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

Yaqin Zhou*, Xiang-yang Li†‡, Fan Li§, Min Liu*, Zhongcheng Li*, Zhiyuan Yin§

*Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
†Illinois Institute of Technology, Chicago, IL, USA
‡School of Software and TNLIST, Tsinghua University, China
§School of Computer Science, Beijing Institute of Technology

Abstract—We consider distributed channel access in multi-hop cognitive radio networks. Previous works on opportunistic channel access using multi-armed bandits (MAB) 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. We formulate the problem as a linearly combinatorial MAB problem that involves a maximum weighted independent set (MWIS) problem with unknown weights which need to learn. Existing methods for MAB where each of \( N \) nodes chooses from \( M \) channels have exponential time and space complexity \( O(M^N) \), and poor theoretical guarantee on throughput performance. 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 \( O(Dm^{\rho}) \) 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. For randomly located networks with an average degree \( d \), the time complexity is \( O(d^\rho) \).

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 to opportunistically access vacant channels in temporal and spatial domain when the primary user is idle. However, due to resource and hardware constraints, at a given time, cognitive radios (CR) can sense only a part of heterogeneous channels with unknown quality before transmission.

Thus, it is core for secondary users to learn and select the best possible channels to access. Several recent results [1], [2], [3], [4], [5], [6], [7] 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. However, these methods generally adopt the simplest form of MAB where only single-hop networks fit the model. Dynamic channel access in multihop cognitive radio networks demands more sophisticated formulation that considers constraints of general interference among users. A naive extension of formulation from the single-hop case to multihop 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, there will be \( O(M^N) \) combinations totally when each user has \( M \) channels to choose. As all these aforementioned works adopt a UCB-type learning policy [8] [9] [10], the upper bound of regret as well as time and space complexity is linear with the number of arms, thus linear with \( O(M^N) \) in multihop networks.

Efficient channel access under multihop networks also requires decentralized design with low computation and communication. Previous decentralized MAB methods [4] [2] [11] pay little attention to these practical challenges around multihop networks. Though there is no communication cost in [4] [2], they require exponential time in a single learning round. Distinct from [4], [2] assumes multiple users can access the same resource, which does not capture conflicts among nearby users. On the other hand, [12] proposes a low-computation learning policy for multi-hop networks, but the policy takes a centralized form and still leaves challenges on distributed implementation unsolved.

Here we investigate the problem of achieving maximum expected throughput through a decentralized learning process with low computation and communication cost. As this problem involves competition among adjacent users, and cooperation for maximum throughput network wide, there may be no effective solutions if we directly formulate the problem into an integer linear programming. We then subtly formulate the problem into a linearly combinatorial 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 \( N \) secondary users. The other benefit is that we can adaptively choose efficient methods to solve the involved NP-hard MWIS problem and still achieve zero-regret.

We propose a decentralized channel access scheme based on robust PTAS [13] to approximately solve the MWIS problem. Our decentralized implementation achieves an approximation ratio of \( \rho = 1 + \epsilon \), but only requires time complexity \( O(Dm^{\rho}) \)
to find the strategy decision after weight is estimated. Here \( r = O(\log_2 M) \) is the hop number required to achieve a robust PTAS \(^{[13]}\), which is a constant for networks with a constant number of channels to choose. It costs time complexity \( O(d^n) \) for a random network. Our simulation results show that our new distributed learning policy indeed outperforms previous policies in terms of average throughput, time and storage cost.

The remainder of this paper is organized as follows. 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 \( G = (V,E,C) \) with a set \( V = \{v_i| i = 1, \ldots, N\} \) of \( N \) nodes (users), a set \( E \) of edges denoting conflicts, and a set \( C = \{c_j| j = 1, \ldots, M\} \) 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 model conflicts between two nodes, where each node is treated as a disk centered on itself. Conflicts happen 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 \( \mu_{i,j} \in [0,1] \). 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 \( i \neq i' \).

