IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 30, 2018, accepted May 24, 2018, date of publication May 29, 2018, date of current version June 20, 2018.

*Digital Object Identifier 10.1109/ACCESS.2018.2841985*

# Evolutionary Sleep Scheduling in Software-Defined Networks

**WEIQI CHEN[1], HAN CHEN[1], QUANSHENG GUAN[1], (Senior Member, IEEE), FEI JI[1], AND BINGYI GUO[2]**

[1]School of Electronic and Information Engineering, South China University of Technology, Guangzhou 501640, China
[2]School of Information Science and Engineering, Shandong Normal University, Jinan 250000, China

Corresponding author: Quansheng Guan (eeqshguan@scut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grants 61431005, 61671208, 61671211, and U1701265, in part by the Guangdong provincial research project under Grants 2016A030308006 and 2016B090918049, and in part by the Pearl River S&T Nova Program of Guangzhou.

**ABSTRACT** The redundant design of communication networks leads to under-utilization of idle devices, which have been reported to consume a significant portion of energy. Thus, it demands a sleep scheduling scheme to improve energy efficiency of communication networks. In this paper, we formulate the optimal sleep scheduling problem from the perspective of routing, which aggregates the traffic loads to fewer active devices by route selection and put the idle devices into sleep to save energy. We then design a genetic algorithm to find out near-optimal sleep scheduling solution, which facilitates the implementation in software-defined networks. Simulation results over network instants from the online database survivable network design library show that our proposed genetic sleep scheduling algorithm outperforms the existing schemes in saving energy.

**INDEX TERMS** Green networks, sleeping scheduling, energy efficiency, genetic algorithm.

## I. INTRODUCTION

As communication networks are designed redundantly without concern on energy consumption to meet the peak communication demands, it has led to an overwhelmingly low energy efficiency [1], [2]. It has been also verified by the fact reported in [3] that the operating factor of network traffic loads reaches only 30% at peak hours and it drops down to 5% at idle time. The development of datacenter networks may further increase the waste of energy. Therefore, it has become an imperative demand to improve the energy efficiency in communication networks.

Avoiding network redundancy and over-provisions can reduce energy wastes in communication networks. An efficient way to improve energy efficiency is to match the network service capability with the network service requirement by dynamically scheduling network resources. The operating power in a network device is related directly to energy consumption in that device. Thus, two fundamental methods including dynamic power adaptation and dynamic sleep scheduling are in common use [4].

Dynamic power adaptation is to adjust the power of CPU in network devices to save energy, since a small CPU power results in a low data rate. Reference [5] suggests an algorithm that relies on the feedback from datacenter network devices to adapt the devices' power to the traffic loads. Reference [6] considers also the failure management costs while saving electricity consumption. A similar mechanism is also applied in cellular orthogonal frequency division multiple (OFDM) access networks [7]. Similarly, power control in the radio transmitter is useful for wireless networks to save energy [8]. However, the scope of energy saving of power adaptation applies only to the CPU in a device. The other function modules in network devices like buses and memory remain working, which are still consuming energy. It has been revealed in [4] that the idle device consumes almost 90% energy of the fully-loaded device. In this sense, power adaptation saves only limited energy, while putting devices into sleep can save more energy.

The idea of sleep scheduling comes from some studies [9]–[11], which exhibit a fact that the activated network devices, rather than traffic loads, are responsible for network energy consumption. Thus we can aggregate traffic loads into some devices and make others sleep for energy-savings. Some of network devices are emerging to support sleep/wakeup primitives [12]–[14], making the sleep scheduling feasible in communication networks.

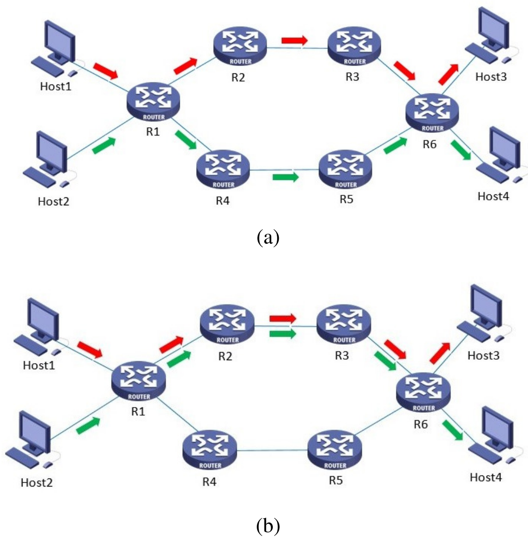



**FIGURE 1. Aggregating the users' demands into fewer nodes and links saves energy consumption in the network. (a) All nodes and links are active in the network. (b) R4 and R5, as well as their associated links are idle to sleep.**

The challenges for sleep scheduling exist in that the sleep/activate operations in each device have to be carried out in a network-wide perspective. The individual sleeping may otherwise downgrade network services or even lose network connectivity. Routing is a feasible method for the distributed network system to deal with the network connectivity and traffic distribution in the network. References [15] and [16] had proposed an energy-aware routing (EAR) algorithm in backbone networks which re-regulates traffic loads in the devices and activates a small part of network devices to serve traffic loads. The EAR algorithm is realized by the existing shortest path routing to distribute traffic loads. Reference [17] partitions the network nodes into clusters. The cluster members can put their routing modules into sleep to save energy.

We study the optimal sleep scheduling problem from the perspective of routing, by distributing the traffic loads to the least number of active nodes and links, subject to network connectivity and service capability. Thus the idle nodes can then be put into asleep to save energy. We use the example in Fig. 1 to demonstrate the routing-based sleep scheduling. There are 6 nodes and 10 links in the network. Two user flows are attached to the network. We usually prefer two separate routes which activate all the nodes and links in the network to avoid network congestion (see Fig. 1(a)). However, 4 nodes and 7 links are enough to meet the users' demands, when we aggregate the users' demands and route their traffic into one path (see Fig. 1(b)). The solution in Fig. 1(b) saves the energy consumption in 2 nodes and 3 links, which become idle after traffic aggregation. Note that aggregating traffic loads into fewer links may incur traffic delay. However, the B4 network using the software-defined network architecture (SDN) [18] has proven that the aggregated traffic loads on a link can almost approach up to 90% of its capacity without significant traffic delay.

The computation of a routing solution having the least active nodes and links in the network for the optimal sleep scheduling desires the centralized implementation, which is a big challenge for the traditional distributed network system. Fortunately, the emerging SDN technology has enabled centralized decision-making in the network [19], [20] and provided flexible control on data transmissions [21]. SDN decouples the control plane (the controller) and the data plane which consists of a number of switches. The controller distributes flow tables to the switches to control the flow forwarding. Network developers can control the construction of flow tables with the help of the APIs provided by the controller. Therefore, we can construct the flow tables according to the routing solution we find. Thus the centralized sleep scheduling then can be readily implemented in the SDN controller. Energy-aware routing has been implemented in SDNs to achieve green networking [22], [23].

