Multi-Armed-Bandit-Based Spectrum Scheduling Algorithms in Wireless Networks: A Survey

Feng Li, Dongxiao Yu, Huan Yang, Jiguo Yu, Holger Karl, and Xiuzhen Cheng

Abstract

Assigning bands of the wireless spectrum as resources to users is a common problem in wireless networks. Typically, frequency bands were assumed to be available in a stable manner. Nevertheless, in recent scenarios where wireless networks may be deployed in unknown environments, spectrum competition is considered, making it uncertain whether a frequency band is available at all or at what quality. To fully exploit such resources with uncertain availability, the multi-armed bandit (MAB) method, a representative online learning technique, has been applied to design spectrum scheduling algorithms. This article surveys such proposals. We describe the following three aspects: how to model spectrum scheduling problems within the MAB framework, what the main thread is following which prevalent algorithms are designed, and how to evaluate algorithm performance and complexity. We also give some promising directions for future research in related fields.

Introduction

Properly scheduling resources (e.g., spectrum) plays a crucial role in improving the efficiency of a wireless network. Especially in the upcoming fifth generation (5G) networks, fully utilized spectrum is expected to guarantee system performance in terms of throughput, delay, and so on [1]. But such a demand is a challenge for resource assignment algorithms, especially when new network applications are deployed in unknown environments for which a prior profile of spectrum state may not be available. Furthermore, the pursuit of better spectrum utilization entails more dynamic and possibly competitive spectrum access (e.g., in cognitive radio networks), while the uncertainty of the spectrum results in more difficulties in developing efficient dynamic spectrum access mechanisms.

Fortunately, artificial intelligence reveals a new option to tackle such uncertainty. Multi-armed bandit (MAB) is a representative reinforcement learning problem. The analogy is that there is a bandit machine with multiple arms, pulling one of which produces some reward. The reward is usually defined by a stochastic process following an unknown probability distribution or is under the control of arbitrary adversaries. Without prior knowledge, the players cannot directly identify the best arm; hence, they have to learn the arms' uncertain rewards with loss in reward gains by trying them out sequentially. In the MAB problem, the players need to choose a sequence of arms to play such that the cumulative loss is minimized.

Considering the spectrum may be of unknown stochastic characteristics or under the control of adversaries, it is promising to apply a learning technique to address such uncertainty in spectrum scheduling; MAB is one of the most promising techniques to achieve this goal. Although quite a few efforts have been made in the past decades on MAB algorithms [2–4], applying them to resolve practical issues in real systems, for example, the spectrum scheduling problem in wireless networks, is just beginning, compared to other relatively mature areas in full-knowledge-based resource scheduling problems. Therefore, we believe that it is the right time to review the existing design methodologies and motivate future research directions.

In the remainder of this article, we first review preliminaries about the MAB problem in the following section. We then introduce the spectrum scheduling problem based on the MAB framework and give a classification. Next, we survey recent publications. We also present some promising research directions based on the insightful discussions on the existing proposals, and finally conclude this article.

Multi-Armed Bandit Problem

In this section, we first briefly introduce the formulations of MAB problems, and then outline the algorithms to tackle these problems. More details about MAB in general can be found in [2–4].

Problem Formulation

In the classical MAB problem, we consider a bandit machine with K arms. Time is divided into slots $t = 1, 2, ...$, and we denote by $r_k(t)$ the reward produced by pulling arm k in slot t. For each arm k, $r_k(t)$ is drawn from an *unknown* probability distribution independently and identically or is controlled by arbitrary adversaries across time. The former case is called *stochastic* MAB, while the latter one is referred to as (non-stochastic) *adversarial* MAB.

In stochastic MAB, we denote by μ_k the mean of the unknown probability distribution from which we draw $r_k(t)$ independently and identically, and let μ^* be the highest expected reward delivered by the

Feng Li and Dongxiao Yu (corresponding author) are with Shandong University; Huan Yang is with Qingdao University; Jiguo Yu (corresponding author) is with Qilu University of Technology and National Supercomputer Center in Jinan; Holger Karl is with Paderborn University; Xiuzhen Cheng is with Shandong University and George Washington University.

Digital Object Identifier:
10.1109/MWC.001.1900280

1536-1284/20/$25.00 © 2020 IEEE

best arm. We aim at designing a randomized policy $\Pi = \{\pi(1), \cdots, \pi(T)\}$ within time horizon T, where $\pi(t) \in \{1, 2, \cdots, K\}$ denotes the arm selected in slot t, such that the following *regret function*

$$F(\Pi) = \mu^* T - \mathbb{E}\left[\sum_{t=1}^{T} r_{\pi(t)}(t)\right] \quad (1)$$

can be minimized without being aware of $\{\mu_k\}_{k=1,\ldots,K}$. In fact, the regret function can be thought of as the expected loss of reward induced by the fact that the policy cannot directly identify the best arm due to the unknown statistics.

