2014 IEEE 34th International Conference on Distributed Computing Systems

Almost Optimal Channel Access in Multi-Hop Networks With Unknown Channel Variables

Yaqin Zhou*, Qiuyuan Huang†, Fan Li‡[1], Xiang-yang Li§¶, Min Liu*, Zhongcheng Li*, Zhiyuan Yin‡
*Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
†Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA
‡School of Computer Science, Beijing Institute of Technology
§Illinois Institute of Technology, Chicago, IL, USA
¶School of Software and TNLIST, Tsinghua University, China

***Abstract*— We consider the problem of online dynamic channel accessing in multi-hop cognitive radio networks. Previous works on online dynamic channel accessing mainly focus on single-hop networks that assume complete conflicts among all secondary users. In the multi-hop multi-channel network settings studied here, there is more general competition among different communication pairs. A simple application of models for single-hop case to multi-hop case with N nodes and M channels leads to exponential time/space complexity $O(M^N)$, and poor theoretical guarantee on throughput performance. We thus novelly formulate the problem as a linearly combinatorial multi-armed bandits (MAB) problem that involves a maximum weighted independent set (MWIS) problem with unknown weights. To efficiently address the problem, we propose a distributed channel access algorithm that can achieve $1/\rho$ of the optimum averaged throughput where each node has communication complexity $O(r^2 + D)$ and space complexity $O(m)$ in the learning process, and time complexity $O(Dm^{\rho^r})$ in strategy decision process for an arbitrary wireless network. Here $\rho = 1 + \epsilon$ is the approximation ratio to MWIS for a local r-hop network with $m < N$ nodes, and D is the number of mini-rounds inside each round of strategy decision.**

I. Introduction

Available spectrum is being exhausted, while a lot of frequency bands are extremely under utilized. As a promising solution to improve dynamic allocation of the under-utilized spectrum, cognitive radio technology allows secondary users (SUs) to opportunistically access vacant channels in temporal and spatial domain when the primary user is idle. However, the available channel qualities to SUs are unknown at the start of transmissions because of uncontrollable external disturbances and primary users' occupancies. Meanwhile, due to resource and hardware constraints, cognitive radios (CRs) can sense only a part of heterogeneous channels at a given time. Thus, it is vital for secondary users to learn and select the best possible channels to access. Several recent results [1], [2], [3], [4], [5], [6], [7], [8] are proposed to take the dynamic spectrum sharing problem as the multi-armed bandits problem, and attempt to find a dynamic channel access policy that results in almost optimal expected throughput (or *zero-regret*) through learning history, compared with the optimal fixed channel policy. Unfortunately, these methods generally adopt the simplest form of MAB where only single-hop cognitive radio networks (CRNs) fit the model.

[1]Fan Li is the contact author

In dynamic channel accessing of multi-hop CRNs, the first key challenge is to model general interference among SUs into a more sophisticated MAB formulation. As a consequence of general interference, a naive extension from the single-hop case to multi-hop case will lead to regret, time and space complexity that is exponential with the number of users in the learning process. More specifically, taken as an arm a strategy consisting of decisions from each of the N users with M available channels, there will be totally $O(M^N)$ combinations. Since all these aforementioned works adopt a UCB-type learning policy [9] [10] [11], the upper bound of regret as well as time and space complexity is linear with the number of arms, thus linear with $O(M^N)$. Such theoretical results are extremely poor in large scale multi-hop CRNs.

As practical efficiency is always the most concerned benchmark in multi-hop networks, another obstacle lies in devising decentralized methods with low computation and communication complexity. Previous decentralized MAB methods [4] [2] [12] pay little attention to these practical challenges around multi-hop networks. Though there is no communication cost in [4] [2], they require exponential computation time in a single learning round, resulting in serious throughput loss. Additionally, [2] assumes multiple users can access the same resource, which does not capture conflicts among near-by SUs. General conflicts in multi-hop CRNs cause severely throughput loss as well. Therefore, it is imperative to design efficient channel accessing policies to overcome these practical challenges.

In this paper, we focus on investigating the zero-regret online channel access problem in multi-hop CRNs. It involves competition among adjacent users, and cooperation for maximum throughput network wide, combined with classical exploration-exploitation tradeoff in learning unknown channel qualities. We give a first attempt to mathematically formulate the challenging problem for theoretical analysis, and provide efficient decentralized solutions for practical benefits. Our main contributions are two fold.

- We formulate the problem into a linearly combinatorial

1063-6927/14 $31.00 © 2014 IEEE
DOI 10.1109/ICDCS.2014.54

MAB problem that shall find a maximum weighted independent set of vertexes where weight is unknown channel quality. This novel formulation facilitates us to utilize a zero-regret learning policy where it only costs time and space complexity $O(MN)$ for a network with M channels and N SUs. The other benefit is that it gives us freedom to design efficient methods on our demands to solve the MWIS problem and retain zero-regret.

- We propose an efficient decentralized channel access scheme with provable theoretical performance. Our decentralized implementation achieves the same approximation ratio of $\rho = 1 + \epsilon$ as the centralized form, but significantly reduce time complexity from $O(N^{\rho^r})$ to $O(Nm^{\rho^r})$. Here $r = O(\log_\rho M)$ is the hop number required to achieve a PTAS. Our simulation results show that our new distributed learning policy indeed outperforms previous policies in terms of average throughput and time.

Remark on assumed models: We realized some apparent limitations that might lead to misunderstanding of our contributions, i.e., i.i.d channel qualities and unit-disk interference models. We argue that our first contribution gives a general model to study zero-regret online channel access problems in various multi-hop CRNs, such as CRNs with Markov or nonstochastic channel qualities, and other graph-based or physical interference models. For general graph-based interference models, it becomes a linearly combinatorial MAB under stochastic/Markov/nonsotchastic unknown variables, while for physical interference models, it becomes a non-linearly combinatorial MAB under stochastic/Markov/nonsotchastic unknown variables. The remaining challenges are to solve the linearly/non-linearly combinatorial MAB with involved MWIS efficiently.

Organization: We present the network model in Section II, problem formulation in Section III, our distributed access policy in Section IV, and our simulation results in Section V. We review related work in Section VI, and conclude the work in Section VII.

II. Network model

Consider a network denoted by conflict graph $G = (V, E, C)$ with a set $V = \{v_i | i = 1, \dots, N\}$ of N nodes (SUs), a set E of edges denoting conflicts, and a set $C = \{c_j | j = 1, \dots, M\}$ of M channels. We assume M is a constant as the number of available frequency bands is fixed in a given network. We use unit disks to reflect conflicts between two nodes, where each node is treated as a disk centered on itself. There is a conflict edge in E if any two intersected disks access the same channel simultaneously. The network is time-slotted with global synchronization. At each round t, node v_i has M choices of channels, where channel c_j having data rate drawn from an i.i.d stochastic process $\xi_{i,j}(t)$ over time with a mean $\mu_{i,j} \in [0, 1]$, as done in many previous works [13] [5] [6] . Without loss of generality, we assume the same channel may demonstrate different channel quality for different users. For the same channel c_j, the random process $\xi_{i,j}(t)$ is independent from $\xi_{i',j}(t)$ if $i \neq i'$.

