth approximation for the max function in literature, for example, the entropy type approximation [17, 18], the two dimensional approximation [19], etc.. All these approaches are known as a special case of the so-called smoothing method, an overview of these approaches can be found in [20]. In this section, we propose a smoothing function to approximate the max function in MLR problem (7) by exploiting the special structure of the network. We derive the necessary and sufficient conditions that are required to achieve the optimality of the smoothed problem.

#### 3.1 Problem Transformation

Note after applying the geometric routing, the original undirected network  $G(N, A)$  is reduced to a *directed acyclic graph(DAG)*  $G(N, A')$ , where  $A'$  is the set of directed links, and sink node  $t$  is the root of the DAG. For any such DAG, we can find a *separation*  $s = (N_A|N_B|N_C)$  to partition the node set  $N$  into three subsets  $N_A, N_B$  and  $N_C$ , where  $N_B$  is the *cut set*,  $N_A$  and  $N_B$  are two disjoint node sets. Without loss of generality, let the sink node be located in subset  $N_C$ .

Normally we can find many such separations given a directed acyclic graph. For a specific *separation*, we can find a set of routing variables  $\phi(s)$  for the nodes in  $N_A$  and  $N_B$  to minimize the maximum energy consumption rate of the subset  $N_B$ , which we denote as  $\tilde{w}(s) = \min \max\{\tilde{w}_l, l \in N_B\}$ . Since there exists multiple separations, we can always find the worst separation  $s^*$  which has the largest minimax energy consumption rate  $\tilde{w}(s)$  among all possible separations, i.e.,  $s^* = \arg \max\{\tilde{w}(s_1), \tilde{w}(s_2), \dots\}$ . We call the corresponding cutset  $N_B^*$  as the bottleneck set since this is the node set that limits the lifetime of the network. The problem is therefore to find a set of routing variables for the bottleneck nodes to achieve the minimax energy consumption  $\tilde{w}(s^*)$ . Formally, we have

$$\begin{aligned} & \text{minimize} \quad \max_{l \in N_B^*} \tilde{w}_l & (8) \\ & \text{subject to} \quad \phi_{ij} \geq 0, \sum_{j \in S_i} \phi_{ij} = 1, \forall i. \end{aligned}$$

Note that (8) is similar to (7) except on the following two points:

- First, the size of the problem for (8) is reduced from  $|N|$  to  $|N_B^*|$ , where  $|N|$  and  $|N_B^*|$  are the size of the set  $N$  and  $N_B^*$  respectively.
- Second, the values  $\tilde{w}_l, l \in N_B^*$  tend to have a small difference between them because they belong to the same cutset.

The max function in (8) is still not differentiable, however, we can approximate it using the smoothing methods taking advantage of the above two points. We define  $\mu = \sum_{l \in N_B^*} \tilde{w}_l / |N_B^*|$  as the mean power consumption of the bottleneck set, and introduce the following smoothing function

$$U = \mu^2 + \frac{c}{|N_B^*|} \sum_{l \in N_B^*} (\tilde{w}_l - \mu)^2 \quad (9)$$

where  $c$  is a positive nondecreasing sequence. The smoothing function is a penalty function consisting of two terms. The physical interpretation of the first term of  $U$  is to minimize the mean power consumption of bottleneck nodes. This is achieved by aggregating the data at upstream as much as possible so as to reduce the overall traffic across the bottleneck nodes. The second term of  $U$  can be understood as a penalty to the total variability of  $\tilde{w}_l$ s, which is achieved by optimal routing to equalize the power consumption of the set of bottleneck nodes.

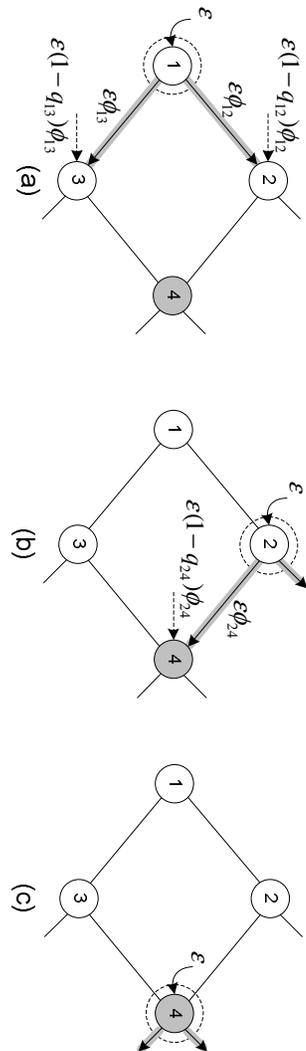


Fig. 1. Three possible relations of node  $i$ ,  $k$  and  $l$ : (a) non-adjacent source node and bottleneck node; (b) adjacent source node and bottleneck node; (c) co-located source node and bottleneck node

### 3.2 Optimality Conditions

In this section, we discuss the necessary and sufficient conditions that must be satisfied to achieve the optimality of the smoothed problem (9). Using the routing variables as the control variables, we extend the techniques in [21] to derive the necessary and sufficient conditions for the optimality. Note that the discussion of this section is applicable to non-bottleneck node cuts as well. So without abusing the notations, we use  $N_A$ ,  $N_B$  and  $N_C$  to denote three corresponding subsets.