Assume the reward yielded by arm k in slot t, namely $r_k(t)$, is under the control of adversaries in the adversarial MAB. Let k^* be the best arm that yields the highest cumulative reward within time horizon T. We compare the performance of our policy with that of the best single arm k^*, and aim at minimizing the so-called weak regret function

$$\tilde{F}(\Pi) = \sum_{t=1}^{T} r_{k^*} T - \mathbb{E}\left[\sum_{t=1}^{T} r_{\pi(t)}(t)\right] \quad (2)$$

To minimize the regret (or weak regret) function, players make sequential decisions to select the arms such that the resulting expected cumulative reward can be as close to the one yielded by the best single arm as possible. To serve this goal, the basis of designing algorithms for the MAB problems is to make a trade-off between *exploration* and *exploitation*. On one hand, we try out each arm to learn more about the distribution of its reward for long-term benefit; on the other hand, we take the "best" arm according to our empirical estimates on the rewards across different arms to maximize immediate return.

Algorithms for MAB

One choice to resolve the MAB problem is to select the arms in a greedy manner, that is, the so-called *ε-greedy algorithm* [2]. Specifically, in each slot, we either select, with probability $1 - \varepsilon$, the arm that has yielded the highest average reward or, with probability ε, choose one of the arms at random. Since ε is a constant, the regret function grows linearly in time. The performance can be improved by shrinking ε such that the resulting regret scales logarithmically with respect to T.

Unfortunately, this greedy algorithm explores the arms in a "purely" randomized manner and does not take into account (readily available) confidence intervals on the empirical estimates of the arms' rewards. This knowledge is exploited by the Upper Confidence Bound (UCB) algorithm, where we calculate a UCB index for each arm. For each arm, its UCB index takes into account the average reward gained by selecting it until now and the total times it has been selected until now, so as to make a trade-off between exploitation and exploration [2, 4]. By selecting the arm that has the largest UCB index in each slot, the expected regret of the UCB algorithm can be upper-bounded by $O(\log T)$.

Another family of approaches to resolve the adversarial MAB problem is the *exponential-weight* algorithm. Similar to the ε-greedy algorithm, it determines which arm to select in each slot by leveraging a probability distribution. The probability combines both uniform distribution for exploration and another one for exploitation that exponentially weights the arms according to the empirically estimated cumulative rewards. For any assignment of rewards set by adversaries, the consequent expected weak regret scales as

$$\mathcal{O}\left(2\sqrt{KT\log K}\right).$$

Although there are many state-of-the-art variations extending the above classical MAB algorithms [2–4], we do not present any more due to space limitations.

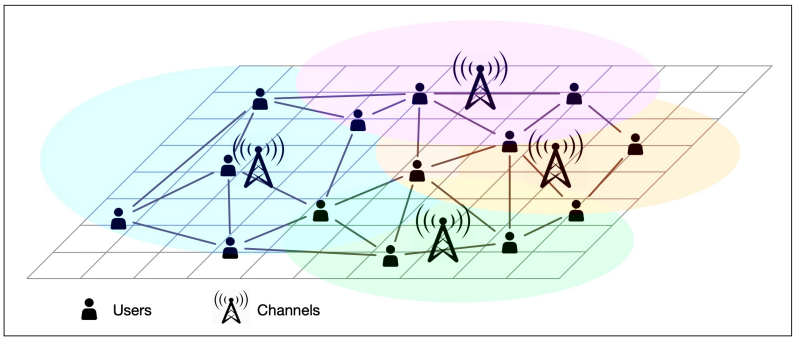

FIGURE 1. Spectrum scheduling. The aim is to allocate the partitions of the spectrum, namely the channels, to a set of users. Since some of the users may not be able to share a channel, we use a conflict graph to present such an interference relationship, where two users are connected by an edge if concurrent co-channel transmissions cannot be deployed to them.

Algorithm Classification

In this section, we first describe the spectrum scheduling problem and then discuss the MAB-based modeling methods. We finally summarize the categorization of the proposals we survey in this article.

Problem Description

As shown in Fig. 1, spectrum is divided into a set of channels (which are usually assumed to be orthogonal in existing proposals), and users access these channels to transmit their data.[1] Scheduling channel access is a very typical issue in wireless networks. Since wireless channels may have different qualities and thus yield different amounts of reward (e.g., in terms of throughput) even with respect to individual users, the main problem we investigate is how to allocate the channels to the users so as to maximize the yielded reward. Note that we use "users" to represent the entities (e.g., transmission links) to which we would like to assign channels in the following.

