

Genetic Algorithm based Resource Management for Cognitive Mesh Networks with Real-time and Non-real-time Services

Hangguan Shan¹, Ziyun Ye¹, Yuanguo Bi² and Aiping Huang¹

¹Department of Information Science and Electronic Engineering, Zhejiang University
Hangzhou, China

[e-mail: {hshan, 3110101549, aiping.huang}@zju.edu.cn]

²College of Information Science and Engineering, Northeastern University
Shenyang, China

[e-mail: biyuanguo@ise.neu.edu.cn]

*Corresponding author: Hangguan Shan

*Received January 19, 2015; revised May 19, 2015; accepted June 18, 2015;
published August 31, 2015*

Abstract

Quality-of-service (QoS) provisioning for a cognitive mesh network (CMN) with heterogeneous services has become a challenging area of research in recent days. Considering both real-time (RT) and non-real-time (NRT) traffic in a multihop CMN, [1] studied cross-layer resource management, including joint access control, route selection, and resource allocation. Due to the complexity of the formulated resource allocation problems, which are mixed-integer non-linear programming, a low-complexity yet efficient algorithm was proposed there to approximately solve the formulated optimization problems. In contrast, in this work, we present an application of genetic algorithm (GA) to re-address the hard resource allocation problems studied in [1]. Novel initialization, selection, crossover, and mutation operations are designed such that solutions with enough randomness can be generated and converge with as less number of attempts as possible, thus improving the efficiency of the algorithm effectively. Simulation results show the effectiveness of the newly proposed GA-based algorithm. Furthermore, by comparing the performance of the newly proposed algorithm with the one proposed in [1], more insights have been obtained in terms of the tradeoff among QoS provisioning for RT traffic, throughput maximization for NRT traffic, and time complexity of an algorithm for resource allocation in a multihop network such as CMN.

Keywords: Cognitive mesh network, heterogeneous traffic, resource allocation, genetic algorithm

This work was supported by the National Hi-Tech R\&D Program of China (No. 2014AA01A702), the National Key Basic Research Program of China (No. 2012CB316104), the National Natural Science Foundation of China (No. 61201228), the National Science and Technology Specific Major Projects (No. 2015ZX03001031), the Specialized Research Fund for the Doctoral Program of Higher Education (No. 20120101120077), and the Fundamental Research Funds for the Central Universities (No. 2015XZZX001-02). We express our thanks to Dr. Li Sun who checked our manuscript.

1. Introduction

To accommodate the explosively growing data traffic in a large coverage area, cognitive radio based wireless mesh network or cognitive mesh network (CMN) has been considered as one of the most promising solutions [1-3]. By using cognitive radio, CMN is capable of utilizing spectrum resource from primary spectrum such as TV-band spectrum. However, to protect the transmission of primary users, a cognitive radio shall switch among primary channels if a primary user returns to the channel it uses, leading to time-varying capacity of a cognitive radio-based network [4]. As such, quality-of-service (QoS) provisioning for such a network is a challenging issue.

In the literature, there exist a number of research works studying resource allocation and/or QoS provisioning for cognitive radio networks (CRNs). [5] studied joint spectrum allocation, scheduling, and flow routing such that the required network-wide radio spectrum in a CRN is minimized. By integrating per-node-based power control, the algorithm proposed in [6] can minimize the network-wide bandwidth-footprint-product (BFP) of a CRN. With physical layer signal-to-interference-and-noise-ratio (SINR) model, [7] studied capacity maximization for CRNs by jointly optimizing power control, scheduling, and flow routing. However, the heterogeneous services which can coexist in a CMN and the support of their differentiated QoS requirements were not taken into account in the aforesaid works. In contrast, QoS provisioning for heterogeneous services in CRNs was studied in some recent works (e.g., [8-10]). In [8], QoS differentiation between real-time (RT) and non-real-time (NRT) secondary users (SUs) was achieved at the call level when allocating channel between the two types of users. Studied in [9] and [10] were joint channel and power allocation with the objective of maximizing network capacity while providing minimum rate guarantee for SUs with RT traffic and fair rate sharing among SUs with NRT traffic. However, as these works mainly focus on single-hop CRNs, directly applying these schemes in a multihop CMN can be ineffective or inefficient.

Differently, by considering heterogeneous traffic consisted of both RT and NRT traffic in a multihop setting of CMN, studied in [1] were cross-layer resource management, including joint access control, route selection, and channel and power allocation. Mathematically, the cross-layer resource allocation problem studied in [1] was formulated as three optimization sub-problems. The first one was to select route requested by any new flow, with the objective of choosing a route with maximal expected transmission rate, and it was addressed by the shortest-path-based algorithm. The other two were on joint channel and power allocation, which, respectively, were for RT flows with the objective of minimizing consumed network resources but serving as many RT flows as possible and for NRT flows with the objective of maximizing the total background throughput. Both were formulated as interrelated NP-hard mixed-integer nonlinear programming (MINLP) problems. The access control for RT flows was implemented accompanied with route selection and channel and power allocation. By using a two-level decoupling approach, a low-complexity yet efficient algorithm was proposed to approximately solve the high-complexity MINLP problems. Simulation results show the effectiveness of the proposed algorithm in terms of improving both the RT flow's mean access ratio and the NRT flow's mean throughput. However, due to the complexity of the formulated channel and power allocation problems, the algorithm proposed in previous works [1] failed to show the optimal performance of a CMN when supporting both RT and NRT traffic.

To have an in-depth understanding of the performance of a CMN supporting heterogeneous services including both RT and NRT traffic, in this work we present an application of genetic algorithm (GA) to re-address the resource allocation problems (channel and power allocation for both RT and NRT traffic) studied in [1]. The newly proposed GA-based algorithm can be treated as the optimal solution, because GA has been proven to be effective in solving NP-hard problems [11, 12]. The main contributions and significance of this work are twofold.

First of all, we propose a novel GA-based algorithm that is tailored for the channel and power allocation for a multihop CMN with both RT and NRT traffic. Novel initialization, selection, crossover, and mutation operations are defined such that solutions with enough randomness can be generated and converge with as less number of attempts as possible, which improves the efficiency of the algorithm effectively. Apart from this, elitism strategy is integrated into the algorithm design to guarantee the performance of the proposed algorithm.

Secondly, the effectiveness of the proposed GA-based algorithm can be observed from simulation results, especially in increasing the access ratio of RT flows. Furthermore, with the GA-based algorithm, the tradeoff between QoS provisioning for RT flows and throughput maximization for NRT flows can be more obvious, because the GA-based algorithm is able to fully utilize network resources to admit as many RT flows as possible. Additionally, by comparing the performance of the newly proposed algorithm with the algorithm designed in [1], more insights have been obtained in terms of the tradeoff among QoS provisioning for RT traffic, throughput maximization for NRT traffic, and time complexity of a resource allocation algorithm.

The remainder of this paper is organized as follows. A short review of applying GA to resource allocation for wireless networks is provided in Section 2. Section 3 introduces the system model reused from [1]. The GA-based algorithm tailored for RT traffic resource allocation (named GA-RT) and that for NRT traffic resource allocation (named GA-NRT) are described in Sections 4 and 5, respectively. After presenting our simulation results in Section 6, we conclude this paper in Section 7.

2. Related Work

In this section, we present a short review of the recent works that apply GA-based algorithm to resource allocation for wireless networks. Considering a multicell scenario with nonuniform traffic distribution in multihop networks, the search for the optimal topology is an NP-hard problem. To address the aforesaid issue, [13] proposed a novel sequential genetic algorithm in which topologies were encoded as a set of chromosomes and special crossover and mutation operations were proposed. Both high performance improvements in the system and fast convergence compared with exhaustive search were shown in the numerical test. In CRNs, to mitigate the impact of spectrum usage termination and switching, minimum interference robust topology construction (MIRTC) is a critical problem. In [14], MIRTC was formulated as an integer programming problem and was addressed by a genetic-algorithm-based channel assignment (GACA) scheme, which could construct a robust CR topology while minimizing interference. In [15], based on genetic algorithm an adaptive intelligent task mapping together with a scheduling scheme was proposed to provide a real-time guarantee for parallel processing in a multihop wireless network. Furthermore, to alleviate power scarcity of a multihop wireless network, a hybrid fitness function was devised and embedded in the algorithm to extend the overall network lifetime. Studied in [16] was NP-hard reinforced

backbone network deployment problem, with the practical consideration of the limitation of available backbone nodes and the enforcement of backbone network connectivity. In that work, GA was treated as the optimal solution to find the performance bound of the proposed heuristic algorithm. In [17], with a goal of maximizing the service provider's revenue taking into consideration user churn behavior, the authors studied power constrained discrete rate allocation (PC-DRA) problem in a CDMA data network. Due to its proved NP-Complete complexity, two GA-based algorithms were proposed to solve the problem. Considering a cochannel deployment of femtocells in a macrocell network, [18] formulated a multiobjective optimization problem with the objectives of maximizing the throughput of all users and increasing the power efficiency of femtocell base stations. The authors solved the problem by using nondominated sorting genetic algorithm version II. In [19], GA-based approach was combined with Karush-Kuhn-Tucker (KKT)-driven approach to solve the joint power-subcarrier-time intra-cluster resource allocation problem in wireless mesh networks, which was proved as an NP-hard problem.

3. System Model

In a multihop CMN as considered in [1], heterogeneous traffic is characterized by two classes of coexisting traffic, namely real-time traffic and non-real-time traffic. The network is aimed at supporting QoS provisioning and service differentiation for both types of traffic. The important symbols defined in [1] and reused in this paper are summarized in Table 1.

