

Research Article

Bicriteria Optimization in Wireless Sensor Networks: Link Scheduling and Energy Consumption

Jian Chen,^{1,2} Jie Jia,² Enliang Dai,² Yingyou Wen,^{1,2} and Dazhe Zhao^{1,2}

¹Key Laboratory of Medical Image Computing of Ministry of Education, Northeastern University, Shenyang 110819, China

²School of Information Science & Engineering, Northeastern University, Shenyang 110819, China

Correspondence should be addressed to Jie Jia; jiajie@ise.neu.edu.cn

Received 25 September 2014; Revised 22 January 2015; Accepted 20 February 2015

Academic Editor: Anand Paul

Copyright © 2015 Jian Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Link scheduling is important for reliable data communication in wireless sensor networks. Previous works mainly focus on how to find the minimum scheduling length but ignore the impact of energy consumption. In this paper, we integrate them together and solve them by multiobjective genetic algorithms. As a contribution, by jointly modeling the route selection and interference-free link scheduling problem, we give a systematical analysis on the relationship between link scheduling and energy consumption. Considering the specific many-to-one communication nature of WSNs, we propose a novel link scheduling scheme based on NSGA-II (Non-dominated Sorting Genetic Algorithm II). Our approach aims to search the optimal routing tree which satisfies the minimum scheduling length and energy consumption for wireless sensor networks. To achieve this goal, the solution representation based on the routing tree, the genetic operations including tree based recombination and mutation, and the fitness evaluation based on heuristic link scheduling algorithm are well designed. Extensive simulations demonstrate that our algorithm can quickly converge to the Pareto optimal solution between the two performance metrics.

1. Introduction

A wireless sensor network usually consists of a large number of tiny sensor nodes with sensing, data processing, and communication components. In general, it is formed by these sensor nodes in a self-organized manner [1]. Recently, wireless sensor network is playing an increasingly important role in many application fields, such as medicine, outer space exploration, volcano surveillance, intelligence applications, intrusion detection, and border surveillance [2, 3].

MAC access is one of the key issues in designing wireless sensor networks [4]. Many performance metrics, such as network lifetime, the accuracy of information perception, the delay of transmission, and location accuracy, are all closely related to the MAC technology [5]. Generally speaking, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and Time Division Multiple Access (TDMA) are two widely used MAC access technologies for wireless networks [6]. As to the former one, due to its competitive channel access mode, the access fairness cannot be supported.

In addition, some strategies applied in the technology, such as idle channel listening and communication conflict avoidance, consume additional energy, which cannot satisfy the demands of large-scale applications. In contrast, in TDMA based MAC access technology, by allocating time slot for each wireless terminal, we can eliminate collisions, guarantee link bandwidth with scheduled access, and support a more reliable data transmission for WSNs. In addition, sensor nodes in TDMA protocols can operate in a low duty cycle if their allocated time slots is not come, thus to save energy and extend network lifetime [7]. For example, in a traffic monitoring application, sensor nodes with TDMA technology can achieve a lifetime of 1,200 days compared with ten days using CSMA/CA technology [8].

How to efficiently allocate time slots to satisfy all routing requirements is the main task in designing TDMA schedule for WSNs. Since the number of scheduling periods is limited, some sensor nodes close to each other may be assigned to the same time slot by random slot allocation. This will cause

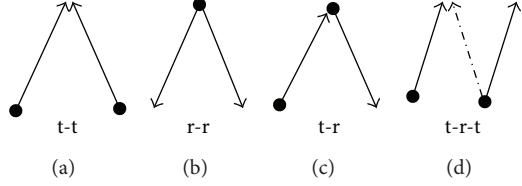


FIGURE 1: Primary interference and secondary interference in TDMA wireless networks.

interference and capacity reduction. Therefore, interference-free link scheduling algorithm should be designed. Besides, since transmission latency depends on the scheduling period, the longer scheduling period, the larger transmission latency. TDMA scheduling algorithm therefore should minimize the number of time slots required. In addition, due to the power-constrained batteries in each sensor node, link scheduling algorithm with energy consumption optimization should also be considered, thus to prolong network survival time.

Recently, link scheduling for wireless networks has received a great attention [9–12]. As wireless conflicts can be modeled as interference (conflict) graphs [9, 10], with the assumption that each node has the same packet arrival rate, some works use graph coloring based heuristic algorithms [11, 12] to find the interference-free link scheduling schemes. However, because of the relay characteristics in WSN, each sensor node has to transmit its own packet as well as to forward its children's; thus the above-mentioned scheduling mechanism with the same slot demands cannot be directly applied to WSN applications. The time complexity in finding an interference-free link scheduling solution for wireless multihop networks is analyzed in [13], and a graph coloring solution with polynomial time complexity is proposed. In order to reduce the message load caused by acquiring the entire network topology, literatures [14, 15] propose a distributed scheduling mechanism based on local topology. However, these scheduling algorithms both are based on the assumption that there are some independent point-to-point routing sessions among the network. In [16], fast slot allocation for linear duty-cycled wireless sensor networks is researched, and centralized optimal as well as distributed scheduling algorithms are proposed. In WSNs, however, data are often transferred from some sensor nodes to one sink node; therefore, the specific many-to-one communication nature is not considered. For this reason, scheduling and routing mechanisms are designed together in some literatures. In [17], the authors used AODV routing protocol to generate multiple paths satisfying the demands of all routing sessions, and the scheduling algorithm is executed only on the best route. Similarly, the literature [18] first used the existed routing algorithm to determine the traffic demand of each node, and then the genetic algorithm and particle swarm algorithm are hybridized to find the optimal slot allocation. However, the route optimization is not considered in their approach. In literature [19], the tree based link scheduling algorithm combined with power control is proposed, which can improve the communication reliability as well as reduce data gathering latency. In literature [20], the authors have

established a cross layer optimization model including physical layer, MAC layer, and the network layer problems, and a convex optimization method is proposed. In [21], link scheduling for a converge-cast routing tree is researched, and a slot allocation algorithm joint with channel assignment is proposed. However, these algorithms discussed above did not take energy consumption into account, which may easily lead to shorter network survival time than expected.