At each round \( t \), an \( N \)-dimensional strategy vector \( s_x(t) = s_x = \{s_{x,i}| i = 1, \ldots, N\} \) 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 \( s_x \). We use \( x = 1, \ldots, X \) 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 \( s_x \) 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, \( R_x(t) = \sum_{i|\in s_x} \xi_{i,s_{x,i}}(t) \). We evaluate policies using regret, which is defined as the difference between the expected throughput that could be obtained by a static optimal policy with the existence of a genie, and that obtained by the given policy. Let \( R_1 = \lambda_1 \) be the optimum fixed channel access strategy, then regret can be expressed as

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

III. PROBLEM FORMULATION

We first analyze the optimum throughput on the assumption that the mean of each random variable is known. We remodel the network \( G = (V,E,C) \) as an extended conflict graph \( H = (\bar{V}, \bar{E}) \), where \( \bar{V} = \{v_{i,j} | i \in [1,N], j \in [1,M]\} \), and show that the problem can be reformulated as a MWIS problem in extended conflict graph \( H \). Define a set of virtual vertices \( \{v_{i,j} | i = 1, \ldots, N\} \) for each node \( v_i \) 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 \( i \neq j \) and \( p \) has an edge in original network \( G \). Then graph \( H \) has \( N \times M \) vertices. We give an instance in Fig. 1 where the original network has 3 available channels and 3 nodes.

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 \( G \). Indeed, an IS of \( H \) one-to-one maps to a 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 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. Let \( \xi_{i,j}(t) \) be weight of virtual vertex \( v_{i,j} \). If the mean of \( \xi_{i,j}(t) \) is known, the optimum strategy is to find a maximum weighted independent set of vertexes from \( H \) as choices made by nodes in \( G \), i.e.,

\[
R_1 = \max_{s_x \in F} \sum_{i=1}^{N} \mu_{i,s_{x,i}}
\]

s.t. \( F \) is feasible strategy set.

However, these random variables are unknown actually, each user needs to learn and estimate the weight of each strategy, denoted by \( W_x(t) = \sum_{i|\in s_x} w_{i,s_{x,i}}(t) \), 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 bandit problem that selects at most \( N \) arms (i.e., vertexes in \( H \)) out of \( K = N \times M \) ones to minimize the regret \( \mathcal{R} \), 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 \( \beta \)-regret \(^{[14]}\), is defined as the difference between the expected throughput that is \( 1/\beta \) of the optimum, and that gained throughput (a \( \beta \)-approximation policy which instead yields a strategy with learned weight at

| Variable | Meaning |
|----------|---------|
| \( J_G(v) \) | \( r \)-hop neighborhood of node \( v \) in \( G \) |
| \( J_H(v) \) | \( r \)-hop neighborhood of vertex \( v \) in \( H \) |
| \( A_v \) | set of Candidate vertexes in \( r \)-hop neighborhood of \( v \) |
| MWIS(I) | maximum weighted independent set for vertex set \( I \) |
| MIS(I) | independent set with maximum cardinality for vertex set \( I \) |
| \( W(I) \) | summed weight of all vertexes in vertex set \( I \) |
| \( s_x(t) \) | strategy decision for round \( t \) |

Fig. 1: Original conflict graph \( G \) to extended conflict graph \( H \)

| TABLE I: Summary of notations | |
|------------------------------|---|
| Variable | Meaning |
| \( J_G(v) \) | \( r \)-hop neighborhood of node \( v \) in \( G \) |
| \( J_H(v) \) | \( r \)-hop neighborhood of vertex \( v \) in \( H \) |
| \( A_v \) | set of Candidate vertexes in \( r \)-hop neighborhood of \( v \) |
| MWIS(I) | maximum weighted independent set for vertex set \( I \) |
| MIS(I) | independent set with maximum cardinality for vertex set \( I \) |
| \( W(I) \) | summed weight of all vertexes in vertex set \( I \) |
| \( s_x(t) \) | strategy decision for round \( t \)|
least $1/\beta$ of the maximum possible weight). Let $R_{\beta,x}(t)$ be the reward of strategy $s_{\beta,x}$ generated by the $\beta$-approximation policy, then $\beta$-regret can be expressed as

$$R_{\beta}(n) = \sum_{R_{\beta,x} < R_1/\beta} \Delta_{\beta,x} E[T_{\beta,x}(n)] + \sum_{R_{\beta,x} \geq R_1/\beta} \Delta_{\beta,x} E[T_{\beta,x}(n)]$$

where $T_{\beta,x}(n)$ is the number of times that strategy $s_{\beta,x}$ has been played by round $n$, and $\Delta_{\beta,x}$ is the distance between $R_1/\beta$ and mean throughput of strategy $s_x$.