Table 1. Summary of Important Symbols

Symbol	Definition	Symbol	Definition
N	Number of nodes	M	Number of channel
Ω_{RT} (Θ_{RT})	Set of nodes (links) transmitting RT flows	Ω_{NRT} (Θ_{NRT})	Set of nodes (links) transmitting NRT flows
\mathbf{x}_{RT} (\mathbf{x}_{NRT})	Channel allocation matrix $\mathbf{x}_{RT} = [\tilde{x}_{i,j}^{m,l}]_{ \Omega_{RT} \times \Omega_{RT} \times M \times \Lambda_{RT} }$ $(\mathbf{x}_{NRT} = [\hat{x}_{i,j}^{m,l}]_{ \Omega_{NRT} \times \Omega_{NRT} \times M \times \Lambda_{NRT} })$ where $\tilde{x}_{i,j}^{m,l} = 1$ ($\hat{x}_{i,j}^{m,l} = 1$) means that channel m of link (i, j) is allocated to RT (NRT) flow and 0 otherwise	\mathbf{p}_{RT} (\mathbf{p}_{NRT})	Power allocation vector $\mathbf{p}_{RT} = [\tilde{p}_n^m]_{ \Omega_{RT} \times M}$ $(\mathbf{p}_{NRT} = [\hat{p}_n^m]_{ \Omega_{NRT} \times M})$ denoting the power allocated to RT (NRT) flows on each channel of nodes in Ω_{RT} (Ω_{NRT})
P_{\max}	Maximum transmit power	Γ_i	Set of available channels at node i
Λ_{RT} (Λ_{NRT})	Set of RT (NRT) traffic flows, including both the ongoing and newly arrived ones	$\Gamma_{i,j}$	Set of commonly available channels between nodes i and j
Δ	Interference cushion at a receiver of RT flow or the tolerable interference caused by nodes in Ω_{NRT}	$\hat{\Gamma}_i$	Set of channels allocatable to NRT flows at node i after resource allocation for RT flows
$g_{i,j}^m$	Channel gain from node i to node j on channel m	\hat{P}_i	Power allocatable for NRT flows at node i after resource allocation for RT flows

$r_{i,j}^m(\gamma)$	Mean effective transmission rate of channel m of link (i, j) given received SINR γ	$\tilde{\gamma}_{i,j}^m(\mathbf{p}_{RT})$	The received SINR on channel m of link (i, j) given RT flow's power allocation vector \mathbf{p}_{RT}
$R(l)$	Minimum rate requirement for RT flow l	$\hat{\gamma}_{i,j}^m(\mathbf{p}_{NRT})$	The received SINR on channel m of link (i, j) given NRT flow's power allocation vector \mathbf{p}_{NRT}
$r(l)$	NRT flow l 's transmission rate	ρ_l	Route path of flow l
Ψ	Rate vector of NRT flows $\Psi = [r(l)]_{1 \times \Lambda_{NRT} }$	$C_{i,j}(\mathbf{x}_{RT}, \mathbf{p}_{RT}, \Lambda_{RT})$	A function representing the total radio resource link (i, j) consumed for RT flows, given channel allocation matrix \mathbf{x}_{RT} , power allocation vector \mathbf{p}_{RT} , and set of RT traffic flows Λ_{RT}

In this paper, we consider slot-based resource allocation, integrating functions, including joint access control, and channel and power allocation, via the following two optimization problems. For more details, interested readers are suggested to refer to [1]. Specifically, RT traffic is allocated channel and power resource ahead of NRT traffic, due to its high priority. For the RT flows, we try to serve as many RT flows as possible. If there are too many RT flows as compared with the amount of network resources, we adopt the strategy of first-in-first-served (FIFS) which means the RT flow that arrived last is ignored and so on. Once the network resources are sufficient for RT flows, we try to minimize network-wide resources consumed by RT flows while guaranteeing their data rates, thus leaving as many network resources as possible for NRT flows to boost background throughput. Resource allocation for RT flows is formulated as:

$$\text{OP1: } \min_{\mathbf{x}_{RT}, \mathbf{p}_{RT}, \Lambda_{RT}} f(\mathbf{x}_{RT}, \mathbf{p}_{RT}, \Lambda_{RT}) = \sum_{(i,j) \in \Theta_{RT}} C_{i,j}(\mathbf{x}_{RT}, \mathbf{p}_{RT}, \Lambda_{RT}) \quad (1a)$$

$$\text{s.t. } \sum_{l_1 \in \Lambda_{RT}}^{k \in \Omega_{RT}} \tilde{x}_{k,n}^{m,l_1} + \sum_{l_2 \in \Lambda_{RT}}^{j \in \Omega_{RT}} \tilde{x}_{n,j}^{m,l_2} \leq 1, \forall n \in \Omega_{RT}, \forall m \in \Gamma_n \quad (1b)$$

$$\sum_{m \in \Gamma_{i,j}} r_{i,j}^m(\tilde{\gamma}_{i,j}^m(\mathbf{p}_{RT})) \tilde{x}_{i,j}^{m,l} \geq R(l), \forall l \in \Lambda_{RT}, \forall (i, j) \in \rho_l \quad (1c)$$

$$\sum_{m \in \Gamma_n} \tilde{p}_n^m \leq P_{\max}, \forall n \in \Omega_{RT} \quad (1d)$$

$$\tilde{x}_{i,j}^{m,l} \in \{0, 1\}, \tilde{p}_n^m \geq 0, \forall (i, j) \in \Theta_{RT}, \forall n \in \Omega_{RT}, \forall l \in \Lambda_{RT}, \forall m \in \Gamma_{i,j} \quad (1e)$$

where (1a) is the cost function representing the total radio resources, including channel \mathbf{x}_{RT} and power \mathbf{p}_{RT} , consumed by the RT flows in Λ_{RT} , (1b) describes that any channel of a node can be used to either transmit or receive one flow from one neighboring node, (1c) means that the actual flow rate must be greater than the minimum rate requirement $R(l)$ for this RT flow, with $\tilde{\gamma}_{i,j}^m(\mathbf{p}_{RT})$ representing the received SINR (taking the maximum interference from the NRT flows into account) on channel m of link (i, j) given RT flows' power allocation \mathbf{p}_{RT} and $r_{i,j}^m(\gamma)$ the mean effective transmission rate of the same channel given received SINR γ (for more details, see [1]), (1d) is the power constraint which means that the total power at a node is limited, and (1e) gives the range of each optimization variable.

Due to the fact that the NRT flows are assigned with a low priority, we only allocate remaining allocatable channel and power to the NRT flows. When allocating resources to the

NRT flows, we adopt the strategy of best effort, which means that we try to use up all remaining resources to boost the overall throughput of the NRT flows. With the objective of maximizing throughput, we formulate the optimization problem for the NRT flows as:

$$\text{OP2: } \max_{\mathbf{x}_{NRT}, \mathbf{p}_{NRT}, \Psi} h(\mathbf{x}_{NRT}, \mathbf{p}_{NRT}, \Psi) = \sum_{l \in \Lambda_{NRT}} r(l) \quad (2a)$$

$$\text{s.t. } \sum_{l_1 \in \Lambda_{NRT}} \hat{x}_{k,n}^{m,l_1} + \sum_{l_2 \in \Lambda_{NRT}} \hat{x}_{n,j}^{m,l_2} \leq 1, \quad \forall n \in \Omega_{NRT}, \quad \forall m \in \hat{\Gamma}_n \quad (2b)$$

$$\sum_{m \in \hat{\Gamma}_{i,j}} r_{i,j}^m(\hat{\gamma}_{i,j}^m(\mathbf{p}_{NRT})) \hat{x}_{i,j}^{m,l} \geq r(l), \quad \forall l \in \Lambda_{NRT}, \quad \forall (i,j) \in \rho_l \quad (2c)$$

$$\sum_{m \in \hat{\Gamma}_n} \hat{p}_n^m \leq \hat{P}_n, \quad \forall n \in \Omega_{NRT} \quad (2d)$$

$$\sum_{u \in \Omega_{NRT}^m} g_{u,n}^m \hat{p}_u^m \leq \Delta, \quad \forall n \in \Omega_{RT}, \quad \forall m \in \Gamma_n \setminus \hat{\Gamma}_n \quad (2e)$$

$$\hat{x}_{i,j}^{m,l} \in \{0,1\}, \quad \hat{p}_n^m \geq 0, \quad r(l) \geq 0 \quad (2f)$$

$$\forall (i,j) \in \Theta_{NRT}, \quad \forall n \in \Omega_{NRT}, \quad \forall l \in \Lambda_{NRT}, \quad \forall m \in \hat{\Gamma}_{i,j}$$

where \mathbf{x}_{NRT} , \mathbf{p}_{NRT} , and $\Psi = [r(l)]_{l \in \Lambda_{NRT}}$ are optimization variables, corresponding to channel allocation, power allocation, and end-to-end flow transmission rate, respectively, with $r(l)$ being the transmission rate of NRT flow l and defined as the lowest rate on all hops of flow l , (2b) is the channel allocation constraint implying that any channel of a node can be used to either transmit or receive one traffic flow from one neighboring node, (2c) means that the transmission rate over each hop of an NRT flow must be no less than the flow's end-to-end transmission rate, (2d) is the power constraint which means that the total power at a node is limited to the remaining power after the allocation for the RT flows, (2e) denotes that the interference from the NRT flows to any RT flow cannot be greater than the tolerable interference Δ (i.e., the interference cushion), (2f) gives the range of each optimization variable. Although here we adopt a two-step approach to decouple the resource allocation for the two types of coexisting RT and NRT traffic, resource allocation for each of them is interrelated with each other through interference cushion involved constraints (1c) and (2e).