The original contributions of this paper are shown as follows: First, we formulate the joint optimization of sensor network scheduling problem including both route selection and energy consumption as a multiobjective optimization problem. Second, by coding the specific many-to-one routing tree as an individual for evolution, we propose a novel link scheduling scheme based on NSGA-II (Non-dominated Sorting Genetic Algorithm II). Third, with the given routing tree and link slot requirement, we propose a heuristic slot allocation algorithm to evaluate the fitness of each individual. Finally, we conduct extensive simulations to demonstrate the effectiveness of our proposed method. The rest of the paper is organized as follows. Section 2 presents the assumptions, network model, and design considerations. NSGA-II based link scheduling algorithm is proposed in Section 3. Section 4 gives the simulation results, and Section 5 concludes our paper.

2. Network Model and Problem Formulation

2.1. Network Model. Assume a WSN is composed of a single sink node and $N - 1$ sensor nodes which periodically generate data with different rate. The sink node and sensor nodes are labeled as $i = 1, \dots, N$. Without loss of generality, each node in the network has a fixed communication radius R_c . The entire network can be represented as a directed topology graph $G(V, E)$, where $V = \{i \mid 1 \leq i \leq N\}$ is the set of nodes, $E = \{(i, j) \mid 1 \leq i, j \leq N, d_{ij} \leq R_c\}$ is the directed edge set, and d_{ij} represents the distance between nodes i and j . Due to the broadcast transmission nature of the wireless radio, a pair of nodes t within interference range may interfere with each other's communication, even if they cannot directly communicate. To schedule two links at the same time slot, we must ensure that the schedule will avoid the interference. Two different types of interference have been imported in the literature, namely, primary interference and secondary interference [22], as shown in Figure 1.

Primary interference happens when a node transmits and receives packets at the same time, as shown in Figures 1(a), 1(b), and 1(c). The transmitter-transmitter ($t-t$) interference is caused by two parallel transmissions that garble

each other at the same receiver, while a single transmitter cannot separate packets intended for two different receivers, which will result in the receiver-receiver (r-r) conflict; the transmitter-receiver (t-r) conflict occurs because the nodes cannot transmit and receive at the same time. Secondary interference, such as transmitter-receiver-transmitter (t-r-t) interference model, occurs when a node receives two or more separate transmissions, but only one transmission is intended for this node, as shown in Figure 1(d). Here, we should note that another secondary interference model, the receiver-transmitter-receiver (r-t-r) interference model, can cause unnecessary delay of transmission in RTS/CTS handshake in 802.11 protocols. However, in TDMA based MAC protocols, by assigning different time slot for each link, the r-t-r conflict does not exist.

We employ a $T \times L$ matrix \mathbf{Y} to represent link schedule for all the links in E , where T is the total number of slots and L is the total number of links in E . For each element $y_{i,j}^t$ in matrix \mathbf{Y} , we have

$$y_{i,j}^t = \begin{cases} 1, & \text{if } e_{ij} \text{ is active in time slot } t \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Let I_e denote the sets of links which will cause interference with e . In order to achieve conflict-free scheduling, these mutual interference links should be assigned to different time slots, and we have

$$\sum_{e_{ij} \in I_e} y_{i,j}^t \leq 1. \quad (2)$$

Define $\langle s_q, r_q \rangle$ as the routing sessions, where s_q and r_q represent the source sensor node and slot demand of session $q \in [1, 2, \dots, Q]$ ($Q \leq V$), respectively. Define $f_{i,j}^{q,t}$ (and $f_{j,i}^{q,t}$) as the traffic flow traveling from i to j (and from j to i) for session q on slot t , and we have

$$f_{i,j}^{q,t} = \begin{cases} 1, & \text{if } q \text{ is transmitting on link } e_{ij} \text{ in slot } t \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

When $f_{i,j}^{q,t} = 1$, $y_{i,j}^t = 1$, thus we get

$$f_{i,j}^{q,t} \leq y_{i,j}^t. \quad (4)$$

In order to achieve conflict-free scheduling, we have

$$\sum_{e_{ij} \in I(e)} f_{i,j}^{q,t} \leq 1. \quad (5)$$

Let $\langle s_q, r_q \rangle$ denote the end-to-end routing session $q \in [1, 2, \dots, Q]$, to sink node, where s_q , r_q represent the source sensor node and its slot demand, respectively. Define $f_{i,j}^q$ (and $f_{j,i}^q$) as the flow travelling from node i to node j

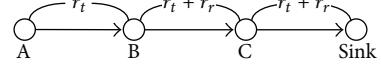


FIGURE 2: The length of path and energy consumption.

(and from node j to node i) for session q , where $i, j \in V$, $i \neq j$. And the routing constraints can be defined as follows:

$$\sum_{t \in T} \sum_{(k,i) \in E} f_{ki}^{q,t} = \sum_{t \in T} \sum_{(i,j) \in E} f_{ij}^{q,t}, \quad (6)$$

$$(i \neq s_q, i \neq \text{sink}, i \in V, q \in Q),$$

$$\sum_{t \in T} \sum_{(i,j) \in E} f_{ij}^{q,t} = r_q, \quad (i = s_q, i \in V, q \in Q), \quad (7)$$

$$\sum_{t \in T} \sum_{(j,i) \in E} f_{ji}^{q,t} = r_q, \quad (i = \text{sink}, i \in V, q \in Q), \quad (8)$$

$$\sum_{t \in T} \sum_{q \in Q} f_{ji}^{q,t} \leq \sum_{t \in T} y_{ij}^t, \quad (9)$$

where (6) states that the amount of incoming flow in all slots is equal to the amount of outgoing flow for each node, except the source and sink node; (7) represents that the outgoing flow in all slots from the source is equal to its traffic demand; (8) states that the incoming flow in all slots to the sink node is equal to its traffic demand; and (9) states that the sum of the flows over all sessions traversing a link cannot exceed the total allocated slots on this link.