First of all, we can rewrite the smoothing function (9) as

$$\begin{aligned}
 U &= \mu^2 + \frac{c}{|N_B|} \sum_{l \in N_B} (\tilde{w}_l - \mu)^2 \\
 &= \mu^2 + \frac{c}{|N_B|} \left( \sum_{l \in N_B} \tilde{w}_l^2 - 2\mu \sum_{l \in N_B} \tilde{w}_l + |N_B| \mu^2 \right) \\
 &= \frac{c}{|N_B|} \sum_{l \in N_B} \tilde{w}_l^2 - \frac{(c-1)}{|N_B|^2} \left( \sum_{l \in N_B} \tilde{w}_l \right)^2
 \end{aligned}$$

Here we use the fact that  $\mu = \sum_{l \in N_B} \tilde{w}_l / |N_B|$ . Since all  $\tilde{w}_l$ s are the function of the routing variable  $\phi = \{\phi_{ij}\}$ , we can differentiate  $U$  directly from the above equation as

$$\begin{aligned}
 \frac{\partial U}{\partial \phi_{ik}} &= \frac{2c}{|N_B|} \sum_{l \in N_B} \tilde{w}_l \frac{\partial \tilde{w}_l}{\partial \phi_{ik}} - \frac{2(c-1)}{|N_B|^2} \sum_{l \in N_B} \tilde{w}_l \sum_{l \in N_B} \frac{\partial \tilde{w}_l}{\partial \phi_{ik}} \\
 &= \frac{2}{|N_B|} \sum_{l \in N_B} \left( c\tilde{w}_l - (c-1)\mu \right) \frac{\partial \tilde{w}_l}{\partial \phi_{ik}}
 \end{aligned} \tag{10}$$

So what we need to do is to find  $\partial \tilde{w}_l / \partial \phi_{ik}$ . To this end, we introduce a set of dummy variables  $r_i$  ( $i \in N$ ), where  $r_i$  can be interpreted as the dummy traffic injected into node  $i$ . This dummy traffic  $r_i$  follows the same set of routing given by  $\phi$ . To derive  $\partial \tilde{w}_l / \partial \phi_{ik}$ , we need to consider three possible topologies of the source node  $i$ , its next-hop neighbor  $k$  and the bottleneck node  $l$ . A simple four-node topology is shown in Fig. 1 to illustrate these three possible scenarios.

#### (a) Non-adjacent source node

If the source node  $i$  is not adjacent to the bottleneck node  $l$ , let us consider a small increment  $\epsilon$  to the input rate  $r_i$ , this will cause an increment  $\epsilon\phi_{ik}$  to the traffic rate of its next-hop neighbor  $k$ . Taking into account the data correlation between node  $i$  and  $k$ , the amount of increment is reduced to  $\epsilon(1 - q_{ik})\phi_{ik}$  from node  $k$  downwards. Since node  $k$  is not a bottleneck node, this extra traffic is equivalent to an increment of  $\epsilon(1 - q_{ik})\phi_{ik}$  to the input rate  $r_k$ . Therefore, the contribution of the increment of  $r_i$  to the power consumption of node  $l$  can be expressed via  $r_k$

as  $\epsilon(1 - q_{ik})\phi_{ik}\partial\tilde{w}_l/\partial r_k$ . Since this analysis is applicable for all next-hop neighbors, summing up over all  $k \in S_i$  gives

$$\frac{\partial\tilde{w}_l}{\partial r_i} = \sum_{k \in S_i} (1 - q_{ik})\phi_{ik} \frac{\partial\tilde{w}_l}{\partial r_k} \quad (11)$$

Now suppose that the traffic  $\lambda_i$  is fixed, an increment  $\epsilon$  to the routing variable  $\phi_{ik}$  will cause an increment  $\epsilon(1 - q_{ik})\lambda_i$  to node  $k$ , which is equivalent to an increment of  $\epsilon\lambda_i(1 - q_{ik})$  to input rate  $r_k$ . Applying the similar analysis as above, we find

$$\frac{\partial\tilde{w}_l}{\partial\phi_{ik}} = \lambda_i(1 - q_{ik}) \frac{\partial\tilde{w}_l}{\partial r_k} \quad (12)$$

