Primary Social Behavior aware Routing and Scheduling for Cognitive Radio Networks

Shouling Ji[†], Zhipeng Cai[‡], Jing (Selena) He[♯], and Raheem Beyah[†]
[†]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30308, USA
[‡]Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA
[♯]Department of Computer Science, Kennesaw State University, Kennesaw, GA 30144-5591, USA
Email: sji@gatech.edu, zcai@cs.gsu.edu, jhe4@kennesaw.edu, rbeyah@ece.gatech.edu

***Abstract*—As an objective reality, the social behavior pattern of Primary Users (PUs) has significant impacts on the design and management of the secondary network. However, most of the existing works overlook this fact by simplifying the spectrum whitespace assumption. In this paper, we study the joint routing and time-domain scheduling problem for Cognitive Radio Networks (CRNs) by considering the social behaviors of PUs. Our main contributions consist of four aspects. First, we analyze the social pattern of PUs based on two practical data traces. According to the obtained social pattern, the available spectrum whitespace is derived for SUs. Subsequently, in terms of previous analysis, we propose a centralized joint routing and time-domain scheduling framework with global provable ϵ-optimality ($\epsilon \in [0, 1]$) by employing the branch-and-bound technique, where ϵ indicates the expected closeness of our solution to the optimum solution. The solution of this centralized algorithm can serve as a theoretical benchmark for developing future routing and scheduling algorithms for CRNs. Third, we design a distributed primary behavior-aware routing and scheduling algorithm with local performance guarantee, where the routing and scheduling fairness, the available bandwidth, the potential interference, *etc*. are taken into account. Finally, simulation results confirm our assertion that primary behaviors have significant impacts on the spectrum whitespace, and demonstrate that primary-behavior-aware joint routing and scheduling design can utilize spectrum whitespace efficiently.**

I. Introduction

Recently, to improve the utilization efficiency of the precious wireless spectrum, Cognitive Radio Networks (CRNs) as a new communication paradigm attract numerous research interests. As in traditional wireless networks, *routing* and *scheduling* are two of the most fundamental issues in CRNs. However, new challenges appear in the design of routing and scheduling algorithms for CRNs due to the dynamical spectrum supply characteristic, *i.e.*, the available spectrum bands for a Secondary User (SU) is dynamic over time and maybe different from its neighbors. Therefore, traditional routing and scheduling algorithms cannot achieve satisfiable performance since they do not consider such spectrum dynamics. Aiming at efficient routing and scheduling for CRNs, many efforts have been spent to propose heuristic and optimization-based routing and scheduling schemes [8]-[13].

However, to the best of our knowledge, the existing routing and scheduling algorithms do not consider the social behaviors of Primary Users (PUs), which are the objective realities in the Primary Network (PN) and have significant impacts on the design and management of the Secondary Network (SN) [15]. For instance, the PN consisting of cell phone users (*e.g.*, AT&T) follows some evident social pattern in weekdays: (i) the primary activity is heavier during [10:00-22:00] (denoted in the 24-hour manner) compared with other time slots; and (ii) the primary activity is heavier during [9:00-17:00] in the activity area and [18:00-23:00] in the residential area [2]-[7]. Furthermore, as indicated by many social studies (see the references in [15]), the social behaviors/patterns of PUs are considerably stable physical phenomenons over a long term. Therefore, the social pattern of PUs in different time periods can guide us to obtain insight on understanding the dynamic available spectrum, namely the *spectrum whitespace*, for SUs, which leads to the possible design of more effective routing and scheduling algorithms.

In this paper, for the first time, we propose to study the joint routing and scheduling problem for CRNs by considering the social behaviors of PUs. Towards this direction, we first investigate the social pattern of PUs by analyzing two well-known typical data traces. Based on the primary social behaviors, more accurate analysis on the existing spectrum whitespace for SUs can be derived which demonstrates that the spectrum whitespace is a function depending on the primary social behaviors, current time, the distributions of PUs and SUs, *etc*. It clearly distinguishes our work from the existing works where the social behaviors of PUs are overlooked [10]-[12].

After deriving the primary social behavior aware spectrum whitespace for SUs, we study the joint routing and time-domain based scheduling issue for CRNs by proposing both a centralized algorithm with global provable ϵ-optimality ($\epsilon \in [0, 1]$) and a distributed algorithm with local performance guarantee. For our centralized algorithm, we first mathematically formalize the problem as a time-dependent Mixed-Integer Linear Program (MILP), which is NP-complete in general. Subsequently, by carefully examining the upper and lower bounds of the time-dependent MILP, we apply the *branch-and-bound* technique to design a joint routing and scheduling algorithm with ϵ-optimality, where ϵ is the expected ratio between our solution and the optimum solution. Similar as [10], the centralized algorithm can work in some long-term-stable CRN applications and serve as a performance benchmark for joint

978-1-4673-7331-9/15/$31.00

routing and scheduling solutions in multihop CRNs. On the other hand, for the CRNs preferring distributed algorithms, we propose a distributed joint routing and scheduling framework. In this framework, the distance to the destination, available spectrum bandwidth, potential traffic, *etc*. are jointly considered for routing, and the data flow rate aware smart carrier-sensing is considered for scheduling.

Note that, different from most of existing works [10]-[13] where scheduling is considered in the frequency-domain, we conduct scheduling in the time domain since the spectrum whitespace is time dependent. Nevertheless, as indicated in [10]-[13], the time-domain based scheduling methods can be extended to the frequency-domain. In addition, our algorithm can work with the frequency-domain scheduling methods in [10]-[13] together in the time-frequency-domain.

Finally, by conducting simulations, we demonstrate that the primary-behavior-aware joint routing and scheduling design can utilize spectrum whitespace efficiently to induce higher network throughput and lower transmission latency.

II. System Model

We consider a dense-scaling SN coexisting with a PN deployed in a square area of size $A = \frac{cn}{\log n}$, where c is a changeable constant value and n is the number of SUs.

PN: The PN consists of N Poisson distributed PUs with density λ denoted by S_i $(1 \leq i \leq N)$. The transmission and interference radii of PUs are assumed to be R and R_I, respectively (without loss of generality, $R_I > R$). The network time is slotted with each time slot of length τ. At the very beginning of each time slot, each PU either transmits some data or keeps silent during that time slot. Furthermore, since we focus on scheduling in the time-domain, we assume there is only one abstract primary spectrum band shared by PUs and SUs, *i.e*. we study the spectrum dynamics over time while do not consider the scheduling in the frequency domain. The bandwidth of the abstract primary spectrum is assumed to be W, which can be treated as the aggregated bandwidth of all the actual primary spectrum bands. This assumption is reasonable since we conduct the time-domain scheduling and it has been widely recognized [1]. According to this assumption, the available spectrum for SUs at time t is denoted by W^t, which is a time and primary behavior dependent parameter (see derivation in Section III).