Another main concern of spectrum scheduling is the interference among users; multiple users accessing the same channel simultaneously may interfere with each other. The most popular interference models include the *full interference* model and the graph-based model. In the former model, it is assumed (e.g., in [5–8]) that any two or more users accessing the same channel simultaneously will induce collisions, such that a channel accommodates at most one user. Nevertheless, the full interference model does not take into account the locality of the interference. For example, we can assign the same channel to geographically dispersed users such that their concurrent co-channel transmissions can be supported. Hence, in the graph-based interference

[1] We use "channel" here as a common shorthand for "frequency band." The extended notion of a channel comprising time and space for communication as well should also be amenable to machine learning approaches, but we leave that discussion for a followup article.

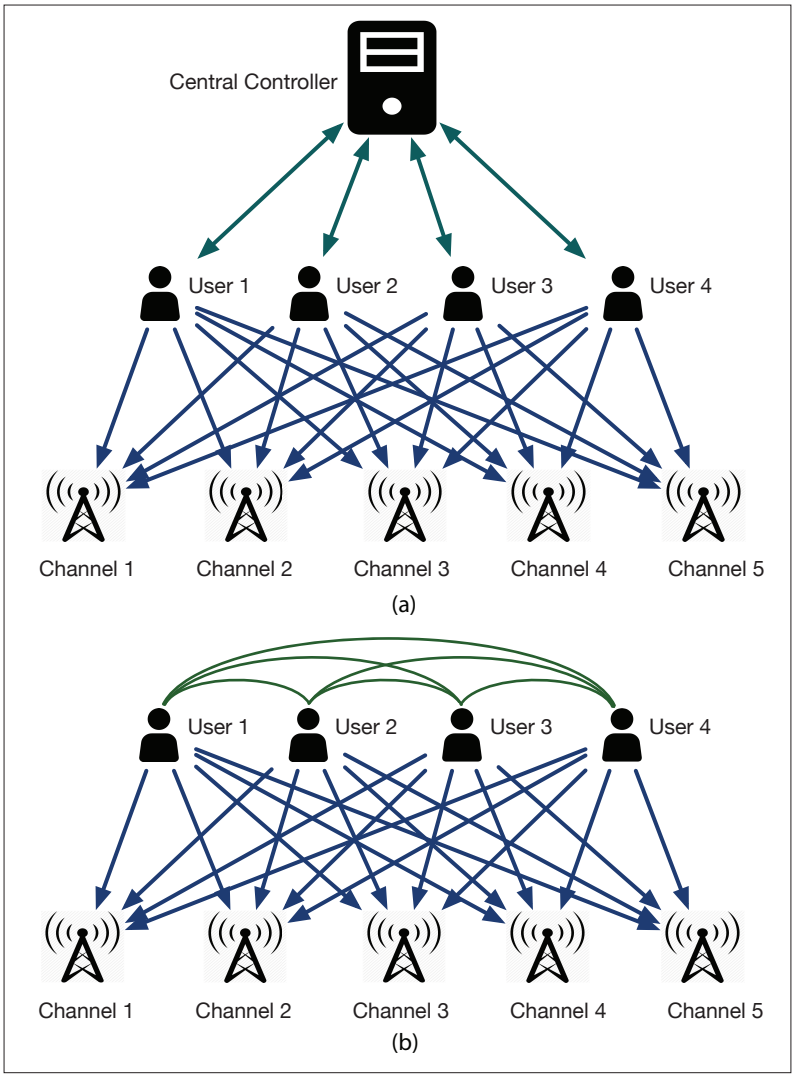

FIGURE 2. An example of spectrum scheduling with four users and five channels.

mode, we usually use a so-called *conflict graph* to represent the interference. As demonstrated in Fig. 1, two users are connected by an edge if we cannot assign the same channel to them due to the resulting interference. In fact, when the conflict graph is complete such that any two users interfere with each other, the graph-based interference model is degenerated into the full interference model.

Even if the statistics of channel qualities were known a priori, calculating an optimal channel allocation is a highly nontrivial issue. For example, under a full interference model, the problem can be transformed into a maximum weight matching (MWM) problem in a bipartite graph, as shown in [5, 8]. In particular, the vertices in the bipartite graph can be divided into two disjoint and independent sets: one is composed of the users, while the other one consists of the channels. For each pair of user and channel, there is an edge between them weighted by the expected reward we can earn by assigning the channel to the user. The goal is to seek the maximum weight user-channel matching in the bipartite graph. Additionally, as demonstrated by [9, 10], when graph-based interference is considered, allocating the channels to the users is equivalent to the coloring problem: assigning different colors (i.e., channels) to the adjacent vertices (i.e., users that interfere with each other) such that the resulting total expected reward is maximized, provided that for any user, assigning different colors to it may result in different amounts of reward. It is apparent that both of the above two problems are NP-hard, and we have to be content with nearly optimal solutions in polynomial time.