For example, in Fig. 1(a), node 1 is the source node and node 4 is the bottleneck node. An increment  $\epsilon$  of the input rate of node 1 leads to an increment  $\epsilon\phi_{12}$  to the traffic of node 2, which is equivalent to an increment of  $\epsilon(1 - q_{12})\phi_{12}$  to the input rate of node 2. Similar analysis is applicable for node 3. So the overall increment of power consumption of node 4 due to the increment of  $r_1$  is given by  $\epsilon\partial\tilde{w}_4/\partial r_1 = \epsilon[(1 - q_{12})\phi_{12}\partial\tilde{w}_4/\partial r_2 + (1 - q_{13})\phi_{13}\partial\tilde{w}_4/\partial r_3]$ . Canceling out  $\epsilon$  gives the result as (11). Similarly, an increment  $\epsilon$  to the routing variable  $\phi_{12}$  gives rise to an equivalent increment of  $\epsilon\lambda_1(1 - q_{12})$  to  $r_2$ , so the corresponding increment of power consumption of node 4 is expressed by  $\partial\tilde{w}_4/\partial\phi_{12} = \lambda_1(1 - q_{12})\partial\tilde{w}_4/\partial r_2$ .

(b) **Adjacent source node**

If the source node  $i$  is adjacent to the bottleneck node  $l$ , in this case, the increment of power consumption of node  $l$  due to the increment of the input rate  $r_i$  consists of two parts. One is for receiving the increased traffic  $\epsilon\phi_{il}$ , which is given by  $\epsilon\phi_{il}(e_r/E_l)$ . The other is for transmitting the traffic  $\epsilon(1 - q_{il})\phi_{il}$ , which is given by  $\epsilon(1 - q_{il})\partial\tilde{w}_l/\partial r_l$  following the similar analysis as above. Taking into account the indirect increment from other neighbor  $k \neq l$ , which can derived as above, we obtain

$$\begin{aligned} \frac{\partial\tilde{w}_l}{\partial r_i} &= \sum_{k \in S_i, k \neq l} (1 - q_{ik})\phi_{ik} \frac{\partial\tilde{w}_l}{\partial r_k} + \phi_{il} \left( \frac{e_r}{E_l} + (1 - q_{il}) \frac{\partial\tilde{w}_l}{\partial r_l} \right) \\ &= \sum_{k \in S_i} (1 - q_{ik})\phi_{ik} \frac{\partial\tilde{w}_l}{\partial r_k} + \frac{\phi_{il}e_r}{E_l} \end{aligned} \quad (13)$$

Similarly, an increment  $\epsilon$  to  $\phi_{ik}$  leads to an increment of  $\epsilon\lambda_i$  to node  $k$ , therefore

$$\frac{\partial\tilde{w}_l}{\partial\phi_{ik}} = \lambda_i \left( \frac{e_r}{E_l} + (1 - q_{il}) \frac{\partial\tilde{w}_l}{\partial r_l} \right) \quad (14)$$

An example is illustrated in Fig. 1(b) where node 2 is the source node and node 4 is the bottleneck node. The increment  $\epsilon$  of the input rate  $r_2$  leads to an increment of  $\epsilon\phi_{24}$  to node 4. The increment of the power consumption of node 4 is therefore given by  $\epsilon\phi_{24}e_r/E_4$  plus  $\epsilon(1 - q_{24})\phi_{24}\partial\tilde{w}_4/\partial r_4$ . Taking into account the increment from other downstream links gives rise to the result of (13). Similarly, the increment of power consumption of node 4 due to the increment  $\epsilon$  of the routing variable  $\phi_{24}$  is given by  $\partial\tilde{w}_4/\partial\phi_{24} = \lambda_2(e_r/E_4 + (1 - q_{il})\partial\tilde{w}_4/\partial r_2)$ .

(c) **Co-located source node**

If the source node  $i$  is also a bottleneck node, note that  $r_i$  is a dummy variable, so we do not consider the power consumption for generating traffic  $\epsilon$ . Taking the derivative directly from (5) we have

$$\frac{\partial\tilde{w}_i}{\partial r_i} = \sum_{k \in S_i} \frac{e_{ik}\phi_{ik}}{E_i} \quad (15)$$

$$\frac{\partial\tilde{w}_i}{\partial\phi_{ik}} = \frac{\lambda_i e_{ik}}{E_i} \quad (16)$$

The corresponding example is illustrated in Fig. 1(c) where source node 4 is also bottleneck node. The analysis is simple and we will not elaborate here.

Another possible case is for  $i \in N_C$  and  $l \in N_B$ . However, this case is not necessary to discuss because both  $\partial\tilde{w}_l/\partial r_i$  and  $\partial\tilde{w}_l/\partial\phi_{ik}$  are zeros as  $\tilde{w}_l$  has no relation with  $r_i$ .