Routing in SDNs has attracted extensive research works in literature. Using SDN, different routing schemes can be designed flexibly to meet different network demands [24], [25]. Reference [24] designed a route selection and update algorithm to find routes that minimize the delay under link capacity constraints. Reference [26] presented a model that jointly considers the node capacity and bandwidth capacity at each links, and then proposed an online routing algorithm. Reference [27] formulated the energy consumption problem into a minimum-cost commodity flow problem and proposed an N-algorithm to solve it. The existing works have shown that the optimal routing problem in communication networks is often NP-hard. We need an efficient method to find the optimal routing strategy to achieve an energy-efficient network.

We apply the idea of genetic algorithm (GA) to find the optimal sleep scheduling solution in this paper. GA is often described as an excellent global search algorithm to search an acceptable suboptimal solution [19]. GA has also been widely applied in communication networks to solve the network design and resource optimization issues [28]–[30].

The contributions of this paper are summarized as follows.

- *Formulation for routing based sleep scheduling*: We formulate the sleep scheduling into a routing problem that finds the least active devices, including nodes and links, to serve traffic loads. Different from the previous works like [25], which consider only the energy consumption of links, we consider also both the capacity constraint and energy consumption in nodes.
- *GA-based sleep scheduling algorithm*: We propose a GA-based algorithm to find the optimal solution for the routing-based sleep scheduling problem. A routing strategy is regarded as an individual, whose chromosome is made up by genes of users. A gene stands for a feasible path of a user. A number of individuals form a population. The fitness function relates directly to the energy consumption in the network. After three basic GA operations, including reproduction, crossover and mutation operation, a population steps to next generation that has

**TABLE 1.** Variables and descriptions.

| Variable | Description |
|---|---|
| $e_{ij}$ | A link from node $i$ to node $j$ |
| $x_{ij}^u$ | Indicates whether user $u$ is using $e_{ij}$. |
| $W_{ij}$ | The bandwidth capacity of link $e_{ij}$. |
| $w_{ij}$ | The loads on link $e_{ij}$. |
| $c_{ij}$ | The energy consumption of link $e_{ij}$. |
| $x_{ij}$ | Indicates whether link $e_{ij}$ is activated in $S$. |
| $v_k$ | Node $k$, $V = \{v_k \| 0 < k \leq \|V\|\}$. |
| $x_k^u$ | Indicates whether user $u$ is using $v_k$. |
| $W_k$ | The bandwidth capacity of node $v_k$. |
| $w_k$ | The load on node $v_k$. |
| $c_k$ | The energy consumption of node $v_k$. |
| $x_k$ | Indicates whether node $v_k$ is activated in $S$. |
| $n$ | The number of users. |
| $w_u$ | The communication demand of user $u$. |
| $p_u$ | A feasible path for user $u$. |
| $P_u$ | All feasible paths for user $u$, $p_u \in P_u$. |
| $S$ | A routing strategy, $S = \{p_1, p_2, \cdots, p_n\}$. |

better fitness. The individual with the best fitness after sufficient generations will approach the optimal sleep scheduling solution.

- *Simulation comparison*: We use the real network scenarios provided by the online database Survivable Network Design Library (SNDlib) [31] to study the energy efficiency of our proposed algorithm in the simulation. The simulation results show that our proposed GA-based sleep scheduling algorithm outperforms the existing schemes and the heuristic algorithm proposed by [25] in terms of energy efficiency.

This paper is structured as follows. Section II presents the system model and problem formulation. Section III proposes the GA-based sleep scheduling algorithm. Simulation results are shown in Section IV. Finally, Section V concludes this paper.

## II. SYSTEM MODEL

In this section, we will describe the network model, the energy consumption model and the problem formulation. The notations used in the discussions are listed in the Table. 1.

### A. NETWORK MODEL

We consider a network $G = (V, E)$, which consists of a set of nodes $V = \{v_k | 0 < k \leq |V|\}$, and a set of edges/links[1] $E = \{e_{ij} | i, j \in V, \ i \neq j\}$. There are $n$ users using this network, denoted by $N = \{1, 2, ..., n\}$.

The source node $s_u$ denotes the ingress node that user $u$'s traffic enters into the network, and the egress node $d_u$ is the destination of user $u$'s traffic. The network has to establish a reachable path between $s_u$ and $d_u$ with a sequence of links and nodes to meet the communication demand of user $u$, which is denoted by $w_u$. One feasible path $p_u = \{v_{u_1}, v_{u_2}, \cdots, v_{u_{(t+1)}}\}$ for user $u$ is composed of $t$ continuous edges $E_u = \{e_{v_{u_1}v_{u_2}}, e_{v_{u_2}v_{u_3}}, \cdots, e_{v_{u_t}v_{u_{(t+1)}}}\}$ and $t+1$ nodes $V_u = \{v_{u_1}, v_{u_2}, \cdots, v_{u_{(t+1)}}\}$. Let $x_{ij}^u \in \{0, 1\}$ be a binary variable to stand for whether link $e_{ij}$ is activated to serve user $u$, and let $x_{ij}^u = 1$ if link $e_{ij} \in E_u$, otherwise, $x_{ij}^u = 0$. Similarly, $x_k^u = 1$ if node $v_k \in V_u$, otherwise, $x_k^u = 0$. All feasible paths for user $u$ are assembled in a set $P_u$. Therefore, we have $p_u \in P_u$. A set $S = (p_1, p_2, ..., p_n)$ is a routing strategy that satisfies all users' demands, with which we can obtain a connected sub-network. The subgraph $G' = \cup_{u \in N} p_u$ may consist of a part of the nodes and edges in the original networks $G$.

The bandwidth capacities of an edge $e_{ij}$ and a node $v_k$ are denoted by $W_{ij}$ and $W_k$ respectively. User $u$ will consume $w_u$ bandwidth of the path $p_u$ when it uses path $p_u$. Thus the total loads on edge $e_{ij}$ are the summation of the bandwidth consumption on this edge by all users, i.e.,

$$w_{ij} = \sum_{u \in N} w_u \cdot x_{ij}^u, \tag{1}$$

subject to $W_{ij} \geq w_{ij} \geq 0$. Similarly, the total loads on node $v_k$ are

$$w_k = \sum_{u \in N} w_u \cdot x_k^u, \tag{2}$$

subject to $W_k \geq w_k \geq 0$. It is worth noting that the loads on edges or nodes cannot be larger than their bandwidth capacities.