SN and Problem Description: The considered SN consists of n independently and identically distributed SUs denoted by s_i $(1 \leq i \leq n)$. The transmission and interference radii of SUs are r and r_I respectively (without loss of generality, $r_I > r$). Let $D(\cdot, \cdot)$ represent the Euclidean distance between two nodes. Then, there exists a link/edge from s_i to s_j if $D(s_i, s_j) \leq r$. Thus, the SN can be modeled as a graph $G = (V, E)$, where $V = \{s_i | 1 \leq i \leq n\}$ is the node set and E is the set of all the possible links formed by nodes in V. For s_i, its *neighbor set* and *interference set* are defined as $V_i = \{s_j | D(s_i, s_j) \leq r, s_j \neq s_i\}$ and $V_i^I = \{s_j | D(s_i, s_j) \leq r_I, s_j \neq s_i\}$, respectively.

For our problem, we study how to route and schedule the data transmission for a set of communication sessions $\mathcal{L}$ within time T in a SN, where $\mathcal{L} = \{l | l$ is a session from source SU l_s to destination SU l_d with data rate requirement $\gamma(l)\}$. Similar to [10], our objective is to maximize a scaling factor $\kappa \geq 0$ such that each session under our algorithm can achieve a data transmission rate of at least $\kappa \cdot \gamma(l)$. As indicated in [10], this objective is more general which can cover the objectives of *maxmin throughput* and *maximizing each session's rate proportional to its minimum rate requirement*.

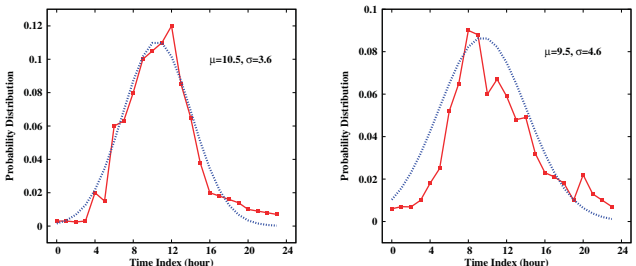
III. Social Behavior Analysis of PUs

In terms of recent empirical studies [2][3], it can be seen that the utilization of primary spectrum is very inefficient. For instance, over 96% (> 542.4 MHz) of spectrum interval $[960\text{MHz}, 1525\text{MHz}]$ is underutilized. Therefore, it is reasonable for SUs to opportunistically access the primary spectrum to improve the spectrum utilization efficiency. Now, we have two natural and related questions: (1) how does the spectrum whitespace/opportunities distribute over time? and (2) how to utilize primary spectrum effectively and meanwhile minimizing harmful impacts on primary activities? Taking the spectrum interval $[806\text{MHz}, 902\text{MHz}]$ (assigned to cell phones, SMR) as an example, on average, 45.2% (43.4 MHz) of this spectrum interval is available for SUs. However, is it appropriate for SUs to access these 43.4 MHz spectrum equi-probably over time? Can the optimal performance be achieved by the existing routing and scheduling algorithms if 43.4 MHz is taken as a fixed available bandwidth parameter? To answer the above two questions, more in-depth research should be conducted on analyzing the behaviors of PUs, which is overlooked in most of the existing routing and scheduling works.

On the other hand, PUs tend to follow some stable social pattern [15] since they are humans or operated by humans, *e.g.*, cell phone users, TV viewers. Consequently, we propose to analyze the spectrum whitespace for SUs from a social perspective, which can provide us new insights on understanding spectrum dynamics over time followed by designing efficient routing and scheduling algorithms. We start our analysis based on two representative data traces, the MIT Reality trace [6] and the UCSD trace [7], which record contacts through Bluetooth or WiFi interfaces among mobile device users on campus.

The MIT Reality trace involves 97 mobile device holders. A contact between two holders is recorded if their devices are connected through Bluetooth. In total, the MIT Reality trace recorded 114046 contacts spanning 246 days. The UCSD trace contains 123225 WiFi contacts generated by 275 mobile devices spanning 77 days. A contact is recorded if two devices are connected via a WiFi access point. These two traces can be viewed as two small PNs on campus (without loss of generality, the contacts in these two traces are very similar to that of cell phone users and PUs in cellular networks [4]).

Since these two traces are recorded on campus and considering students' behaviors, intuitively, the users are more active during the daytime. By a close check on the MIT



(a) MIT Reality trace. (b) UCSD trace.

Fig. 1. Probability distribution of a user to be active. The dot lines show the normal distributions of the MIT Reality trace and UCSD trace, respectively.

reality trace, we find that the data generally follows a one-day-based periodical distribution, *i.e.*, the contacts distributions over each day are almost the same. On average, over 90% contracts happened during time [8:00-20:00]. Particularly, the peak hours are [10:00-18:00] during which over 80% contacts happened. Similar skew distribution can also be found in the UCSD trace.

In terms of the active user distribution, the probability distributions of a user to be active over time in the MIT Reality trace and UCSD trace are shown in Fig.1. In Fig.1, we make a translation on the time index (x-axis) by starting from 4:00 with the purpose of making a better curve fitting. From Fig.1, the probability distribution of a user to be active in the two traces can be approximated by normal distributions $\mathcal{N}(\mu = 10.5, \sigma = 3.6)$ and $\mathcal{N}(\mu = 9.5, \sigma = 4.6)$ respectively, where μ is the *expectation* and σ is the *standard deviation* of the corresponding normal distribution. By checking the real distributions and the approximate normal distributions, the Root-Mean-Square Error (RMSE) is 0.0084 for the MIT Reality trace and 0.0140 for the UCSD trace. Therefore, it is reasonable to exploit a normal distribution to approximately describe the primary behaviors.

Based on the analysis of the primary social behaviors, it is reasonable to assume that the probability distribution of a PU to be active follows a normal distribution $\mathcal{N}(\mu, \sigma)$. In the following, we derive the available spectrum whitespace for SUs in terms of the primary behaviors. First, we give some preliminary knowledge on Poisson distribution and Normal distribution as follows. For a random variable X, we use $X \sim \mathcal{P}(\lambda)$, $X \sim \mathcal{B}(k, p)$, and $X \sim \mathcal{N}(\mu, \sigma)$ to denote that X follows a Poisson distribution with parameter λ, a binomial distribution with parameters k and p, and a normal distribution with parameters μ and σ, respectively. Then, we have the following properties.

Property 1: If $X \sim \mathcal{P}(\lambda)$ and Y conditional on $X = k$ is $Y|(X = k) \sim \mathcal{B}(k, p)$, then $Y \sim \mathcal{P}(\lambda \cdot p)$.

Property 2: Let $X \sim \mathcal{N}(\mu, \sigma)$, and $f(X = x)$ and $F(X = x)$ be the *probability density function* (pdf) and *Cumulative Distribution Function* (CDF) of the normal distribution, respectively. Then, $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ and $F(x) = \int_{-\infty}^{x} f(t) \cdot dt = \Phi(\frac{x-\mu}{\sigma})$, where $\Phi(\cdot)$ is the CDF of $\mathcal{N}(0, 1)$.

Now, we are ready to derive the spectrum whitespace for SUs by considering the primary social behaviors, where the probability distribution of a PU to be active is given by a normal distribution $\mathcal{N}(\mu, \sigma)$. Let $p([t_1, t_2])$ denote the probability of a PU to be active during time $[t_1, t_2]$, and variable Y^t denote the number of active PUs during the t-th time slot. Then, we have the following lemma. Due to the space limitation, we omit the proof of Lemma 1.