TABLE I: Summary of notations

Variable	Meaning
N	number of SUs
M	number of channels
$J_{G,r}(v)$	r-hop neighborhood of node v in G
$J_{H,r}(v)$	r-hop neighborhood of vertex v in H
$A_r(v)$	set of Candidate vertexes in r-hop neighborhood of v
MWIS(I)	maximum weighted independent set for vertex set I
$W(I)$	summed weight of all vertexes in vertex set I
$\mathbf{s}_x(t)$	strategy decision for round t
$T_x(n)/T_{\beta,x}(n)$	number of times that strategy $s_x/s_{\beta,x}$ has been played so far
$\Delta_x/\Delta_{\beta,x}$	distance between $R_1/\frac{1}{\beta}R_1$ and mean throughput of strategy $\mathbf{s}_x$
$\Delta_{\min}/\Delta_{\beta,\min}$	minimum of $\Delta_x/\Delta_{\beta,x}$
$\xi_{i,j}(t)$	data rate for channel c_j experienced by user v_i
$\mu_{i,j}$	mean of $\xi_{i,j}(t)$ over time

At each round t, an N-dimensional *strategy* vector $\mathbf{s}_x(t) = \mathbf{s}_x = \{s_{x,i} | i = 1, \dots, N\}$ is selected under some *policy* from the *feasible strategy set* F. Here $s_{x,i}$ is the index of channel selected by node v_i in strategy $\mathbf{s}_x$. We use $x = 1, \dots, X$ to index strategies of feasible set F in the decreasing order of average throughput $\lambda_x = \sum_{i=1}^{N} \mu_{i,s_{x,i}}$. By feasible we mean that all nodes can transmit simultaneously without conflict. When a strategy $\mathbf{s}_x$ is determined, each node v_i observes the data rate $\xi_{i,s_{x,i}}(t)$ of its selected channel, and then the total throughput of the network at t is defined as, $R_x(t) = \sum_{i \in \mathbf{s}_x} \xi_{i,s_{x,i}}(t)$. We evaluate policies using *regret*, which is defined as the difference between the expected throughput that could be obtained by a static optimal policy, and that obtained by the given policy. Let $R_1 = \lambda_1$ be the optimum fixed channel access strategy, then regret up to current round n can be expressed as

$$\Re(n) = nR_1 - E\left[\sum_{t=1}^{n} R_x(t)\right] = \sum_{x: R_x < R_1} \Delta_x E\left[T_x(n)\right].$$

III. Problem formulation

To model the multi-channel scenario, we remodel the original conflict network $G = (V, E, C)$ as an extended conflict graph $H = (\widetilde{V}, \widetilde{E})$, where $\widetilde{V} = \{v_{i,j} \mid i \in [1, N], j \in [1, M]\}$, and show that the problem can be reformulated as a MWIS problem in extended conflict graph H. For each node i, define a set of virtual vertices $\{v_{i,j}, j = 1, \dots, M\}$ and connect $v_{i,j}$ with $v_{i,k} (j \neq k)$ for all j, k. Node v_i is master node of virtual vertex $v_{i,j}$, while $v_{i,j}$ is slave of v_i. Connect $v_{i,j}$ with $v_{p,j}$ if i and p has an edge in original network G. Then graph H has $N \times M$ vertexes. We give an instance in Fig. 1 where the original network G has 3 available channels and 3 nodes.

We first analyze the optimum solution assuming each random variable is known. Let each vertex associate with a weight of $\xi_{i,j}(t)$. As each node of G has a clique of virtual vertexes in H, and vertexes with the same channel index retain the conflict relationships of master nodes in G, then it is straightforward that a MWIS of H is a throughput-optimal allocation of channels in G. Moreover, an IS of H one-to-one maps to a

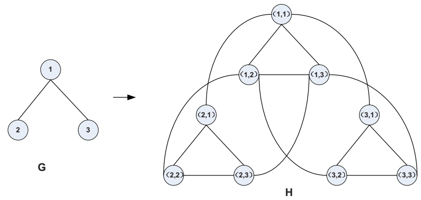


Fig. 1: Original conflict graph G to extended conflict graph H

feasible strategy in F. Therefore, the feasible strategy set F consists of all independent sets (**IS**) of vertexes in H. Here note that the *independence number*[1] of H is less than N if the chromatic number of G is greater than M, and is N otherwise. The actual length of a feasible strategy may be smaller than N if some nodes do not choose any channel. If the mean of $\xi_{i,j}(t)$ is known, the optimum strategy is to find a MWIS from H as choices made by nodes in G, i.e,

$$R_1 = \max_{\mathbf{s}_x \in F} \sum_{i=1}^{N} \mu_{i,s_{x,i}}$$
$$\text{s.t. } F \text{ is feasible strategy set.} \tag{1}$$

Recall that these random variables are unknown actually, each user needs to learn and estimate the weight of each strategy, denoted by $W_x(t) = \sum_{x_i \in \mathbf{s}_x} w_{i,s_{x,i}}(t)$, where $w_{i,s_{x,i}}(t)$ is estimated weight of random variable $\xi_{i,s_{x,i}}(t)$. Thus, our problem becomes a NP-hard combinatorial multi-armed bandits problem that selects at most N arms (i.e., vertexes in H) out of $K = NM$ ones to minimize the regret $\mathfrak{R}$, such that these arms are independent from each other in H. For brevity, we map the channel index $s_{x,i}$ of node v_i to arm index $k = (i-1)N + s_{x,i}$.

For NP-hard combinatorial multi-armed bandits problems, a weaker vision of regret, called β*-regret* [14], is defined as the difference between the expected throughput that is $1/\beta$ of the optimum, and that gained throughput (a β-approximation policy which instead yields a strategy with learned weight at least $1/\beta$ of the maximum possible weight) . Let $R_{\beta,x}(t)$ be the reward of strategy $\mathbf{s}_{\beta,x}$ generated by the β-approximation policy, then β-regret can be expressed as

$$\mathfrak{R}_\beta(n) = \sum_{R_{\beta,x} < R_1/\beta} \Delta_{\beta,x} E\big[T_{\beta,x}(n)\big] + \sum_{R_{\beta,x} \ge R_1/\beta} \Delta_{\beta,x} E\big[T_{\beta,x}(n)\big].$$

Here in feasible strategy set of a β-approximation policy, strategies can be divided into two sets, i.e., a set of β-approximation strategies and a set of non-β-approximation strategies. A β-approximation strategy is a strategy with mean throughput of R_1/β at least, and a non-β-approximation strategy is one with mean throughput less than R_1/β. Thus we have negative $\Delta_{\beta,x}$ for β-approximation strategies and positive $\Delta_{\beta,x}$ for non-β-approximation strategies.

IV. Channel access

We divide each round of channel accessing into two sequent parts, one for strategy decision and the other for data transmission. In the former part, we learn to select the best strategy for current time. In the later part, users access corresponding channels to transmit data, and observe real data rate after transmission. We assume a common control channel for control message passing in strategy decision.