We can now combine the above results to derive  $\partial U/\partial\phi_{ik}$  of (10) by considering the following four cases:

- If  $i, k \in N_A$ , then none of the bottleneck nodes are adjacent to node  $i$ , so we can obtain  $\partial w_l/\partial\phi_{ik}$  from (12) for all  $l \in N_B$ . Substituting these into (10) we have

$$\frac{\partial U}{\partial\phi_{ik}} = \frac{2\lambda_i(1-q_{ik})}{|N_B|} \sum_{l \in N_B} [c\tilde{w}_l - (c-1)\mu] \frac{\partial\tilde{w}_l}{\partial r_k} \quad (17)$$

- If  $i \in N_A, k \in N_B$ , then node  $k$  is a bottleneck node adjacent to node  $i$ . Therefore,  $\partial w_k/\partial\phi_{ik}$  is given by (14), while for other bottleneck nodes  $l \neq k$ ,  $\partial w_l/\partial\phi_{ik}$  is given by (12). Substituting these into (10), we have

$$\frac{\partial U}{\partial\phi_{ik}} = \frac{2\lambda_i}{|N_B|} \left( (1-q_{ik}) \sum_{l \in N_B} [c\tilde{w}_l - (c-1)\mu] \frac{\partial\tilde{w}_l}{\partial r_k} + \frac{e_r}{E_k} [c\tilde{w}_k - (c-1)\mu] \right) \quad (18)$$

- If  $i, k \in N_B$ , then node  $i$  and  $k$  are adjacent bottleneck nodes, so  $\partial w_i/\partial\phi_{ik}$  and  $\partial w_k/\partial\phi_{ik}$  are given by (16) and (14) respectively. Therefore

$$\frac{\partial U}{\partial\phi_{ik}} = \frac{2\lambda_i}{|N_B|} \left( (1-q_{ik}) \sum_{l \in N_B} [c\tilde{w}_l - (c-1)\mu] \frac{\partial\tilde{w}_l}{\partial r_k} + \frac{e_{ik}}{E_i} [c\tilde{w}_i - (c-1)\mu] + \frac{e_r}{E_k} [c\tilde{w}_k - (c-1)\mu] \right) \quad (19)$$

- If  $i \in N_B, k \in N_C$ , the source node  $i$  is also a bottleneck node, so  $\partial w_i/\partial\phi_{ik}$  is given by (16), therefore

$$\frac{\partial U}{\partial\phi_{ik}} = \frac{2\lambda_i}{|N_B|} \left( (1-q_{ik}) \sum_{l \in N_B} [c\tilde{w}_l - (c-1)\mu] \frac{\partial\tilde{w}_l}{\partial r_k} + \frac{e_{ik}}{E_i} [c\tilde{w}_i - (c-1)\mu] \right) \quad (20)$$

Now all that are required is to find a stationary point for the routing variable  $\phi$  to minimize  $U$ . We summarize it as the necessary condition in the following theorem.

**Theorem 1.** (Necessary Condition) *Let  $\partial U/\partial\phi_{ik}$  given by (17)-(20), the necessary condition for a minimum of  $U$  with respect to  $\phi^*$  for all  $i \in N_A \cup N_B, k \in S_i$  is*

$$\frac{\partial U}{\partial\phi_{ik}^*} = \begin{cases} = \nu_i, \phi_{ik}^* > 0; \\ \geq \nu_i, \phi_{ik}^* = 0. \end{cases} \quad (21)$$

*Proof.* Let us define the following Lagrange function

$$U(\phi, \nu, \mu) = U + \sum_{i \in N} \nu_i \left( 1 - \sum_{k \in S_i} \phi_{ik} \right) - \sum_{i \in N, k \in S_i} \mu_{ik} \phi_{ik} \quad (22)$$

Where  $\nu = (\nu_1, \dots, \nu_N)$  and  $\mu = \{\mu_{ik}\}$  are the Lagrange multipliers. According to Kuhn-Tucker theorem, the necessary condition for a  $\phi^*$  to be a local minimum for  $U(\phi, \nu, \mu)$  is that there exist Lagrange multipliers  $\nu_i^*, i \in N$  and  $\mu_{ik}^*, i \in N, k \in S_i$  such that

$$\frac{\partial U}{\partial\phi_{ik}^*} - \nu_i^* - \mu_{ik}^* = 0 \quad (23)$$

$$\mu_{ik}^* = 0, \text{ if } \phi_{ik}^* > 0, \forall i, k; \quad (24)$$

$$\mu_{ik}^* > 0, \text{ if } \phi_{ik}^* = 0, \forall i, k. \quad (25)$$

Rearranging (23) to  $\partial U/\partial\phi_{ik}^* = \nu_i^* + \mu_{ik}^*$ , and taking into accounts of (24) and (25) will complete the proof of (21).

The necessary condition (21) states that all links  $(i, k)$  for which  $\phi_{ik} > 0$  must have the same value of  $\partial U / \partial \phi_{ik}$ , and this value must be less than or equal to the value of  $\partial U / \partial \phi_{ik}$  for the links on which  $\phi_{ik} = 0$ . However, as illustrated in [21], the condition (21) is not sufficient to minimize  $U$  because it is automatically satisfied if the traffic rate  $\lambda_i$  is zero, even though the routing can still be improved. To overcome this problem, we prove next that after removing the factor  $\lambda_i$  from (17)-(20), the sufficient condition to minimize  $U$  with respect to  $\phi$  for all  $i \in N_A \cup N_B, k \in S_i$  is given by the following theorem.