On the basis of the scheduling period T , the variable $f_{i,j}^{q,t}$ and the length of routing path of session q , Γ_q can be calculated as

$$\Gamma_q = \sum_{t \in T} \sum_{(i,j) \in E} f_{ij}^{q,t}. \quad (10)$$

Assume that the energy consumption is only dominated by communication costs, as opposed to sensing and processing costs. A sensor consumes e_1 units of energy when sending data in one slot, while it depletes e_2 units of energy when receiving data in one slot. If we ignore the energy consumed in the sink node, therefore, for each routing session q , given its length of routing path Γ_q , its energy consumption can be calculated as

$$E_q = r_t \Gamma_q + r_r (\Gamma_q - 1). \quad (11)$$

Figure 2 shows a simple example of network energy consumption when node A transmits one unit of sensing data to sink node. According to Figure 2, we can get the length of the route path as 3. Based on the energy consumption model defined in (11), the route has to send the data 3 times and receive the data twice; we can get the total energy consumption of this route as $3r_t + 2r_r$.

2.2. Problem Formulation. Now we are ready to define the joint optimization problem: given the topology graph $G(V, E)$ and the number of routing sessions Q , our goal is to find an interference-free link scheduling scheme to satisfy all routing requirements, and the scheduling length and the total energy consumption are minimized. Formally, this problem can be expressed as the following multiobjective problem:

$$\begin{aligned}
& \min \left\{ \sum_{q \in Q}^T E_q, \right. \\
& \quad \left. y_{i,j}^t = \begin{cases} 1, & \text{if } e_{ij} \text{ is active in time slot } t \\ 0, & \text{otherwise,} \end{cases} \right. \\
& \quad \sum_{e_{ij} \in I_e} y_{i,j}^t \leq 1, \\
& f_{ij}^{q,t} = \begin{cases} 1, & \text{if } q \text{ is transmitting on link } e_{ij} \text{ in slot } t \\ 0, & \text{otherwise,} \end{cases} \\
& f_{ij}^{q,t} \leq y_{ij}^t, \quad \sum_{e_{ij} \in I(e)} f_{ij}^{q,t} \leq 1, \\
& \sum_{t \in T} \sum_{(k,i) \in E} f_{ki}^{q,t} = \sum_{t \in T} \sum_{(i,j) \in E} f_{ij}^{q,t}, \\
& \quad (i \neq s_q, i \neq \text{sink}, i \in V, q \in Q), \\
& \sum_{t \in T} \sum_{(i,j) \in E} f_{ij}^{q,t} = r_q, \quad (i = s_q, i \in V, q \in Q), \\
& \sum_{t \in T} \sum_{(j,i) \in E} f_{ji}^{q,t} = r_q, \quad (i = \text{sink}, i \in V, q \in Q), \\
& \sum_{t \in T} \sum_{q \in Q} f_{ji}^{q,t} \leq \sum_{t \in T} y_{ij}^t, \\
& \Gamma_q = \sum_{t \in T} \sum_{(i,j) \in E} f_{ij}^{q,t}, \\
& E_q = r_t \Gamma_q + r_r (\Gamma_q - 1), \tag{12}
\end{aligned}$$

where r_q ($q = 1, 2, \dots, Q$) are input parameters. Note that $y_{i,j}^t$ and $f_{ij}^{q,t}$ are optimization variables to determine network slot allocation and route selection. By convention, the optimization problem is in the form of *bicriteria-objective integer programming* (MOIP) problem, which is NP-hard [13] in general and cannot be solved by CPLEX. In the next section, we will develop a nested optimization framework based on NSGA-II to solve this problem.

3. Multiobjective Optimization Based on NSGA-II

In this section, we first briefly review the fundamentals of multiobjective optimization. Then, we give the description of our approach based on NSGA-II. The mathematical description of multiobjective optimization can be given as

$$\begin{aligned}
& \min_{x \in F} [f_1(x), f_2(x), \dots, f_m(x)]^T, \\
& F : \{x : h(x) = 0, g(x) \leq 0, a \leq x \leq b\},
\end{aligned} \tag{13}$$

where x is a solution to the MOP and the number of objective functions $n \geq 2$. F is the feasible set, which is subjected to the equality and inequality constraints denoted as $h(x)$ and $g(x)$ and explicit variable bounds $[a, b]$. In a multiobjective minimization problem, a solution x_1 dominates another solution x_2 ($x_1 \succcurlyeq x_2$) if and only if

$$\begin{aligned}
& f_u(x_1) \leq f_u(x_2) \quad \exists u \in \{1, 2, \dots, m\}, \\
& f_v(x_1) < f_v(x_2) \quad \exists v \in \{1, 2, \dots, m\}.
\end{aligned} \tag{14}$$

For such a multiobjective optimization problem, a solution x^* is called a Pareto optimal solution if and only if there is no $x \in F$ that dominates x^* . A well-known technique for solving MOPs is the weighting method, which converts the original MOP into a single optimization problem for forming a linear combination of the objectives

$$\begin{aligned}
& \min_{x \in F} w_1 f_1(x) + w_2 f_2(x) + \dots + w_m f_m(x), \\
& F : \{x : h(x) = 0, g(x) \leq 0, a \leq x \leq b\},
\end{aligned} \tag{15}$$

where w_i , $i \in [1, n]$, is the weight for the i th objective f_i . Usually, $0 < w_i < 1$ and $\sum_{i=1}^m w_i = 1$. However, the weighting method in solving MOP suffers from two main disadvantages [23]. First of all, it cannot find Pareto optimal solutions in MOPs with a nonconvex Pareto optimal front, which may reduce the number of design alternatives. As shown in Figure 3, the weighting method can only find the Pareto optimal solutions of point A and point D; however, other three solutions B, C, and E cannot be found. Second, it needs to choose appropriate weights for each optimization function in priori; the decision on preferences may not be clear to the user until the solution is generated.

Recently, multiobjective genetic algorithms (MOGAs) are recognized to be better qualified to tackle multiobjective optimization problems. Compared with weighting method, MOGA can capture multiple Pareto optimal solutions in a single run and exploit similarities of solutions by genetic operation. In addition, MOGA is less susceptible to the shape of continuity of the Pareto optimal front. Therefore, in our paper, we apply the most popular studied MOGA, Non-dominated Sorting Genetic Algorithm II (NSGA-II) [24], to find the Pareto optimal solution.