In the setting of the MAB framework, the problem becomes much more difficult, since we cannot directly figure out even a nearly optimal (stationary) channel allocation scheme due to the uncertainty of the channel qualities. Instead, we resort to *dynamic* ones where each user selects one (or some) of the channels in each decision round[2] so as to gradually learn the uncertainty and gain reward by accessing the selected channel. Hence, the channel scheduling scheme without prior knowledge induces regret within the time horizon; we aim at minimizing that regret.

Multi-Armed Bandit Models for Spectrum Scheduling

Single-Player MAB vs. Multi-Player MAB: From the perspective of MAB modeling, we hereby focus on the following two categories according to the number of players: single-player MAB (SMAB) and multi-player MAB (MMAB). In the former case, the single player can choose one (or some) of the arms in each decision round, while in the latter one, there are multiple players that can operate the MAB simultaneously, and we may additionally assume that, in each decision round, an arm cannot be pulled by more than one player. Our aim is thus to calculate the optimal, sequential player-arm matching to maximize the total reward.

In a single-user multi-channel network, the user is the player sequentially selecting the arms, while the arms usually correspond to the channels, as shown in [11, 12]. Nevertheless, it is worth noting that the players may be different from the "users" to which we assign channels when considering multi-user multi-channel networks. For example, the SMAB model also can be used in multi-user multi-channel networks [5, 8, 9] where a centralized network controller usually corresponds to the "player" of the MAB, as shown later.

Centralized vs. Decentralized: Depending on whether it entails a central infrastructure (e.g., a central network controller that can gather global information over the system) or not, we categorize the MAB models into *centralized* MAB and *decentralized* MAB. In fact, SMAB usually corresponds to centralized algorithms since it is the single player executing multi-armed bandit algorithms, whereas the players (i.e., users) in the MMAB model are supposed to be well synchronized, such that they can collaborate to operate the bandit machine, especially considering that an arm cannot be pulled by multiple players simultaneously, and avoiding such a conflict demands coordination among the players.

We hereby show an example with four users and five channels in Fig. 2 to illustrate the spectrum scheduling problem under different MAB models. As demonstrated in Fig. 2a, for centralized spectrum scheduling algorithms (e.g., [5, 8, 9, 11, 12]), a central network controller is neces-

[2] Decision rounds can be regarded as the periods during which users select their channel(s). A decision round may be composed of a number of consecutive time slots, within which the users negotiate for coordinating channel selections (e.g., in [7].

Reference	MAB	Interference	Centralized/decentralized	Complexity	Regret
Gai *et al.*, (2010) [5]	SMAB	Full interference	Centralized	$O(MN(M+N)^3)$	$O(N^4M\log T)$
Gai *et al.*, (2011) [6]	MMAB	Full interference	Decentralized	$O(M)$	$O(N(M-N)\log T)$
Lelarge *et al.*, (2013)[9]	SMAB	Graph-based	Centralized	–	$O\left(\sqrt{N^3T\log N}\right)\Big/O(N^3\log T)$
Zhou *et al.*, (2014) [13]	MMAB	Graph-based	Decentralized	$O(Dm^{(1-\varepsilon)^r})$	$1/(1-\varepsilon)$
Li *et al.*, (2014) [11]	SMAB	–	Centralized	$O(M)$	$O(Mn\log T)$
Zhang *et al.*, (2016) [10]	MMAB	Graph-based	Decentralized	$O(N^3)$	$O(T)$
Avner *et al.*, (2016) [7]	MMAB	Full interference	Decentralized	$O(N)$	–
Kang *et al.*, (2018) [8]	SMAB	Full interference	Centralized	$O(M)$	$O(SN!N^3\log T)$
Cai *et al.*, (2018) [12]	SMAB	–	Centralized	$O(M)$	$O(nM\log(M)T^{2/3})$
Stahlbuhk *et al.*, (2019) [14]	MMAB	Graph-based	Decentralized	$O(N)$	$O(N\log T)$

TABLE 1. Classification of algorithms.

sary to serve as the (single) player. The controller observes the rewards the users obtain by accessing the channels and makes sequential decisions by running MAB algorithms to guide the channel selections. For decentralized spectrum scheduling without a central controller (e.g., [6, 7, 10, 13, 14]), it is usually assumed that the users can communicate with each other to coordinate their channel selections.