Lemma 1: (1) $p([t_1, t_2]) = \Phi(\frac{t_2-\mu}{\sigma}) - \Phi(\frac{t_1-\mu}{\sigma})$; and (2) $Y^t \sim \mathcal{P}(\lambda \cdot p_p^t)$, where $p_p^t = \Phi(\frac{t\tau-\mu}{\sigma}) - \Phi(\frac{(t-1)\tau-\mu}{\sigma})$ is the probability that a PU is active during the t-th time slot.

Based on Lemma 1, we can obtain the expected available spectrum whitespace, for a secondary link from s_i to s_j at time slot t, denoted by W_{ij}^t, as shown in Lemma 2. For convenience, we define a function $A(x, y) = x^2 \cos^{-1}(\frac{y}{x}) - y\sqrt{x^2 - y^2}$. Furthermore, due to the space limitation, we omit the proof of Lemma 2.

Lemma 2: Let $d_1 = \frac{D(s_i,s_j)^2 + r_I^2 - R_I^2}{2D(s_i,s_j)}$, $d_2 = \frac{D(s_i,s_j)^2 - r_I^2 + R_I^2}{2D(s_i,s_j)}$, and $a_{ij} = A(r_I, d_1) + A(R_I, d_2)$. Then, $W_{ij}^t = W \cdot p_{ij}^t$, where $p_{ij}^t = e^{-p_p^t \lambda(\pi(r_I^2 + R_I^2) - a_{ij})}$ is the probability of a spectrum opportunity from s_i to s_j without causing interference to PUs.

Remarks: From Lemma 2, the expected whitespace for a secondary link is time and primary social behavior dependent. Compared with the traditional static assumption on the spectrum whitespace, W_{ij}^t is more accurate by considering the dynamic primary social behaviors. Note that, it is possible that PUs might follow some other social pattern, which practically depends on the considered PN. Our primary social behavior aware idea and analysis could be extended to other types of PNs to aid the design of more efficient protocols.

IV. Joint Routing and Time-Domain Scheduling

A. Time-Domain based Scheduling

We consider the scheduling in the time domain which consists of T time slots. Let $\mathbb{N}^T = \{1, 2, \cdots, T\}$. To determine whether a link from s_i to s_j is scheduled during time slot $t \in \mathbb{N}^T$, we define an *indicator* θ_{ij}^t as follows. $\theta_{ij}^t = 1$ if s_i is scheduled to transmit data to s_j during time slot t; otherwise, $\theta_{ij}^t = 0$.

From the transmission/reception perspective, every transmitter/receiver s_i can only transmit/receive data to/from one node during any time slot t, *i.e.*,

$$\sum_{s_j \in V_i} \theta_{ij}^t \leq 1, \quad \sum_{s_k \in V_i} \theta_{ki}^t \leq 1, \ (t \in \mathbb{N}^T). \tag{1}$$

Furthermore, $\forall s_i \in V$, s_i cannot be both transmitter and receiver simultaneously, *i.e.*,

$$\sum_{s_j \in V_i} \theta_{ij}^t + \sum_{s_k \in V_i} \theta_{ki}^t \leq 1, \ (t \in \mathbb{N}^T). \tag{2}$$

By checking constraints (1)and (2) carefully and considering θ-variables (the sum of θ-variables is also a non-negative integer), it can be verified that constraint (2) implies constraint (1), *i.e.*, if (2) is satisfied, then (1) is also satisfied.

To obtain a feasible schedule, we have to guarantee all the scheduled transmissions in a time slot are interference-free, *i.e.*,

$$\theta_{ij}^t + \sum_{s_q \in V_k} \theta_{kq}^t \leq 1, (s_k \in V_j^I, s_k \neq s_i, t \in \mathbb{N}^T). \quad (3)$$

B. Routing

Similar as the objective in [10], our routing and scheduling objective is to maximize a scaling factor κ such that each session l in $\mathcal{L}$ can achieve a data transmission rate of at least $\kappa \cdot \gamma(l)$. We define a variable $f_{ij}^t(l)$ denoting the allocated data rate on the link from s_i to s_j for session l in time slot t. Then, from the routing perspective, the balance of data flow of each session should be maintained at each node. Consequently, for $l \in \mathcal{L}$, if $s_i = l_s$, then we have constraint

$$\sum_{t=1}^{T} \sum_{s_j \in V_i} f_{ij}^t(l) = \kappa \cdot \gamma(l), (l \in \mathcal{L}, s_i = l_s). \quad (4)$$

If s_i is an intermediate forwarding node for $l \in \mathcal{L}$, we have

$$\sum_{t=1}^{T} \sum_{s_j \in V_i} f_{ij}^t(l) = \sum_{t=1}^{T} \sum_{s_k \in V_i} f_{ki}^t(l), (s_i \neq l_s, s_i \neq l_d). \quad (5)$$

If $s_i = l_d$, we have

$$\sum_{t=1}^{T} \sum_{s_k \in V_i} f_{ki}^t(l) = \kappa \cdot \gamma(l), \ (l \in \mathcal{L}, s_i = l_d). \quad (6)$$

Besides the flow balance constraints, the scheduled data transmission on a link during a time slot should not exceed the affordable transmission capability of that link, *i.e.*,

$$\sum_{l \in \mathcal{L}} f_{ij}^t(l) \leq c_0 \cdot \theta_{ij}^t \cdot W_{ij}^t, \ (s_i \neq l_d, s_j \neq l_s, t \in \mathbb{N}^T), \quad (7)$$

where W_{ij}^t is the dynamic primary social behavior aware spectrum whitespace (Lemma 2), and c_0 is a user-defined adjustable constant value which can provide routing and scheduling flexibility depending on the transmission path loss, noise, *etc*.

C. Mathematical Formulation

By considering routing and scheduling together, our problem can be mathematically formalized as follows.

$$\begin{array}{lll} \max & \kappa & \\ \text{s.t.} & \text{constriants } (2)-(7), W_{ij}^t = W \cdot p_{ij}^t & (t \in \mathbb{N}^T) \\ & \kappa \geq 0, f_{ij}^t(l) \geq 0, \theta_{ij}^t = \{0,1\} & (l \in \mathcal{L}, s_j \in V_i) \end{array}$$

In the above formulation, κ and $f_{ij}^t(l)$ are nonnegative variables, θ_{ij}^t are 0/1 variables, and W_{ij}^t are time-dependent nonnegative variables. Thus, the optimization problem is a Mixed-Integer NonLinear Program (MINLP) problem, which is NP-hard in general. Fortunately, the social pattern of individuals is a pretty stable physical phenomenon in a long term [15]. Therefore, the social pattern of the PN can be derived from historical PU behaviors. It follows that W_{ij}^t can be obtained beforehand in terms of similar analysis shown in Section III and be viewed as known values in the optimization problem. After such an analysis and derivation, the above MINLP problem can be reduced to a Mixed-Integer Linear Program (MILP) problem, denoted by **P**. However, **P** is still NP-complete in general. Hence, we will design dedicated provable centralized solution and workable distributed solution in the following sections.

V. ϵ-Optimal Solution

A. Solution Framework

Our ϵ-optimal solution is based on the classical *branch-and-bound* technique. Here, we extend this technique to seek an ϵ-optimal solution.