Though a previous learning policy in [11] achieves zero-regret, the upper bound of regret heavily depends on the distribution of strategies in feasible set $F$ (or $F_{\beta}$ which is all strategies whose throughput is at least $R_1/\beta$). That is, the upper bound of regret $R_{\beta}$ (or $\beta$-regret $R_{\beta}(\beta)$) including a factor of $\frac{\Delta_{\beta,\min}}{2\Delta_{\beta,\min}}$ becomes vacuous if $\Delta_{\min}$ (or $\Delta_{\beta,\min}$) $\to 0$. We expect a zero-regret policy without dependency on $\Delta_{\min}$. Meanwhile, it admits distributed implementation with low computation and communication complexity to guarantee efficiency of the channel access process.

IV. CHANNEL ACCESS

Each round is divided into two separate parts, one for strategy decision and the other for data transmission. In the strategy decision part, it utilizes the learned information in history to determine which strategy shall be selected for current time. In the data transmission 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

To learn for the best possible strategy, we adopt the learning policy proposed 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 (4) it utilizes estimated weight of each vertex in $\tilde{V}$ to select a maximum weighted independent set as strategy decision for next channel access. The estimated value for actual weight $\tilde{\xi}_{i,j}(t+1)$ of vertex $v_{i,j}$ is

$$w_{i,j}(t+1) = \hat{\mu}_{i,j}(t) + \sqrt{\frac{\max \left( \frac{\ln \frac{16}{eK}}{K^{m_{x,i}}}, 0 \right)}{m_{x,i}}}.$$  \hspace{1cm} (3)

Algorithm 1 Learning policy

1: For each round $t$, select a strategy $s_x$ by maximizing

$$\max_{s_x \in F} \sum_{s_{\beta,x} \in s_x} \left( \hat{\mu}_{s_{\beta,x}}(t) + \frac{\max \left( \frac{\ln \frac{16}{eK}}{K^{m_{x,i}}}, 0 \right)}{m_{x,i}} \right).$$  \hspace{1cm} (4)

Consequently, it only requires to store and update estimation for $M N$ vertices that costs storage and computation linear with $M N$, instead for $M N$ strategies in $F$ that costs storage and computation linear with $M N$. More specially, we need two $1 \times K$ vectors to store and update the estimated weight. One is $\hat{\mu}_k$ where $\hat{\mu}_k$ is observed mean of $\tilde{\xi}_k$ up to the current round, and the other is $\bar{m}_k$ where $\bar{m}_k$ is the number of times that channel $\tilde{\xi}_k$ has been selected so far. After data transmission on the channels of chosen strategy $s_x$ in slot $t$, actual weight $\tilde{\xi}_{s_x,i}(t)$ is observed for all $s_{x,i} \in s_x$. Then $\hat{\mu}_k$ and $\bar{m}_k$ are updated in the following way:

$$\hat{\mu}_k(t) = \begin{cases} \hat{\mu}_k(t-1), & \text{if } k \notin s_x, \\ \hat{\mu}_k(t-1) + \xi_{s_x,i}(t), & \text{if } k \in s_x, \end{cases}$$  \hspace{1cm} (5)

$$\bar{m}_k(t) = \begin{cases} \bar{m}_k(t-1) + 1, & \text{if } k \in s_x, \\ \bar{m}_k(t-1), & \text{else}. \end{cases}$$  \hspace{1cm} (6)

Due to NP-hardness of the MWIS problem in [4], it is desirable to solve it approximately while retaining zero-regret. The following theorem shows that, for any algorithm with approximation ratio at least $1/\beta$ for the MWIS problem, the regret on the achieved throughput is bounded.