Since $\tilde{x}_{i,j}^{m,l}$ and (1c) in OP1 are integer variable and nonlinear constraint (for more details, see [1]), respectively, OP1 is a mixed-integer nonlinear programming (MINLP), which is NP-hard in general [20]. Similarly, we can find that OP2 is also a MINLP problem. Since GA has been shown to be an effective way in finding the global optimal solution [11, 12], we will present the application of GA in seek of the solution that can achieve the optimum, i.e., the performance bound of the network.

4. GA-based RT Traffic Resource Management

Due to the high priority, we first allocate resources to RT traffic. Accordingly, in this section, a GA-based algorithm (named GA-RT) is firstly proposed to solve the RT traffic resource allocation problem formulated as OP1. It starts with an initial population of individuals characterized by chromosomes, and then improves the quality of the population through evolution. In each generation, three operations are carried out one by one to yield a new population, which are: 1) selection; 2) crossover; and 3) mutation. Due to the multiple complex constraints of the optimization problem, novel approaches are designed to facilitate the implementation of the proposed GA-based algorithm. In the following, we first explain the encoding scheme on which the algorithm is based, and then introduce our design for each of the aforesaid operations.

A. Encoding Scheme

Before the resource allocation, the gateway node which plays the role of resource manager, needs to collect sensing results and traffic demands in the network [1]. The results about all n_l available links in the network in current slot are recorded in a matrix *chromosomes_format_info* (see Fig. 1). Specifically, the information on each link, namely, the sender and the receiver of this link, and the number of available channels and that of flows allocated over this link are recorded in *sender*, *receiver*, *available_chl_num*, and *flow_num*, respectively. Besides, the index of available channels and that of flows over each link are recorded in matrices *available_chl_idx* and *flow_id*, respectively. The collected information facilitates the following coding for chromosomes.

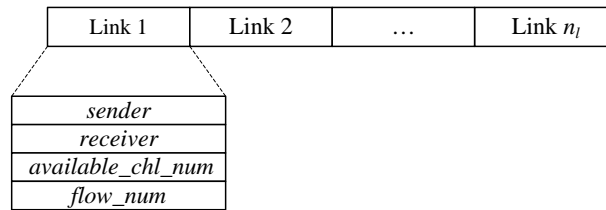


Fig. 1. The structure of *chromosomes_format_info*

A chromosome or an individual of the population corresponds to a channel-power allocation that needs to be optimized. It is coded as a vector shown in Fig. 2, where n_{chl}^i denotes the number of available channels over available link i . Moreover, each available channel of a link should be allocated to at most one single flow with constrained transmit power for the sender.

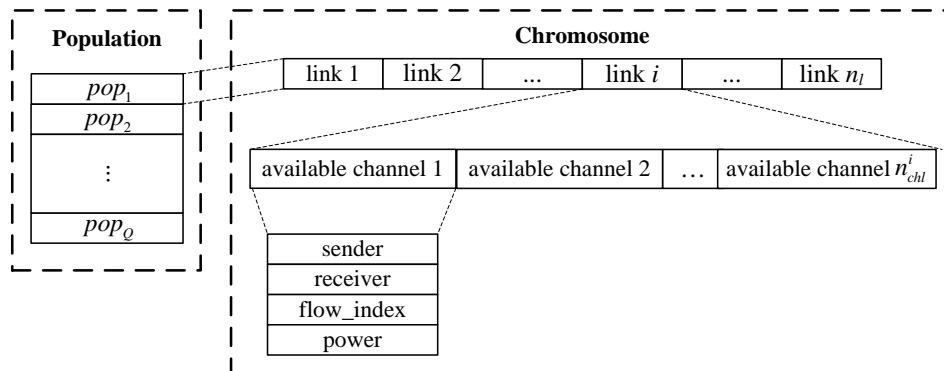


Fig. 2. A genetic representation of a set of feasible solutions to the optimization problem

B. Initialization

To start the GA we need to set up an initial population of a number of chromosomes. In Fig. 2, Q chromosomes are generated. For OPI, the initialization process includes both channel and power allocations. Normally, both the allocations in an initial solution should be generated randomly. But, in this case, due to the channel constraint (1b) and the rate requirement (1c), total randomness causes the initialization to fail frequently, leading to many attempts, which

reduces the efficiency of the algorithm significantly. To address this issue some novel strategies are adopted when allocating channel and power.

Considering the interference between RT flows, first we finish channel allocation (i.e., flow assignment) for all links, and then allocate power to all channels with flow assignment. To ensure sufficient randomness, we allocate channel to different chromosomes in a random order of links. Further, for any chromosome, when allocating channel, we record the used channels at each node, and only allocate unused channels in order to satisfy (1b). For any link (say link i), a necessary condition for a successful channel allocation is $n_{flow}^i \leq \xi_{chl}^i$ where n_{flow}^i and ξ_{chl}^i ($\leq n_{chl}^i$) denote the number of flows and the number of allocatable channels on link i , respectively. Specially, if $n_{flow}^i > n_{chl}^i$, the channels are definitely insufficient and the RT flow that arrived last should be suspended. Besides, to support all the flows on link i , each flow is allocated a channel firstly. For the remaining unused channels on link i , we allocate each of them randomly to one of the flows on the link, with probability p_{init} . Accordingly, the probability that this channel is not used is $(1 - p_{init})$.

When the channel allocation is successful, we allocate power to the channels in use. Due to the rate requirement (1c), we find that, in contrast to random power allocation, equal power allocation is more likely to satisfy (1c). Thus we count the number of times each node is used as a sender to transmit RT flows and equally allocate power to the channels that have the same sender, so that (1d) is satisfied. However, if equal power allocation meets requirement (1c), to generate sufficient randomness we reduce the power used on each channel by a random value up to d_1 (in percent) which is preset, otherwise we regard power allocation as failed and try to reallocate channel again and begin to delete RT flows after certain number of attempts. We keep reducing the power for at most N_p times until (1c) is unsatisfied and use the latest feasible result, because using as less power for RT flows as possible can reserve more available resources for NRT flows in general.

The initialization stops if the total Q chromosomes are generated. If some chromosomes are failed to be generated after a sufficient number of generation attempts but at least one chromosome has been initiated, we can randomly duplicate the required chromosomes from those that have been initiated. It should be noticed that, if the network resources are too limited, OPI can be infeasible and therefore it is possible that no chromosome can be initiated in this case. The algorithm designed for initialization is summarized in Algorithm 1, which consists of a triple-nested loop (see Lines 3, 5, 30). Notice that the outmost loop at Line 30 corresponds to the case that we apply FIFO to reduce the number of RT flows due to the limited amount of network resources. Besides, in the innermost loop, three sequential subloops are enclosed, for flow assignment (Line 6), equal power allocation to all channels with flow assignment at any nodes (Line 18), and power reduction attempt (Line 22), respectively. Therefore, the time complexity of the algorithm is on the order of $O(N_t N_c (n_l + NM + N_p) |\Lambda_{RT}|)$, where N_t , N_c , N , and M are the maximum number of generation attempts (defined in Line 3), the maximum number of channel allocation attempts (defined in Line 5), the number of nodes and that of channels in the network, respectively.

Algorithm 1: Initialization in GA-RT

```

1  Input: chromosomes_format_info, available_chl_idx, flow_id, Q, Nt, Nc, Np
2  Initiate  $q = 0$ ;
3  For generation_attempt = 1 to Nt // Nt=max_attempt
4    Set chl_alt_attempt = 1 and generate a random order of links  $\pi(i)$ ;
5    While chl_alt_attempt  $\leq N_c$  // Nc=max_chl_alt_num
6      For link  $\pi(i) = 1$  to  $n_l$ 
7        Identify the number and index of the allocatable channels on link  $i$ ;
8        If  $n_{flow}^i \leq \xi_{chl}^i$ 
9          Randomly allocate each flow one basic channel;
10         If there are still rest unused channels, allocate each of them randomly to one
            of the flows on the link, with probability  $p_{init}$ ;
11         Else if  $n_{flow}^i > n_{chl}^i$ 
12           Suspend the last arrived RT flow and go to Line 3 to restart initialization;
13         Else
14           Fail to generate a solution and  $chl\_alt\_attempt = chl\_alt\_attempt + 1$ ;
15           Break and go to line 5 to reallocate channel for the solution;
16         End
17       End
18     For a solution with successful channel allocation, first try equal power allocation to
        all channels with flow assignment at any nodes;
19     If equal power allocation dissatisfies (1c)
20       Fail to generate a solution and  $chl\_alt\_attempt = chl\_alt\_attempt + 1$ ;
21     Else
22       Keep reducing the power used on each channel by a random value up to  $d_1$  until
        (1c) dissatisfies or power has been reduced for  $N_p$  times;
23       Allocate each channel the latest feasible power result, update  $q = q + 1$ ;
24       Break;
25     End
26   End
27   Break if  $q = Q$ ;
28 End
29 If no solution generates
30   Suspend the last arrived RT flow and go to Line 3 to restart initialization;
31 Else if  $q < Q$ 
32   Randomly duplicate the required solution(s) from those that have been initiated;
33 End
34 Output:  $\{pop_i\}_{i=1}^Q$ 