When employing the branch-and-bound technique to resolve the MILP problem, *e.g.*, **P** in the previous section, there are three key points: determining the upper bound of **P**, determining the lower bound of **P**, and partitioning **P** into subproblems. The basic idea of branch-and-bound is to narrow down the gap between the upper bound and the lower bound by iterations, *i.e.*, decreasing the optimization space, until a satisfiable and feasible solution is achieved. During each iteration, from the perspective of *branching*, if we cannot find a satisfiable solution for the current problem, we partition this problem into two subproblems by giving more constraints, *e.g.*, fixing the value of an integer variable in **P** to be 0 or 1 in each subproblem and add the new generated problems to the problem list to start a new iteration. For *bounding*, we ignore the problem that cannot produce a desired solution during each iteration.

Our Branch-and-Bound (BB) framework is shown in Alg. 1. In Alg. 1, Ψ is the set of problems which initially contains **P**. ω and o denote the current lower and upper bounds of **P**. β_ω is the corresponding solution of ω. $LP(P)$ represents the linearized version of P (Section V-A1). $o(P)$ denotes the upper bound of P and we show how to obtain $o(\cdot)$ in Section V-A1. $\omega(P)$ and $\beta(P)$ represent the lower bound and the corresponding solution of P, which can be obtained by Alg. Lower-Bound(P) (Section V-A2). Alg. Branching(P, P_1, P_2) partitions P into subproblems P_1 and P_2 (Section V-A3).

Algorithm 1: Branch-and-Bound (BB) framework.

```
Ψ = {P}, ω = −∞, β_ω = ∅, o = o(P);
while ω < εo ∧ Ψ ≠ ∅ do
    ∀P ∈ Ψ, if ∄ a feasible solution of LP(P) or
    ω ≥ ε · o(P), then Ψ = Ψ \ {P};
    select P s.t. o(P) = max{o(x)|x ∈ Ψ}, let o = o(P);
    obtain β(P) and ω(P) by Lower-Bound(P);
    if ω(P) > ω, then ω = ω(P), β_ω = β(P);
    if ω ≥ ε · o, then return the ε-optimal solution β_ω;
    create two subproblems by Branching(P, P_1, P_2);
    Ψ = (Ψ \ {P}) ∪ {P_1, P_2};
return β_ω;
```

The general idea of BB is as follows. We first initialize the lower and upper bounds of **P** and then start the iteration

process. During each iteration, the problems without feasible solutions or cannot produce desired solutions are removed for further consideration (bounding). This bounding technique distinguishes this method from the brute-force exhaustive searching, by which many unqualified problems are removed so that the problem space is reduced [10]. Subsequently, the problem with the maximum upper bound in Ψ, denoted by P, is selected for consideration. By employing Lower-Bound(P), a feasible solution $\beta(P)$ and the corresponding lower bound $\omega(P)$ can be obtained for P. $\omega(P)$ is then compared with the current lower bound ω to see whether ω can be improved. Next, if the current best feasible solution β_ω is an ϵ-optimal solution, it is returned. Otherwise, two new subproblems P_1 and P_2 are generated from P by adding more constraints (branching).

1) Upper Bound of P: As discussed before, the formulated problem is NP-complete. The difficulty lies on the integral constraints on variables θ_{ij}^t. To obtain an upper bound of the problem, we can relax θ_{ij}^t in P to real numbers in $[0, 1]$. Evidently, the relaxed version of P, denoted by $LP(P)$, is a Linear Program, which can be addressed efficiently within polynomial time. The solution of $LP(P)$ is an upper bound of P since the optimization space is enlarged.

2) Lower Bound of P: Although every feasible solution of P corresponds to a lower bound of P, we want to find a lower bound of P closer to the overall optimal solution of **P** during each iteration, and thus Alg. 1 can be further accelerated. Therefore, we have two concerns when seeking $\beta(P)$ and $\omega(P)$: $\beta(P)$ is a feasible solution and $\omega(P)$ is close to the optimal solution of **P** (equivalently, close to $o(P)$). To guarantee the feasibility of $\beta(P)$, the difficulty lies on satisfying the integral constraints, *i.e.*, to obtain a feasible link scheduling. On the other hand, the optimality of $\beta(P)$ is determined by how optimal the link scheduling and routing are. Since the optimal routing part can be obtained by a Linear Program as long as the scheduling is fixed, we propose a fast greedy algorithm Lower-Bound(P) as shown in Alg. 2 to determine $\beta(P)$ and $\omega(P)$ in Alg. 1. In Alg. 2, c_1 and c_2 are user-defined adjustable constants. f_{ij}^t is defined as the aggregate traffic from s_i to s_j during time slot t, i.e., $f_{ij}^t = \sum_{l \in \mathcal{L}} f_{ij}^t(l)$.

The basic idea of Alg. 2 is as follows. To obtain a feasible solution of P with a nice lower bound, multiple iterations might be needed. During each iteration, a greedy idea is employed to determine one scheduling link and many non-scheduling links. In each iteration, we first solve the linear version of P determining its upper bound. If the solution of $LP(P)$ happens to satisfy all the integral constraints of P, $\beta(P)$ and $\omega(P)$ are determined and returned. Otherwise, there are some θ variables having non-integer values, *i.e.*, with values in $(0, 1)$. Then, we set the θ variable with the maximum $c_1 \frac{f_{ij}^t}{\max\{1, W_{ij}^t\}} + c_2 \frac{f_{ij}^t}{\theta_{ij}^t}$ value with integer 1 and add it to problem P as a new constraint for the next iteration. For determining the scheduling link θ_{xy}^t, we consider the following two factors: (i) $c_1 \frac{f_{ij}^t}{\max\{1, W_{ij}^t\}}$ which reflects how much available bandwidth can be used if we schedule this link (the link with higher spectrum utilization ratio is preferred); and (ii) $c_2 \frac{f_{ij}^t}{\theta_{ij}^t}$ which indicates the efficiency contributed to the scheduling scheme if we schedule this link. Basically, the greedy criteria is to determine a scheduling link which can maximize the scheduling and data transmission efficiency during each iteration until no more links can be scheduled in time slot t. After determining the scheduling link θ_{xy}^t, all the θ variables corresponding to the links which are interfered with s_y or interfered by s_x are set to be 0 and added to P as constraints. In this manner, many θ variables will be assigned fixed appropriate integer values, which can significantly accelerate the iteration process and thus obtain a feasible solution and a lower bound for P (mathematical analysis is available in Section V-B).

Algorithm 2: Lower-Bound(P).