Theorem 1: [14] The $\beta$-approximation learning policy has

$$\sup_{t} R_{\beta}(t) \leq N K \left( \sqrt{\frac{\ln \frac{16}{eK}}{K^{m_{x,i}}}} + \frac{1}{K^{m_{x,i}}} \right)^{1/2} \left( \frac{1}{K^{m_{x,i}}} \right) + 1 \epsilon^2 \left( 1 + \frac{4\epsilon^2 K^{m_{x,i}}}{K^{m_{x,i}}} \right)^{1/2} K n^{1/2}$$

(7)

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 $\beta$ is, the more sacrifice on overall throughput it causes. Given that, we choose the robust PTAS proposed in [13] to solve the MWIS problem. Though centralized, the robust PTAS is elegant, and more importantly, it requires no geometric information as other PTAS schemes [15] [16]. This feature is very attractive as it is expensive to get and maintain exact locations of each node in multi-hop wireless networks, not to say negative effect of errors by location methods. We will show how to implement it in a distributed manner later. For better understanding, we first introduce the centralized method.

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 $\|u, v\| \leq 2$. For a subset $I$ of nodes in $V$, let $W(I)$ denote the total weight of $I$, i.e., $W(I) = \sum_{v \in I} w_i$, and MWIS($I$) denote a maximum weighted independent set for $I$. The independent set with maximal cardinality (MIS) for $I$ is written as MIS($I$). Let $d_{C} (u, v)$ be the minimum hop of any path connecting $u$ and $v$ in $G$. Define

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

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 $w_{\text{max}} = \{ \max w_v | v \in V \}$, and then computes $\text{MWIS}(J_{G,r})$ as long as $W(\text{MWIS}(J_{G,r+1})) > \rho W(\text{MWIS}(J_{G,r}))$ holds. Let $r$ denote the smallest $r$ for which the criterion is violated. It has been proved that $r$ is a constant for a specific $\rho$, i.e., $\rho^r \leq (2r + 1)^2$. We then remove $\text{MWIS}(J_{G,r}(v_{\text{max}}))$ 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 $\rho$-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

$$J_{H,r} = J_{H,r}(v) := \{ u \in \bar{V} | d_H(u,v) \leq r \}.$$ 

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 $H$.

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

**Proof:** Robust PTAS can be equally extended to other intersection graphs as long as the graph is growth-bounded, where the number of independent vertexes in a vertex's $r$-hop neighborhood is constantly bounded $[17, 13]$. Though the extended graph $H$ is not a strict unit graph, it is straightforward to 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 vertex 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 multihop networks. We design a distributed implementation that takes low communication and computation complexity.

The main framework of our distributed implementation is shown in Algorithm 2 which is run round by round, where each round consists of a strategy decision part and a data transmission part (see Fig. 2). The strategy decision part 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 $s_r(t - 1)$), to ensure computation of MWIS with newest weight. In our protocol, these vertexes in $s_r(t - 1)$ 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 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 selected in a strategy decision observe new values, and utilize the observation to update estimated weight (i.e., plugging (5) and (6) into (3)). If each vertex performs weight broadcast sequentially, obviously 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 [18, 19, 20], by which number of mini-timeslots can be reduced to $O((2r + 1)^2)$.