Algorithm Categorization

Table 1 classifies the algorithms considered here according to the MAB models. As mentioned earlier, the centralized algorithms usually adopt SMAB (e.g., [5, 8, 9, 11, 12]), while the decentralized ones are usually based on MMAB (e.g., [6, 7, 10, 13, 14]. Moreover, with respect to the interference models, except when only a single user is considered (e.g., [11, 12]), channel allocation in multi-user networks is supposed to take into account the interference among the users through either a full interference model or graph-based one.

We also give complexities and upper bounds on the regrets to illustrate the efficiencies of the algorithms. We denote by M and N the number of channels and the number of users, respectively. These two notations are used throughout the remainder of this article, while the others are introduced when we survey the corresponding proposals. In fact, most proposals design their algorithms based on the UCB policy. For example, in [5–8, 11, 14], the UCB-like index for each arm can be calculated by taking into account the estimated average reward we have obtained by selecting that arm as well as the number of times the arm has been selected. In each decision round, the arm with the highest index value will be selected. Although different algorithms define the arms uniquely, as introduced later (e.g., an arm may correspond to a user-channel matching in [5] or a channel in [6–8]), their regrets are all logarithmically upper-bounded with respect to T and scale as a polynomial of M and N. On the other hand, the exponential-weight approach can be utilized to design algorithms in adversarial settings, e.g., as shown in [9], resulting in a regret scaling as $\mathcal{O}\left(\sqrt{T}\right)$.

Survey of Spectrum Access Scheduling in Wireless Networks

Centralized Algorithms

Given an N-user, M-channel network, the channel allocation problem is usually formulated as an MWM problem in a bipartite graph, where we seek the optimal matching between the users and the channels. The difficulty of the problem lies in that we do not know the weight (even its statistics) with respect to each user-channel combination. Gai *et al.* [5] built an SMAB model where each arm corresponds to a channel allocation scheme (i.e., a matching in the bipartite graph). Since there is an exponentially large number of feasible channel allocation schemes, directly applying the UCB approach to the bandit results in significant complexity in both storage and computation. Inspired by the fact that one user-channel combination can be involved in different channel allocation schemes, maintaining the historical observations for each user-channel combination is sufficient to calculate the UCB index for each arm, resulting in a polynomial storage space $O(MN)$. The arm with the largest index value is chosen in each decision round by taking the Hungarian algorithm as a subroutine to resolve an MWM problem. The computation in each decision round has a polynomial complexity $O(MN(M+N)^3)$, and the growth of the regret is upper-bounded by $O(N^4M\log T)$.

The above proposal is based on an assumption of full interference among the users, while this model may not be the case in some real scenarios. Adopting the conflict graph-based interference model,[3] [9] investigates the channel allocation problem in the context of both adversarial MAB and stochastic MAB. In the former case, the authors extend the exponential-weight algorithm by learning permutations [15]. When the conflict graph is complete, the upper bound of the regret achieves

$$\mathcal{O}\left(\sqrt{N^3T\log N}\right).$$

[3] Although the general conflict graph is used, [9] actually pays more attention to the full interference case.

One promising thread to further reduce the regret is to allow the users to exchange their historical observations (instead of only control signals for coordinating the behaviors of the users) with each other. The key behind this approach is to understand the trade-off between the regret and the communication overhead. This issue is highly nontrivial and necessitates further studies.

This article also utilizes the e-greedy algorithm to resolve the stochastic MAB problem. When $N = M$, an upper bound for the regret of $O(N^3 \log T)$ is guaranteed for the full interference setting.

Existing proposals usually assume that a user senses and accesses only one channel in each slot, but in reality, sensing a channel actually demands a much shorter time than accessing it. Therefore, it is possible to sense multiple channels sequentially in one slot, which has considerable potential to exploit instantaneous opportunities among the channels. Li *et al.* [11] proposed the so-called *Sequencing Multi-Armed Bandit* problem where each arm corresponds to a sequence of channels sensed by a user in a time slot. Once the user chooses an arm, it senses the channels in the corresponding order until achieving an available one. It is apparent that directly applying the UCB approach leads to significant overhead in both storage and computation due to the exponentially large number of arms. Therefore, [5, 11] fully exploited the overlapped sub-sequence between two arms, so as to reduce the storage overhead and the computation complexity down to a reasonable level. In particular, a novel UCB-based index, the so-called sequencing confidence bound (SCB), is designed such that the resulting channel sensing and accessing algorithm has both storage and computation overhead linear in M. Moreover, thanks to the exploitation-exploration trade-off brought by the UCB-based index, we have the expected regret upper-bounded by $O(Mn\log T)$, where n is the number of the channels that a user can sense in one time slot.