MOGA algorithms cannot deal with the variables of the optimization problem directly. They work with chromosome-like codes representing the solutions. Thus, the first issue

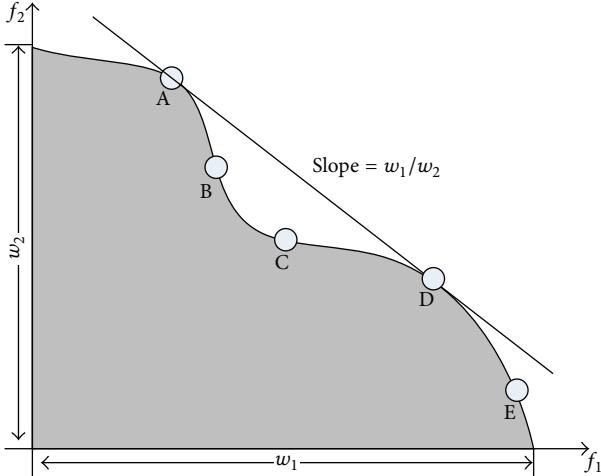


FIGURE 3: Graphical interpretation of the weighting method for solving multiobjective problems.

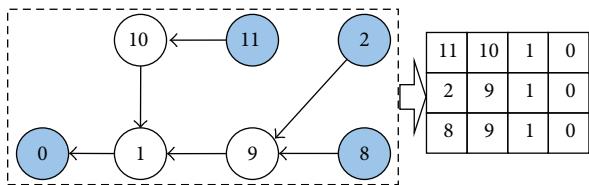


FIGURE 4: Illustration of the array representation of the routing tree.

is how to represent solutions of the problem, that is, how to use a data structure that is appropriate for the problem [25]. From the description of the optimization problem described in (12), we could notice that the difficulty of the optimization problem comes from the determination of the routing tree. Specifically, if the routing tree can be obtained by some techniques, the energy consumption can also be obtained. Therefore, in our paper, we use the routing tree as the problem representation. In this representation, the path of each routing session is coded as a string of positive integers. Since there are Q routing paths and the many-to-one communication nature, these routing paths are further combined to a reverse multicast tree and are represented as an integer array, in which row i lists node IDs routing from s_i ($1 \leq i \leq Q$) to sink node. Figure 4 illustrates an example of 3 routing paths and their chromosome representation, where the sink node is denoted as node 0. In order to explore the genetic diversity, the routing path for any routing session q is randomly generated. For each source node s_q , we first randomly select a node v_1 from $N(s_q)$, the neighborhood of s_q . Then we randomly select a node v_2 from $N(v_1)$. This process is repeated until the sink node is reached. Thus, we get a random path $P(s_q, \text{sink}) = \{s_q, v_1, v_2, \dots, \text{sink}\}$. Since no loop is allowed on the reverse multicast tree, the nodes that are already included in the tree should be excluded, thereby avoiding reentry of the same node.

Generally, MOGA begins with an initial population, which is composed of N randomly generated channel assignment schemes. This initial population will execute selection,

crossover, and mutation operators to generate new individuals in a search space. For each channel assignment scheme in the population, namely, an individual, a fitness value is applied to reflect the goodness degree for solving the problem. During each cycle of genetic evolution, chromosomes with better fitness values will have a higher chance of surviving in the subsequent generations, emulating the survival-of-the-fittest phenomenon in nature. Therefore, the first important aspect in designing MOGA is how to calculate the individual fitness. In our algorithm, the fitness is defined as the minimum scheduling period and total energy consumption. When the routing tree is represented in a chromosome, we can calculate its total length of all routing paths. Therefore, the total energy consumption of a chromosome can be directly calculated. Since finding the minimum scheduling period with the given chromosome representation is NP-hard [14], a heuristic based link scheduling algorithm for fitness evaluation is proposed, as shown in Algorithm 1.

From Algorithm 1, we can see that our heuristic link scheduling algorithm mainly includes two parts, the link scheduling order selection and time slot allocation. In the link scheduling order selection part, one link with the least interference degree will be served first, as shown in line (2)~line (6). In the slot allocation part, we first allocate as many as possible time slots in scheduling period T to satisfy its slot demands for each being served link, as shown in (8) to (16); if we cannot allocate enough slots to meet its slot demands, the scheduling period of the network T will be updated, and more $(T - r_{e_k} + a(e_k), T]$ will be allocated to this link, which can be shown in lines (17) to (20).

Crossover and mutation are two other fundamental operations involved in the evolution process of GA. The purpose of crossover operation is to generate offspring chromosomes (new individuals) by performing certain genetic recombination on parent chromosomes. These offspring chromosomes share genetic similarities with their parents but from a different generation. As to the crossover operation, considering that the chromosomes are expressed by tree data structure, we adopt single-point crossover to exchange partial chromosome, as shown in Figure 5. For two chromosomes Pa_1 and Pa_2 for crossover, we first find at least one common node (node 11) between these two chromosomes of the same source node (node 7). And then one common node, denoted as v , is randomly selected. For example, in Pa_1 , there is a path consisting of two parts: $(pa_1^1 : r \rightarrow v)$ and $(pa_1^2 : v \rightarrow \text{sink})$. In Pa_2 , there is also a path consisting of two parts: $(pa_2^1 : r_i \rightarrow v)$ and $(pa_2^2 : v \rightarrow \text{sink})$. The crossover operation then exchanges the paths $(pa_1^1 : r_i \rightarrow v)$ and $(pa_1^2 : r_i \rightarrow v)$ with the crossover probability p_c . The population will undergo the mutation operation after the crossover operation is carried out. With the mutation probability p_m , each time we select one chromosome Ch_i in which one source node r_i is randomly selected. On the path $(r_i \rightarrow \text{sink})$ one node is selected as the mutation point, denoted as v . The mutation will replace the path $(r_i \rightarrow v)$ by a new random path. Since the new generated chromosomes after crossover and mutation may be infeasible, an additional repair execution to

Input: Routing graph $G'(V', E')$, and the slot demands r_k of the k th link ($k = 1, \dots, m$)
Output: The scheduling period T and slot allocation