After WB, each vertex then runs distributed Robust PTAS presented 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.
Algorithm 3: Distributed robust PTAS for strategy decision at each vertex

Initialization: ∀ vertex $v \in \tilde{V}$, marked as status Candidate, have collected newest weights of all $(2r + 1)$-hop neighbors $J_{H,2r+1}(v)$. 
1: for mini-round $\tau = 1, 2, \ldots, N$ do 
2: if $v$ is Candidate then 
3: if $w_u \geq \max\{w_v | u \in A_{2r+2}(v)\}$ then 
4: $v$ is marked as LocalLeader and declare in $(2r + 1)$-hop neighborhood. 
5: end if 
6: end if 
7: if $v$ is LocalLeader then 
8: Compute a local MWIS($A_r(v)$) using enumeration. 
9: 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. 
10: Locally broadcast the results within $(3r + 1)$-hop neighborhood of the LocalLeader. 
11: Update its own status accordingly. 
12: else if $v$ is Candidate then 
13: if $v$ receives determination messages then 
14: Update status of itself and $(2r + 1)$-hop neighbors accordingly. 
15: end if 
16: end if 
17: end for

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

The execution of Algorithm 3 completes, and a new round follows.

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.

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. 5 and local computation in a single mini-round in Fig. 4 for the network presented in Fig. 1.

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, which may require mini-rounds $O(N)$. Actually, a constant number of mini-rounds is enough to output a good decision. That is why we set $D$ mini-rounds of Algorithm 3 in main framework Algorithm 2.

Herein we first present the achieved approximation ratio after $O(N)$ mini-rounds. The results on mini-rounds $D$ will be presented in later analysis.

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

Proof: Let $V_\tau(\tau)$ be a LocalLeader selected at mini-round $\tau$. In each mini-round, a LocalLeader $V_\tau(\tau)$ 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 $\rho$-approximation to MWIS($A_{r+1}$). Let MWIS($V$) be
the global optimum, and \( I \) be intersection of MWIS(\( \tilde{V} \)) and \( A_{r+1}(v_i(\tau)) \), we have \( W(I) \leq \rho W(MWIS(A_{r+1})) \). As union of \( A_r \) in all mini-rounds is exactly \( \tilde{V} \) and any two distinct \( A_r \) do not intersect, we have the union of all MWIS(\( A_r \)) output by all LocalLeaders is \( \rho \)-approximation to the global optimum MWIS(\( \tilde{V} \)) in weight.

Complexity. We summarize complexity in a complete round. 

Communication complexity: As shown in Fig. 3, 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 \( \sum_{i=0}^{\infty} \frac{1}{2^i} = O(1) \) 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 \( |MWIS(A_r(v))| \leq M(2r+1)^2 \), there are totally \( C_{Mm}^{M(2r+1)^2} \) enumerations. Using \( M(2r+1)^2 < \rho^2 \) for \( r < \tilde{r} \), we have

\[
C_{Mm}^{M(2r+1)^2} \leq \left( \frac{Mme}{M(2r+1)^2} \right)^{M(2r+1)^2} \leq \left( \frac{me}{(2r+1)^2} \right)^{\rho^2}. \tag{8}
\]

Hence, it requires polynomial time \( O(m\rho^2) \) per mini-round, and \( O(Dm\rho^2) \) 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: It is \( O(m) \), as each vertex has to store weight of neighbors within \((2r+1)\)-hop neighborhood.

In our protocol, \( r \) and \( D \) is constant, then the communication, computation, and space complexity is \( O(1) \), \( O(m\rho^2) \), and \( O(m) \) respectively.

**Fig. 5:** Worst case of strategy decision caused by linear effect where \( w_1 > w_2 \cdots > w_N \)

D. Improve to constant-time-complexity strategy decision

As mentioned previously, the distributed implementation of strategy decision requires \( O(N) \) mini-rounds to get all vertexes marked. We then show a simple instance of the worst case in Fig. 5. In the figure we use a linear network where all vertexes are aligned uniformly along a line within 1-hop distance. One can easily figure out that when the weight of each vertex is in a decreasing order from the start vertex to the end vertex, at the beginning only the start vertex can be LocalLeader since no other vertexes are locally maximum weighted. And, in each next round, still only one could be LocalLeader sequently. Thus it would take \( N \) mini-rounds in a single round.

We then analyze the time complexity under random networks where location of each vertex is uniformly random distributed. We assume a random network has an average degree of \( d \). We expect to show it is possible to achieve a slightly smaller constant-approximation ratio if the algorithm terminates after a fixed number of \( D \ll N \) mini-rounds, no matter whether there remains vertexes unmarked or not. Surprisingly, we find that it is indeed the case. The following theorem presents our results.