### B. ENERGY CONSUMPTION MODEL

Since the idle device consumes up to 90% energy of the fully-loaded device, we make no difference on energy consumption of active devices with traffic loads that are larger than 0. Any node or link which is included in the routing strategy should be active to serve the traffic loads. We denote $c_{ij}$ as the energy consumption of link $e_{ij}$. Let $x_{ij} \in \{0, 1\}$ as the state of link $e_{ij}$, i.e., $x_{ij} = 1$ if link $e_{ij}$ is active, otherwise, $x_{ij} = 0$. Similarly, we denote $c_k$ as the energy consumption of node $v_k$, and $x_k \in \{0, 1\}$ as the state of node $v$, i.e., $x_k = 1$ if node $v_k$ is active, otherwise, $x_k = 0$.

For $n$ users using the network with a strategy set $S = (p_1, p_2, \cdots, p_n)$, the total energy cost can be the summation of the energy cost of links and nodes as follows:

$$C_S = C_{link} + C_{node}, \tag{3}$$

where

$$C_{link} = \sum_{i \in V} \sum_{j \in V, i \neq j} c_{ij} \cdot x_{ij}, \tag{4}$$

$$C_{node} = \sum_{v_k \in V} c_k \cdot x_k. \tag{5}$$

[1]Edges and links are considered as the same concept, and are interchangable in the paper.

### C. PROBLEM FORMULATION

As aforementioned, the sleep scheduling saves energy by putting the idle network devices into sleep. Since whether nodes/links are idle or not is determined by the traffic distribution, we intend to find out a routing strategy $S = (p_1, p_2, \cdots, p_n)$ that activates the least network devices, which can be described as

$$\begin{aligned} &\textit{find}: \; S, \\ &\textit{minimize}: \; C_S, \\ &\textit{subject to}: \; \eta W_{ij} \geq w_{ij}, && (6)\\ &\qquad\qquad \eta W_k \geq w_k, && (7)\\ &\qquad\qquad \sum_{u\in N}\sum_{i\in V, i\neq v} w_u \cdot x_{iv}^u = \sum_{u\in N}\sum_{j\in V, j\neq v} w_u \cdot x_{vj}^u. && (8) \end{aligned}$$

The capacity constraints for a link and a node are presented in in equations (6) and (7), where the traffic loads carried by a node or a link cannot be larger than their bandwidth limits. The coefficient $\eta$ ($0 \leq \eta \leq 1$) is used to prevent the long queuing delay introduced by the overloaded link,[2] e.g., $\eta = 0.9$ means that the link or node utilization is up to 0.9.

The traffic loads on a link or a node are calculated by eqs. (1) and (2), respectively. The intermediate nodes for a flow will not generate new traffic loads. Thus, the incoming flow should be equal to the outgoing flow for every intermediate node, which is presented by the constraint (8).

## III. GENETIC ALGORITHM FOR SLEEP SCHEDULING

Searching a routing strategy for a user to minimize the energy cost of a given network is often an NP-hard problem [28]. Thus, searching a routing strategy for multiple users to minimize the energy cost is also an NP-hard problem. Since GA has excellent performance in finding solution for the optimization problem, we design a GA-based sleep scheduling algorithm to find the solution.

### A. GA-BASED SLEEP SCHEDULING ALGORITHM

To begin with, we should encode the chromosome of an individual $S$, which is a routing strategy for the network. A number of individuals form a population $\mathbb{S}$ ($S \in \mathbb{S}$). A fitness function is introduced to evaluate an individual. The *reproduction* operation selects two individuals according to their fitness values calculated by the fitness function, and the *crossover* operation switches part of the chromosome of the two individuals, after which the *mutation* operation is applied to each individual to change one gene of the chromosome. After sufficient generations' evolution with the above three operations, we can obtain an individual with the best fitness, which is an acceptable solution for the problem.

The components of GA are described as follows:

- *Encoding the chromosome*: Encoding the chromosome for individuals reasonably is a crucial step for genetic algorithm. To make sure the strategy denoted by the chromosome is a feasible solution, we encode the chromosome with $n$ genes, and a gene denotes a feasible path for a user. The SDN controller often has the whole picture of the network connections, i.e., $G(V, E)$. Thus, it can calculate all the feasible paths for users. To do this, the SDN controller has to first collect the source nodes $s_u$ ($\forall u \in N$) and the destination nodes $d_u$ ($\forall u \in N$) for all users, as well as their communication demands $w_u$ ($\forall u \in N$).
  Afterwards the SDN controller finds out all feasible paths for all users by applying the depth-first search algorithm [33]. Denote the feasible solution space by $T = \{P_1, P_2, \cdots, P_n\}$, where $P_u = \{p_1^u, p_2^u, \cdots, p_k^u\}$ ($1 \leq u \leq n$) stands for all feasible paths for user $u$ and $k$ is the number of the feasible paths. For example, $P_1 = \{p_1^1, p_2^1, \cdots, p_k^1\}$ means that there are $k$ feasible paths meeting user 1's communication demand. Note that a path has been regarded as a gene. Thus, the paths $p_1^1, p_2^1, \cdots, p_k^1$ are allele to each other. Suppose $p_1^1 = \{1, 3, 5, 7\}$, which denotes that nodes 1, 3, 5, 7 and edges (1, 3), (3, 5), (5, 7) are activated to serve the communication demand of user 1.
  The selected path for a user is encoded as a gene in our GA-based algorithm. Thus, the chromosome of an individual $S_i$ consists of $n$ genes ($p_1, p_2, \cdots, p_n$), each of which is selected from $P_i$ ($1 \leq i \leq n$) randomly. So we can describe an individual as $S_i = \{p_1, p_2, \cdots, p_n\}$. Every gene in $S_i$ is not allele to each other. For example, suppose an individual's chromosome is denoted by $S_i = \{1, 3, \cdots, 4\}$, we can know that the network controller selects the $1st$ path in $P_1$ for user 1, the $3rd$ path in $P_2$ for user 2 and the $4th$ path in $P_n$ for user $n$. Similarly, another $m-1$ individuals are selected to form a population with $m$ individuals, that is $\mathbb{S} = \{S_1, S_2, \cdots, S_m\}$.
- *Fitness function*: A fitness function is introduced to judge whether an individual's gene-type is good or not. Our objective is to search a routing strategy that costs least energy. For simplicity, we denote the fitness function as the following formulation that is generated from the objective function:

$$f(S_i) = \begin{cases} \dfrac{1}{C_{S_i}}, & \text{(6) and (7) are both satisfied,} \\ 0, & \text{(6) or (7) are unsatisfied.} \end{cases} \tag{9}$$

The cost of energy of an individual is set to infinity if nodes or edges included in the routing strategy are overloaded according to (6) and (7), which means the fitness of the individual is 0. The larger the fitness value of an individual means the better energy efficiency it achieves. To make sure the individual with larger fitness value has a higher chance to be selected, we adopt the Roulette Wheel method [34] to denote the probability of an individual to be selected, that is

$$pro(S_i) = \frac{f(S_i)}{\sum_{S_i\in\mathbb{S}} f(S_i)}. \tag{10}$$

[2] Suppose the link utilization is $U$, the average queuing delay is proportional to $\frac{U}{1-U}$ [32, Ch. 3].