```

C. Fitness Function and Selection Operation

During each successive generation, chromosomes that are fit enough are selected to survive and to breed a new generation. The well-known Roulette-Wheel scheme [21] is used here, where each solution is picked with the probability related to a fitness function of the current solution. However, as the Roulette-Wheel scheme cannot be directly applied to a minimization problem such as OP1 and negative fitness values are not allowed (thus simply maximizing the negative value of (1a) is invalid), it is necessary to map the objective function (1a) to an appropriate fitness function form. The following transformation is considered [22]

$$F(\mathbf{x}_{RT}, \mathbf{p}_{RT}, \Lambda_{RT}) = \begin{cases} C_{\max} - f(\mathbf{x}_{RT}, \mathbf{p}_{RT}, \Lambda_{RT}), & f(\mathbf{x}_{RT}, \mathbf{p}_{RT}, \Lambda_{RT}) < C_{\max} \\ 0, & f(\mathbf{x}_{RT}, \mathbf{p}_{RT}, \Lambda_{RT}) \geq C_{\max} \end{cases} \quad (3)$$

where C_{\max} is the maximum value of the objective function (1a) till current generation. More specifically, we define the objective function $f(\mathbf{x}_{RT}, \mathbf{p}_{RT}, \Lambda_{RT})$ in OP1 as:

$$f(\mathbf{x}_{RT}, \mathbf{p}_{RT}, \Lambda_{RT}) = \sum_{i,j,m,l} \tilde{x}_{i,j}^{m,l} + \sum_{n,m} \tilde{p}_n^m. \quad (4)$$

After fitness calculation for the Q solutions, the fitness function of pop_i is recorded as F_i . Following elitism strategy [23], we reserve the best chromosome satisfying $pop_{best} = \arg \max_{1 \leq i \leq Q} F_i$ for the next generation directly. To improve the performance of the GA, the best chromosome will be kept throughout this generation. For the other $(Q-1)$ solutions in the next generation, we randomly pick them from the current generation. According to the Roulette-Wheel scheme, the probability that pop_i in the current generation is picked is

$$\Pr(pop_i) = \frac{F_i}{\sum_{i=1}^Q F_i}. \quad (5)$$

The algorithm designed for selection operation is summarized in Algorithm 2. The time complexity of the algorithm depends mainly on the operation to find the best chromosome (Line 2) and the chromosome selection between Lines 4 and 11, thus is on the order of $O(Qn_{chl} + Q^2) = O(Q(M + Q))$, where $n_{chl} = \sum_{i=1}^{n_l} n_{chl}^i$ is the number of total available channels on all links that we have to check to derive F_i for a chromosome.

Algorithm 2: Selection operation in GA-RT

- 1 Input: $\{pop_i\}_{i=1}^Q, \{F_i\}_{i=1}^Q, C_{\max}, Q$
 - 2 Find $pop_{best} = \arg \max_{i \in \{1, Q\}} F_i$ and assign $C_{\max} = \max\{C_{\max}, \max_{1 \leq i \leq Q} F_i\}$;
 - 3 Normalize F_i by $\hat{F}_i = F_i / \sum_{i=1}^Q F_i$;
 - 4 For $i = 1$ to $Q-1$
 - 5 Generate a random value $r \in [0,1]$;
 - 6 For $j = 1$ to Q
 - 7 If $\sum_{i=1}^{j-1} \hat{F}_i < r \leq \sum_{i=1}^j \hat{F}_i$
 - 8 Select pop_j to generate the next generation by $pop_i^{next} = pop_j$;
 - 9 End
 - 10 End
 - 11 End
 - 12 For $i = 1$ to $Q-1$
 - 13 Assign $pop_i = pop_i^{next}$;
 - 14 End
 - 15 Keep the best solution by $pop_Q = pop_{best}$;
 - 16 Output: $\{pop_i\}_{i=1}^Q, C_{\max}, pop_{best}$
-

D. Crossover Operation

To produce next generation, single point crossover method is applied in our algorithm. For solutions in current population, every two of adjacent ones make up a group (denoted as

$parent_1$ and $parent_2$) and go through crossover operation with probability p_{co} . At the beginning of crossover, we randomly select link k as a crossover point. Two children solutions denoted as $children_1$ and $children_2$ are generated by firstly duplicating the channel-power allocation of the links before link k from $parent_1$ and $parent_2$, respectively, i.e.,

$$children_i(link\ 1:link\ k) = parent_i(link\ 1:link\ k),\ i = 1, 2 \quad (6)$$

and then exchanging the channel allocation of the remaining parts from the two parents, i.e.,

$$flow_assignment_{children_1}(link\ k+1:link\ n_1) = flow_assignment_{parent_2}(link\ k+1:link\ n_1) \quad (7)$$

$$flow_assignment_{children_2}(link\ k+1:link\ n_1) = flow_assignment_{parent_1}(link\ k+1:link\ n_1). \quad (8)$$

Obviously, the first step of direct duplication will never violate channel constraint (1b); however, the second step is prone to make mistakes. Therefore, any violation of (1b) should be detected in the exchange process. If so, a new crossover point should be reselected.

The exchange of power allocation is implemented after a successful exchange of channel allocation. The blending method for power exchange follows the way for continuous value [23]. If the power levels allocated to $parent_1$ and $parent_2$ at the same channel (say channel j) over the same link (say link q) are $p_{parent_1}(q, j)$ and $p_{parent_2}(q, j)$, respectively, the power levels for their children at the same channel over the same link are

$$p_{children_1}(q, j) = \alpha \times p_{parent_1}(q, j) + (1 - \alpha) \times p_{parent_2}(q, j) \quad (9)$$

$$p_{children_2}(q, j) = (1 - \alpha) \times p_{parent_1}(q, j) + \alpha \times p_{parent_2}(q, j) \quad (10)$$

where $q \in \{k+1, k+2, \dots, n_1\}$, and α is a random value on the interval $[0, 1]$. However, remedy might be still needed if power constraint (1d) is violated with such power blending. If so, we can re-adjust power allocation by taking into account both the maximal power a node can use and the allocated power of the link sender,

$$p_{children_i}(q, j) = \left(P_{\max} - \sum_{u=1}^{q-1} \sum_{v=1}^{n_{ch}^u} x_{u,q} p_{children_i}(u, v) - \sum_{v=1}^{j-1} p_{children_i}(q, v) \right) \beta_i, \quad i = 1, 2 \quad (11)$$

where $x_{u,q}$ is an indicator which is equal to one if links u and v have the same transmitter and zero otherwise, and β_i is a random value on the interval $[0, 1]$.

After power allocation, we check whether or not the children solutions satisfy rate requirement (1c). If not, we can first try different power allocations for a number of attempts and then turn to change the crossover position if necessary. Finally, if all possible crossover positions are used and failed, or we skip crossover operation at the beginning (with probability $(1-p_{co})$), we duplicate parent solutions directly. But, the best chromosome is kept from crossover due to elitism strategy. The algorithm for crossover operation is summarized in Algorithm 3. For the time complexity of the algorithm, we know that: 1) for a population of size Q the crossover operation is executed $Q/2$ times (Line 2); 2) to finish channel exchange, any channel crossover that violates (1b) initiates a new crossover link point selection (Line 10), thus contributing at most $n_l M$ times of channel exchanges; and 3) to finish power exchange, any power crossover that violates (1c) initiates a new power allocation for every channel over the links after crossover link point, as long as the limit of power allocation attempts N_p is not reached (Line 14), thus generating at most $N_p n_l M$ times of power allocations. To sum up, the time complexity of Algorithm 3 is thus on the order of $O(Q/2 \cdot (n_l M + N_p n_l M)) = O(Q n_l M N_p)$.

Algorithm 3: Crossover operation in GA-RT

```

1  Input: chromosomes_format_info, flow_id,  $\{pop_i\}_{i=1}^Q$ ,  $pop_{best}$ 
2  For every two of adjacent solutions  $pop_{2i-1}$  and  $pop_{2i}$ 
3       $parent_1 = pop_{2i-1}$ ;  $parent_2 = pop_{2i}$ ;
4      Generate a random value  $r \in [0,1]$ ;
5      If  $r < p_{co}$ 
6          Select a crossover point for this parent pair and denote it as link  $k$ ;
7          Duplicate genes by  $children_i(link\ 1:link\ k) = parent_i(link\ 1:link\ k)$ ,  $i = 1, 2$ ;
8          Exchange the channel allocation by  $flow\_assignment_{children_1}(link\ k+1:link\ n_l) =$ 
            $flow\_assignment_{parent_2}(link\ k+1:link\ n_l)$  and  $flow\_assignment_{children_2}(link\ k+1:link\ n_l) =$ 
            $flow\_assignment_{parent_1}(link\ k+1:link\ n_l)$ , while check whether (1b) is violated;
9          If (1b) is violated for any channel and not all crossover points have been tested
10             Go to line 6;
11          Else if crossover still fails after trying all crossover points
12             Go to line 28;
13          End
14          For  $power\_alt\_attempt = 1$  to  $N_p$  //  $N_p = max\_power\_alt\_num$ 
15              For link  $q = k+1$  to  $n_l$ 
16                  For link  $q$ 's available channel  $j = 1$  to  $n_{chl}^q$  which has flow assignment
17                      Generate  $\alpha$  and exchange power allocation by
18                       $p_{children_1}(q, j) = \alpha \times p_{parent_1}(q, j) + (1 - \alpha) \times p_{parent_2}(q, j)$ ,
19                       $p_{children_2}(q, j) = (1 - \alpha) \times p_{parent_1}(q, j) + \alpha \times p_{parent_2}(q, j)$ ;
20                      If (1d) is violated for children  $i$ 
21                          Generate  $\beta_i$  and re-adjust power allocation by
22                          
$$p_{children_i}(q, j) = \left( P_{max} - \sum_{u=1}^{q-1} \sum_{v=1}^{n_{chl}^u} x_{u,q} P_{children_i}(u, v) - \sum_{v=1}^{j-1} P_{children_i}(q, v) \right) \beta_i$$
;
23                          End
24                          End
25                          End
26                          If both children solutions satisfy (1c)
27                              Break;
28                              End
29                              End
30                              End
31                          End
32                          If  $r \geq p_{co}$  or crossover fails // failed after testing all crossover points or trying the
                           // maximum number of power allocation attempts
                           Duplicate parent solutions directly;
33                      End
34                  End
35              End
36          End
37      End
38  Keep the best chromosome by  $pop_Q = pop_{best}$ ;
39  Output:  $\{pop_i\}_{i=1}^Q$ 