Kang *et al.* [8] focused on designing a low-complexity learning algorithm for the channel allocation problem. As mentioned above, for a multi-user multi-channel network with the full interference model, the channel allocation problem is formulated as a matching problem based on a bipartite graph. Given a permutation of the users, the maximum weight matching can be calculated in a greedy manner by taking the UCB indices (associated with the user-channel combinations) as the weights. In each decision round, a candidate matching is calculated according to a random permutation of the users. The canditate matching is then compared to the one produced in the previous round, and the one with higher UCB value is adopted. Although the algorithm has linear complexity, it results in a compromised regret bounded by $O(SN!N^3 \log T)$, where S is the number of all feasible matchings.

Although existing proposals only consider calculating optimal decisions to minimize the regret (e.g., in throughput) with no concern toward guaranteeing constraints in other performance metrics, Cai *et al.* [12] proposed an online learning framework to realize network optimization with constraints also guaranteed. For example, in a hierarchical cognitive radio network, a channel is available for the secondary users only when there is no primary user accessing it. The goal is thus to maximize the rewards gained by the secondary user while ensuring the average number of the accessed primary-free channels is above a threshold. In this framework, sublinear bounds are achieved for both regret and constraint violation.

Decentralized Algorithms

A decentralized channel allocation algorithm was developed in [6] where the users decide in each round without information exchange during the channel selections. Unlike the SMAB model used in [5], the algorithm designed in [6] relies on an N-player M-armed bandit. Since the full interference model is employed, the users are prioritized such that each of them selects the maximally UCB-valued channel from the ones that have not been selected by the others with higher priorities. By generalizing the well-known UCB policy, the resulting regret is upper-bounded by $O(N(M - N)\log T)$.

The general interference model was used in [13]; this article nevertheless adapts the conflict graph to facilitate the algorithm design. Specifically, each vertex denotes a user-channel combination, and the edges represent the interference relationship among the vertices. Each vertex is associated with an unknown weight that represents the reward the corresponding user can obtain by accessing the channel. Then a feasible channel allocation scheme corresponds to an independent set in the (adapted) conflict graph, and the goal is to find the one with the maximum weight without prior knowledge about the weights of the vertices. Taking the feasible channel allocations as the arms, the MAB problem can be resolved by employing a distributed polynomial time approximation scheme (PTAS) as a subroutine to address the maximum weight independent set (MWIS) problem. The PTAS relies on a local $(2r + 1)$-hop broadcast in D slots and has an approximation ratio of $r = 1 - \varepsilon$, while the channel allocation algorithm results in a computational complexity of $O(Dm^{\rho^r})$ in each decision round, where m is the maximum number of r-hop neighboring users. Concerning regret, the algorithm achieves $1/\rho$ of the rewards yielded by the optimal solution.

Zhang *et al.* [10] built a conflict graph similar to the one used in [9]. The graph takes the users as the vertices, and two users are connected by an edge if they interfere with each other. In fact, allocating the given channels to the users in the conflict graph can be formulated as a coloring problem where we assign different colors (i.e., channels) to adjacent vertices (i.e., users). Each color-vertex pair is associated with an unknown weight, and the aim is to design a coloring scheme maximizing the resulting total weight. By leveraging the e-greedy scheme based on an N-player M-armed bandit model, Zhang *et al.* [10] utilize a distributed graph coloring algorithm to realize decentralized channel allocation with the regret scaling linearly as $O(T)$. The distributed coloring algorithm is implemented by message exchange among the users, resulting in a message complexity of $O(N^3)$.

Avner *et al.* [7] also proposed a decentralized channel allocation algorithm where only very limited information exchange among the users (i.e., the players) is allowed. Nevertheless, the maximization of the reward cannot be guaranteed. Hence, this proposal aims at achieving a *stable marriage configuration* where no two users would like to swap channels since they would not both benefit. In particular, in each decision round, each user ranks the channels according to the UCB indices. Then an initiator is elected from the users to signal others sequentially, according to its ranking of the

channels. Channel swapping occurs only when it benefits both users. This algorithm results in a very low communication complexity of $O(N)$, but logarithmic growth of the regret cannot be ensured.