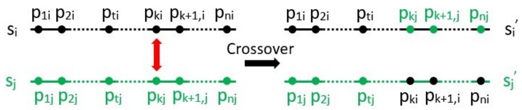



**FIGURE 2.** The crossover operation in the GA-based sleep scheduling algorithm.


**FIGURE 3.** The mutation operation in the GA-based sleep scheduling algorithm.

We can see that the probability that an individual with infinite energy consumption is selected as the routing strategy is zero, which means that this individual becomes ineffective. As the number of ineffective individuals increases, the search space of the GA algorithm decreases, which will slow down the convergence speed. A repair function is used to solve this problem. The repair function first finds out which gene causes the strategy overloaded, and then replaces the gene with another gene that is the shortest path meeting the user's demand.

- *Reproduction*: The reproduction operation imitates the mechanism for the survival of the fittest to select individuals to mate. The individuals with better fitness value have a higher probability to be selected. We realize this operation by using (10).
- *Crossover*: In the crossover operation, we adopt single-point crossing method, which can deal with two individuals $S_i = \{p_{1_i}, p_{2_i}, \cdots, p_{n_i}\}$ and $S_j = \{p_{1_j}, p_{2_j}, \cdots, p_{n_j}\}$ that are selected by the reproduction operation in a time. As shown in Fig. 2, we first randomly select the same point in two individuals' chromosomes, e.g., $p_{k_i}$ and $p_{k_j}$. Then we exchange genes in two chromosomes from the selected gene to the last gene. Then we have two new individuals $S_i' = \{p_{1_i}, p_{2_i}, \cdots, p_{k_j}, \cdots, p_{n_j}\}$ and $S_j' = \{p_{1_j}, p_{2_j}, \cdots, p_{k_i}, \cdots, p_{n_i}\}$. The genes in $S_i'$ and $S_j'$ are still not allele to each other.
- *Mutation*: The mutation operation only deals with one individual generated by the crossover operation in a time. The mutation operation will be carried out at a randomly selected gene with a probability called mutation rate. We use Fig. 3 to demonstrate how the mutation operation works. We select a point randomly on the chromosome $S_i' = \{p_{1_i}, p_{2_i}, \cdots, p_{k_j}, \cdots, p_{n_j}\}$, e.g., $p_{t_i}$, and then replace the pointed gene with its best allele $p_{t_i'}$. The best allele $p_{t_i'}$ of a given gene $p_{t_i}$ is defined as a gene in $P_t$ that makes the fitness of the individual maximum when other genes of a chromosome are settled. Thus, we can get $S_i'' = \{p_{1_i}, p_{2_i}, \cdots, p_{t_i'}, \cdots, p_{k_j}, \cdots, p_{n_j}\}$. The same operation will be also applied to $S_j'$.

### B. ALGORITHM DESIGN

Based on the operations in Section III-A, the GA-based sleep scheduling algorithm is described in Algorithm 1.

**Algorithm 1** GA-Based Sleep Scheduling Algorithm

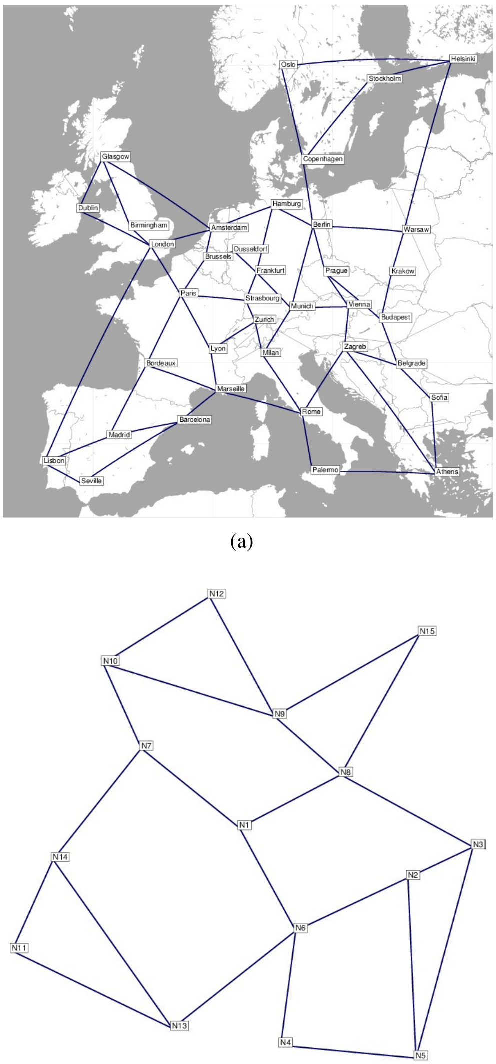
1: The network controller collects all users' $s_j, d_j, w_j, j \in N$, and calculates all feasible paths $T = \{P_1, P_2, \cdots, P_n\}$
2: Select $m$ individuals as the first generation population $\mathbb{S} = \{S_1, S_2, \cdots, S_m\}$, where $S_i = \{p_1, p_2, \cdots, p_n\}$, $i \in m$
3: Initialize the number of generations $g$, and the mutation rate $u$
4: **for** The generation is less than $g$ **do**
5: Calculate the fitness of each individual $S_i$
6: **for** the amount of new individuals is less than $m - 2$ **do**
7: Do Reproduction Operation
8: Do Crossover Operation
9: Do Mutation Operation
10: Get two new individuals and put them into $\mathbb{S}_{new}$
11: **end for**
12: Copy the individual with the maximum fitness twice into $\mathbb{S}_{new}$
13: Replace the population $\mathbb{S}$ with $\mathbb{S}_{new}$ and step to the next generation
14: **end for**
15: Output the individual with the maximum fitness in $\mathbb{S}$, e.g., $S_j$
16: The network controller establishes the network $G' = \cup_{j \in N} p_j, p_j \in S_j$
17: Construct flow tables according to $G'$ and then distribute them to switches by controller.
18: Schedule nodes and edges in $G - G'$ into asleep

As described in Algorithm 1, the network controller collects information from all users to calculate all feasible paths in Step 1. Each path is regarded as a gene, namely the unit of a chromosome.

In Step 2, the individuals' chromosomes are encoded. We select $m$ individuals and get a population $\mathbb{S}$ as the first generation. In order to accelerate the convergence of the GA-based algorithm, one of the individuals is formed with the shortest paths meeting users' demands.

Step 3 initializes the parameters of the algorithm. Steps 7 to 9 are the core of GA, including three operations: reproduction, crossover and mutation. They make two new individuals of next generation. Repeat Steps 7 to 10 until we get $m-2$ new individuals. We copy the individual with maximum fitness in population $\mathbb{S}$ twice to be the last two new individuals, such that the next generation's maximum fitness value will not be less than current one's [35], as shown in Step 12. Thus we get a new population $\mathbb{S}_{new}$ with $m$ new individuals. Replace $\mathbb{S}$ with $\mathbb{S}_{new}$ and repeat Steps 4 to 14 until we reach the $G$-th generation.