**Theorem 2.** (Sufficient Condition) *Let  $\partial U / \partial \phi_{ik}$  given by (17)-(20), and define  $\partial U / \partial r_k = \sum_{l \in N_B} [c\tilde{w}_l - (c-1)\mu] \partial \tilde{w}_l / \partial r_k$ , it is sufficient for a  $\phi^*$  to be a minimizer of  $U$  if for all  $i \in N_A \cup N_B, k \in S_i$ , there is*

$$(1 - q_{ik}) \frac{\partial U}{\partial r_k} \geq \frac{\partial U}{\partial r_i} \quad (26a)$$

$$(1 - q_{ik}) \frac{\partial U}{\partial r_k} + \frac{e_r}{E_k} [c\tilde{w}_k - (c-1)\mu] \geq \frac{\partial U}{\partial r_i} \quad (26b)$$

$$(1 - q_{ik}) \frac{\partial U}{\partial r_k} + \frac{e_{ik}}{E_i} [c\tilde{w}_i - (c-1)\mu] + \frac{e_r}{E_k} (c\tilde{w}_k - (c-1)\mu) \geq \frac{\partial U}{\partial r_i} \quad (26c)$$

$$(1 - q_{ik}) \frac{\partial U}{\partial r_k} + \frac{e_{ik}}{E_i} [c\tilde{w}_i - (c-1)\mu] \geq \frac{\partial U}{\partial r_i} \quad (26d)$$

where (26a)-(26d) correspond to the four cases given by (17)-(20) respectively.

*Proof.* Suppose that there is a set of routing variables  $\phi^*$  satisfying (26), the corresponding node flows are  $\lambda^*$  and link flows are  $f^*$ , where  $f_{ik} = \lambda_i \phi_{ik}, i \in N, k \in S_i$ . Let  $\phi$  be any other set of routing variables with the corresponding node flows  $\lambda$  and link flows  $f$ . Define  $f(\theta)$  as the convex combination of  $f^*$  and  $f$  with respect to a variable  $\theta$ , that is,

$$f_{ik}(\theta) = (1 - \theta)f_{ik}^* + \theta f_{ik} \quad (27)$$

Therefore, each  $\tilde{w}_l, l \in N_B$  can be represented by the link flow  $f$ , which in turn is a function of  $\theta$ , so  $U$  is also a function of  $\theta$ . We rewrite the smoothing function (9) as

$$U(\theta) = \frac{c}{|N_B|} \sum_{l \in N_B} \tilde{w}_l^2(\theta) - \frac{(c-1)}{|N_B|^2} \left( \sum_{l \in N_B} \tilde{w}_l(\theta) \right)^2 \quad (28)$$

Since each  $w_l(\theta)$  is a convex function of the node flow  $f$ , therefore  $U(\theta)$  is also a convex function with respect to  $\theta$ , so it is obvious

$$\left. \frac{dU(\theta)}{d\theta} \right|_{\theta=0} \leq U(\phi) - U(\phi^*) \quad (29)$$

Since  $\phi$  is an arbitrary set of routing variable, it will complete the proof by proving that  $dU(\theta)/d\theta \geq 0$  at  $\theta = 0$ .

From (5) and (27), it is straightforward to express  $\tilde{w}_l$  as a function of the link flow  $f(\theta)$  as

$$\tilde{w}_l(\theta) = \frac{1}{E_l} \left( e_s + e_g r_l + \sum_{i \in A_l} f_{il}(\theta) e_r + \sum_{k \in S_l} f_{lk}(\theta) e_{lk} \right) \quad (30)$$

Differentiating  $\tilde{w}_l$  directly from (27) and (30), we get

$$\frac{\partial \tilde{w}_l}{\partial \theta} = \sum_{i \in A_l} \frac{e_r}{E_l} (f_{il} - f_{il}^*) + \sum_{k \in S_l} \frac{e_{lk}}{E_l} (f_{lk} - f_{lk}^*) \quad (31)$$

We can calculate  $dU(\theta)/d\theta$  directly using (28) and (31)

$$\begin{aligned}
\left. \frac{dU(\theta)}{d\theta} \right|_{\theta=0} &= \frac{2c}{|N_B|} \sum_{l \in N_B} \tilde{w}_l \frac{\partial \tilde{w}_l}{\partial \theta} - \frac{2(c-1)}{|N_B|^2} \sum_{l \in N_B} \tilde{w}_l \sum_{l \in N_B} \frac{\partial \tilde{w}_l}{\partial \theta} \\
&= \frac{2}{|N_B|} \sum_{l \in N_B} \left[ c\tilde{w}_l - \frac{(c-1)}{|N_B|} \sum_{l \in N_B} \tilde{w}_l \right] \cdot \frac{\partial \tilde{w}_l}{\partial \theta} \\
&= \frac{2}{|N_B|} \sum_{l \in N_B} [c\tilde{w}_l - (c-1)\mu] \cdot \left( \sum_{i \in A_l} \frac{e_r}{E_l} (f_{il} - f_{il}^*) + \sum_{k \in S_l} \frac{e_{lk}}{E_l} (f_{lk} - f_{lk}^*) \right)
\end{aligned}$$