1 solve $LP(P)$, denote the solution by $\beta(P) = \{\theta_{ij}^t, f_{ij}^t(l) | D(s_i, s_j) \le r, t \in \mathbb{N}^T\}$;
2 **if** $\beta(P)$ *satisfies all the integral constraints in* P **then**
3 **return** $\beta(P)$ and the corresponding$\omega(P)$;
4 let $\theta_{xy}^t = \arg\max_{\theta_{ij}^t} \{c_1 \frac{f_{ij}^t}{\max\{1, W_{ij}^t\}} + c_2 \frac{f_{ij}^t}{\theta_{ij}^t} \mid 0 < \theta_{ij}^t < 1, \theta_{ij}^t \in \beta(P)\}$;
5 $\beta(P) = (\beta(P) \setminus \{\theta_{xy}^t\}) \cup \{\theta_{xy}^t = 1\}$;
6 $P = P \cup \{\theta_{xy}^t = 1\}$;
7 for $\forall \theta_{ij}^t \in \beta(P)$ s.t. $s_i \in V_y^I \wedge s_i \neq s_x$ and $\forall \theta_{ij}^t \in \beta(P)$ s.t. $s_j \in V_x^I \wedge s_j \neq s_y$, do $P = P \cup \{\theta_{ij}^t = 0\}$;
8 go to step 1;

3) Partition of P: To design an effective partition/branching algorithm, the first step is to find an appropriate partition variable. In our branching algorithm, as shown in Alg. 3, our criteria to choose the partition variable is based on the impacts on the scheduling and routing solution if this variable is determined. Specifically, we select the most free θ variable (whose value is close to 0.5 while far from the desired integer 0/1) to partition problem P. In Alg. 3, to accelerate the overall branch-and-bound process, we add more integral constraints derived from $\theta_{xy}^t = 1$ to subproblem P_2 (see the analysis in Section V-B).

Algorithm 3: Branching(P, P_1, P_2).

1 solve $LP(P)$, denote the solution by $\phi(P)$;
2 choose θ_{xy}^t s.t. $\theta_{xy}^t = \arg \min_{0 < \theta_{ij}^t < 1} \{|\theta_{ij}^t - 0.5| \cdot \frac{W_{ij}^t}{\max\{1, f_{ij}^t\}} | \theta_{ij}^t \in \phi(P)\}$;
3 $P_1 = P \cup \{\theta_{xy}^t = 0\}$, $P_2 = P \cup \{\theta_{xy}^t = 1\}$;
4 for $\forall \theta_{ij}^t \in \phi(P)$ s.t. $s_i \in V_y^I \wedge s_i \neq s_x$ and $\forall \theta_{ij}^t \in \phi(P)$ s.t. $s_j \in V_x^I \wedge s_j \neq s_y$, $P_2 = P_2 \cup \{\theta_{ij}^t = 0\}$;

B. Performance Analysis

In this subsection, we analyze the correctness and optimality of the designed BB framework. First, we give the fastness analysis on Lower-Bound(P).

Lemma 3: In Lower-Bound(P), (i) if we set $\theta_{xy}^t = 1$, the number of θ variables set to 0 in one iteration, *i.e.*, the number of $\theta_{ij}^t = 0$, is lower bounded by $\Omega(\log^2 n)$; and (ii) the number of iterations is upper bounded by $\min\{O(\frac{nT}{(r_I-r)^2 \log n}), \frac{nT}{2}\}$.

Proof Sketch: To analyze how many θ variables are set to 0 in one iteration in Alg. 2, we start from deriving the number of nodes in V_y^I. Since all the SUs are deployed in a square area with size A, then $\odot(s_y, r_I) \cap A \geq \frac{\pi r_I^2}{4}$. Furthermore, considering that SUs are independently and identically distributed, the probability that a SU is located at the interference area of s_y is $p = \Pr(s_i \in V_y^I) = \frac{\pi r^2}{4A} = \frac{\pi r^2 \log n}{4cn}$. Let X be a random variable denoting $|V_y^I|$. Then, $X \sim \mathcal{B}(n, p)$ (here, we do not consider s_x and s_y particularly which is reasonable for large n). In addition, let $c_3 \max_{\xi<0} \frac{\pi r^2 (e^\xi - 1)}{4c\xi} + \frac{2}{\xi}$ be a positive value depending on c. Applying the Chernoff bound and for any $\xi < 0$, we have $\Pr(X \leq c_3 \log n) \leq \min_{\xi<0} \frac{E[e^{\xi X}]}{e^{\xi c_3 \log n}} = \min_{\xi<0} \frac{(1+(e^\xi-1)p)^n}{e^{\xi c_3 \log n}} \leq \min_{\xi<0} \frac{\exp((e^\xi-1)np)}{e^{\xi c_3 \log n}} \leq \exp(-2 \ln n) = \frac{1}{n^2}$. $\sum_{n>0} \frac{1}{n^2} = \frac{\pi^2}{6}$ is an upper bounded *Riemann zeta function* with parameter 2. Thus, according to the Borel-Cantelli Lemma, the event $X \geq c_3 \log n$ happens with probability 1. Employing the similar technique, we can prove $|V_i| \geq c_4 \log n$, where c_4 is a constant value. Therefore, the number of $\theta_{ij}^t = 0$ for $s_i \in V_y^I \wedge s_i \neq s_x$ in one iteration of Alg. 2 is lower bounded by $\Omega(c_3 c_4 \log^2 n)$. Similarly, the number of $\theta_{ij}^t = 0$ for $s_j \in V_x^I \wedge s_j \neq s_y$ in one iteration of Alg. 2 is lower bounded by $\Omega(\log^2 n)$.

We now prove the upper bound on the number of iterations in Alg. 2. To this end, we only have to figure out how many 1's can be set since in each iteration of Alg. 2, only one θ variable is set to 1. We consider one specific time slot t. In t, each transmitter s_x corresponds to one interference disk $\odot(s_x, r_I)$, within which no other receiver should appear except for s_y. Therefore, there is no other transmitter in disk $\odot(s_x, r_I - r)$ except for s_x, *i.e.*, for any two concurrent transmitters s_x and s_q, $\odot(s_x, \frac{r_I - r}{2})$ and $\odot(s_q, \frac{r_I - r}{2})$ are not overlapping. Furthermore, any disk $\odot(s_x, \frac{r_I - r}{2})$ must locate in a square area with side length of $\sqrt{A} + r_I - r$. It follows that the number of transmitters during a time slot is upper bounded by $\frac{(\sqrt{A} + r_I - r)^2}{\pi (r_I - r)^2 / 4} = O(\frac{n}{(r_I - r)^2 \log n})$. Consequently, within T time slots, the number of transmitters (1's or iterations) in Alg. 2 is upper bounded by $O(\frac{nT}{(r_I - r)^2 \log n})$. When $r_I \approx r$, we can conclude that the number of transmitter-receiver pairs is upper bounded by $n/2$ in a time slot, which implies the number of iterations in Alg. 2 is also upper bounded by $\frac{nT}{2}$. □

From Lemma 3, we can see that the lower bound decision process for problem P can be accelerated by fixing the values of more θ variables. Now, we show the correctness of Lower-Bound(P) as follows. Due to the space limitation, we omit the proof of Lemma 4.

Lemma 4: Alg. 2 produces a feasible solution $\beta(P)$ with lower bound $\omega(P)$ for problem P.

Now, we analyze the fastness of Branching(P, P_1, P_2). Based on similar proof technique in Lemma 3, the following corollary can be proved. Corollary 1 indicates the designed branching scheme can accelerate the overall branch-and-bound process by considering more derived constraints[1].

Corollary 1: The number of $\theta_{ij}^t = 0$ constraints added to subproblem P_2 is lower bounded by $\Omega(\log^2 n)$ in Alg. 3.