- (1) Construct the conflict graph $G'_c(V'_c, E'_c)$ from $G'(V', E')$
- (2) **while** G'_c is not empty do
- (3) find the vertex with the smallest degree in G'_c ;
- (4) remove this vertex from G'_c and all its incident edges;
- (5) Let e_k denote the $(m - k + 1)$ th removed vertex.
- (6) **end while**
- (7) Let $T = 1$;
- (8) **for** each link $e_k \in E'$, $k = 1 : m$
- (9) let $assign = \text{false}$ and $a(e_k) = 0$;
- (10) **for** slot $t = 1 : T$
- (11) **if** e_k do not conflict its neighbors in G'_c in slot t
- (12) assign e_k with slot t ;
- (13) **if** $(a(e_k) ++ == r_{e_k})$
- (14) $assign = \text{true}$;
- (15) break;
- (16) **end for**
- (17) **if** $assign == \text{false}$
- (18) $T = T + r_{e_k} - a(e_k)$;
- (19) assign e_k slot $(T - r_{e_k} + a(e_k), T]$;
- (20) **end for**

ALGORITHM 1: Heuristic Link Scheduling Algorithm.

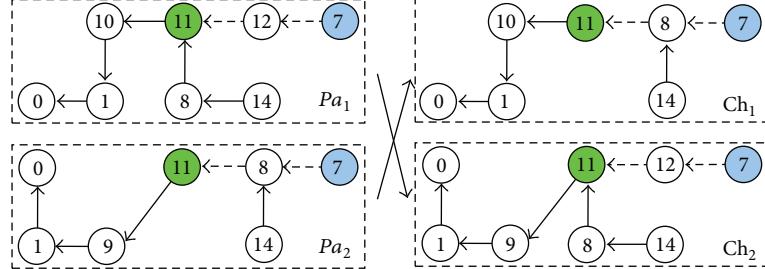


FIGURE 5: Illustration of the crossover operation.

eliminate the loops will be executed, which can help to speed up the convergence.

Unlike traditional GA, NSGA-II is built on nondomination in each front. Each individual in the population is assigned two fitness evaluations (scheduling length and energy consumption). In addition, if two solutions have the same fitness values, they will further be measured by the crowding distance, which can be calculated by the distance between its neighboring solutions along each of the objectives (see [24] for details). Algorithm 2 describes the detailed procedures of NSGA-II link scheduling algorithm. From Algorithm 2, we can see that after the generation of initial population P (shown in step 1), the fast nondominated sorting operation is executed first to generate nondominated fronts of the initial generation P , as shown in step 2. Then, the initial population is combined with new generated population P' , and controlled elitism operation is executed, which can update P by selecting the solutions starting from the first front. If the number of solutions in the remaining allowable front is larger than the available places in the population, then

the solutions with larger crowding distance are selected first. These procedures can be seen in step 3. Finally, similar to link GA, crossover and mutation operation is also applied to generate new population P' , as shown in step 4. After several iterations, the entire population only containing the solutions near or at the Pareto optimal front will be selected.

4. Performance Evaluation

In this section, we present the simulation results of our algorithm. All simulations are executed in Matlab 7.0. Two metrics, including the total energy consumption and scheduling length, are imported to evaluate the performance of our proposed algorithm. For all the simulations, we set the energy consumption for receiving and transmitting one unit data r_r and r_t as 1 and 2, respectively. The size of the population is set as 100; the crossover probability p_c and mutation probability p_m are set as 0.9 and 0.1. The controlled elitism ρ is set as 0.5. In our simulations, two types of node distributions are adopted: random distribution and hotspot distribution,

Input: Routing graph $G'(V', E')$, the slot demands r_k of the k th link ($k = 1, \dots, m$), the given number of generation G , the population size K , the recombination probability p_r , the mutation probability p_m , and the reduction rate of the controlled elitism ρ

Output: Non-dominated solutions in P

- (1) **Step 1.** (initialization)
- (2) Set $g = 0, P' = \emptyset$;
- (3) Generate an initial population P using coding and initialization strategy;
- (4) Calculate T using Algorithm 1 and $\sum_{q \in Q} E_q$ for each individual;
- (5) **Step 2.** (Non-dominated sorting)
- (6) $P = P \cup P'$;
- (7) **Do fast non-dominated sorting algorithm**, resulting non-dominated fronts (F_1, F_2, \dots, F_R)
- (8) **Step 3.** (controlled elitism)
- (9) Set $r = 1, P = \emptyset$;
- (10) **While** $|P| < K$ **do**
- (11) Calculate n_r according to the controlled elitism scheme;
- (12) Sort F_r in descending order using crowded comparison;
- (13) Put the first n_r members of F_r in P , that is, $P = P \cup F_r[1 : n_r]$;
- (14) $r = r + 1$;
- (15) **Step 4.** (Reproduction)
- (16) Generate an offspring P' (with size of K) from P according crossover and mutation operation;
- (17) Calculate T and $\sum_{q \in Q} E_q$ for each individual in P' ;
- (18) **Step 5.** (Termination)
- (19) $g = g + 1$;
- (20) **if** $g \geq G$ or the required T and $\sum_{q \in Q} E_q$ are met **then** terminate;
- (21) **else** go to Step 2

ALGORITHM 2: NSGA-II based link scheduling.