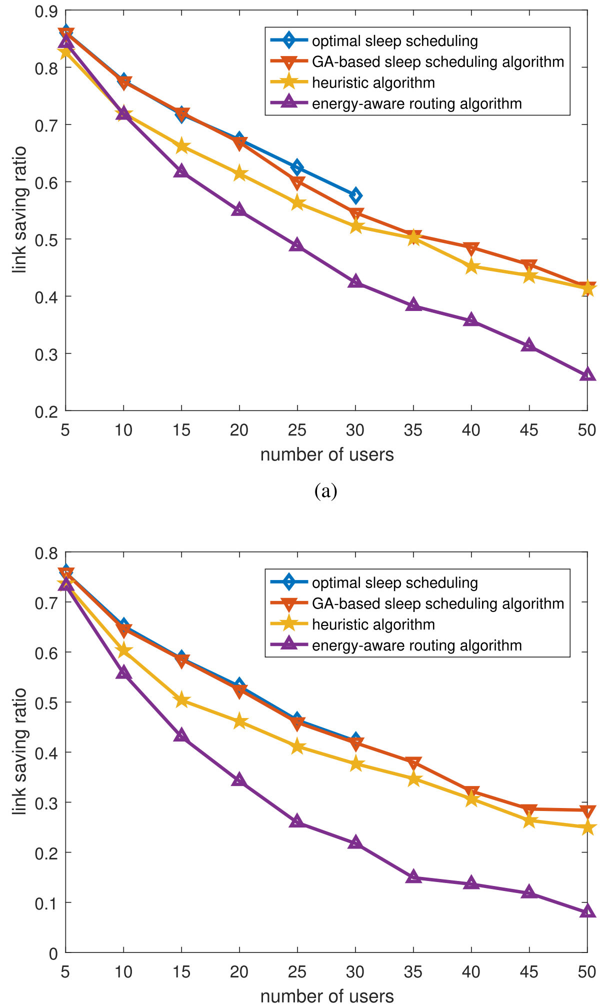
Step 15 shows that the individual with the maximum fitness in the final generation, e.g., $S_j$, will be the output as the routing strategy. Based on $S_j$, the network controller constructs flow tables, activates nodes and edges in $S_j$, and shuts down the other ones as shown in Steps 16 to 18.




(a)

(b)

**FIGURE 4.** **Network instances from the database SNDlib. (a) Europe network. (b) Atlanta network.**




(a)

(b)

**FIGURE 5.** **Link saving ratio vs. the number of users. (a) Europe. (b) Atlanta.**

## IV. SIMULATION RESULTS AND DISCUSSIONS

We now present the simulation results and study the performance of the proposed algorithm in this section.

### A. SIMULATION NETWORK INSTANCES

We import two network instances as simulation scenarios from the Survivable Network Design Library (SNDlib) [31], which is a fixed telecommunication network data library available at [36]. The chosen instances in the simulations are *Europe* and *Atlanta*, which stand for the realistic networks in Europe and Atlanta respectively, as shown in Fig. 4. There are 37 nodes and 57 links in network *Europe* (Fig. 4(a)) and 15 nodes and 22 links in network *Altanta* (Fig. 4(b)). Each network has unbalanced traffic and the bandwidth capacity as well as different energy consumptions in links or nodes.

### B. SIMULATION SETTINGS AND PERFORMANCE METRICS

In the simulations, the parameters of GA is set as follows: $g = 50$, $u = 0.5$, $m = 50$.

We use the following metrics to study the performance of our proposed sleep scheduling algorithm.

- *Link saving ratio*: the ratio of the number of idle links to the number of all links in the network, i.e.,

$$\text{link saving ratio} = \frac{\text{\# of idle links}}{\text{\# of all links}}. \tag{11}$$

- *Link utilization*: the ratio of the used bandwidth of a link to the capacity of that link, i.e.,

$$\text{link utilization} = \frac{\text{the used bandwidth of a link}}{\text{the capacity of a link}}. \tag{12}$$

  Similarly, the *average link utilization* is the average link utilization of active links, i.e.,

$$\text{average link utilization} = \frac{\sum \text{link utilization}}{\text{\# of active links}}. \tag{13}$$

- *Node saving ratio*: the ratio of the number of idle nodes to the number of all nodes in the network, i.e.,

$$\text{node saving ratio} = \frac{\text{\# of idle nodes}}{\text{\# of all nodes}}. \tag{14}$$