```

E. Mutation Operation

To maintain genetic diversity and increase convergence rate, mutation is performed to alter some positions or genes in chromosomes randomly. But the solution with the highest fitness (i.e., pop_{best}) is kept from mutation. For the other solutions, we alter the channel-power allocation in them with probability p_{mut} , which means that totally $\lfloor p_{mut} \times n_{chl} \times (Q-1) \rfloor$ positions are changed in mutation operation.

When mutating, we first randomly choose a mutation position. Let m_{RT} denote the index of the channel related to the selected position. We record the channel and power allocation of other positions in this solution. Under constraint (1b), if the channel at m_{RT} is not usable, we regard mutation in m_{RT} as finished. Otherwise, we check whether or not current flow on this channel can quit by setting the power over channel m_{RT} as zero and checking rate requirement (1c). If the requirement is not satisfied, we also consider this mutation as finished. Otherwise, we randomly choose one of the flows on the link including 0 which means this channel is not to be used after mutation.

Then we update power allocation. If the power over channel m_{RT} is nonzero before mutation, we keep reducing the power by a random value up to d_2 (in percent) which is preset until rate requirement (1c) is not satisfied or we have tried at most N_p times, and use the last feasible power result. However, if the channel is not used before, we firstly allocate a random part of remaining power to this node and check (1c). On condition that (1c) is unsatisfied, we again try to reduce the power by a random value up to d_2 (in percent) until we have tried at most N_p times. If the attempt fails or there is no remaining power at first, we withdraw flow assignment. The algorithm for mutation operation is described in Algorithm 4. The time complexity of the algorithm depends mainly on the number of mutation attempts (see Line 4) and that of power adjustment attempts N_p (see Lines 14 and 20) and is thus on the order of $O(\lfloor p_{mut} \times n_{chl} \times (Q-1) \rfloor N_p) = O(MQN_p)$.

Algorithm 4: Mutation operation in GA-RT

- 1 Input: *chromosomes_format_info*, *available_chl_idx*, *flow_id*, $\{pop_i\}_{i=1}^Q$;
 - 2 Find pop_{best} from $\{pop_i\}_{i=1}^Q$;
 - 3 Exclude pop_{best} from further mutation process;
 - 4 For $i = 1$ to $\lfloor p_{mut} \times n_{chl} \times (Q-1) \rfloor$
 - 5 Randomly choose a mutation position m_{RT} , and record the channel and power allocation of other positions in this solution;
 - 6 If *channel at m_{RT} is not usable under constraint (1b)*
 - 7 This mutation attempt ends;
 - 8 Else *check whether (1c) will dissatisfy if setting the power over channel m_{RT} as zero*
 - 9 If so this mutation attempt ends;
 - 10 End
 - 11 Randomly allocate one of the flows on the link (including 0) to channel m_{RT} ;
 - 12 If *a flow is assigned to channel m_{RT}*
 - 13 If *the power over channel m_{RT} is nonzero before mutation*
 - 14 Keep reducing the power by a random value up to d_2 until (1c) is unsatisfied or we have tried at most N_p times;
 - 15 Use the last feasible power result;
 - 16 Else if *the remaining power is zero*
-

```

17      Withdraw flow assignment;
18      Else allocate a random part of the remaining power to this node and check (1c);
19      If (1c) is unsatisfied
20          Try to reduce the power by a random value up to  $d_2$  for at most  $N_p$  times;
21          If (1c) continues to be unsatisfied until the preset number of attempt times
            is reached, then withdraw flow assignment;
22      End
23  End
24 End
25 End
26 Output:  $\{pop_i\}_{i=1}^Q$ 

```

To sum up the GA-RT algorithm, it is composed of an initialization operation and several iterations to improve the quality of the population. Each iteration includes three operations: selection, crossover, and mutation. As such, the time complexity of GA-RT is on the order of $O(N_i N_c (n_i + NM + N_p) |\Lambda_{RT}| + T(Q(M + Q) + Qn_i MN_p + MQN_p))$, where T is the generation number.

5. GA-based NRT Traffic Resource Management

After allocating resource to RT traffic, to solve the NRT traffic resource allocation problem formulated as OP2, we design a GA-based algorithm (named GA-NRT), similar to the one proposed for RT traffic. However, as OP2 is to maximize the overall throughput of all NRT flows and more constraints (e.g., interference constraint (2e)) appear in the optimization problem, the algorithm proposed for OP1 cannot be directly applied to OP2. In the following, we introduce our design in applying GA to solve OP2 while mainly focusing on the difference between GA-RT and GA-NRT. It is also worth mentioning that the encoding scheme for GA-RT can be reused for GA-NRT.

A. Initialization

To increase throughput of all NRT flows, in the initialization of GA-NRT we try to use up all remaining resources that are unoccupied by RT flows. Moreover, for different chromosomes, we allocate all remaining channels to NRT flows one by one in a random flow permutation order, which helps to generate sufficient randomness. When allocating these channels, in each allocation round we only assign at most one usable channel to each hop of a flow. Further, we separate disconnected NRT flows from connectable or connected NRT flows, and keep allocating channel to connectable or connected NRT flows until all of them cannot be allocated more channels. A flow will be marked as disconnected if channel allocation fails on a link of the flow before it achieves an end-to-end connection. The channels allocated to a disconnected flow should be withdrawn and reused by other flows with certain probability.

When channel allocation finishes, we do power allocation under constraints (2d) and (2e). To ensure randomness under constraint (2d), for any sending node, with probability p_{eq} we equally allocate the remaining power to the channels with NRT flow assignment, which is similar to the equal power allocation adopted in GA-RT's initialization. On the other hand, with probability $(1-p_{eq})$ we randomly allocate part of the remaining power to the channels with NRT flow assignment. Besides, if RT flows pose nonzero interference constraint on the

transmitters of NRT flows (i.e., $\Delta \neq 0$), we might need to further reduce the power allocation by taking constraint (2e) into account. Thus we can ensure that both constraints are satisfied and we use as many channel and power resources as possible yet still generate enough randomness.

B. Fitness Function and Selection Operation

Similar to GA-RT, the selection operation of GA-NRT is based on the Roulette-Wheel scheme. The only difference lies in the fitness function. Because OP2 is to maximize the total NRT traffic throughput $h(\mathbf{x}_{NRT}, \mathbf{p}_{NRT}, \Psi) = \sum_{l \in \Lambda_{NRT}} r(l)$, where $r(l)$ is the transmission rate of NRT flow l and is defined as the lowest rate on all hops of flow l , the fitness function $H(\mathbf{x}_{NRT}, \mathbf{p}_{NRT}, \Psi)$ for GA-NRT can follow the form [21]

$$H(\mathbf{x}_{NRT}, \mathbf{p}_{NRT}, \Psi) = \begin{cases} h(\mathbf{x}_{NRT}, \mathbf{p}_{NRT}, \Psi) + C_{\min}, & h(\mathbf{x}_{NRT}, \mathbf{p}_{NRT}, \Psi) + C_{\min} > 0 \\ 0, & h(\mathbf{x}_{NRT}, \mathbf{p}_{NRT}, \Psi) + C_{\min} \leq 0 \end{cases} \quad (12)$$

where C_{\min} is the minimum value of the objective function (2a) till current generation. Again, following elitism strategy, the best chromosome will be recorded and kept throughout the generation.

C. Crossover Operation

The crossover operation of GA-NRT is designed as the same for GA-RT in terms of the crossover of spectrum resource. The difference lies in the crossover of power allocation, because of the newly added interference constraint (2e). After channel crossover, the power levels for the children solutions at an available channel (say channel j) of a link (say link q) after the crossover point (say link k) are derived (denoted as $p_{children_1}(q, j)$ and $p_{children_2}(q, j)$ respectively) through (9) and (10). If $p_{children_i}(q, j)$ is too large to satisfy (2d) and (2e), we re-adjust it by

$$p_{children_i}(q, j) = \begin{cases} \hat{P}'_{t(q)} \times \delta_i, & \Delta = 0 \\ \min \left\{ \hat{P}'_{t(q)}, \Delta / \max_{\sigma \in \Omega_{RT}^j} \left\{ \sum_{\zeta \in \Omega_{NRT}^j} g_{\zeta, \sigma}^j \right\} \right\} \times \delta_i, & \Delta \neq 0 \end{cases} \quad (13)$$

where $q \in \{k+1, k+2, \dots, n_l\}$, δ_i is a random value on the interval $[0, 1]$, and $\hat{P}'_{t(q)} = \hat{P}_{t(q)}$

$-\sum_{u=1}^{q-1} \sum_{v=1}^{n_{hi}^u} x_{u,q} p_{children_i}(u, v) - \sum_{v=1}^{j-1} p_{children_i}(q, v)$ denotes the remaining power the sender of link q can use for NRT traffic but excluding the power this node has already used for NRT traffic in links 1 to q , with $t(q)$ representing the sender of link q and $x_{u,q}$ an indicator implying whether or not links u and q have the same transmitter.