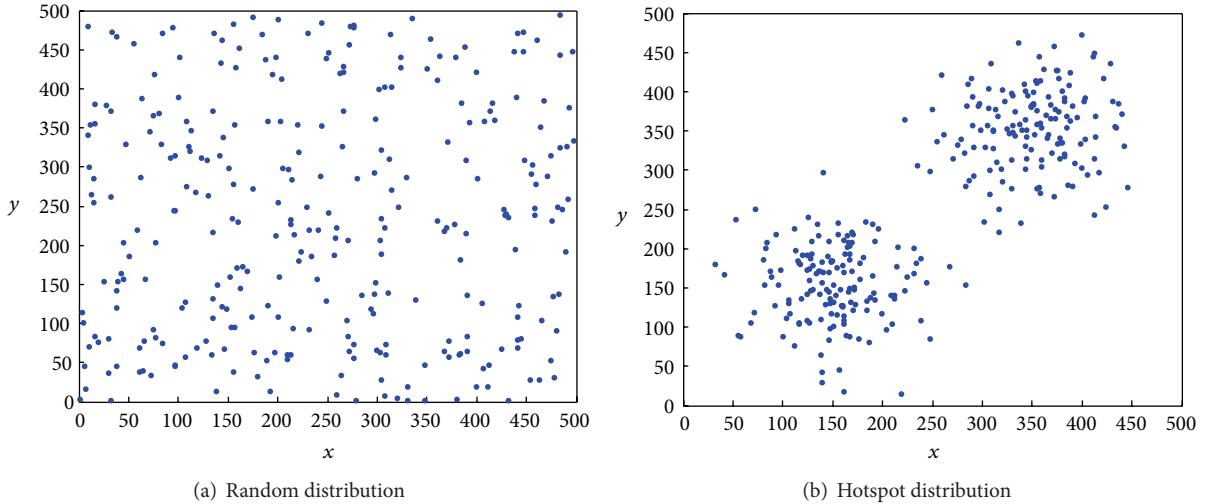


FIGURE 6: Two different sensor node distributions.

as shown in Figure 6. In random distribution, all nodes are randomly scattered in the target area, while in hotspot distribution, all nodes are scattered in a Gaussian random distribution.

First, 50 nodes are randomly deployed in a $120 * 120$ target area, as shown in Figure 9. The sink node (with ID 37) is located in the center of the area. The transmission range R_c and interference range R_I of each sensor node are equal to 20. Assume each sensor node has a unit slot demand routed to sink node. Figure 7 shows the initial solution distribution

of the randomly generated population. Figure 8 shows Pareto optimal solutions obtained after running the algorithm for a number of generations. From Figures 7 and 8, we can see that many more nondominated solutions are found after 30 generations. And after 600 generations, we can find the Pareto optimal solution with the minimum scheduling length and energy consumption. Figure 9 shows the topology graph of the routing tree in the last generation, where the number beside each link is the link number. The final slot allocation of each link is shown in Table 1. From the description of

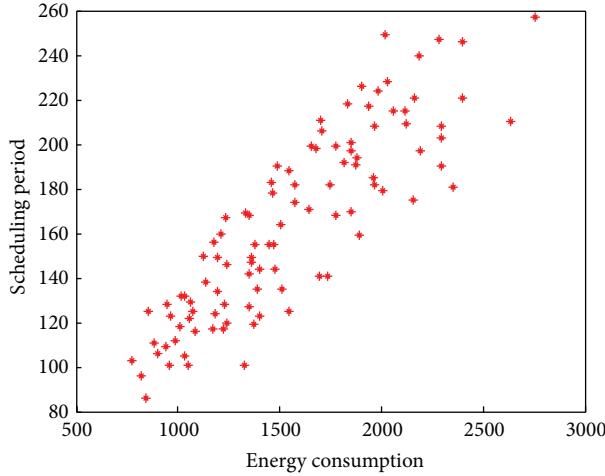


FIGURE 7: Distribution of initial population.

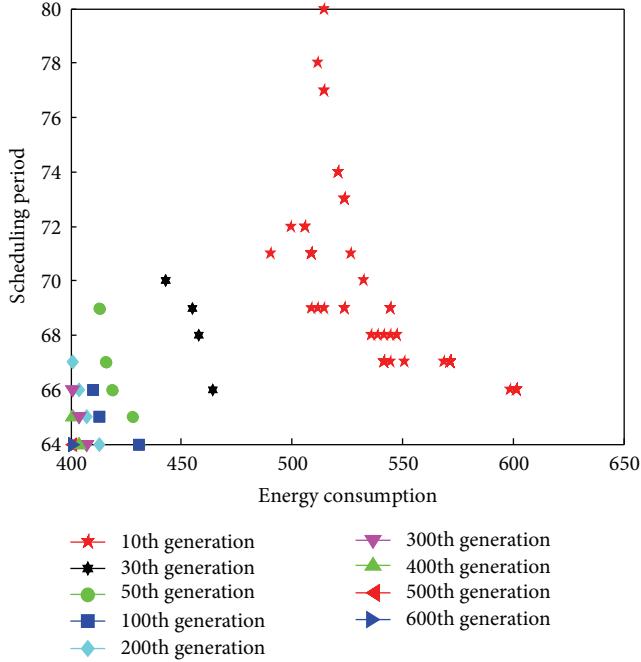


FIGURE 8: Distribution of population in different generations.

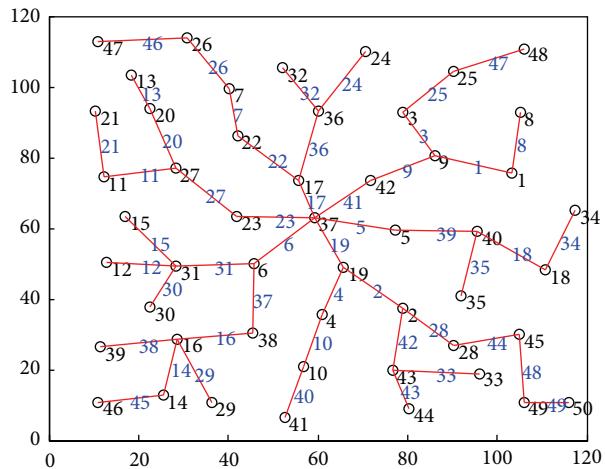


FIGURE 9: Topology of the optimal routing tree in the last generation.

TABLE 1: Time slot allocation table.