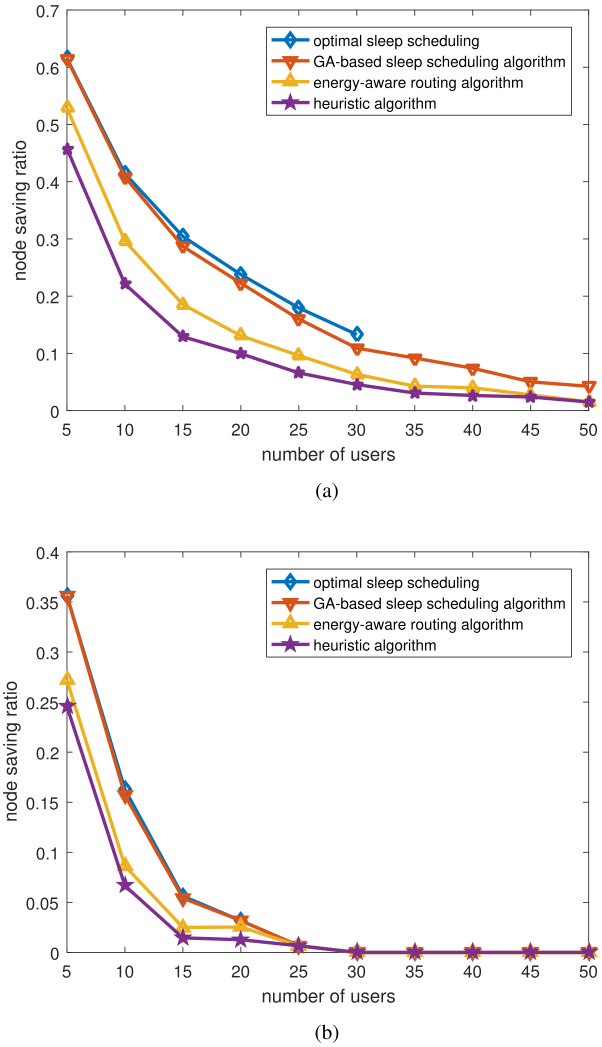



**FIGURE 6.** **Node saving ratio vs. the number of users. (a) Europe. (b) Atlanta.**

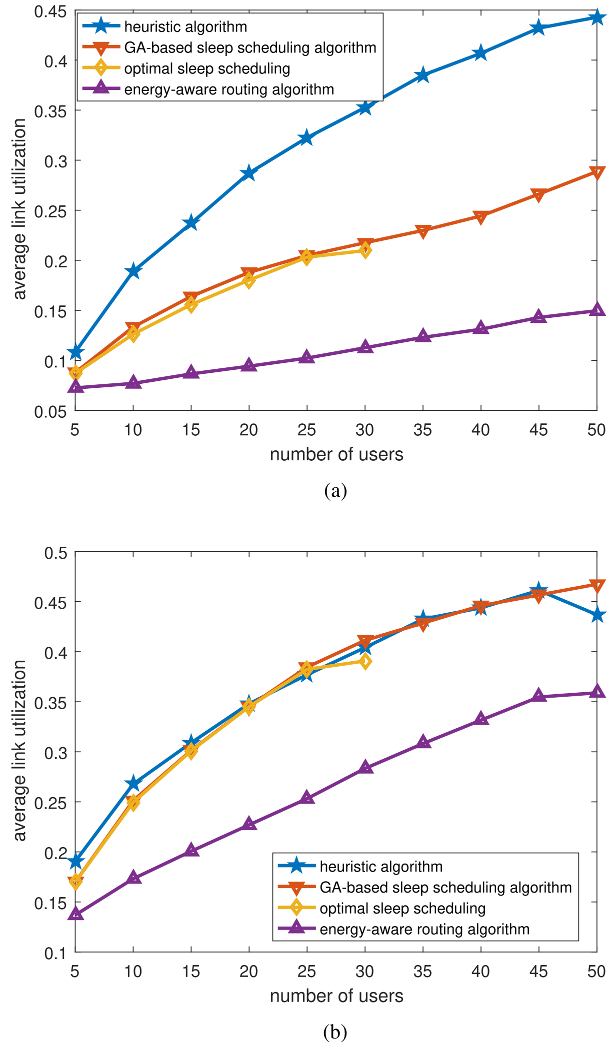



**FIGURE 7.** **Average link utilization vs. the number of users. (a) Europe. (b) Atlanta.**

- *Node utilization*: the ratio of the number of active neighbor links of a node to the total number of neighbor links of that node, i.e.,

$$\text{node utilization} = \frac{\#\text{ of active neighbor links}}{\#\text{ of all neighbor links}}. \tag{15}$$

The *average node utilization* is the average node utilization of active nodes, i.e.,

$$\text{average node utilization} = \frac{\sum \text{node utilization}}{\#\text{ of active nodes}}. \tag{16}$$

- *Energy saving ratio*: the ratio of the energy saved by the sleep scheduling scheme to the energy consumption without sleep scheduling, i.e.,

$$\text{energy saving ratio} = \frac{\text{the energy that saved}}{\text{total energy}}. \tag{17}$$

The link saving ratio and the node saving ratio indicate how many links and nodes that can be freed from working and then can be put into sleep. A routing strategy with larger link saving ratio and node saving ratio means that it can satisfy users' demands with fewer network devices, which frees more network devices that can be scheduled to sleep and saves more energy. The link utilization shows how much the bandwidth of an active link is used. We are also interested in the link and node utilization because the utilization shows how efficiently the link and node resources are utilized. As for the energy saving ratio, it stands for how much energy can be saved. With higher energy saving ratio, a routing strategy can save more energies.

### C. RESULTS AND DISCUSSIONS

We study our GA-based algorithm by comparing it to the optimal sleep scheduling, the existing EAR algorithm and the heuristic method proposed by [25]. The existing EAR algorithm finds a feasible shortest path for each user. We use exhaustive search to find the optimal sleep scheduling strategy. We only plot the data of the optimal sleep scheduling under 30 users because the running of the exhaustive search takes lots of the time.

Fig. 5 shows the link saving ratio of the two networks. From Fig. 5(a) we know that our GA-based algorithm