We then first prove that

$$\sum_{l \in N_B} [c\tilde{w}_l - (c-1)\mu] \cdot \left( \sum_{i \in A_l} \frac{e_r f_{il}}{E_l} + \sum_{k \in S_l} \frac{e_{lk} f_{lk}}{E_l} \right) \geq \sum_{i \in N_A \cup N_B} r_i \frac{\partial U}{\partial r_i} \quad (32)$$

Note that from (26a)-(26d), multiplying both sides of these equations with  $\lambda_i$  and  $\phi_{ik}$ , summing over all  $i \in N_A \cup N_B$  and  $k \in S_i$ , and using the fact that  $\lambda_i = r_i + \sum_{j \in A_i} \lambda_j \phi_{ji} (1 - q_{ji})$ , we can obtain the result for the left-hand side as

$$\begin{aligned}
\text{LHS} &= \sum_{i \in N_A \cup N_B} \sum_{k \in S_i} \lambda_i \phi_{ik} (1 - q_{ik}) \frac{\partial U}{\partial r_k} \\
&+ \sum_{i \in N_A} \sum_{k \in S_i, k \in N_B} \left( \frac{\lambda_i \phi_{ik} e_r}{E_k} [c\tilde{w}_k - (c-1)\mu] \right) \\
&+ \sum_{i \in N_B} \sum_{k \in S_i, k \in N_B} \left( \frac{\lambda_i \phi_{ik} e_{ik}}{E_i} [c\tilde{w}_i - (c-1)\mu] + \frac{\lambda_i \phi_{ik} e_r}{E_k} [c\tilde{w}_k - (c-1)\mu] \right) \\
&+ \sum_{i \in N_B} \sum_{k \in S_i, k \in N_C} \left( \frac{\lambda_i \phi_{ik} e_{ik}}{E_i} [c\tilde{w}_i - (c-1)\mu] \right)
\end{aligned} \quad (33)$$

and the right-hand side as

$$\text{RHS} = \sum_{i \in N_A \cup N_B} r_i \frac{\partial U}{\partial r_i} + \sum_{i \in N_A \cup N_B} \sum_{j \in A_i} \lambda_j \phi_{ji} (1 - q_{ji}) \frac{\partial U}{\partial r_i} \quad (34)$$

Now let look at the first term of LHS in (33), which sums over all links directed from nodes  $i \in N_A \cup N_B$ . Similarly, the second term of RHS in (34) sums over all in links directed to nodes  $i \in N_A \cup N_B$ . Recalling that the network is directed acyclic, canceling the common part of these two terms, the remaining part of the first term of (33) is the sum over all links  $(i, k)$ ,  $i \in N_B, k \in N_C$ , which is zero because  $\partial \tilde{w}_i / \partial r_k$  are zero for these links. In other words, we can totally cancel out the first term of (33) and the second term of (34).

Re-arranging the summation of the second, third and the fourth terms of lefthand side in (33), and recalling the inequality between (33) and (34), we obtain

$$\sum_{l \in N_B} [c\tilde{w}_l - (c-1)\mu] \left( \sum_{i \in A_l} \frac{e_r}{E_l} \lambda_i \phi_{il} + \sum_{k \in S_l} \frac{e_{lk}}{E_l} \lambda_l \phi_{lk} \right) \geq \sum_{i \in N_A \cup N_A} r_i \frac{\partial U}{\partial r_i} \quad (35)$$

Note that  $f_{il} = \lambda_i \phi_{il}$ , substituting this into (35) we can obtain (32).

Following the same derivation procedure, if  $\lambda^*$  and  $\phi^*$  are substituted for  $\lambda$  and  $\phi$ , this becomes an equality from the equations for  $\partial U / \partial r_i$  in (26). That is,

$$\sum_{l \in N_B} [c\tilde{w}_l - (c-1)\mu] \left( \sum_{i \in A_l} \frac{e_r f_{il}^*}{E_l} + \sum_{k \in S_l} \frac{e_{lk} f_{lk}^*}{E_l} \right) = \sum_{i \in N_A \cup N_A} r_i \frac{\partial U}{\partial r_i} \quad (36)$$

Substituting (32) and (36) into (32), we see that  $dW(\theta)/d\theta \geq 0$  at  $\theta = 0$ , which complete the proof.