For problem P which is partitioned into subproblems $\{P_i\}$, we define this partition as *meaningful and scheduling-consistency preserved* if (i) at least one new constraint is added to each subproblem P_i; and (ii) the newly added constraints to each subproblem are confliction-free with the constraints in P. Now, we show the correctness of Alg. 3 as follows. Due to the space limitation, we omit the proof of Lemma 5.

Lemma 5: Branching(P, P_1, P_2) can correctly and meaningfully partition P with scheduling-consistency preservation.

Let o_i and ω_i be the upper and lower bounds of **P** in the i-th iteration, respectively. Then, the following lemma indicates the decreasing property of o_i and increasing property of ω_i. Due to the space limitation, we omit the proof of Lemma 6.

Lemma 6: In the BB framework, $o_{i+1} \leq o_i$ and $\omega_{i+1} \geq \omega_i$.

Now, we are ready to show the correctness and optimality of the designed BB framework.

Theorem 1: The BB framework obtains an ϵ-optimal solution for problem **P** within finite time.

Proof Sketch: This theorem can be proved in three steps: (i) each iteration can be correctly executed; (ii) the number of iterations is finite; and (iii) the final solution is ϵ-optimal. First, based on the correctness analysis of Alg. 2 and 3 in Lemmas 4 and 5, we can conclude that each iteration in Alg. 1 can be correctly executed. Second, if all the θ variables in **P** is determined, **P** is reduced to an LP problem which can be solved in polynomial time. During each iteration in Alg. 1, either an ϵ-optimal solution is found or some θ variable can be determined by generating new subproblems. Furthermore, the number of θ variables is upper bounded by $O(n(n-1)T)$ and the possible value of a θ variable is 0/1, which implies the number of iterations in Alg. 1 is finite. Third, since the number of iterations is finite and the upper bound (respectively, lower bound) of **P** is non-increasing (respectively, non-decreasing) as shown in Lemma 6, we can conclude that the ϵ-optimal solution can be found by Alg. 1. □

Discussion: Note that, in Alg. 1, the time consumption for each iteration is of polynomial complexity (mainly caused by solving $LP(P)$). However, the problem space $|\Psi|$ can be of exponential order in the worst case. Fortunately, by applying the multi-fold bounding techniques, *e.g.*, removing some problems for further consideration (Alg. 1), adding more derived integral constraints to the subproblem (Alg. 3, Corollary 1, Lemma 5), *etc*, the branch-and-bound process

[1]Note that, in Alg. 3, although the constraints $\theta_{ij}^t = 0$ added to subproblem P_2 might already exist, it is still meaningful when a $\theta_{ij}^t = 0$ constraint is explicitly added to a subproblem for the first time.

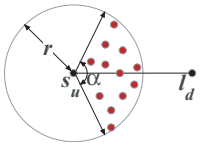


Fig. 2. Forwarding sector.

can be significantly accelerated to find the desired ϵ-optimal solution. As indicated in [10], the solution of Alg. 1 can be exploited as a reference optimal result/performance benchmark to guide the development of future routing and scheduling algorithms. Specifically, it has been shown in [10] that this kind of branch-and-bound based algorithms are effective and feasible in offering performance benchmark for static CRN-s. Therefore, as the first work on joint routing and time-domain scheduling with PU social behavior consideration, the designed BB framework is meaningful for static multi-hop CRNs.

VI. Distributed Solution

A. Solution Framework

As the BB framework which provides a provable ϵ-optimal solution is a centralized algorithm, in this section, we design a distributed joint routing and time-domain scheduling algorithm with PU social behavior consideration.

At time t, the amount of data of session $l \in \mathcal{L}$ at s_u ($s_u \neq l_d$) waiting for transmission is $d_u^t(l)$. Define $\mathcal{L}_u^t = \{l | d_u^t(l) > 0\}$. For $l \in \mathcal{L}_u^t$, the elapsed time since the last time that s_u transmits data for l is denoted by $\tau_u(l)$. If s_u never transmits data for l before, $\tau_u(l)$ is defined as the elapsed time since the time that s_u received/generated data of l.

To relay data for $l \in \mathcal{L}_u^t$, we define an *α-forwarding sector* as follows[2]. An *α-forwarding sector* $A_u^\alpha(l)$ for l at s_u is a *sector* of disk $\odot(s_u, r)$ with central angle α ($\alpha \in (0, 2\arccos\frac{r}{2D(s_u,l_d)}]$)[3] and angle bisector $\overline{s_u l_d}$ as shown in Fig.2. We also define an *α-forwarding set* of l at s_u as $N_u^\alpha(l) = \{s_i | s_i \in A_u^\alpha(l)\}$. When s_u transmits data for l at time slot t, it selects the next relay s_v from $N_u^\alpha(l)$ with the highest *forwarding score* $f_{uv}^t(l)$ defined as $f_{uv}^t(l) = \frac{c_5(D(s_u,l_d)-D(s_v,l_d))(W_{uv}^t+1)}{c_6 \max\{1,|V_v^I|\}+c_7 \max\{1,\sum_{\ell\in\mathcal{L}} d_v^t(\ell)\}}$, where c_i ($5 \leq i \leq 7$) are some user defined adjustable positive constant values, and term $W_{uv}^t + 1$ is to guarantee $f_{uv}^t(l) > 0$ when $D(s_v, l_d) < D(s_u, l_d)$. From the definition of the forwarding score, we can see that s_u prefers the next hop s_v such that s_v is closer to the destination, more available bandwidth from s_u to s_v, less interference at s_v, and less traffic at s_v.

To deal with the interference in scheduling, we employ the previous CSMA-like controlling strategy [8]. Based on the previous work [8], we let SUs work on the *re-start* mode (a receiver can switch to receive stronger signal on the re-start mode) and set the *Carrier Sensing Range* (CSR) of each SU as $\max\{(1+c_8)\frac{R}{r}, 1+c_9\} \cdot r$ where c_8 and c_9 are some constant values depending on the *path loss exponent* and the *signal-to-interference-plus-noise ratio* at SUs and PUs. Then, as proven in the previous work (Lemmas 2 and 3 in [8]), (i) a SU only has to carrier sense the communication environment within its CSR; and (ii) it can successfully conduct data transmission without interfering other concurrent primary and secondary transmitters as long as there is a spectrum opportunity within its CSR.

[2]We assume $D(s_u, l_d) > r$. Otherwise, s_u forwards data to l_d directly.

[3]Note that, since $2\arccos\frac{r}{2D(s_u,l_d)} > \frac{\pi}{2}$, α can be easily determined.

Algorithm 4: DRS framework at s_u

```
s_u updates L_u^t;
select l such that τ_u(l) = max{τ_u(ℓ)|ℓ ∈ L_u^t};
select forwarding node s_v from N_u^α such that
  s_v = arg max_{s_v} {f_uv^t(l) > 0 | s_v ∈ N_u^α(l)};
s_u carrier senses with CSR max{(1 + c_8) R/r, 1 + c_9} · r;
if there is a spectrum opportunity then
    s_u transmits min{W_uv^t, d_u^t(l)} data to s_v;
    d_u^t(l) = d_u^t(l) − min{W_uv^t, d_u^t(l)}, τ_u(l) = 0;
```

Our Distributed Routing and Scheduling (DRS) algorithm at s_u during time slot t is shown in Alg. 4. From DRS, we can see that both scheduling and routing fairness are considered. It is because that when scheduling data transmission at s_u, the session waiting for the longest time has the highest priority, and when selecting the routing relay, the SU with lower traffic is preferred. Furthermore, the available spectrum bandwidth, potential interference, and distance to the destination are also considered in the routing. Note that, DRS also preserves the dynamic (multi-path) routing characteristic since s_u might choose different relays based on $f_{uv}^t(l)$ at different time slots.