A. The learning policy

Though the learning policy in [12] achieves zero-regret, the upper bound of regret $\mathfrak{R}$ (or β-regret $\mathfrak{R}_\beta$) including a factor of $\frac{1}{\Delta_{\min}}$ (or $\frac{1}{\Delta_{\beta,\min}}$) becomes vacuous if $\Delta_{\min}$ (or $\Delta_{\beta,\min}$) $\to 0$. Thus we adopt the learning policy in [14], where the upper bound of regret is independent with $\Delta_{\min}$ (or $\Delta_{\beta,\min}$). The centralized form of the learning policy is shown in Algorithm 1, where in (3) a independent set of vertexes with maximum estimated weight are selected among $\widetilde{V}$ as the strategy decision. The estimated value for actual weight $\xi_{s_{x,i}}(t+1)$ of vertex $v_{s_{x,i}}$ is

$$w_{s_{x,i}}(t+1) = \tilde{\mu}_{s_{x,i}}(t) + \sqrt{\frac{\max\left(\ln \frac{t^{2/3}}{K m_{s_{x,i}}}, 0\right)}{m_{s_{x,i}}}}. \tag{2}$$

where $\tilde{\mu}_{s_{x,i}}$ is observed mean of $\xi_{s_{x,i}}$ up to the current round, and $m_{s_{x,i}}$ is the number of times that channel $\xi_{s_{x,i}}$ has been selected so far.

Algorithm 1 Learning policy

1: For each round t, select a strategy $\mathbf{s}_x$ by maximizing

$$\max_{\mathbf{s}_x \in F} \sum_{s_{x,i} \in \mathbf{s}_x} \left(\tilde{\mu}_{s_{x,i}}(t) + \sqrt{\frac{\max\left(\ln \frac{t^{2/3}}{K m_{s_{x,i}}}, 0\right)}{m_{s_{x,i}}}} \right). \tag{3}$$

Clearly, it only requires to store and update estimation for MN vertexes that costs storage and computation linear with MN, instead for M^N strategies in F that costs storage and computation linear with M^N. It costs two $1 \times K$ vectors to store and update the estimated weight. One is $(\tilde{\mu}_k)_{1\times K}$, and the other is $(m_k)_{1\times K}$. After data transmission on the channels of chosen strategy $\mathbf{s}_x$ in round t, actual weight $\xi_{s_{x,i}}(t)$ is observed for all $s_{x,i} \in \mathbf{s}_x$. Then $(\tilde{\mu}_k)_{1\times K}$ and $(m_k)_{1\times K}$ are updated in the following way:

$$\tilde{\mu}_k(t) = \begin{cases} \frac{\tilde{\mu}_k(t-1)\cdot m_k(t-1) + \xi_k(t)}{m_k(t)} & \text{if } k \in \mathbf{s}_x, \\ \tilde{\mu}_k(t-1) & \text{else.} \end{cases} \tag{4}$$

$$m_k(t) = \begin{cases} m_k(t-1) + 1 & \text{if } k \in \mathbf{s}_x, \\ m_k(t-1) & \text{else.} \end{cases} \tag{5}$$

Due to NP-hardness of the MWIS problem in (3), it is desirable to solve it approximately while retaining zero-regret. The following theorem shows that, for *any* algorithm with approximation ratio at least $1/\beta$, the regret on the achieved throughput is bounded.

Theorem 1: [14] The β-approximation learning policy has

$$\sup \mathfrak{R}_\beta(n) \le \frac{1}{\beta} NK + \left(\sqrt{eK} + \frac{16}{e\beta}(1+N)N^3 \right) n^{\frac{2}{3}}$$
$$+ \frac{1}{\beta}\left(1 + \frac{4\sqrt{K}N^2}{e\beta^2}\right) N^2 K n^{\frac{5}{6}} \tag{6}$$

[1]The cardinality of the maximum independent set that is an independent set of largest possible size for a given graph.

without dependency on $\Delta_{\beta,\min}$. The supremum is taken over all X-tuple of probability distributions on $[0, 1]$.

B. Centralized approximation solution for channel access

Intuitively, the greater the value β is, the more loss on overall throughput it causes, compared to the optimal throughput. Given that, it is beneficial to employ a PTAS to solve the MWIS problem. Compared with other existing PTAS schemes [15] [16], the robust PTAS in [17] seems the most suitable candidate for its elegance and non- requirement on geometric information. The key problem is that the theoretical results are no longer directly applicable to the extended conflict graph H because of its complex structures. We need to prove the correctness and new theoretical bounds of the robust PTAS in H. For better understanding, first we summarize the basic idea of robust PTAS for unit disk graphs.

Robust PTAS. We begin with some notations. Given a unit disk graph $G = (V, E)$ with a set V of nodes and a set E of edges, an edge $(u, v) \in E$ if the Euclidean distance $\|u, v\| < 2$. For a subset I of nodes in V, let $W(I)$ denote the total weight of I, i.e., $W(I) = \sum_{v_i \in I} w_i$, and MWIS$(I)$ denote a maximum weighted independent set for I. Let $d_G(u, v)$ be the minimum hop of any path connecting u and v in G. Define

$$J_{G,r}(v) := \{u \in V \mid d_G(u, v) \leq r\}$$

be the *r-hop neighborhood* of v in G. The *r-hop distance of* G, $L_{G,r}(v)$, is the maximum Euclidean distance between v and neighbors in $J_{G,r}$. Clearly $L_{G,r}(v) < 2r$.

Let $\epsilon > 0$ and $\rho := 1 + \epsilon$ denote the desired approximation guarantee. In graph G, the algorithm starts with a node of maximal weight $w_{\max} = \{\max w_v | v \in V\}$, and then computes MWIS$(J_{G,r})$ as long as $W(\text{MWIS}(J_{G,r+1})) > \rho W(\text{MWIS}(J_{G,r}))$ holds. Let $\bar{r}$ denote the smallest r for which the criterion is violated. It has been proved that $\bar{r}$ is a constant for a specific ρ, i.e., $\rho^r \leq (2r + 1)^2$. We then remove MWIS$(J_{G,\bar{r}}(v_{\max}))$ and all the adjacent vertices from G, and repeat the above process on the remaining graph. Then the union of all removed independent sets form an independent set, and it is proved that it is ρ-approximation for the MWIS of unit disk graph G.

As the extended conflict graph H is not a strict unit disk graph, we distinguish some notations. Define *r-hop neighborhood in extended graph* H as

$$J_{H,r} = J_{H,r}(v) := \{u \in \widetilde{V} \mid d_H(u, v) \leq r\},$$

where $d_H(u, v)$ be the minimum hop of any path connecting u and v in H. The *r-hop distance of* H, $L_{H,r}(v)$, is the maximum Euclidean distance between v and neighbors in $J_{H,r}$. Note that two vertexes that belong to the same master node of G has 0 Euclidean distance geometrically, but they are 1-hop neighbors in H. The r-hop distance of H also satisfies $L_{H,r}(v) = L_{H,r} < 2r$. We then have the following theorem on approximation ratio achieved by robust PTAS in H.