### 3.3 Algorithm

Let us define two indicator functions  $I_i$  and  $I_k$ , where  $I_i$  is 1 if  $i \in N_B$  and 0 otherwise, and  $I_k$  is 1 if  $k \in N_B$  and 0 otherwise. Let  $Z_{ik} = I_i e_{ik} [c\bar{w}_i - (c-1)\mu] / E_i + I_k e_r [c\bar{w}_k - (c-1)\mu] / E_k$ , then the sufficient condition stated in (26) can be simplified as

$$(1 - q_{ik}) \frac{\partial U}{\partial r_k} + Z_{ik} \geq \frac{\partial U}{\partial r_i} \quad (37)$$

for all  $i \in N_A \cup N_B, k \in S_i$ , where equality is achieved for  $k$  whose routing variable  $\phi_{ik}$  is greater than 0. That is, when the optimality is achieved, only those links with the smallest  $(1 - q_{ik})\partial U/\partial r_k + Z_{ik}$  have nonzero traffic.

Based on the sufficient conditions, we design a gradient descent algorithm for each node to locally update its routing variables according to the received information from downstream neighbors. Instead of presenting the whole algorithm, we just present the routing variable update procedure here and refer the readers to standard textbooks such as [22] [23] for implantation details. Firstly,  $(1 - q_{ik})\partial U/\partial r_k + Z_{ik}$  is computed for every neighbor  $k \in S_i$ . The best neighbor  $k_{min}$  with the smallest  $(1 - q_{ik})\partial U/\partial r_k + Z_{ik}$  will have its routing variable increased while that of other neighbors' will be decreased accordingly. The next step is to compute the amount of reduction  $\Delta_{ik}$  to each  $\phi_{ik} (k \neq k_{min})$ . Let  $a_{ik}$  be the gradient difference between each neighbor  $k$  and neighbor  $k_{min}$ , that is

$$a_{ik} = (1 - q_{ik}) \frac{\partial U}{\partial r_k} + Z_{ik} - \min_{k \in S_i} \left\{ (1 - q_{ik}) \frac{\partial U}{\partial r_k} + Z_{ik} \right\}, \quad k \in S_i \quad (38)$$

Then the amount of traffic reduction  $\Delta_{ik}$  is proportional to  $a_{ik}$  with the constraint that the routing variable  $\phi_{ik}$  cannot be negative. That is, for each  $k \in S_i, k \neq k_{min}$ ,

$$\Delta_{ik} = \min \left\{ \phi_{ik}, \frac{\eta \phi_{ik} a_{ik}}{\max_{k \in S_i} a_{ik}} \right\} \quad (39)$$

and

$$\phi_{ik} \leftarrow \phi_{ik} - \Delta_{ik} \quad (40)$$

where  $\eta$  is a positive scalar. Finally, the total amounts of reduction are added to  $\phi_{ik_{min}}$  as

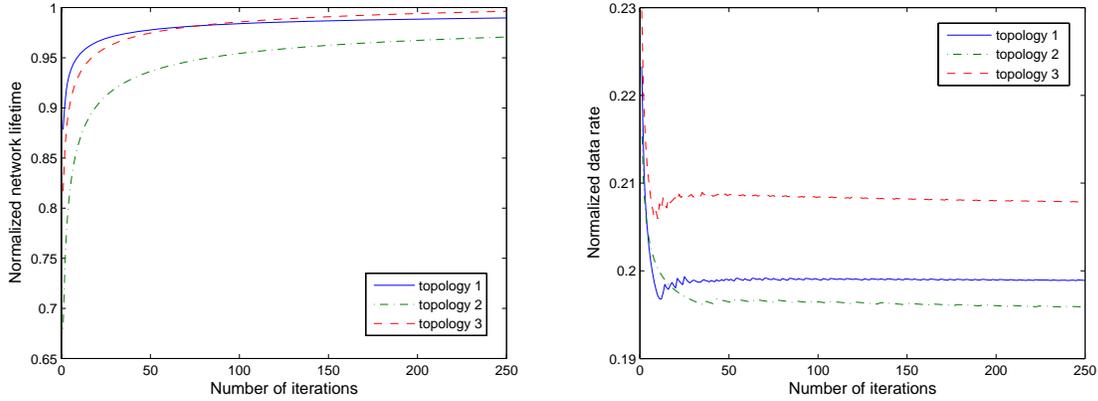
$$\phi_{ik_{min}} \leftarrow \phi_{ik_{min}} + \sum_{k \in S_i, k \neq k_{min}} \Delta_{ik} \quad (41)$$

Using this algorithm, each node  $i$  gradually decreases the routing variables for which the value  $(1 - q_{ik})\partial U/\partial r_k + Z_{ik}$  is large, and increases those for which it is small until the sufficient condition (37) is satisfied.

## 4 Performance Evaluation

We simulate the MLR algorithm over a set of sensor networks with the number of nodes varying from 50 to 100. Each network has its nodes randomly distributed over a square of 100 units by 100 units. All the nodes are assumed to have equal initial battery energy and equal traffic generating rate. For data correlation settings, we adopt the *gaussian random field* model [11] where the correlation coefficient  $q_{ik}$  decreases exponentially with the distance between nodes, or  $q_{ik} = \exp(-\alpha d_{ik}^2)$ . Here  $\alpha$  is the correlation exponent and varies from  $\alpha = 0.001$  (high correlation) to  $\alpha = 0.01$  (low correlation) in the simulations. Also, a decreasing sequence of step size  $\eta$  and an increasing sequence of  $c$  are used in the simulations.