D. Mutation Operation

The mutation operation in GA-NRT starts with fitness calculation, which identifies pop_{best} and further restricts the range of mutation positions in the remaining solutions. Similar to the mutation in GA-RT, the one here at a specific position m_{NRT} includes both channel and power reallocation. To reallocate channel, we shall find out whether or not the channel at a mutation position can be used under channel constraint (2b). If so, to utilize the channel, we also need to know whether there is enough remaining power under power constraint (2d). Besides, if the channel at position m_{NRT} is used by a flow before mutation, we check whether this channel is

the only channel that the flow uses at this hop as well. If the channel is unique, it cannot be reassigned to the other flow, or the original flow will break at this hop and its rate will fall into zero. Finally, under the circumstance that the channel at position m_{NRT} can be allocated to the other flow, we randomly reallocate it to one of the flows on the link including 0 which means this channel is not to be used after mutation operation.

Power reallocation is implemented for any channel with successful flow reassignment. By considering the power constraints (2d) and (2e), power reallocated for a channel (say channel j) of a link (say link q) in a solution (say pop_i) can be given by

$$P_{pop_i}(q, j) = \begin{cases} \hat{P}_{t(q)}'' \times \rho, & \Delta = 0 \\ \min \left\{ \hat{P}_{t(q)}'', \Delta / \max_{\sigma \in \Omega_{RT}^j} \left\{ \sum_{\zeta \in \Omega_{NRT}^j} g_{\zeta, \sigma}^j \right\} \right\} \times \rho, & \Delta \neq 0 \end{cases} \quad (14)$$

where ρ is a random value on the interval $[0, 1]$, $\hat{P}_{t(q)}'' = \hat{P}_{t(q)} - \sum_{u=1}^{n_t} \sum_{v=1}^{n_{ch}^u} y_{u,q,v} P_{pop_i}(u, v)$ is the remaining power the sender of link q can use for NRT traffic but excluding the power this node has already allocated to the channels of NRT traffic without requiring mutation, with indicator $y_{u,q,v}$ representing whether or not links u and q have the same transmitter and channel v at link u requires mutation.

For the time complexity of the GA-NRT algorithm, given the same generation number and the same order for the number of RT flows and that of NRT flows, it should not be larger than that of the GA-RT algorithm, because iterative power reduction adopted in the initialization and mutation operations in GA-RT, to reserve as many power resources as possible, are avoided in the counterparts of GA-NRT. Moreover, no NRT flows have to be suspended and thus no iterative resource allocation as we do for RT flows is required, which further reduces the time complexity of GA-NRT. As such, the overall time complexity of the GA-based resource allocation for both RT and NRT flows in CMNs is on the order of that of GA-RT as derived in Section 4.

6. Performance Evaluation

We consider a CMN with 9 nodes randomly located in 375m×375m coverage area. The spectrum is of 2.5MHz bandwidth and is partitioned into 12 channels, with each having a bandwidth of 208.33kHz. The impact of primary activity is studied in terms of the mean idle time of each channel. In the simulation, the mean idle time of each channel on any link is set to be the same value λ . The RT and NRT traffic flows are generated according to a Poisson process with mean rate set as λ_{RT} flow/s and 0.5 flow/s, respectively. To restrict the running time of the proposed GA-based resource allocation algorithm, we set the maximal number of generations in both GA-RT and GA-NRT as 20 and population size as 10. Further, p_{init} , p_{co} , p_{mut} , p_{eq} , d_1 , d_2 , and N_p used in GA-RT and GA-NRT are 0.5, 0.8, 0.1, 5%, 5%, and 2000, respectively. Other system parameters are chosen according to [1]. To compare the performance of the newly proposed GA-based algorithm, the algorithm proposed in [1], which utilizes a two-level decoupling approach (i.e., decouple power allocation from channel allocation and channel allocation from link allocation) to allocate channel and then geometric program to solve power allocation, and the greedy algorithm adopted in [24] for comparison, which is based on greedy channel-link mapping and uniform power allocation, are utilized.

We perform the simulation by matlab on an Intel Xeon 3.47GHz machine for 40 times, and each run lasts for 20s of network time. The 95% confidence interval is also given.

Fig. 3 shows the relationship between the mean access ratio of RT flows and the mean idle time λ of each channel given different values of interference cushion. It can be seen that with an increase of λ , the access ratio of RT flows increases for any of the three algorithms, due to more opportunities to find an idle channel. Also, the access ratio decreases if the interference cushion Δ increases, since more network resources are reserved for NRT flows. From **Fig. 3**, we can find out that the proposed GA-based algorithm is better than the other algorithms in all simulated cases in terms of the mean access ratio of RT flows. For example, when $\Delta = 0$ and $\lambda = 9$, the access ratios with the GA-based algorithm, with the algorithm proposed in [1], and with the greedy algorithm are 84.33%, 69.78%, and 56.77%, respectively. Moreover, when $\Delta = 0$, the access ratio with the GA-based algorithm is on average 15.79% higher than that with the algorithm proposed in [1] and 42.5% higher than that with the greedy algorithm.

In contrast to the access ratio of RT traffic, **Fig. 4** shows the simulation results of the mean throughput of NRT traffic. As observed, the throughput of NRT flows with each of the three algorithms increases as λ increases, due to the fact that increasing interference cushion reserves more network resources for NRT flows. Obviously, the GA-based algorithm performs better than the other two algorithms. When $\Delta = 10^{-8}$ W and $\lambda = 9$ s, the throughputs with the GA-based algorithm, with the algorithm in [1], and with the greedy algorithm are 6.50Mbps, 4.67Mbps, and 3.09Mbps, respectively. Moreover, when $\Delta = 0$, the throughput with the newly proposed algorithm is on average 39.07% (104.33%) higher than that with the algorithm in [1] (the greedy algorithm). However, it is also noteworthy that the throughput with the algorithm in [1] at $\Delta = 10^{-8}$ W is higher than the throughput with the GA-based algorithm at $\Delta = 0$. The reason is that, when $\Delta = 0$, GA allocates a large amount of resources for RT flows, limiting the remaining resources for NRT flows. Similar observation will be made in **Fig. 11** when studying the impact of traffic load.

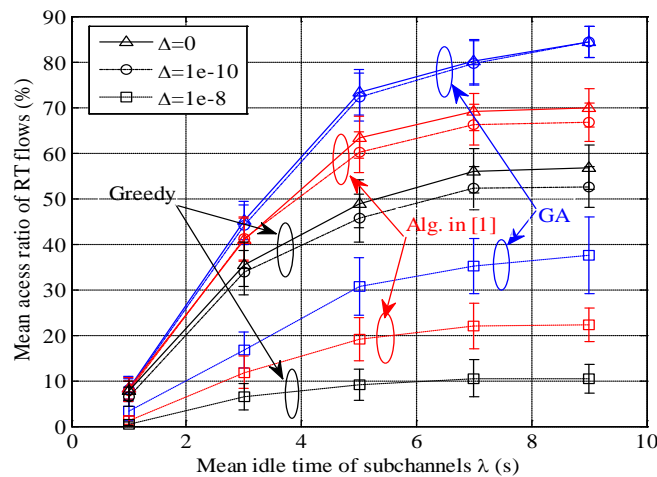


Fig. 3. Mean access ratio of RT flows vs. λ

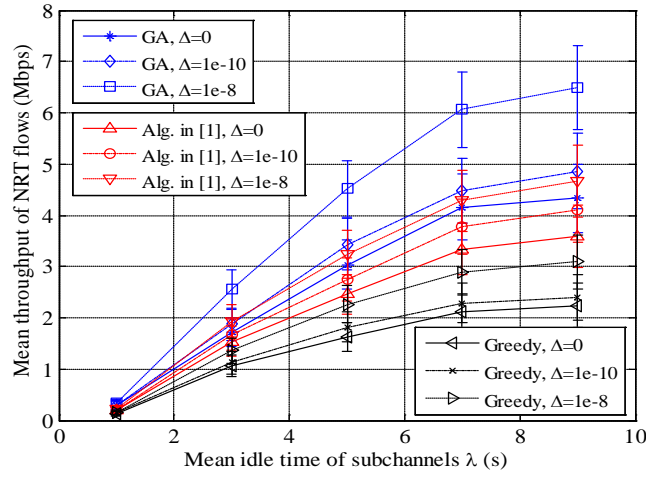


Fig. 4. Mean throughput of NRT flows vs. λ

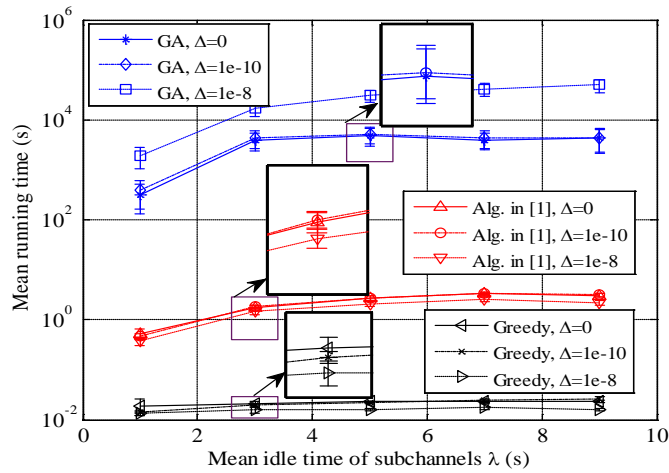


Fig. 5. Mean running time of the three algorithms vs. λ

Fig. 5 compares the mean running time of the three algorithms. From the figure, we can find out that the mean running time of the algorithm in [1] is two orders of magnitude higher than the greedy algorithm but four orders of magnitude lower than the newly proposed GA-based algorithm. With an increase of λ , the mean running time increases slightly and converges because the network is saturated. Thus, we can conclude that GA improves the performance at the cost of time complexity. In addition, Figs. 6-9 show the relationship between the average fitting value of allocating resource for RT traffic as well as that for NRT flows with the GA-based algorithm and the number of generations, at $\lambda = 3$ s and 9s, respectively. From the figures, we can find out that, GA is easy to converge when λ is small and hard to converge when it is large. The reason comes from the fact that the number of the idle channels on all links increases with an increase of λ , which enlarges the searching space accordingly. In other words, for the GA-based algorithm, it is important to select an appropriate generation number in order to well balance the tradeoff between performance improvement and running time reduction.