Algorithm 2 Main framework of distributed channel access

for Round $t = 1, \ldots, n$, $\forall v \in \widetilde{V}$ **do**
  if v belongs to strategy decision $\mathbf{s}_x(t-1)$ in previous round **then**
    Broadcast its new weight in $(2r + 1)$-hop neighborhood.
  end if
  Receive all updated weight on $(2r + 1)$-hop neighbors, and update corresponding weight.
  Perform distributed Robust PTAS as Algorithm 3 within D mini-rounds.
  if v is marked as *Winner* **then**
    Access the channel to transmit data.
    Observe actual data rate.
    Update estimated weight using Equation (4), (5) and (2).
  end if
end for

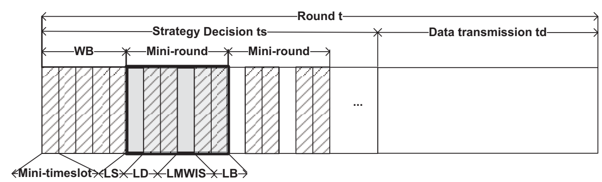


Fig. 2: Structure of a single round:**WB**-weight update; **LS**-LocalLeader selection; **LD**-LocalLeader declaration; **LMWIS**-local computation of MWIS; **LB**-local broadcast of status determination.

Theorem 2: Robust PTAS applies to extended conflict graph H with approximation ratio ρ, where $\rho^r = M \cdot (2r + 1)^2$.

Proof: We note that Robust PTAS can be extended to other intersection graphs as long as the graph is growth-bounded, i.e., the number of independent vertexes in a vertex's r-hop neighborhood is constantly bounded [18] [17]. As H is not a strict unit graph, we then verify that H is growth-bounded. Note that a set of virtual vertexes that belong to the same master node form a clique in H. For a node $v_i \in V$ in G, the independent number of $J_{G,r}(v_i)$ is upper bounded by $(2r + 1)^2$. As each node in G will define M slave vertexes in H, a simple pigeonhole principle shows that the number of independent vertexes in the r-hop neighborhood $J_{H,r}(v)$ of graph H is bounded from above by $M \cdot (2r + 1)^2$. Thus we say H is also growth-bounded, and the approximation ratio achieved in H satisfies $\rho^r \leq M \cdot (2r + 1)^2$. ■

C. Distributed channel access

As the centralized form of robust PTAS algorithm requires centralized computation and global collection of weight/observed information, it costs high computation (i.e., $O(N^{\rho^r})$) and communication complexity that is unwelcome in multi-hop networks. We design a distributed PTAS that can achieve the same approximation as the centralized form, but with much lower cost.

1. Main framework

The main framework of our distributed design is shown in Algorithm 2. For better explanation, we describe it in the role of virtual vertexes, actual computations can be executed by their master nodes instead. Each virtual vertex maintains information on local neighbors in H from the initial round.

In the strategy decision part, Algorithm 2 includes an initiation step called Weight Broadcast (**WB**), where each vertex broadcasts its new weight if it accessed channel in previous round (i.e., included in previous strategy decision $\mathbf{s}_x(t-1)$), to ensue computation of MWIS with newest weight. These vertexes in $\mathbf{s}_x(t-1)$ require to broadcast updated weight information within hops $(2r+1)$ to ensure independence of the final output, for which we will explain later. Let *mini-timeslot* be the time unit required for a round of communication between two connected vertexes. In the first round, the initial weight of each vertex is 0, so vertexes can be randomly selected as LocalLeader, or they can use their IDs as weight. In the later case, it will cost $O(N)$ mini-timeslots to collect IDs of all neighbors even in a local neighborhood. In next rounds, however, it costs only $O((2r+1)^2)$ mini-timeslots to finish the WB process. The key observation is that within any $(2r+1)$-hop neighborhood of any vertex, at most $O((2r+1)^2)$ vertexes are selected as independent vertexes. Only independent vertexes included in a strategy decision observe new values, and utilize the observation to update estimated weight (i.e., plugging (4) and (5) into (2)). If each vertex performs weight broadcast sequently, obliviously it will take $O((2r+1)^3)$ mini-timeslots to finish the whole procedure in a $(2r+1)$-hop neighborhood. As an alternative, these selected vertexes can efficiently broadcast their weight using pipeline methods such as constructing a connected dominating set or scheduling local broadcast [19] [20] [21], by which number of mini-timeslots can be reduced to $O((2r+1)^2)$.

After WB, each vertex then runs distributed Robust PTAS presented in Algorithm 3 to compute MWIS with updated weight. In our protocol, we will run D *mini-rounds* to output a final IS with a good approximation ratio to the optimum. When finishing execution of Algorithm 3, the vertexes included in current strategy decision access channels for data transmission, where they obtain new observation to update estimation of weight for the next round. Until now a full round of Algorithm 2 completes, and a new round follows.

2. Distributed robust PTAS

Now we describe distributed Robust PTAS in Algorithm 3. We introduce four statuses in Algorithm 3: *Candidate*, *LocalLeader*, *Winner* and *Loser*. A *Candidate* is one vertex that is not marked as Winner or Loser, and thus has opportunity to be a Winner. Initially, at the start of each round, each node is marked as *Candidate*. A *LocalLeader* is a Candidate that has the maximum weight among all its Candidate neighbors in $(2r+1)$-hop neighborhood. Each LocalLeader will compute the maximum weighted independent set using all *Candidate* vertexes in its r-hop neighborhood. A *Winner* is a vertex that is included in the final resulting IS computed from LocalLeader, while a *Loser* is a vertex that is neither Candidate nor Winner.

Notice that here we use the $(2r+1)$-hop neighborhood to find a *LocalLeader* while use r-hop neighborhood to compute an IS. This approach will assure that the union of all the independent sets computed by all selected LocalLeaders form an independent set, as the hop-distance between any two LocalLeaders is at least $2r+2$ and the hop-distance between any two vertexes from the computed independent sets by two LocalLeaders is at least 2. It is straightforward that the hop-distance between any two LocalLeaders is at least $2r+2$, as a LocalLeader has

---

Algorithm 3 Distributed robust PTAS for strategy decision at each vertex

Initialization: $\forall$ vertex $v \in \widetilde{V}$, marked as status *Candidate*, have collected newest weights of all $(2r+1)$-hop neighbors $J_{H,2r+1}(v)$.

for mini-round $\tau = 1, 2, \ldots, N$ **do**
  if v is *Candidate* **then**
    if $w_v \geq \max\{w_u | u \in A_{2r+1}(v)\}$ **then**
      v is marked as *LocalLeader* and declare in $(2r+1)$-hop neighborhood.
    end if
  end if
  if v is *LocalLeader* **then**
    Compute a local MWIS$(A_r(v))$ using enumeration.
    Determine status of r-hop neighbors. For any *Candidate* vertex in $A_r(v)$, marked as *Winner* if it is in MWIS$(A_r(v))$, or marked as *Loser* otherwise.
    Locally broadcast the results within $(3r+1)$-hop neighborhood of the LocalLeader.
    Update its own status accordingly.
  else if v is *Candidate* **then**
    if v receives determination messages **then**
      Update status of itself and $(2r+1)$-hop neighbors accordingly.
    end if
  end if
end for