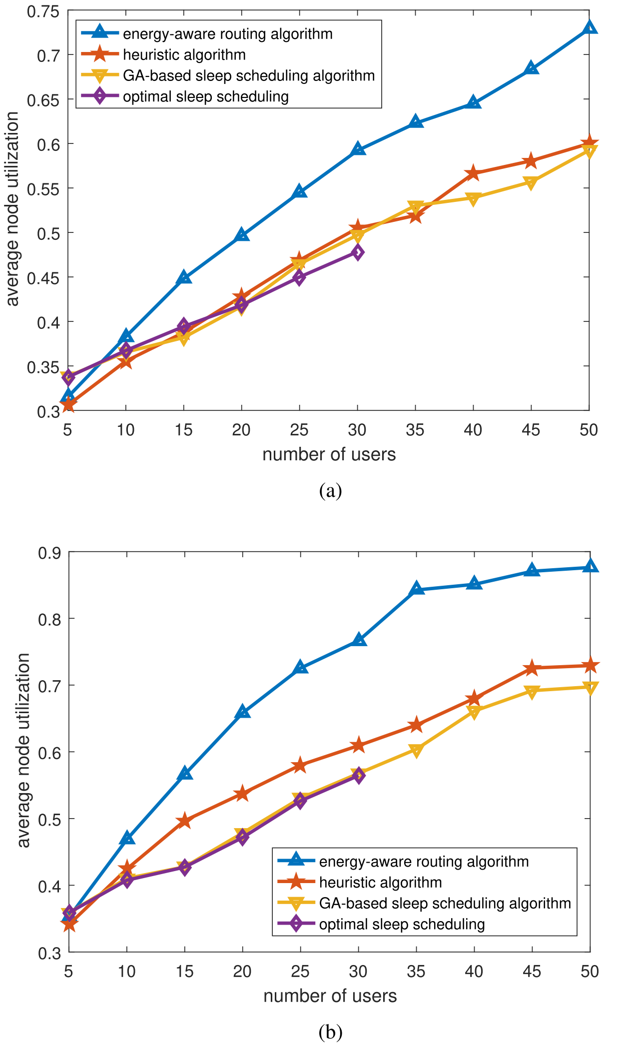



**FIGURE 8.** **Average node utilization vs. the number of users. (a) Europe. (b) Atlanta.**

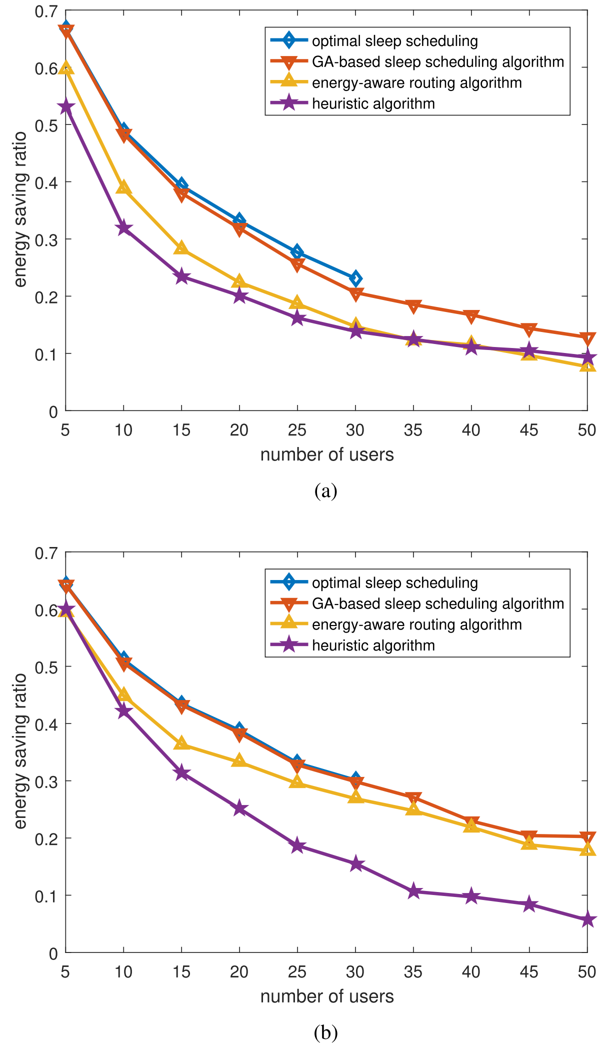



**FIGURE 9.** **Energy saving ratio vs. the number of users. (a) Europe. (b) Atlanta.**

achieves higher link saving ratio than the existing energy-aware routing algorithm in [15] and the heuristic algorithm in [25] in the network *Europe*. The link saving ratio by our GA-based algorithm is very close to that by the optimal sleep scheduling. We can also see that as the number of users increases, the link ratio decreases, since more links are needed to satisfy the users' demands. The gap of the link saving ratios between GA-based algorithm and the existing energy-aware routing algorithm becomes increasingly obvious as the number of users increases. The more users using the network service, the more chances that users' demands will be gathered into some of the links, which leads to a high link saving ratio in our proposed GA-based algorithm. We can have a similar observation in network *Atlanta*, as shown in Fig. 5(b). In addition, our GA-based algorithm achieves a higher link saving ratio in network *Europe* than in network *Atlanta*. It is the network size that causes this difference. The small-sized network *Atlanta* has less alternative paths for users to route their traffic. Most of links have to be activated to satisfy the users' demands. We can observe a similar result for the node saving ratio in Fig. 6. The heuristic algorithm performs worse in the node saving ratio than the link saving ratio, since the heuristic algorithm considers only the energy consumption in links.

Activating less links results in a high link utilization in serving the same traffic demands. As shown in Fig. 7, the GA-based algorithm has a similar average link utilization with the optimal sleep scheduling scheme in both networks *Europe* and *Atlanta*. The existing energy-aware routing algorithm has the lowest link utilization since it activates more links in the network to serve the traffic as shown in Fig. 5(a). The heuristic algorithm which considers links achieves the highest link utilization. The reason exists in that the heuristic algorithm may activate links which have small capacity and have small energy consumption. Fig. 8 shows that our proposed GA-based algorithm also achieves a similar node utilization with the optimal sleep scheduling scheme in both networks *Europe* and *Atlanta*, since less nodes are activated. Both the link and node utilizations are increased with the number of users, which stands for the traffic loads in the network.

Fig. 9 shows the energy saving ratio of the four algorithms both in network *Europe* and in network *Atlanta*. Our proposed GA-based algorithm outperforms the existing energy-aware routing algorithm and the heuristic algorithm in terms of energy saving ratio. Our proposed GA-based algorithm has achieved almost the same energy saving ratio as the optimal sleep scheduling solution in the network *Atlanta*. In the large-size network *Europe*, our proposed GA-based algorithm has still achieved a near-optimal energy saving ratio that closely approaches the optimal sleep scheduling solution. Although the heuristic algorithm achieves a higher link saving ratio, it has lower energy saving ratio than the energy-aware routing algorithm, since the energy-aware routing algorithm has lower node saving ratio.

## V. CONCLUSION

In this paper we have presented a sleep scheduling mechanism in software-defined networks (SDNs) to achieve energy efficient networking. The network devices are the main reason for energy consumption. Therefore, it can save energy if the network aggregates users' demands into fewer network devices and put the idle ones into asleep. We have formulated the sleep scheduling from the perspective of routing, which can re-regulate the traffic load distribution in the network. We have also proposed a GA-based algorithm to find out an acceptable routing strategy, according to which the controller can construct flow tables to ensure flow forwarding and put idle network devices into asleep. Our simulations carried out on the network scenarios from SNDlib have shown that our proposed GA-based sleep scheduling algorithm performs better than the existing energy-aware routing algorithm and the heuristic algorithm.

## REFERENCES


[1] A. P. Bianzino, C. Chaudet, D. Rossi, and J.-L. Rougier, "A survey of green networking research," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 1, pp. 3–20, 1st Quart., 2012.

[2] P. Gandotra and R. K. Jha, "A survey on green communication and security challenges in 5G wireless communication networks," *J. Netw. Comput. Appl.*, vol. 96, pp. 39–61, Oct. 2017.

[3] V. Venkatachalam and M. Franz, "Power reduction techniques for microprocessor systems," *ACM Comput. Surv.*, vol. 37, no. 3, pp. 195–237, 2005.

[4] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *Proc. NSDI*, vol. 8, 2008, pp. 323–336.

[5] D. Kliazovich, P. Bouvry, and S. U. Khan, "DENS: Data center energy-efficient network-aware scheduling," *Cluster Comput.*, vol. 16, no. 1, pp. 65–75, 2013.

[6] C. Natalino, L. Chiaraviglio, F. Idzikowski, L. Wosinska, and P. Monti, "Joint optimization of failure management costs, electricity costs, and operator revenue in optical core networks," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 1, pp. 291–304, Mar. 2018.

[7] L. Venturino, A. Zappone, C. Risi, and S. Buzzi, "Energy-efficient scheduling and power allocation in downlink OFDMA networks with base station coordination," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 1–14, Jan. 2015.

[8] W. Dargie, "Dynamic power management in wireless sensor networks: State-of-the-art," *IEEE Sensors J.*, vol. 12, no. 5, pp. 1518–1528, May 2012.

[9] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A power benchmarking framework for network devices," in *Proc. NETWORKING*, Aachen, Germany: Springer, 2009, pp. 795–808.

[10] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *Proc. IEEE INFOCOM,* Phoenix, AZ, USA, Apr. 2008.

[11] F. Dabaghi, Z. Movahedi, and R. Langar, "A survey on green routing protocols using sleep-scheduling in wired networks," *J. Netw. Comput. Appl.*, vol. 77, pp. 106–122, Jan. 2017.