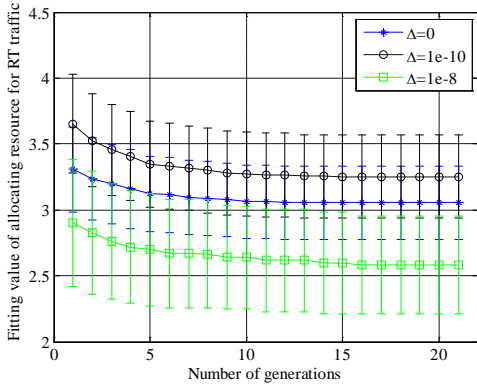


Fig. 6. Fitting value of allocating resource to RT flows vs. number of generations at $\lambda=3$

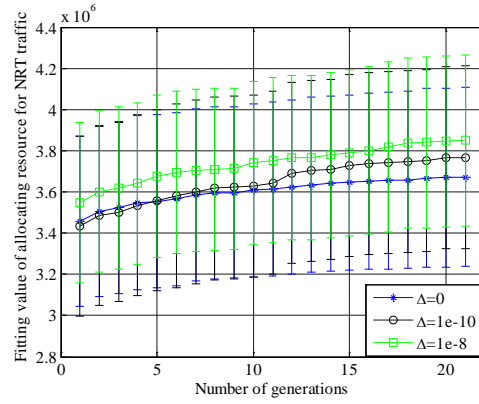


Fig. 7. Fitting value of allocating resource to NRT flows vs. number of generations at $\lambda=3$

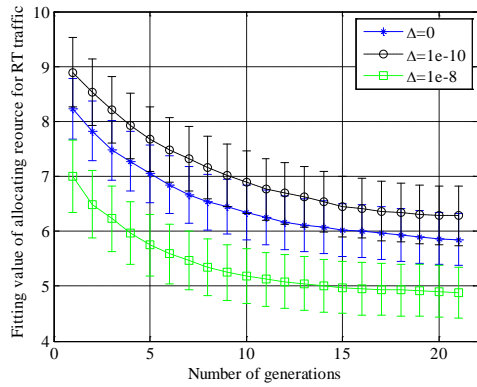


Fig. 8. Fitting value of allocating resource to RT flows vs. number of generations at $\lambda=9$

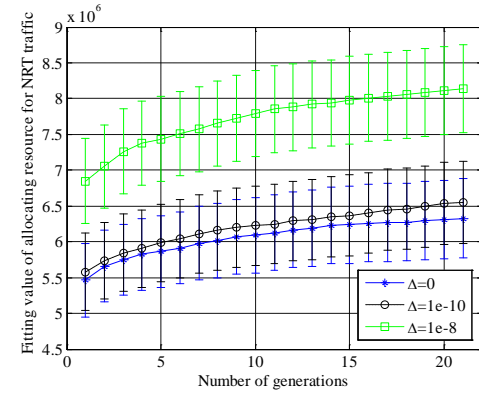


Fig. 9. Fitting value of allocating resource to NRT flows vs. number of generations at $\lambda=9$

Furthermore, we perform simulation to observe the impact of traffic load on the network performance by the proposed GA-based algorithm. **Fig. 10** shows the relationship between the mean access ratio of RT flows and the mean arrival rate λ_{RT} of RT flows given different values of interference cushion. From the figure, we can find that the mean access ratios of RT flows with the three algorithms firstly increase then decrease as λ_{RT} increases. The reason for the increase when $\lambda_{RT} \leq 0.5$ flow/s is that, as compared with the increased traffic load, the network might still have resources to serve a part of the increased RT flows. However, if the traffic load further increases, the network tends to be saturated. As such, the increasing RT flows are hard to be admitted, making the overall access ratio decline. The access ratio also decreases when interference cushion increases, because more network resources are reserved for NRT flows. Moreover, the GA-based algorithm obtains the highest access ratio as compared with the other two algorithms. For instance, when $\Delta = 0$ and $\lambda_{RT} = 1$, the access ratios with the GA-based algorithm, with the algorithm in [1], and with the greedy algorithm are 62.88%, 54.64%, and 45.99%, respectively. On average, the access ratio with the GA-based algorithm is 12.88% higher than that with the algorithm in [1] and 30.36% higher than that with the greedy algorithm, if considering $\Delta = 0$.

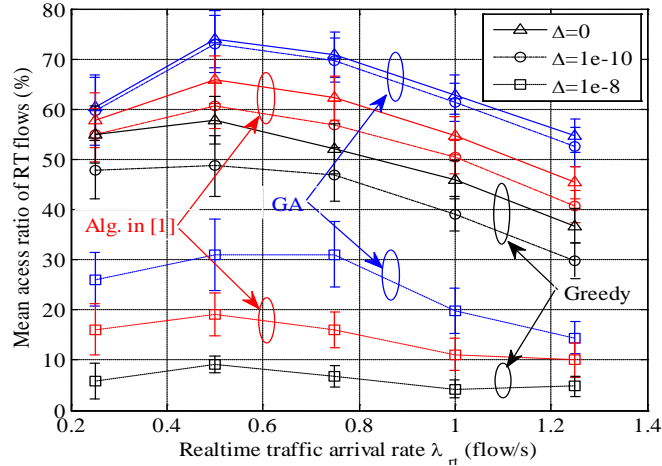


Fig. 10. Mean access ratio of RT flows vs. λ_{RT}

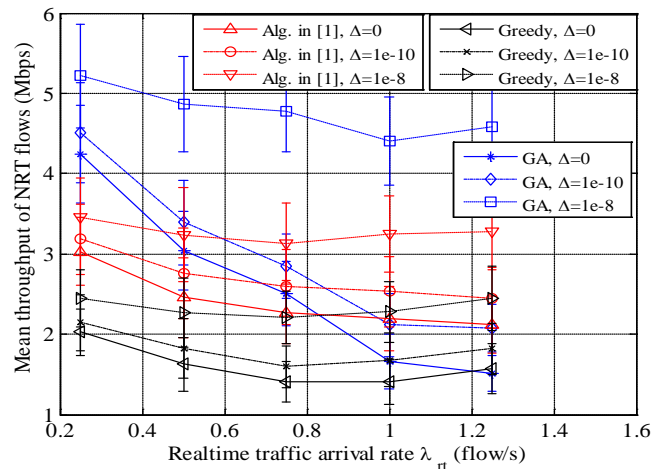


Fig. 11. Mean throughput of NRT flows vs. λ_{RT}

Fig. 11 shows the simulation results of the mean throughput of NRT traffic when the traffic load of RT flows changes. In general, the throughput of NRT flows with each of the three algorithms decreases as λ_{RT} increases, because more resources are used by RT flows. To boost the throughput of NRT traffic, one can increase Δ when allocating resource to RT traffic, as it reserves more resources for NRT traffic. Noticeably, the throughput with the GA-based algorithm decreases more dramatically than the other algorithms when λ_{RT} increases. The reason may stem from the fact that due to the superiority of the genetic algorithm, when allocating resources to RT traffic, the newly proposed GA-based algorithm is able to use up more network resources to admit more RT flows and leaves less network resources for NRT traffic. Therefore, the throughput degradation for NRT traffic can be much more obvious in the GA-based algorithm. Further, as compared to **Fig. 5**, similar observation on the running time of the three algorithms has been made when simulating the impact of traffic load.

It is worth mentioning that the proposed GA-based algorithm improves the network performance but at the cost of time complexity, which is one of the drawbacks of genetic algorithms [11, 12, 23]. To further improve the feasibility of both GA-RT and GA-NRT, similar to [19], we can adopt the strategy that combines a GA-based algorithm and a low-complexity algorithm (e.g., the one in [1]), via independently obtaining the resource allocation solutions using both algorithms and then simply choosing the better one out of the two solutions. Another strategy is using the solution of the low-complexity algorithm as an initial chromosome of the GA-based algorithm. Obviously, with either strategy, if we reduce the generation number of the GA-based algorithm, the time-complexity of the algorithm can be reduced; yet, due to the comparison mechanism or elitism strategy used in the GA-based algorithm, we can guarantee that the final solution is at least not worse than the one of the low-complexity algorithm. In other words, feasibility of the proposed GA-based algorithm can be improved if combined with other low-complexity algorithms, but at the cost of a certain extent of performance.