B. Correctness Analysis

First, we show that the routing strategy in DRS is *proper*, which implies that in DRS (i) an intermediate node can always find a next-hop relay; and (ii) there is no routing loop. First, we show that a next-hop relay can be found in Lemma 7. We omit the proof due to the space limitation.

Lemma 7: In DRS, s_u can find a forwarding node s_v such that $s_v = \arg\max_{s_v}\{f_{uv}^t(l) > 0 | s_v \in N_u^\alpha(l)\}$.

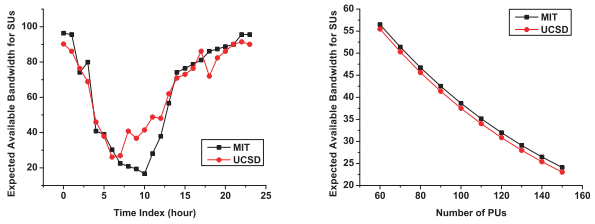
We show that DRS does not produce any routing loop in Lemma 8.

Lemma 8: DRS does not produce any routing loop.

Proof: Suppose for contradiction there is a routing loop, denoted by $\cdots \to s_u \to s_v \to s_w \to \cdots \to s_i \to s_u, \to \cdots$, on the route of session l. Then, according to DRS, we must have $f_{uv}^{t_u}(l) > 0, f_{vw}^{t_v}(l) > 0, \cdots, f_{iu}^{t_i}(l) > 0$. It follows we have $D(s_u, l_d) < D(s_v, l_d) < D(s_w, l_d) < \cdots < D(s_i, l_d) < D(s_u, l_d)$. It is a contradiction. Thus this lemma holds. □

Based on Lemmas 7 and 8, the routing in DRS is proper. Furthermore, according to the previous result (Lemmas 2 and 3 in [8]), interference can be effectively avoided by letting SUs work with a proper CSR. Consequently, we can demonstrate the correctness of DRS as follows.

Theorem 2: DRS produces proper routing and interference-free scheduling for the communication sessions in $\mathcal{L}$.



(a) (b)

Fig. 3. Expected available bandwidth for SUs.

VII. Simulations and Analysis

In this section, we validate the impacts of PUs' behaviors on the available spectrum bandwidth for SUs and the performance of the proposed algorithm. The centralized BB framework is based on LP and mainly for providing a provable theoretical benchmark. Currently, we already have multiple successful commercial softwares to solve LP. Similar as [10], our BB framework is easy to be implemented on top of these softwares, *e.g.*, CPLEX. Therefore, we focus on examining the performance of our distributed solution DRS in this section. We consider a randomly distributed SN coexisted with a Poisson distributed PN in a square area with size A. We employ the MIT trace and the UCSD trace to model the social behaviors of PUs, *i.e.*, PUs follow the same social/active pattern of users in MIT and UCSD. As analyzed in Section III, the activities in MIT and UCSD generally follow a one-day-based periodical distribution. Therefore, we assume the network time is slotted with each time slot of length normalized to 1 minute, which implies a periodical circle has $T_c = 24 \times 60 = 1440$ time slots. The *density* of SUs is defined as $\rho = \frac{n}{A}$. We assume there are $|\mathcal{L}|$ random sessions in the network. For session $l_i \in \mathcal{L}$ $(1 \leq i \leq |\mathcal{L}|)$, its source generates $(10+5i)$ amount of data every Γ time slots, which is defined as the *data generation interval*. For all the other system parameters, they are specified in each group of simulations and the default settings are: $W = 100$, $N = 100$, $A = 10 \times 10$, $\rho = 5$, $r = 1$, $r_I = R_I = 2$, $\delta = 0.8$, $|\mathcal{L}| = 6$, and $\Gamma = 10$.

The compared algorithm is Coolest [14], which is a recently proposed routing algorithm. In Coolest, the path with the lowest spectrum utilization by PUs is preferred for data transmission. For fairness, we incorporate the same scheduling algorithm as in DRS into Coolest for scheduling and interference avoidance. In the following, each group of simulations is repeated for 100 times and the results are the average values.

Available Bandwidth Analysis: The available spectrum bandwidth for SUs under different circumstances is shown in Fig.3. In Fig.3(a), we show the bandwidth for SUs during one periodical circle T_c (24 hours). The results confirm our assertion since primary behaviors do have significant impacts on the available bandwidth for SUs (changes from > 95 to < 20 for MIT and from > 90 to < 30 for UCSD). Therefore, it is more reasonable to design routing and scheduling algorithms for CRNs with consideration of primary social behaviors.

We also examine the average available bandwidth at a SU when *Time Index* $= 5$ versus the number of PUs N (Fig.3(b)). From Fig.3(b), we can see that the available bandwidth for a SU decreases when N increases. The reason is straightforward. More PUs imply more primary activities, and thus less spectrum whitespace is left for SUs.

Throughput Analysis: We analyze the throughput performance of DRS and Coolest versus the number of PUs N, network size A, the number of sessions $|\mathcal{L}|$, and the licensed spectrum width W as shown in Fig.4(a)-(d) respectively (by default $|\mathcal{L}| = 6$). Here, we measure the throughput by the *success delivery ratio* δ within one PU behavior periodical circle T_c. δ is defined as the average ratio between the amount of data been successfully delivered to destinations and the amount of data been generated by sources. From Fig.4(a), we can see that when the number of PUs increases, δ shows a decreasing trend for DRS and Coolest under both the MIT pattern and the UCSD pattern. The reason is that more PUs imply less spectrum opportunities and bandwidth for SUs and thus low δ is induced. Furthermore, DRS has better performance than Coolest. The reasons are as follows: (i) Coolest prefers the path with the lowest spectrum utilization by PUs. Consequently, many SUs might find overlapped pathes when routing which causes the data congestion and accumulation problem followed by low data delivery ratio; and (ii) on the other hand, when DRS determines its routing, it considers the routing and scheduling fairness (including traffic at the next-hop relay), available spectrum bandwidth (primary social behaviors), potential interference, *etc*. Furthermore, DRS also preserves the dynamic routing property. Therefore, the pathes selected in DRS have higher transmission concurrency, followed by high data delivery ratio.

From Fig.4(b), when A increases, δ demonstrates decreasing trend for DRS and Coolest. The reason is that we randomly generate these communication sessions. When the network becomes larger, the average distance from sources to destinations increases while T_c is fixed. Therefore, δ decreases. Again, DRS leads to higher δ than Coolest since primary behavior is considered and the paths in DRS are more balanced.

From Fig.4(c), when $|\mathcal{L}| = 1$, both DRS and Coolest can successfully deliver all the data to the destination. This is because the traffic in the SN is light. However, with the increase of $|\mathcal{L}|$, δ decreases for DRS and Coolest. This comes from the fact that more traffic appears. DRS produces better performance than Coolest since it is primary behavior-aware and utilizes spectrum opportunities more effectively.