Fig. 2 shows three traces of the network lifetime for three network topologies with 50 nodes. The network lifetime is computed at each iteration and normalized with respect to the optimal value obtained by the centralized solution to the MLR problem. We can see that the distributed algorithm can converge efficiently. For the same three sets of experiments, Fig. 3 shows the aggregated data



**Fig. 2.** Normalized network lifetime as a function of the number of iteration ( $N = 50, \alpha = 0.005$ ) **Fig. 3.** Normalized data rate at sink node as a function of the number of iterations ( $N = 50, \alpha = 0.005$ )

rate at the sink node normalized to the total raw data rate of all source nodes. We observe that the traffic rate converges to a stable value in about 25 iterations, but from Fig. 2, we see that the network lifetime continue to increase after that. This is clearly due to the route optimization of the algorithm.

## 5 Conclusion

In this paper we have exploited the data correlation and optima routing to maximize the lifetime of a sensor network with a single sink node. We have proposed a smoothing function to overcome the nondifferentiability of the max function so that a distributed solution is possible. The optimality conditions are derived and a gradient decent algorithm is developed for every node to locally compute the routing variables. Simulation results show that the algorithm can converge to the optimal value efficiently and is scalable to the network size. Extension of our work for multiple sink nodes and for nodes with sleeping mode would be of interest, but these are beyond the scope of this paper.

## Acknowledgment

This work is supported in part by the Hong Kong Research Grants Council under Grant CUHK 4220/03E.

## References

1. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. International Conference on System Sciences*, 2000.
2. S. Singh, Mike Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *MobiCom'98*, pages 181–190, 1998.
3. Teresa H. Meng V. Rodoplu. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, pages 1333–1344, August 1999.
4. J.-H Chang and L. Tassiulas. Routing for maximum system lifetime in wireless ad-hoc networks. In *Proc. of Allerton Conference on Communication, Control and Computing*, Sep. 1999.
5. J.-H Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *Infocom'00*, pages 22–31, 2000.
6. A. Sankar and Z. Liu. Maximum lifetime routing in wireless ad-hoc networks. In *Infocom'04*, March 2004.

7. R. Madan and S. Lall. Distributed algorithms for maximum lifetime routing in wireless sensor networks. In *Global Telecommunications Conference(GLOBECOM '04)*, IEEE, volume 2, Nov 2004.
8. J. Pan, Y. Thomas Hou, L. Cai, Y. Shi, and Sherman X. Shen. Topology control for wireless sensor networks. In *MobiCom'03*, pages 286–299, 2003.
9. K. Kalpakis, K. Dasgupta, and P. Namjoshi. Maximum lifetime data gathering and aggregation in wireless sensor networks. In *Proc of ICN'02*, Aug. 2002.
10. Sundeep Patten, Bhaskar Krishnamachari, and Ramesh. The impact of spatial correlation on routing with compression in wireless sensor networks. In *IPSN'04*, pages 28–35. ACM Press, 2004. Berkeley, California, USA.
11. R. Cristescu, B. Beferull-Lozano, and M. Vetterli. On network correlated data gathering. In *Infocom'04*, Hong Kong, 2004.
12. M. Gastpar and M. Vetterli. Source-channel communication in sensor networks. In *IPSN'03*, 2003.
13. Anna Scaglione and Sergio D. Servetto. On the interdependence of routing and data compression in multi-hop sensor networks. In *MobiCom '02*, pages 140–147. ACM Press, 2002. Atlanta, Georgia, USA.
14. Mehmet C. Vuran, Ozgur B. Akan, and Ian F. Akyildiz. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Computer Networks*, 45(3):245, 2004.
15. F. Kuhn, R. Wattenhofer, and A. Zollinger. Asymptotically optimal geometric mobile ad-hoc routing. In *Proc. of DIALM'99*, pages 24–33, September 2002.
16. M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *IEEE Network Magazine*, 15(6):30–39, November 2001.
17. A. Ben-Tal and M. Teboulle. A smoothing technique for nondifferentiable optimization problems. In *Proceedings of the international seminar on Optimization*, pages 1–11, New York, NY, USA, 1988. Springer-Verlag New York, Inc.
18. X. Li. An entropy-based aggregate method for minimax optimization. *Engineering Optimization*, 18:277–285, 1992.
19. Chunhui Chen and O. L. Mangasarian. Smoothing methods for convex inequalities and linear complementarity problems. *Math. Program.*, 71(1):51–69, 1995.
20. L. Qi and D. Sun. Smoothing functions and a smoothing newton method for complementarity and variational inequality problems. *Journal of Optimization Theory and Applications*, 113:121–147, 2002.
21. Robert G. Gallager. A minimum delay routing algorithm using distributed computation. *IEEE Transaction on Communications*, 25(1):73–85, Jan. 1977.
22. Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.
23. S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2003.