Link No.	Slot allocation
1	4, 8
2	10, 11, 13, 16, 17, 18, 19, 20
3	2, 5, 9
4	1, 5, 9
5	2, 3, 4, 8, 21
6	22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32
7	4, 6, 8
8	3
9	1, 10, 11, 13, 16, 19
10	3, 4
11	1, 2
12	4
13	5
14	3, 4
15	6
16	6, 8, 10, 11, 13
17	33, 34, 35, 36, 37, 38, 39, 40
18	1, 6
19	41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52
20	3, 7
21	4
22	5, 9, 12, 14
23	53, 54, 55, 56, 57, 58
24	1
25	6, 7
26	1, 2
27	10, 11, 13, 16, 19
28	2, 6, 12, 14
29	1
30	2
31	1, 5, 7, 9
32	3
33	1
34	2
35	5
36	15, 17, 18
37	12, 14, 15, 17, 18, 20
38	5
39	7, 12, 14, 15
40	2
41	6, 59, 60, 61, 62, 63, 64
42	7, 8, 15
43	5
44	3, 4, 9
45	2
46	3
47	1
48	7, 8
49	2

routing tree and the slot allocation, we can get the conclusion that any mutual interference links in the routing tree are assigned different time slots, and the total number of slots

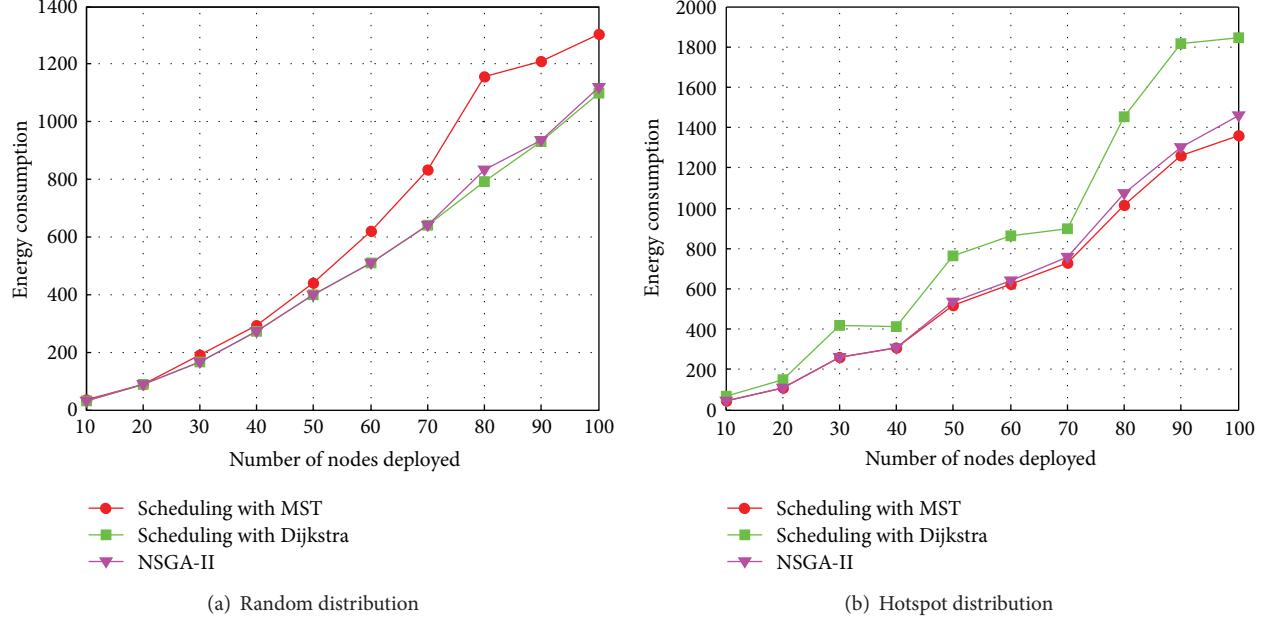


FIGURE 10: Comparison of energy consumption with different number of nodes deployed in different distributions.

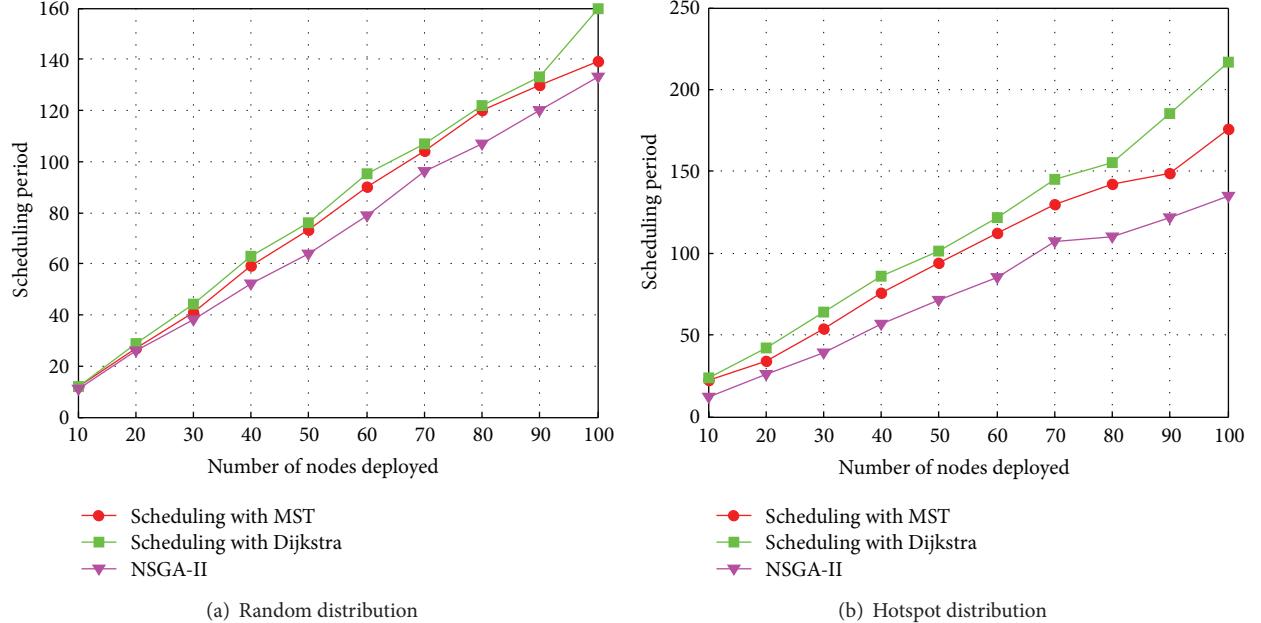


FIGURE 11: Comparison of scheduling period with different number of nodes deployed in different distributions.

assigned to any link also satisfies its routing requirement, which demonstrate the effectiveness of our algorithm.