7. Conclusion

To have an in-depth understanding of the performance of a CMN with heterogeneous services as well as to explore the feasibility of any new approach to solve complex resource allocation problems in such a network, we have designed a new algorithm based on genetic algorithm to address the resource allocation problems studied in [1], which are MINLP. The newly designed GA-based algorithm can be treated as the optimal solution, because genetic algorithm is usually a robust search method requiring little information to search effectively in a large or poorly-understood search space. Novel initialization, selection, crossover, and mutation operations are designed such that solutions with enough randomness can be generated and converge with as less number of attempts as possible, improving the efficiency of the algorithm effectively. The algorithm designed in this paper can provide a comprehensive example of designing GA-based algorithm for many other application scenarios where similar complex wireless resource allocation problems are formulated. Simulation results show the superiority of the newly proposed GA-based algorithm, in increasing both the access ratio of RT flows and the throughput of NRT flows. However, it has also been found that in a CMN with both the RT and NRT traffic, due to the high priority of RT flows, the GA-based algorithm tends to fully utilize network resources to serve as many RT flows as possible, leaving fewer resources to NRT flows. Further, simulation results also reveal that as compared with algorithm in [1] it is also important to balance the tradeoff among QoS provisioning for RT traffic, throughput maximization for NRT traffic, and time complexity of a resource allocation algorithm. For the future work, we will optimize the design of the proposed GA-based algorithm to further reduce its time complexity.

References

- [1] N. Zhou, H. Shan, A. Huang, and X. Wang, "Resource management for heterogeneous services in cognitive mesh networks," in *Proc. of IEEE WCSP'13*, Hangzhou, China, 2013.
[Article \(CrossRef Link\)](#)
- [2] Y.-C. Liang, K.-C. Chen, G. Y. Li, and P. Mahonen, "Cognitive radio networking and communications: An overview," *IEEE Trans. Veh. Technol.*, vol. 60, no. 7, pp. 3386-3407, 2011.
[Article \(CrossRef Link\)](#)
- [3] T. Chen, H. Zhang, M. Matinmikko, and M. D. Katz, "CogMesh: Cognitive wireless mesh

- networks,” in *Proc. of IEEE GLOBECOM workshops*, 2008. [Article \(CrossRef Link\)](#)
- [4] S. Gunawardena and W. Zhuang, “Capacity analysis and call admission Control in distributed cognitive radio networks,” *IEEE Trans. Wireless Comm.*, vol. 10, no. 9, pp. 3110-3120, Sept. 2011. [Article \(CrossRef Link\)](#)
- [5] Y. T. Hou, Y. Shi, and H. D. Sherali, “Spectrum sharing for multi-hop networking with cognitive radios,” *IEEE J. Sel. Area. Comm.*, vol. 26, no. 1, pp. 146-155, 2008. [Article \(CrossRef Link\)](#)
- [6] Y. Shi, Y. T. Hou, and H. Zhou, “Per-node based optimal power control for multi-hop cognitive radio networks,” *IEEE Trans. Wireless Comm.*, vol. 8, no. 10, pp. 5290-5299, 2009. [Article \(CrossRef Link\)](#)
- [7] Y. Shi, Y. T. Hou, S. Kompella, H. D. Sherali, “Maximizing capacity in multihop cognitive radio networks under the SINR model,” *IEEE Trans. Mobile Computing*, vol. 10, no. 7, pp.954-967, 2011. [Article \(CrossRef Link\)](#)
- [8] A. Alshamrani, X. Shen, and L.-L. Xie, “QoS provisioning for heterogeneous services in cooperative cognitive radio networks,” *IEEE J. Sel. Area. Comm.*, vol. 29, no. 4, pp. 819- 830, 2011. [Article \(CrossRef Link\)](#)
- [9] S. Wang, Z.-H. Zhou, M. Ge, and C. Wang, “Resource allocation for heterogeneous cognitive radio networks with imperfect spectrum sensing,” *IEEE J. Sel. Area. Comm.*, vol. 31, no. 3, pp. 464-475, 2013. [Article \(CrossRef Link\)](#)
- [10] R. Xie, F. R. Yu, and H. Ji, “Dynamic resource allocation for heterogeneous services in cognitive radio networks with imperfect channel sensing,” *IEEE Trans. Veh. Technol.*, vol. 61, no. 2, pp. 770-780, 2012. [Article \(CrossRef Link\)](#)
- [11] L. Davis, *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1991.
- [12] J. H. Holland, *Adaptation Natural and Artificial Systems*. The MIT Press, 1975.
- [13] B. Lorenzo and S. Glisic, “Optimal routing and traffic scheduling for multihop cellular networks using Genetic Algorithm,” *IEEE Trans. Mobile Computing* , vol. 12, no. 11, pp. 2274-2288, 2013. [Article \(CrossRef Link\)](#)
- [14] P.-K. Tseng, W.-H. Chung, and P.-C. Hsiu, “Minimum interference topology construction for robust multi-hop cognitive radio networks,” in *Proc. of IEEE WCNC'13*, pp. 101-105, 2013. [Article \(CrossRef Link\)](#)
- [15] Y. Jin, J. Jin, A. Gluhak, K. Moessner, and M. Palaniswami, “An intelligent task allocation scheme for multihop wireless networks,” *IEEE Trans. Parallel and Distributed Systems*, vol. 23, no. 3, pp. 444-451, 2012. [Article \(CrossRef Link\)](#)
- [16] S. Chu, P. Wei, X. Zhong, X. Wang, and Y. Zhou, “Deployment of a connected reinforced backbone network with a limited number of backbone nodes,” *IEEE Trans. Mobile Computing*, vol. 12, no. 6, pp. 1188-1200, 2013. [Article \(CrossRef Link\)](#)
- [17] M. Chatterjee, H. Lin, and S. K. Das, “Rate allocation and admission control for differentiated services in CDMA data networks,” *IEEE Trans. Mobile Computing*, vol. 6, no. 2, pp. 179-191, 2007. [Article \(CrossRef Link\)](#)
- [18] N. Sharma, D. Badheka, and A. Anpalagan, “Multiobjective subchannel and power allocation in interference-limited two-tier OFDMA femtocell networks,” *IEEE Systems Journal*, vol. no. 99, pp. 1-12, 2014. [Article \(CrossRef Link\)](#)
- [19] H. T. Cheng and W. Zhuang, “Novel packet-level resource allocation with effective QoS provisioning for wireless mesh networks,” *IEEE Trans. Wireless Communications*, vol. 8, no. 2, pp. 694-700, 2009. [Article \(CrossRef Link\)](#)
- [20] M. R. Garey and D. S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, 1979.
- [21] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1989.
- [22] Q. Wu, Z. Zheng, and W. Deng, *Operations Research and Optimization with MATLAB Programming*, 2009.
- [23] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, 2004.
- [24] H. T. Cheng and W. Zhuang, “Joint power-frequency-time resource allocation in clustered wireless mesh networks,” *IEEE Network*, vol. 22, no. 1, pp. 45-51, 2008. [Article \(CrossRef Link\)](#)



Hanguan Shan received the B.Sc. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2004 and the Ph.D. degree in electrical engineering from Fudan University, Shanghai, China, in 2009.

From 2009 to 2010, he was a Postdoctoral Research Fellow with the University of Waterloo, Waterloo, ON, Canada. In February 2011, he joined the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China, as an Assistant Professor. His research focuses on cross-layer protocol design, resource allocation, and QoS provisioning in wireless networks.

Dr. Shan was a co-recipient of the Best Industry Paper Award from the IEEE Wireless Communications and Networking Conference (WCNC) 2011, Quintana Roo, Mexico.



Ziyun Ye received the B.Sc. degree in information and communication engineering from Zhejiang University, Hangzhou, China, in 2015. He is pursuing the M.Sc. degree in electrical engineering from University of California, Los Angeles, Los Angeles, U.S. His research interests include wireless networks, cognitive radio networks, cooperative networks, resource management, and heterogeneous networks.



Yuanguo Bi received the B.Sc. degree in Liaoning University, Shenyang, China in 2003 and the M.Sc. and Ph.D. degrees in Northeastern University, Shenyang, China in 2006, both in computer science. From 2007 to 2009, he was a visiting PhD student at the Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is an Associate Professor at the College of Information Science and Engineering, Northeastern University, China. His current research interests include medium access control, QoS routing, multihop broadcast, and transmission control in vehicular networks.



Aiping Huang graduated from Nanjing Institute of Post and Telecommunications, China in 1977, received the M.Sc. degree from Nanjing Institute of Technology (current name: Southeast University), China in 1982, and received Licentiate of Tech. degree from Helsinki University of Technology (HUT), Finland in 1997. She worked from 1977 to 1980 as an engineer at Design and Research Institute of Post and Telecom. Ministry, China. From 1982 to 1994, she was with Zhejiang University (ZJU), China as an Assistant Professor and then Associate Professor in the Dept. of Scientific Instrumentation. She was a Visiting Scholar and then a Research Scientist at HUT (current name: Aalto University) from 1994 to 1998. From 1998, she is a full professor in the College of Information and Electronics Engineering at ZJU. She serves as the vice chair of IEEE ComSoc Nanjing Chapter. She published a book and more than 160 papers in refereed journals and conferences on communications and networks, signal processing. Her current research interests include heterogeneous networks, performance analysis and cross-layer design, planning and optimization of cellular mobile communication networks.