[12] M. Gupta and S. Singh, "Greening of the Internet," in *Proc. ACM SIGCOMM,* Karlscuhe, Germany, 2003, pp. 19–26.

[13] R. Bolla, F. Davoli, R. Bruschi, K. Christensen, F. Cucchietti, and S. Singh, "The potential impact of green technologies in next-generation wireline networks: Is there room for energy saving optimization?" *IEEE Commun. Mag.*, vol. 49, no. 8, pp. 80–86, Aug. 2011.

[14] Q. Guan, T. Chen, S. Jiang, F. Ji, and F. Chen, "User-network cooperation-based sleep scheduling for communication networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3313–3325, Dec. 2016.

[15] D. Coudert, A. Kodjo, and T. K. Phan, "Robust energy-aware routing with redundancy elimination," *Comput. Oper. Res.*, vol. 64, pp. 71–85, Dec. 2015.

[16] F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut, "Minimization of network power consumption with redundancy elimination," *Comput. Commun.*, vol. 59, pp. 98–105, Mar. 2015.

[17] S. Jo, Y. Kim, H. Lee, J. Kangasharju, and M. Mülhäuser, "Novel packet switching for green IP networks," *Electron. Telecommun. Res. Inst. J.*, vol. 39, no. 2, pp. 275–283, 2017.

[18] S. Jain *et al.*, "B4: Experience with a globally-deployed software defined WAN," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, 2013.

[19] J. Rexford, "Route optimization in IP networks," in *Handbook of Optimization in Telecommunications*, Springer, 2006, pp. 679–700.

[20] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 114–119, Feb. 2013.

[21] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 27–51, 1st Quart., 2014.

[22] H. Zhu, X. Liao, C. de Laat, and P. Grosso, "Joint flow routing-scheduling for energy efficient software defined data center networks: A prototype of energy-aware network management platform," *J. Netw. Comput. Appl.*, vol. 63, pp. 110–124, Mar. 2016.

[23] F. Giroire, N. Huin, J. Moulierac, and T. K. Phan, "Energy-aware routing in software-defined network using compression," *Comput. J.*, 2018, doi: 10.1093/comjnl/bxy029.

[24] H. Xu, Z. Yu, X.-Y. Li, L. Huang, C. Qian, and T. Jung, "Joint route selection and update scheduling for low-latency update in SDNs," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 3073–3087, Oct. 2017.

[25] A. Fernandez-Fernandez, C. Cervello-Pastor, and L. Ochoa-Aday, "Achieving energy efficiency: An energy-aware approach in SDN," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–7.

[26] M. Huang, W. Liang, Z. Xu, W. Xu, S. Guo, and Y. Xu, "Dynamic routing for network throughput maximization in software-defined networks," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.

[27] R. Chai, H. Li, F. Meng, and Q. Chen, "Energy consumption optimization-based joint route selection and flow allocation algorithm for software-defined networking," *Sci. China Inf. Sci.*, vol. 60, no. 4, p. 040306, 2017.

[28] B. Dengiz, F. Altiparmak, and A. E. Smith, "Local search genetic algorithm for optimal design of reliable networks," *IEEE Trans. Evol. Comput.*, vol. 1, no. 3, pp. 179–188, Sep. 1997.

[29] C.-F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 2, pp. 997–1006, Apr. 2004.

[30] A. Riedl and D. A. Schupke, "Routing optimization in IP networks utilizing additive and concave link metrics," *IEEE/ACM Trans. Netw.*, vol. 15, no. 5, pp. 1136–1148, Oct. 2007.

[31] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, "SNDlib 1.0—Survivable network design library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010.

[32] D. P. Bertsekas and R. G. Gallager, *Data Networks*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1992.

[33] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM J. Comput.*, vol. 1, no. 2, pp. 146–160, 1972.

[34] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.

[35] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[36] *Survivable Network Design Library (SNDlib)*. Accessed: Jun. 1, 2018. [Online]. Available: http://sndlib.zib.de

**WEIQI CHEN** received the B.E. degree in communication engineering from Sun Yat-sen University, Guangdong, China, in 2008, and the M.S. degree from the South China University of Technology, Guangdong, in 2011, where she is currently pursuing the Ph.D. degree with the School of Electronic and Information Engineering. Her main interests include wireless multi-hop networks and underwater acoustic sensor networks.

**HAN CHEN** received the B.S. degree in information engineering from the South China University of Technology, Guangzhou, China, in 2016, where he is currently pursuing the master's degree. His research interests include green networking, routing algorithm, and software defined network.

**QUANSHENG GUAN** (S'09–M'11–SM'17) received the B.Eng. degree in electronic engineering from the Nanjing University of Post and Telecommunications, China, in 2006, and the Ph.D. degree from the South China University of Technology (SCUT) in 2011. From 2009 to 2010, he was a Visiting Ph.D. Student with The University of British Columbia, Canada. From 2012 to 2013, he was a Post-Doctoral Researcher with The Chinese University of Hong Kong. He was a Visiting Scholar with the Singapore University of Technology and Design in 2013. He was also a Visiting Professor with Polytech Nantes, France, in 2016. He is currently a Professor with the School of Electronic and Information Engineering, SCUT. His research interests include wireless communications and networking, and networked interactions and economics.

His main research interests are in the areas of wireless networks, underwater acoustic networks, and network games and economics. He is a TPC member of conferences, and a reviewer for journals and conferences. He was a co-recipient of Best Paper Awards from IEEE ICCC 2014 and IEEE ICNC 2016. He is a guest editor for mobile information system, an Associate Editor for the IEEE Access and *International Journal of Distributed Sensor Networks*.

**FEI JI** received the B.S. degree in applied electronic technologies from Northwestern Polytechnical University, Xi'an, China, in 1992, and the M.S. degree in bioelectronics and the Ph.D. degree in circuits and systems from the South China University of Technology, Guangzhou, China, in 1995 and 1998, respectively. She was a Visiting Scholar with the University of Waterloo, Canada, from 2009 to 2010. She was with the City University of Hong Kong, as a Research Assistant from 2001 to 2002 and a Senior Research Associate in 2005. She is currently a Professor with the School of Electronic and Information Engineering, South China University of Technology. Her research focuses on wireless communication systems and networking. She was the Registration Chair and the Technical Program Committee Member of the IEEE 2008 International Conference on Communication System.

**BINGYI GUO** received the B.Eng. degree in electronic engineering from the China University of Mining and Technology, Xuzhou, China, in 2007 and the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 2014. From 2012 to 2013, he was a Visiting Ph.D. Student with The University of British Columbia, Vancouver, BC, Canada. From 2014 to 2016, he was with Huawei, Shenzhen, China, where he involved in the research and development of future network systems. He is currently a Lecturer with the Shandong Normal University. His research interests include green communications and incentive mechanism data sharing.