Further, we test the performance of our algorithm under various node densities and compare the performance of the proposed algorithms with existing routing trees combined with link scheduling algorithm proposed in [14]. The comparison is executed with different node densities, where the number of sensor nodes is increased from 10 to 100. For each deployment, we have generated 10 instances and applied our algorithm on each instance with 20 independent

tests. The simulation results of random hotspot distribution are shown in Figures 10 and 11, which record the average solution of these 20 tests. In this simulation, the routing tree generation and link scheduling algorithms are performed jointly in our algorithm. Since the routing optimization is not considered in [14], first the routing tree is generated by Dijkstra and MST algorithms separately, and then the scheduling algorithm is applied with the given routing tree. As shown from Figures 10 and 11, it is clear to see that more traffic loads lead to an increase in scheduling period and

energy consumption as expected. And we also can see that the Dijkstra routing algorithm can achieve the minimized energy consumption, which is mainly because when the Dijkstra routing algorithm is applied, the minimized length of the routing tree can be achieved. However, the largest node degree in the routing tree of Dijkstra routing algorithm results in the longest scheduling period. Compared with the above two algorithms, our algorithm can achieve the shortest scheduling period, and the energy consumption is almost the same as that of Dijkstra routing algorithm, regardless of the node distribution, which further testifies the effectiveness of our algorithm.

5. Conclusion and Future Work

In this paper, we have investigated the challenging problem of joint optimization on link scheduling and energy consumption for wireless sensor networks. By considering route selection and link scheduling together, we have carried out the analysis of energy consumption and link scheduling for wireless sensor network and formulated it as a *bicriteria-objective integer problem*. Considering the specific many-to-one nature of WSNs, we have proposed an efficient optimization scheme based on NSGA-II to solve this problem. In order to ensure the individual validity and fast convergence, the routing tree based representation scheme as well as the corresponding crossover and mutation rules is also designed to satisfy the constraints in the problem. Through extensive simulations, we have demonstrated that our optimization framework based on NSGA-II is quite promising for solving complex multiobjective design models for WSNs.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants no. 61402096, no. 61173153, and no. 60903159 and the Fundamental Research Funds for the Central Universities under Grant nos. N110318001, N130504007, and N140407001.

References

- [1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, no. 4, pp. 445–487, 2005.
- [2] J. Yu, N. Wang, G. Wang, and D. Yu, "Connected dominating sets in wireless ad hoc and sensor networks—a comprehensive survey," *Computer Communications*, vol. 36, no. 2, pp. 121–134, 2013.
- [3] J. Jia, X. Wu, J. Chen, and X. Wang, "An autonomous redeployment algorithm for line barrier coverage of mobile sensor networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 16, no. 1, pp. 58–69, 2014.
- [4] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, "Z-MAC: a hybrid MAC for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 511–524, 2008.
- [5] J. Jia, G. Zhang, X. Wang, and J. Chen, "On distributed localization for road sensor networks: a game theoretic approach," *Mathematical Problems in Engineering*, vol. 2013, Article ID 640391, 9 pages, 2013.
- [6] S. Gandham, Y. Zhang, and Q. Huang, "Distributed time-optimal scheduling for convergecast in wireless sensor networks," *Computer Networks*, vol. 52, no. 3, pp. 610–629, 2008.
- [7] M. Arifuzzaman, M. Matsumoto, and T. Sato, "An intelligent hybrid MAC with traffic-differentiation-based QoS for wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 6, pp. 2391–2399, 2013.
- [8] S. C. Ergen and P. Varaiya, "PEDAMACS: power efficient and delay aware medium access protocol for sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 7, pp. 920–930, 2006.
- [9] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pp. 66–80, ACM, September 2003.
- [10] G. Sharma, R. Mazumdar, and N. Shroff, "On the complexity of scheduling in wireless networks," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom '06)*, pp. 227–238, ACM, New York, NY, USA, 2006.
- [11] T. Salonidis and L. Tassiulas, "Distributed dynamic scheduling for end-to-end rate guarantees in wireless ad hoc networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '05)*, pp. 145–156, ACM, May 2005.
- [12] E. Uysal-Biyikoglu, B. Prabhakar, and A. El Gamal, "Energy-efficient packet transmission over a wireless link," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 487–499, 2002.
- [13] S. C. Ergen and P. Varaiya, "TDMA scheduling algorithms for wireless sensor networks," *Wireless Networks*, vol. 16, no. 4, pp. 985–997, 2010.
- [14] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 181–192, ACM, November 2003.
- [15] I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAND: distributed randomized TDMA scheduling for wireless ad-hoc networks," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '06)*, pp. 190–201, ACM, May 2006.
- [16] Z. Shen, H. Jiang, and Z. Yan, "Fast data collection in linear duty-cycled wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 4, pp. 1951–1957, 2014.
- [17] S. Liu, S. Feng, W. Ye, and H. Zhuang, "Slot allocation algorithms in centralized scheduling scheme for IEEE 802.16 based wireless mesh networks," *Computer Communications*, vol. 32, no. 5, pp. 943–953, 2009.
- [18] J. Mao, Z. Wu, and X. Wu, "A TDMA scheduling scheme for many-to-one communications in wireless sensor networks," *Computer Communications*, vol. 30, no. 4, pp. 863–872, 2007.
- [19] D. Gong and Y. Yang, "Low-latency SINR-based data gathering in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 6, pp. 3207–3221, 2014.
- [20] S. Cui, R. Madan, A. J. Goldsmith, and S. Lall, "Cross-layer energy and delay optimization in small-scale sensor networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 10, pp. 3688–3699, 2007.

- [21] O. Durmaz Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, "Fast data collection in tree-based wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 86–99, 2012.
- [22] P. Djukic and S. Valaee, "Delay aware link scheduling for multi-hop TDMA wireless networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, pp. 870–883, 2009.
- [23] I. Das and J. E. Dennis, "Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multi-criteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.
- [24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [25] J. Jia, J. Chen, G. Chang, and Z. Tan, "Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm," *Computers & Mathematics with Applications*, vol. 57, no. 11-12, pp. 1756–1766, 2009.