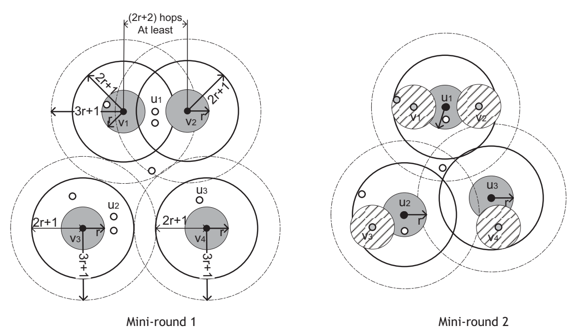


Fig. 3: Illustration of Algorithm 3 in two sequent mini-rounds, where vertexes $v_i, i = 1, 2, 3, 4$ are selected as LocalLeader at mini-round 1, and vertexes $u_i, i = 1, 2, 3$ become LocalLeader at mini-round 2 after neighbors with bigger weight excluded.

the maximum weight in its $(2r+1)$-hop neighborhood. For the case that there are more than 1 maximum weighted vertexes, they can further use IDs to break the tie. As to the minimum 2-hop distance between any two vertexes from the computed independent sets by two LocalLeaders, it ensures that, in the case where the two vertexes are selected to access the same channel, the two vertexes can still keep independent form each other, i.e., they are not in each other's 1-hop neighborhood. Please note that here the hop-distance between two vertexes u and v is $d_G(u, v)$, not Euclidean distance geometrically. Our distributed implementation preserves the merit of Robust PTAS that does not require any geometrical information.

Let $A_r = A_r(v)$ be the set of all Candidate vertexes in $J_{H,r}(v)$ to exclude vertexes that have been marked as Winner or Loser. The algorithm begins with the process called LocalLeader selection (Line $2-6$). To ensure independency of the union of all local computed results, each LocalLeader compute local MWIS within r-hop neighborhood. A LocalLeader has to broadcast its computed MWIS results among $(3r+1)$-hop neighborhood (Line 10), so that Candidate vertexes in the next round have complete status information on its $(2r+1)$-hop neighbors to correctly continue the algorithm. Notice that a Candidate vertex, say u, in the current round could become a LocalLeader in the next round. For this to happen, it must be the case that 1) at current round, there is a virtual vertex, say x, within its $(2r+1)$-hop whose weight is larger, 2) after this round, the virtual vertexes with larger weight change their status (either they are LocalLeaders or they are decided by other LocalLeaders as Winner or Loser). Thus, to assure correct operation, the status of a virtual vertex, say u, should be broadcast by its LocalLeader, say v, to the $3r+1$ hops, as the hop distance between u and v could be as large as $3r+1$.

For better understanding, we illustrate distributed execution of Algorithm 3 in two sequent mini-rounds in Fig. 3, and local computation in a single mini-round in Fig. 4 for the network presented in Fig. 1. In Fig. 3, vertex v_1, v_2, v_3, v_4 (in black) are initially selected as LocalLeader since they have the maximum weight in their $(2r+1)$-hop neighborhood. After LocalLeader declaration, the four LocalLeader vertexes respectively compute local MWIS and determine status for candidates within their r-hop neighborhood, i.e., white vertexes in gray circles. As shown in figure of mini-round 2, these white vertexes as well as LocalLeader vertexes changed their status, and new LocalLeader vertexes are selected among those remaining Candidate vertexes. For example, we can see that u_1 becomes LocalLeader when $(2r+1)$-hop neighbors (e.g., v_1 and v_2) that have greater weight changed their status from Candidate to Winner after mini-round 1. The same happens for u_2 and u_3, which shift their status from Candidate to LocalLeader.

At each mini-round, vertexes either marked as Winner or Loser will be excluded and stop executing the algorithm. The algorithm terminates when no Candidates exist, i.e., all vertexes are marked as either Winner or Loser. The following theorem shows the theoretical bounds of Algorithm 3.

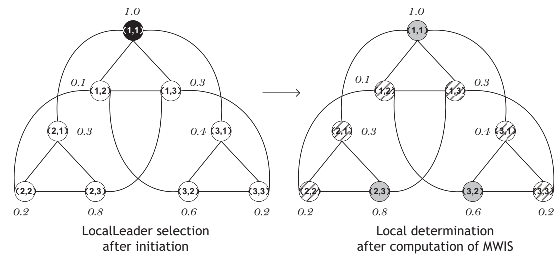


Fig. 4: Local computation of Algorithm 3 in extended graph H with estimated weight: black vertex-*LocalLeader*, white vertex-*Candidate*, stripe vertex-*Loser*, gray vertex-*Winner*.

Theorem 3: Algorithm 3 achieves the same approximation ratio ρ as the centralized robust PTAS in H.

Proof: Let $v_i(\tau)$ be a LocalLeader selected at mini-round τ. In each mini-round, a LocalLeader $v_i(\tau)$ utilizes the robust PTAS to find MWIS(A_r) in its effective r-hop neighborhood. Thus, each MWIS(A_r) computed by a LocalLeader is ρ-approximation to MWIS(A_{r+1}). Let MWIS$(\widetilde{V})$ be the global optimum, and I be intersection of MWIS$(\widetilde{V})$ and $A_{r+1}(v_i(\tau))$, we have $W(I) \leq \rho W(\text{MWIS}(A_{r+1}))$. As union of A_r in all mini-rounds is exactly $\widetilde{V}$ and any two distinct A_r do not intersect, we have the union of all MWIS(A_r) output by all LocalLeaders is ρ-approximation to the global optimum MWIS$(\widetilde{V})$ in weight. ■

3. Complexity analysis

We summarize complexity in a complete round.

Communication complexity: As shown in Fig. 2, local broadcast happens 3 times in each round, respectively for WB, LD, and LB. WB could be finished within mini-timeslots $O((2r+1)^2)$, which costs each vertex $O((2r+1)^2)$ number of messages in worst case. LD is done by a LocalLeader in its $(2r+1)$-hop neighborhood, then it costs $O(2r+1)$ mini-timeslots, and each vertex $O(1)$ passing messages. In LB, each LocalLeader has to broadcast the results within its $(3r+1)$-hop neighborhood. There are at most $\frac{2\pi \cdot 2r}{2r+1} = O(1)$ number of LocalLeaders within any $(2r+1)$-hop neighborhood of any vertex. Thus it costs mini-timeslots $O(3r+1)$, and communication complexity $O(1)$. Totally, it requires mini-timeslots $O(r^2 + Dr)$, and each vertex number of passing messages $O(r^2 + D)$.

Computation complexity: The main computation cost is caused by LMWIS, as LS can be finished instantly. In every mini-round, we use complete enumeration to compute local MWIS in each $A_r(v)$. Suppose there are m nodes in corresponding r-hop neighborhood of G, then $|A_r(v)| \leq Mm$. Since $|\text{MWIS}(A_r(v))| \leq M(2r+1)^2$, there are totally $C_{Mm}^{M(2r+1)^2}$ enumerations. Using $M(2r+1)^2 < \rho^r$ for $r < \bar{r}$, we have

$$C_{Mm}^{M(2r+1)^2} \leq \left(\frac{Mme}{M(2r+1)^2}\right)^{M(2r+1)^2} \leq \left(\frac{me}{(2r+1)^2}\right)^{\rho^r}. \quad (7)$$

Hence, it requires polynomial time $O(m^{\rho^r})$ per mini-round, and $O(Dm^{\rho^r})$ per round. In practice, we can use more efficient constant approximation algorithm instead, the communication complexity reduces to $O(D)$ with a worse approximation ratio.