The MAB framework also can be applied to schedule transmissions in multihop wireless networks [14]. The multihop network can be modeled as a general network graph where the vertices and edges denote the network nodes and transmission links, respectively. Each link has a capacity that is not known a priori; the objective is to select a subset of links to access the channel in each slot and thus maximize network throughput, subject to the interference constraints among the links (which are represented by a conflict graph). By associating each edge with a UCB index as the weight, the above problem can be addressed by calculating the maximum weight matching in a greedy manner in the network graph. Updating the UCB indices and greedily calculating the (nearly) optimal matching can be performed by individual nodes in a distributed fashion. In fact, the transmission links can be treated as the "users" accessing the channel for transmitting data and learning channel conditions. Assuming there are N links in the network graph, the proposed algorithm has a computational complexity of $O(N)$, and the regret is upper-bounded by $O(N\log T)$.

Future Directions

Although many MAB-based algorithms have been proposed for scheduling channel access in wireless networks, there are quite a few unexplored issues. We here present a few possible future directions.

Learning with Budget Constraints: According to the above survey, the overhead in pulling the arms (i.e., sensing and accessing the channels) is not often considered in existing proposals. In reality, wireless network systems (e.g., sensor network or cognitive radio networks) may be constrained by limited energy resources; hence, the energy spent on channel sensing and accessing should be considered.

Observation Sharing for Users: One promising thread to further reduce regret is to allow users to exchange their historical observations (instead of only control signals for coordinating the behaviors of the users [7]) with each other. The key behind this approach is to understand the trade-off between the regret and the communication overhead. This issue is highly nontrivial and requires further studies.

Fault Tolerance: As shown previously, one of the ways to coordinate user behaviors is information exchange [7, 10, 13], especially for decentralized application scenarios. To deploy wireless networks in uncertain environments, one main concern is fault tolerance. For example, the rewards of the arms are initially drawn from some probability distribution but are then (partially) alternated by oblivious or adaptive adversaries during information exchange. We believe that designing fault-tolerant algorithms under such a mixed model would be of considerable interest.

Conclusion

Next generation wireless networks have a high demand on exploiting spectrum resources, while spectrum uncertainty is an inevitable issue we have to address to serve this goal. Considering MAB as a typical reinforcement learning method by which the uncertainty can be learned from a sequence of trails, we have surveyed recently proposed algorithms in this article, which apply the MAB approach to schedule channel accessing. Specifically, we have revealed the key techniques in problem modeling and algorithm design. It has been demonstrated by these state-of-the-art proposals that the centralized algorithms emphasize achieving polynomial computation and storage overhead during the learning processes, while the decentralized ones have to pay additional attention to coordinating the individual users, such that logarithmic or polynomial upper bound on regret can be ensured. Finally, we have outlined some potential future research directions.

It has been demonstrated by these state-of-the-art proposals that, the centralized algorithms is with an emphasis on achieving polynomial computation and storage overhead during the learning processes, while the decentralized ones have to pay additional attentions on coordinating the individual users, such that logarithmic or polynomial upper-bound on the regret can be ensured.

Acknowledgments

This work is partially supported by the National Key R&D Program of China (Grant No. 2019YFB2102600), the NSFC (Grant Nos. 61702304, 61971269, 61832012, 61672321, and 61771289), the Shandong Provincial Natural Science Foundation (Grant No. ZR2017QF005), the China Postdoctoral Science Foundation (Grant No. 2017M622136), and project NICCI as part of the German Research Foundation's SPP 1914 program.