From Fig.4(d), when W increases, δ increases for both DRS and Coolest. This is because a larger W implies more potential available bandwidth for SUs when the activity pattern of PUs is fixed, followed by higher data delivery ratio. Again, DRS has better performance than Coolest.

Latency Analysis: We also examine the latency performance of DRS and Coolest by fixing the expected δ of each session to be no less than 0.8. The results are shown in Fig.4(e)-(h). From Fig.4(e), we can see that when N increases, the number of the consumed time slots of DRS and Coolest

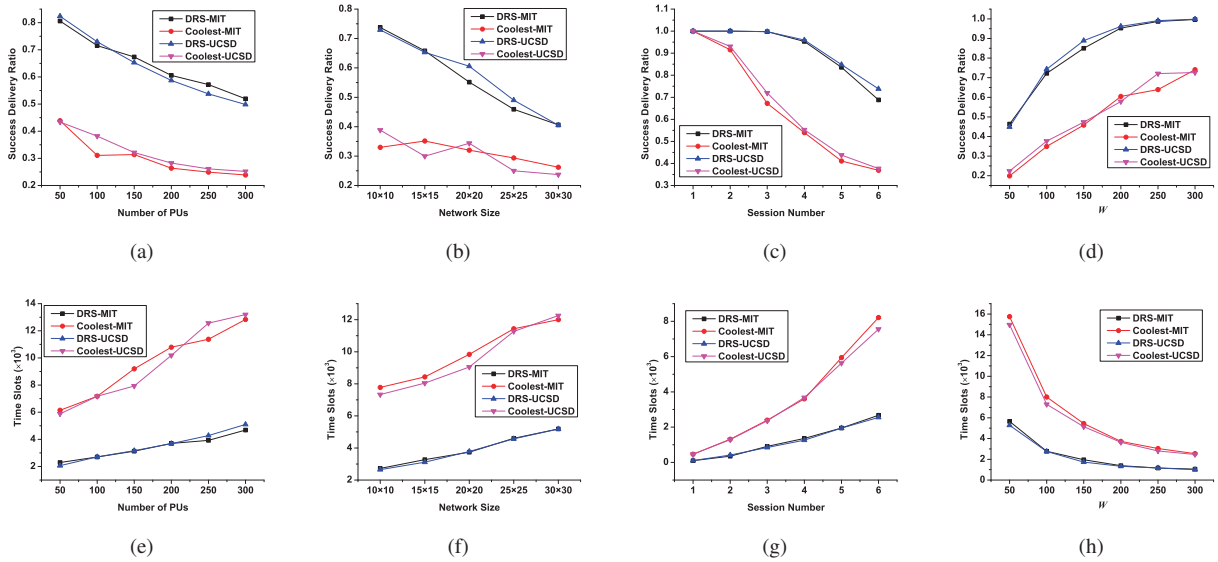


Fig. 4. Throughput and latency of DRS and Coolest.

also increase. This is because more PUs imply less spectrum whitespace for SUs and thus high latency for data delivery. Since DRS considers routing and scheduling fairness as well as traffic balance and potential interference, DRS consumes fewer time slots than Coolest. In Fig.4(f), when A increases, the average transmission distance of each session becomes longer, thus more hops are needed from the source to the destination, inducing higher latency for both DRS and Coolest. DRS produces better performance than Coolest because of its fair routing and scheduling scheme and the consideration of the primary behavior pattern. Fig.4(g) shows that when $|\mathcal{L}|$ increases, there is more traffic in the network. Consequently, the induced latency of DRS and Coolest increases. When $|\mathcal{L}|$ becomes larger, the advantage of DRS over Coolest is more significant. This confirms that DRS incorporates routing and scheduling fairness into the design. Again, DRS has better performance because of efficient utilization of spectrum opportunities. Fig.4(h) shows that larger W implies more spectrum whitespace followed by less time consumption on delivering data for both algorithms. Due to the reasons analyzed before, DRS consumes less time than Coolest.

VIII. Conclusion

In this paper, we study the joint routing and scheduling problem for CRNs by considering the social behaviors of PUs. We first analyze the social pattern of PUs based on two practical data traces and then derive the available spectrum whitespace for SUs. Subsequently, we study the joint routing and time-domain scheduling problem for CRNs by proposing both a centralized algorithm with global provable ϵ-optimality and a distributed algorithm with local performance guarantee. Finally, we conduct extensive simulations to validate our assertion as well as the performance of the proposed algorithm.

Acknowledgment

This work was partly supported by the National Science Foundation (NSF) under grant No. CNS-1252292, the Kennesaw State University's Office of the Vice President of Research (OVPR) Pilot/Seed Grant, and the College of Science and Mathematics Interdisciplinary Research Opportunities (IDROP) Program.

References

[1] I. F. Akyildiz et al., NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey, Computer Networks, 2006.
[2] M. A. McHenry et al., Chicago Spectrum Occupany & Analysis and a Long-Term Studies Proposal, TAPAS 2006.
[3] M. McHenry, Spectrum White Space Measurements, New America Foundation Broadband Forum, 2003.
[4] D. Willkomm et al., Primary User Behavior in Cellular Networks and Implications for Dynamic Spectrum Access, IEEE CM 2009.
[5] D. Chen, S. Yin, Q. Zhang, M. Liu, and S. Li, Mining Spectrum Usage Data: a Large-scale Spectrum Measurement Study, Mobicom 2009.
[6] N. Eagle et al., Reality Mining: Sensing Complex Social Systems, PUC 2006.
[7] M. McNett and G. Voelker, Access and Mobility of Wireless PDA Users, ACM SIGMOBILE MCCR 2005.
[8] Z. Cai, S. Ji, J. (S.) He, and A. G. Bourgeois, Optimal Distributed Data Collection for Asynchronous Cognitive Radio Networks, ICDCS 2012.
[9] S. Ji, R. Beyah, and Z. Cai, Minimum-Latency Broadcast Scheduling for Cognitive Radio Networks, IEEE SECON 2013.
[10] Y. Shi, Y. T. Hou, S. Kompella, and H. D. Sherali, Maximizing Capacity in Multihop Cognitive Radio Networks under the SINR Model, TMC 2011.
[11] Y. T. Hou, Y. Shi, and H. D. Sherali, Spectrum Sharing for Multi-hop Networking with Cognitive Radios, JSAC 2008.
[12] Y. Shi, Y. T. Hou, H. Zhou, and S. F. Midkiff, Distributed Cross-layer Optimization for Cognitive Radio Networks, IEEE TVT 2010.
[13] M. Pan, C. Zhang, P. Li, Y. Fang, Joint Routing and Link Scheduling for Cognitive Radio Networks under Uncertain Spectrum Supply, Infocom 2011.
[14] X. Huang, D. Lu, P. Li, and Y. Fang, Coolest path: spectrum mobility aware routing metrics in cognitive ad hoc networks, ICDCS 2011.
[15] Y. Zhu, B. Xu, X. Shi, and Y. Wang, A Survey of Social-based Routing in Delay Tolerant Networks: Positive and Negative Social Effects, CST 2013.