Space complexity: Each vertex has to store information on weight and status of neighbors within $(2r+1)$-hop neighborhood, therefore the space complexity is $O(m)$ at each vertex.

D. Practical regret

Now we analyze *practical regret* (or *effective throughput*) that considers the missed throughput due to time spent on learning. Let t_a and t_m respectively be length of a single round and mini-round. Time for strategy decision and data transmission denoted by t_s and t_d. In the strategy decision, supposing it requires c mini-rounds, one for weight update, others for strategy decision, then $t_a = t_s + t_d = ct_m + t_d$. The actual data rate gained at each round is $R_x(t) \cdot t_d/t_a = \theta R_x(t)$, where $\theta = t_d/t_a$. The actual distance between R_1/α and a strategy $\mathbf{s}_x$ is $R_1/\alpha - \theta\lambda_x = \theta\Delta_{\theta\alpha,x}$. Thus in a round, the more time for learning, the larger regret it will be. In practice, we cannot use very long round as t_a shall be smaller than channel coherence time.

Using $\beta = \theta\alpha$ as the approximation ratio, and $\Delta_{\beta,X} = \theta\Delta_X$ as the maximum distance between the actual mean throughput of $R_1/\theta\alpha$ and $\mathbf{s}_X$, we obtain the practical regret is less than $\theta \cdot \mathfrak{R}_{\theta\alpha}(n)$ according to [14], i.e.,

Theorem 4: The practical regret of Algorithm 2 satisfies

$$\sup \theta\mathfrak{R}_{\theta\alpha}(n) \le \frac{1}{\alpha}NK + \left(\theta\sqrt{eK} + \frac{16}{e\alpha}(1+N)N^3\right)n^{\frac{2}{3}} + \frac{1}{\alpha}\left(1 + \frac{4\sqrt{K}N^2}{e(\theta\alpha)^2}\right)N^2Kn^{\frac{5}{6}}. \tag{8}$$

Then our channel allocation scheme can guarantee an effective throughput of $R_1/(\theta\alpha) - \theta\mathfrak{R}_{\theta\alpha}(n)$.

V. Simulations

Now we conduct simulations for our proposed channel accessing scheme under various networks. In all simulations we run Algorithm 3 with $r = 2$. We set 8 types of channels with data rates (units kbps) 150, 225, 300, 450, 600, 900, 1200, and 1350 respectively [8]. Each channel evolves as a distinct i.i.d Gaussian stochastic process over time. We set each round as length of a unit time slot. Referring to a cognitive radio system [8], we list the values of time parameters of a round in Table II. We set decision time $t_s = 4t_m$. Let t_b be time to finish local broadcast and t_l be the total time for local computation, we then have $t_m = 2t_b + t_l = 250ms$. According to Fig. 2, in our setting the actual throughput gained at each round would be at most $\frac{t_d}{t_d+4t_m}R_x(t) = 0.5R_x(t)$ even if we can compute optimal MWIS in the limited time.

TABLE II: PARAMETER VALUES FOR SIMULATION

round t_a	2000ms	local broadcast t_b	100ms
local computation t_l	50ms	data transmission t_d	1000ms

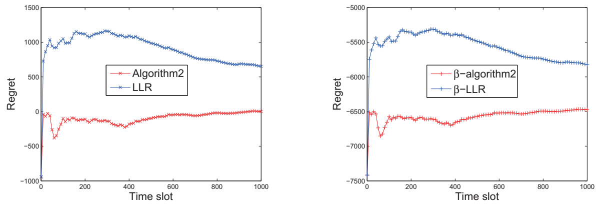
We also do comparison with LLR learning policy [12]. The LLR learning policy works as follows. At each round, the LLR policy select a strategy maximizing

$$\max_{\mathbf{s}_x \in F} \sum_{s_{x,i} \in \mathbf{s}_x} \left(\tilde{\mu}_{s_{x,i}}(t) + \sqrt{\frac{(MN+1)\ln t}{m_{s_{x,i}}}}\right).$$

A. Regret analysis

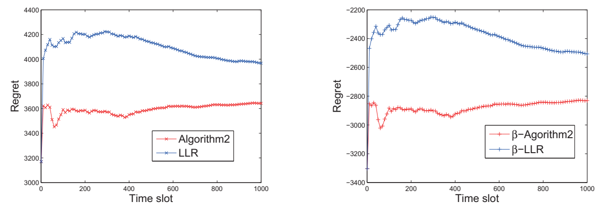
We will study ideal regret/β-regret, and practical regret/β-regret of our proposed algorithm, and compare these metrics with LLR learning policy [12]. According to definition of regret and β-regret, we need to compute the static optimal throughput. As the MWIS problem is NP-hard, we construct a small network where we could find the optimum by brute force easily. Here we randomly generate a connected network with 15 users, each having 3 channels available. Using mean data rate of each channel as weight, we obtain the static optimum of 7282.90.

We then compare the optimal throughput, $1/\beta$ of the optimal throughput, respectively with the observed throughput for ideal regret without consideration of loss by time on strategy decision, and with effective throughput for practical regret with consideration of loss by time on strategy decision. The results of ideal regret/β-regret are shown in Fig. 6, and that of practical regret/β-regret are shown in Fig. 7, both of which plots changes of corresponding regret/β-regret as time increases, In both figures, our proposed algorithm outperforms the LLR learning policy. The ideal regret of our algorithm converges to 0 much faster than the LLR learning policy, and thus produces much less regret than it, e.g., the regret by our learning policy is around 0 after 200 time slots, while it is more than 1000 by the LLR learning policy. For the practical regret, it is far beyond 0 for both of the learning policies, which indicates a significant impact caused by the time on learning. However, the result of our policy coincides with the theoretical analysis that the effective throughput is half of the idea optimal throughput in our setting. The regret of LLR policy is much greater than ours, indicating a smaller practical throughput. As to β-regret and practical β-regret, recall that when the reward of selected strategy is greater than $1/\beta$ of the best reward, the corresponding regret could be negative. Fig. 7 (b) also shows that the practical β-regret converges to a negative value, indicating that the achieved throughput by both algorithms is much better than $1/\beta$ of the idea optimum, even considering missed throughput on learning.



(a) Ideal regret (b) Ideal β-regret

Fig. 5: Ideal regret/β-regret with every-time-slot update



(a) Practical regret (b) Practical β-regret

Fig. 6: Practical regret/β-regret with every-time-slot update

B. Throughput performance under unfrequent update

We evaluate the effective throughput under different frequencies of weight update in this series of simulation, where meanwhile we compare performance of our learning policy with LLR policy. We conduct experiments in two networks, one is a uniformly random located network with 100 users and 10 channels, the other is a part of Citysee wireless network that has been deployed in City Wuxi, China. The part of Citysee topology we adopt has 446 users, and each one is assumed to have 5 available channels. In the random network, as the distribution and density is well controlled, we can finish local MWISL using enumeration within short time. While in the Citysee topology, as it is not a regularly random network with uniform and low density, we have to choose faster greedy algorithm in computation of local MWISL for each LocalLeader, so that time spent on local computation of MWIS is in t_l.