References

[1] S. Maghsudi and E. Hossain, "Multi-Armed Bandits with Application to 5G Small Cells," *IEEE Wireless Commun.*, vol. 23, no. 3, June 2016, pp. 64–73.
[2] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-Time Analysis of the Multiarmed Bandit Problem," *Machine Learning*, vol. 47, no. 2-3, 2002, p. 235–56.
[3] V. Kuleshov and D. Precup, "Algorithms for Multi-Armed Bandit Problems," *J. Machine Learning Research*, vol. 1, 2014, pp. 1–48.
[4] S. Bubeck and N. Cesa-Bianchi, "Regret Analysis of Stochastic and Nonstochastic Multi-Armed Bandit Problems," *Foundations and Trends in Machine Learning*, vol. 5, 2012.
[5] Y. Gai, B. Krishnamachari, and R. Jain, "Learning Multiuser Channel Allocations in Cognitive Radio Networks: A Combinatorial Multi- Armed Bandit Formulation," *Proc. IEEE DySPAN*, 2010, pp. 1–9.
[6] Y. Gai and B. Krishnamachari, "Decentralized Online Learning Algorithms for Opportunistic Spectrum Access," *Proc. IEEE GLOBECOM*, 2011, pp. 1–6.
[7] O. Avner and S. Mannor, "Multi-User Lax Communications: A Multi- Armed Bandit Approach," *Proc. 35th IEEE INFOCOM*, 2016, pp. 1–9.
[8] S. Kang and C. Joo, "Low-Complexity Learning for Dynamic Spectrum Access in Multi-User Multi-Channel Networks," *Proc. 37th IEEE INFOCOM*, 2018, pp. 1–9.
[9] M. Lelarge, A. Proutiere, and M. Talebi, "Spectrum Bandit Optimization," *Proc. Info. Theory Wksp.*, 2013, pp. 1–5.
[10] Y. Zhang *et al.*, "Learning Temporal-Spatial Spectrum Reuse," *IEEE Trans. Commun.*, vol. 64, no. 7, 2016, pp. 3092–3103.
[11] B. Li *et al.*, "Almost Optimal Dynamically-Ordered Channel Sensing and Accessing for Cognitive Networks," *IEEE Trans. Mobile Computing*, vol. 13, no. 10, 2014, pp. 2215–28.
[12] K. Cai *et al.*, "An Online Learning Approach to Network Application Optimization with Guarantee," *Proc. 37th IEEE INFOCOM*, 2018, pp. 2006–14.
[13] Y. Zhou *et al.*, "Almost Optimal Channel Access in Multi-Hop Networks with Unknown Channel Variables," *Proc. 34th IEEE ICDCS*, 2014, pp. 461–70.
[14] T. Stahlbuhk, B. Shrader, and E. Modiano, "Learning Algorithms for Scheduling in Wireless Networks with Unknown Channel Statistics," *Ad Hoc Networks*, vol. 85, 2019, pp. 131–44.
[15] D. Helmbold and M. Warmuth, "Learning Permutation with Exponential Weights," *J. Machine Learning Research*, vol. 10, 2009, pp. 1705–36.

Biographies

Feng Li received his B.S. and M.S. degrees in computer science from Shandong Normal University, China, in 2007, and Shandong University, China, in 2010, respectively. He received his Ph.D. degree (also in computer science) from Nanyang Technological University, Singapore, in 2015. From 2014 to 2015, he worked as a research fellow at the National University of Singapore. After that, he jointed the School of Computer Science and Technology, Shandong University, where he is currently an associate professor. His research interests include distributed algorithms and systems, wireless networking, mobile computing, and the Internet of Things.

Dongxiao Yu received his B.Sc. degree in 2006 from the School of Mathematics, Shandong University, and his Ph.D. degree in 2014 from the Department of Computer Science, University of Hong Kong. He became an associate professor in the School of Computer Science and Technology, Huazhong University of Science and Technology, in 2016. He is currently a professor in the School of Computer Science and Technology, Shandong University. His research interests include wireless networks, distributed computing, and graph algorithms.

Huan Yang received her Ph.D. degree in computer science from Nanyang Technological University in 2015, her M.S. degree in computer science from Shandong University in 2010, and her B.S. degree in computer science from the Heilongjiang Institute of Technology, China, in 2007. Currently, she is an associate professor in the College of Computer Science and Technology, Qingdao University, China. Her research interests include applied optimization, stochastic modeling, and machine learning.

Jiguo Yu [SM] received his Ph.D. degree from the School of Mathematics at Shandong University in 2004. He became a full professor in the School of Computer Science, Qufu Normal University, Shandong, China, in 2007. Currently he is a full professor at Qilu University of Technology (Shandong Academy of Sciences), and the Shandong Computer Science Center (National Supercomputer Center in Jinan). His main research interests include privacy-aware computing, wireless networking, distributed algorithms, peer-to-peer computing, and graph theory. He is particularly interested in designing and analyzing algorithms for many computationally hard problems in networks. He is a member of ACM and a Senior Member of the China Computer Federation.

Holger Karl graduated from the University of Karlsruhe in 1996 and obtained his Ph.D. from Humboldt University of Berlin in 1999, both in computer science. Since 2004, he has headed the Computer Networks Research Group, Paderborn University. He has two main research interests; first, advanced wireless networks, for example, cooperative diversity techniques and resource management in factory-floor automation; and second, future Internet, specifically the design and architecture of protocol stacks and unifying concepts like SDN and NFV across different scenarios.

Xiuzhen Cheng [F] received her M.S. and Ph.D. degrees in computer science from the University of Minnesota — Twin Cities in 2000 and 2002, respectively. She is a professor in the School of Computer Science and Technology, Shandong University. Her current research interests include cyber physical systems, wireless and mobile computing, sensor networking, wireless and mobile security, and algorithm design and analysis. She has served on the Editorial Boards of several technical journals and the Technical Program Committees of various professional conferences/workshops. She also has chaired several international conferences. She worked as a Program Director for the U.S. National Science Foundation (NSF) from April to October 2006 (full time), and from April 2008 to May 2010 (part time). She received the NSF CAREER Award in 2004. She is a member of ACM.