**Theorem 4:** Given a random network \( G \) with an average degree \( d \), Algorithm \( \mathcal{A} \) achieves \( \alpha \)-approximation to the optimum if we set the number of mini-rounds as a constant \( D \ll N \). \( \alpha \) is a constant with constant probability.

**Proof:** The proof is omitted here due to space limit. \( \blacksquare \)

By Theorem 4 and 8, time complexity of Algorithm \( \mathcal{A} \) reduces to \( O(d^{\rho^2}) \) regarding to random networks.

E. 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 \( R_1/\alpha \) and a strategy \( s_x \) is \( R_1/\alpha - \theta \lambda_x = \theta \Delta \Delta \). 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

---

**Fig. 4:** Local computation of Algorithm \( \mathcal{A} \) in extended graph \( H \) with estimated weight: black vertex-LocalLeader, white vertex-Candidate, stripe vertex-Loser, gray vertex-Winner.
of $R_1/\theta_0$ and $s_X$, we obtain the practical regret is less than $	heta \cdot \mathcal{G}_{\theta_0}(n)$ according to [14], i.e.,

**Theorem 5:** The practical regret of Algorithm 2 satisfies

\[
\sup_{\theta, X, n} \theta \mathcal{G}_{\theta_0}(n) \leq \frac{1}{\alpha} NK + \left( \theta \sqrt{eK} + \frac{16}{e\alpha} (1 + N)^3 \right) n^{\frac{3}{2}} + \frac{1}{\alpha} \left( 1 + \frac{4\sqrt{KN^2}}{e(\theta_0)^2} \right) N^2 Kn^{\frac{3}{2}}.
\]

(9)

Then our channel allocation scheme can guarantee an effective throughput of $R_1/(\theta_0) - \mathcal{G}_{\theta_0}(n)$.

V. SIMULATIONS

Now we conduct simulations for our proposed channel accessing scheme under random networks. We set three series of simulations to respectively study efficiency, regret, and influence of stale weight. 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 [12]. Each channel evolves as a distinct i.i.d Gaussian stochastic process over time. We set each round has length of a unit time slot. Referring to a cognitive radio system [12], we list the values of time parameters of a round in Table II. In strategy decision of each round, we set $t_s = 4t_m$. Let $t_b$ be time to finish local broadcast and $t_l$ be the total time for local computation (LocalLeader selection and local MWIS computation). We have $t_m = 2t_b + t_l = 250ms$. According to Fig. 2 the actual throughput gained at each round is $\frac{t_l}{t_s + 4t_m} R_c(t) = 0.5R_c(t)$ in our setting.

**A. Efficiency of Algorithm 3**

We first set a series of experiments to show efficiency of Algorithm 3. We plot the summed weight of all output MWISs by mini-round 1 to $N$ for various $N \times M$ random networks. The value of $N \times M$ is respectively set as $50 \times 5, 100 \times 5, 200 \times 5, 50 \times 10, 100 \times 10$, and $200 \times 10$. From the Fig. 6 we can see that every line converges to a fixed value after the 4th mini-round, no matter how many vertexes there are in the extended graphs. This indicates that all vertexes are marked by that time. The results coincide with Theorem 4 where we claim that our proposed algorithm converges to a constant approximation ratio that is almost optimal under random graphs.

**B. Regret analysis**

In the second series of experiments, we study practical regret and $\beta$-regret caused by our proposed distributed learning scheme. We compare our method with LLR learning policy.