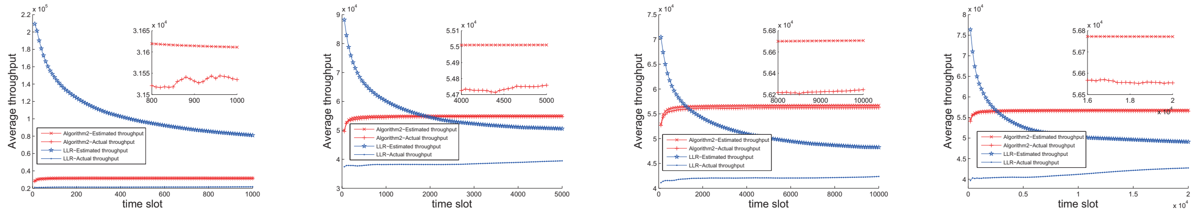
In our proposed algorithm, initially each vertex has to collect weight of neighbors inside $(2r+1)$-hop neighborhood. If weight as well as corresponding strategy decision is updated at every time slot, it will cause high communication and communication cost that significantly affects effective throughput of data transmission. Instead, we can update weight every period P that consists of y time slots. Then we just need to do strategy decision at the beginning, and repeat data transmission y times. The length of a period is $t_P = yt_a$. The actual average throughput gained at the z^{th} period is $R_P(z) = \frac{R_x(zy+1)t_d + \sum_{t=zy+2}^{(z+1)y} R_x(t)t_a}{yt_a}$. For large scale networks, we will not compute the best static strategy as it can not be finished instantly. Instead, we record the average observed throughput up to z period $\widetilde{R}_P(z)$, where $\widetilde{R}_P(z) = \frac{(z-1)\widetilde{R}_P(z-1)+R_P(z)}{z}$, and average estimated throughput $\widetilde{W_P}(z)$ (i.e., average estimated weight of all selected strategies throughput up to z). Let $W_P(z)$ be average estimated throughput at z^{th} period, we have $W_P(z) = \frac{[(y-1)t_a+t_d]W_x(zy+1)}{yt_a}$, and $\widetilde{W}_P(z) = \frac{(z-1)\widetilde{W}_P(z-1)+W_P(z)}{z}$. The difference between $\widetilde{R}_P(z)$ and $\widetilde{W}_P(z)$ can also indicate the throughput performance of algorithms.

We study the frequent case with $y = 1$, and unfrequent cases with stale weight that is updated periodically with $y = 5, 10, 20$ time slots. We conduct each experiment respectively in $1000, 5000, 10000, 20000$ time slots, each updating weight 1000 times. The actual effective throughput will be around $1/2, 9/10, 19/20, 39/40$ of the ideal throughput without time consuming on strategy decision. In Fig. 8 and Fig. 9, we can find that the average actual throughput achieved by both of the algorithms grows to the ideal throughput as a period lasts more time slots. For instance, in Fig. 8, the average actual throughput by our algorithm grows along $31535, 54757, 56245$ to 56554, and that by LLR policy grows along 21165, 39446, 42378 to 42490, when the number of time slots in a period increases from 1 to 20. Especially, a significant improvement can be seen between the frequent case (Fig. 8(a) and Fig. 9(a)) and the unfrequent case of $y = 5$ (Fig. 8(b) and Fig. 9(b)). In the later two cases, further improvement is not so obvious as the proportion of time on learning decreases much more slowly. They collaboratively show that unfrequent update has negligible impact on accuracy of estimation, but significantly improvement on effective throughput.

We then analyze throughput performance of the two learning policies. In each case, we can find that our adopted learning policy is much more accurate than the LLR learning policy. The difference between the estimated average throughput and the actual throughput is quite small in our adopted learning policy, while it is large in the LLR policy. In the figures, the difference between estimated throughput and actual throughput of our algorithm is not clear. Thus we show a zoom-in part of the difference on the upper right of each figure, where we can see small divergency in the last 200 updates. In these figures, it shows that the actual throughput achieved by our learning policy is much better the LLR policy. The huge difference between our policy and LLR policy may be contributed by the convergence speed of the two learning algorithms. From these figures and theoretical design of the two algorithms, it is straightforward that the learning policy we adopted can distinct better strategies much more quickly, while the LLR policy needs much longer time on learning for the best strategy.

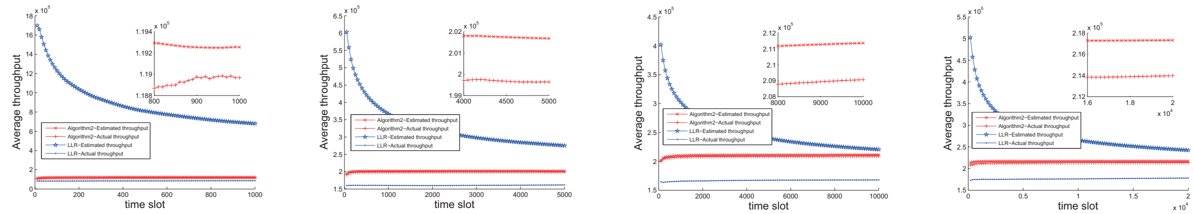
VI. Related works

There is a rich body of results on dynamic spectrum access in CRNs. As channel availability and quality is unknown to SUs, they need to conduct a learning process to select good channels. Several literatures address this problem from sequential



(a) 1 time slot every period (b) 5 time slots every period (c) 10 time slots every period (d) 20 time slots every period

Fig. 7: Estimated v.s. actual average effective throughput with different period update under random topology



(a) 1 time slot every period (b) 5 time slots every period (c) 10 time slots every period (d) 20 time slots every period

Fig. 8: Estimated v.s. actual average effective throughput different period update under Citysee topology

decision perspective by MAB approaches, and several from a game theoretic perspective by convergence of equilibrium.

The results using MAB start from single-user play [22] [23], where each channel evolves as independent and identically distributed Markov process with good or bad state. The results are then extended to multi-user play where $N > 1$ secondary users select channels among M ones [1], [2], [3], [4], [5], [6], [7]. These works basically assume channel quality evolving with i.i.d stochastic process over time, and a single-hop network setting where conflict happens if any pair of users choose the same channel simultaneously. For instance, Shu and Krunz [13] propose a throughput-optimal decision strategy with stochastic homogeneous channels. This optimal strategy has a threshold structure that indicates whether the channel is good or bad. Anandkumar et al. [6] propose two distributed learning and allocation schemes respectively for the case of pre-allocated ranks among SUs and non such prior information.

On the other hand, some results consider dynamic spectrum access from an adaptive, game theoretic learning perspective. M. Maskery, et al. [24] model the dynamic channel process as a non-cooperative game for stochastic homogeneous channels, and basically rely on CSMA mechanism to estimate probability of channel contention. In the case of heterogeneous channel quality, Xu et al. [25] construct a potential game to maximize the expected throughput of all secondary users. They implicitly assume a single-hop network case where all users have the same probability to access channels.

We also review the results on network capacity, and related link scheduling problem that maximizes the channel capacity. There are numerous literatures in this line of work [26], [27], [28], [29], originating from the milestone work by Tassiulas et al. [30]. Though both maximizing throughput, the main difference of capacity problems is that they study throughput performance under a known environment with fixed channel quality. While the problem considered in our work focuses on throughput maximization under unknown and changing link quality. We need to minimize loss of throughput caused by learning, as well as time and communication complexity of learning and their impact on throughput performance.

VII. Conclusion

We conduct a throughout study on online throughput optimal channel accessing scheme for multi-hop CRNs. We believe that the generalization of our model sheds light on studying various visions of the problem regarding to different channel state and interference models. For the specified models in our paper, we successfully tackle the challenges of distributed design and exponential computation/communication/space complexity that severely impair efficiency and actual throughput performance of dynamic channel accessing schemes. It remains hard to address these practical challenges in other types of multi-hop CRNs. Besides, it is of much significance to optimize other performance metrics such as delay or fairness, and characterize tradeoff among these benchmarks. We leave these as future works.

ACKNOWLEDGEMENT

This work has been partially supported by China Scholarship Council, the National Natural Science Foundation of China (No.61132001, No. 61120106008, No.61272474 and

No. 61202410). The work of Professor Fan Li is partially supported by the National Natural Science Foundation of China under Grant No. 61370192 and 60903151, and the Beijing Natural Science Foundation under Grant No. 4122070. Professor Fan Li is the contact author. The research of Professor Xiang-yang Li is partially supported by NSF CNS-1035894, NSF ECCS-1247944, NSF ECCS-1343306, National Natural Science Foundation of China under Grant No. 61170216, No. 61228202.

REFERENCES

[1] K. Liu and Q. Zhao, "Distributed learning in multi-armed bandit with multiple players," *IEEE Transactions on Signal Processing*, vol. 58, no. 11, pp. 5667–5681, 2010.

[2] C. Tekin and M. Liu, "Online learning in decentralized multiuser resource sharing problems," *arXiv preprint arXiv:1210.5544*, 2012.

[3] D. Kalathil, N. Nayyar, and R. Jain, "Decentralized learning for multi-player multi-armed bandits," in *IEEE 51st Annual Conference on Decision and Control*, 2012, pp. 3960–3965.

[4] H. Liu, K. Liu, and Q. Zhao, "Learning in a changing world: Restless multiarmed bandit with unknown dynamics," *IEEE Transactions on Information Theory*, vol. 59, no. 3, pp. 1902–1916, 2013.

[5] A. Anandkumar, N. Michael, and A. Tang, "Opportunistic spectrum access with multiple users: learning under competition," in *Proc. IEEE Infocom,*, 2010, pp. 1–9.

[6] A. Anandkumar, N. Michael, A. K. Tang, and A. Swami, "Distributed algorithms for learning and cognitive medium access with logarithmic regret," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 731–745, 2011.

[7] Y. Gai and B. Krishnamachari, "Decentralized online learning algorithms for opportunistic spectrum access," in *Proc. IEEE Globecom*, 2011, pp. 1–6.

[8] X.-Y. Li, P. Yang, Y. Yan, L. You, S. Tang, and Q. Huang, "Almost optimal accessing of nonstochastic channels in cognitive radio networks," in *Proc. IEEE Infocom*, 2012, pp. 2291–2299.

[9] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[10] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[11] R. Agrawal, "Sample mean based index policies with O (log n) regret for the multi-armed bandit problem," *Advances in Applied Probability*, pp. 1054–1078, 1995.

[12] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1466–1478, 2012.

[13] T. Shu and M. Krunz, "Throughput-efficient sequential channel sensing and probing in cognitive radio networks under sensing errors," in *Proc. ACM Mobihoc*, 2009, pp. 37–48.

[14] X.-Y. Li and Y. Zhou, "Multi-armed bandits with combinatorial strategies under stochastic bandits," *arXiv preprint: http://arxiv.org/abs/1307.5438*.

[15] T. Erlebach, K. Jansen, and E. Seidel, "Polynomial-time approximation schemes for geometric intersection graphs," *SIAM J. Comput.*, vol. 34, no. 6, pp. 1302–1323, Jun. 2005.

[16] F. Kammer, T. Tholey, and H. Voepel, "Approximation algorithms for intersection graphs," pp. 260–273, 2010.

[17] T. Nieberg, J. Hurink, and W. Kern, "A robust ptas for maximum weight independent sets in unit disk graphs," pp. 214–221, 2005.

[18] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer, "Fast deterministic distributed maximal independent set computation on growth-bounded graphs," in *Distributed Computing*. Springer, 2005, pp. 273–287.

[19] S.-H. Huang, P.-J. Wan, J. Deng, and Y. S. Han, "Broadcast scheduling in interference environment," *IEEE Transactions on Mobile Computing*, vol. 7, no. 11, pp. 1338–1348, 2008.

[20] Y. Wang, W. Wang, and X.-Y. Li, "Distributed low-cost backbone formation for wireless ad hoc networks," in *Proc. ACM MobiHoc*, vol. 5, 2005, pp. 25–27.

[21] F. Zou, Y. Wang, X.-H. Xu, X. Li, H. Du, P. Wan, and W. Wu, "New approximations for minimum-weighted dominating sets and minimum-weighted connected dominating sets on unit disk graphs," *Theoretical Computer Science*, vol. 412, no. 3, pp. 198–208, 2011.

[22] Q. Zhao, B. Krishnamachari, and K. Liu, "On myopic sensing for multi-channel opportunistic access: Structure, optimality, and performance," *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 5431–5440, 2008.

[23] S. Ahmad, M. Liu, T. Javidi, Q. Zhao, and B. Krishnamachari, "Optimality of myopic sensing in multichannel opportunistic access," *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 4040–4050, 2009.

[24] M. Maskery, V. Krishnamurthy, and Q. Zhao, "Decentralized dynamic spectrum access for cognitive radios: cooperative design of a non-cooperative game," *IEEE Transactions on Communications*, vol. 57, no. 2, pp. 459–469, 2009.

[25] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y.-D. Yao, "Opportunistic spectrum access in unknown dynamic environment: A game-theoretic stochastic learning solution," *IEEE Transactions on Wireless Communications*, vol. 11, no. 4, pp. 1380–1391, 2012.

[26] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks," in *Proc. IEEE Infocom*, 2008, pp. 1103–1111.

[27] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *Proc. ACM MobiCom*, New York, NY, USA, 2005, pp. 73–87.

[28] L. Jiang and J. Walrand, "A distributed csma algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 960–972, 2010.

[29] S.-J. Tang, X.-Y. Li, X. Wu, Y. Wu, X. Mao, P. Xu, and G. Chen, "Low complexity stable link scheduling for maximizing throughput in wireless networks," in *Proc. IEEE SECON*, 2009, pp. 1–9.

[30] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE/ACM Transactions on Automatic Control*, vol. 37, pp. 1936–1948, 1992.