**C. Throughput performance under unfrequent update**

We evaluate the effective throughput under different frequencies of weight update in the third series of simulation, where
Meanwhile, we compare performance of our learning policy with LLR policy. 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 $R_P(z) = \frac{P_x(z+1)y_d+\sum_{i=\frac{zy}{y}}^{\frac{z(y+1)}{y}} P_x(i)y_d}{yt_a}$. We conduct experiments in a random network with 100 users and 10 channels. For such a large scale network, 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 $\bar{R}_P(z)$, where $\bar{R}_P(z) = \sum_{i=\frac{z(y+1)}{y}}^{\frac{z(y+2)}{y}} R_x(i)y_d$, and average estimated throughput $\bar{W}_P(z)$ (i.e., average estimated weight of all selected strategies up to $z$). Let $\bar{W}_P(z)$ be average estimated throughput at $z$th period, we have $W_P(z) = (\frac{\bar{y}}{y})^{\frac{z}{y}} \frac{1}{\bar{W}_P(z) + P(x)}$ and $W_P(z) = (\frac{\bar{y}}{y})^{\frac{z}{y}} \frac{1}{\bar{W}_P(z) + P(x)}$. The difference between $\bar{R}_P(z)$ and $\bar{W}_P(z)$ can also indicate the throughput performance of the algorithm.

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 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. Especially, a significant improvement can be seen between the frequent case (Fig.8(a)) and the unfrequent case of $y = 5$ (Fig.8(b)). In the later two cases, further improvement is not so obvious as the proportion of time on learning decreases much more slowly. We then compare 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. Except the line of estimated throughput by LLR, difference among other three lines is not obvious in the figures. Thus we show a zoom-in part of the difference on the upper right of each figure. In these figures, it shows that the actual throughput achieved by our learning policy is better the LLR policy. They collaboratively show that unfrequent update has negligible impact on accuracy of estimation, but significantly improve effective throughput.

VI. RELATED WORKS

There is a rich body of results on dynamic spectrum access in cognitive radio networks. As channel availability and quality is unknown to secondary users, they need to conduct a learning process to select good channels. Several literatures address this problem from sequential 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 [21, 22], 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 more than $N > 1$ 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 [23] propose a throughput-optimization 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 for secondary users 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 without uncertainty of channel quality. The concerned issue is that interference among links constrains the maximum supportable arrival rate at each link that is assumed to have unit capacity. While the problem considered in our work focuses on throughput maximization under unknown and changing link quality, as well as existence of interference. 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 proposed an almost throughput optimal channel accessing scheme for multihop cognitive networks. Our scheme consists of a distributed learning process with low computation and space complexity, and a strategy decision process with low computation and communication complexity. Our distributed
implementation does not need extra predefined information on network parameters.

Our works have assumed i.i.d stochastic gain of channels, which is an easy-to-analyze model. Future work will take consideration of adversary case where gains are generated by an adversary that may obliviously or adaptively learn to play against our strategies. Additionally, most works as well as ours minimize weak regret compared to the best static policy, it will be challenging to computation-efficiently minimize strong regret compared to the best dynamic policy.

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 multi-user 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 Proc. of IEEE CDC, 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. of 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. of IEEE GLOBECOM, 2011, pp. 1–6.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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. of IEEE INFOCOM, 2012, pp. 2291–2299.

[13] T. Nieberg, J. Hurink, and W. Kern, “A robust ptas for maximum weight independent sets in unit disk graphs,” Graph-theoretic concepts in computer science, pp. 214–221, 2005.

[14] Y. Zhou and X.-Y. Li, “Multi-armed bandits with combinatorial strategies under stochastic bandits,” arXiv preprint: http://arxiv.org/abs/1307.5438, 2013.

[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] 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.

[18] 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.

[19] Y. Wang, W. Wang, and X.-Y. Li, “Distributed low-cost backbone formation for wireless ad hoc networks,” in Proc. of ACM MOBIHOC, 2005, pp. 25–27.

[20] 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.

[21] 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.

[22] S. Ahmed, 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.

[23] T. Shu and M. Krunz, “Throughput-efficient sequential channel sensing and probing in cognitive radio networks under sensing errors,” in Proc. of ACM MobiCom, 2009, pp. 37–48.

[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. of Mobicom, 2005, pp. 73–87.

[28] L. Zhang 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. Xu, M. Xiao, P. Xu, and G. Chen, “Low complexity stable link scheduling for maximizing throughput in wireless networks,” in Proc. of 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.