

# Multi-user lax communications: a multi-armed bandit approach

Orly Avner      Shie Mannor

December 3, 2015

## Abstract

Inspired by cognitive radio networks, we consider a setting where multiple users share several channels modeled as a multi-user multi-armed bandit (MAB) problem. The characteristics of each channel are unknown and are different for each user. Each user can choose between the channels, but her success depends on the particular channel chosen as well as on the selections of other users: if two users select the same channel their messages collide and none of them manages to send any data. Our setting is fully distributed, so there is no central control. As in many communication systems, the users cannot set up a direct communication protocol, so information exchange must be limited to a minimum. We develop an algorithm for learning a stable configuration for the multi-user MAB problem. We further offer both convergence guarantees and experiments inspired by real communication networks, including comparison to state-of-the-art algorithms.

## 1 Introduction

The inspiration for this paper comes from the world of distributed multi-user communication networks, such as cognitive radio networks. These networks consist of a set of communication channels with different characteristics, and independent users whose goal is to transmit over these channels as efficiently as possible.

Modern networks, such as cognitive radio networks, must cope with several challenges. First and foremost, the networks' distributed nature prohibits any form of central control. In addition, many users operate on an "ad hoc" basis, preventing them from forming inter-user communication. In fact, they probably do not even know how many users share their network.

On top of these issues of multi-user coordination, the channel characteristics may be initially unknown, and differ between users. Thus, learning must be integrated into the solution.

## 1.1 Cognitive radio networks

Cognitive Radio Networks (CRNs), introduced in [1], have attracted considerable attention in recent years. The idea that lies at the heart of CRNs is that advanced sensing mechanisms and increased computation power may enable radio devices to dramatically improve their performance in terms of resource utilization, resilience and more. Networks of such users are usually dynamic and stochastic, giving rise to many interesting problems [2,3]. We focus on developing a sensing and transmission scheme that enables users to learn a stable, orthogonal configuration without communicating directly.

## 1.2 Multi-armed bandits

A well known framework for learning in CRNs is the classical Multi-Armed Bandit (MAB) model. MABs offer a simple, intuitive framework for learning the characteristics of a number of unknown options in an online manner, while balancing exploration and exploitation. A MAB problem consists of a single user repeatedly choosing between arms with different characteristics, that are initially unknown. After every round, the user acquires a reward that depends on the arm she chose. Her goal in most setups is to maximize the expected sum of rewards acquired over time.

As suggested in [4], the channels of a CRN are naturally cast as the arms of a bandit, with different performance measures (bandwidth, ACK signals, bit rate) serving as the reward.

Many papers propose solutions for the stochastic MAB problem (see, e.g., [5–7]) and its adversarial version (see, e.g., [8]), but they all assume a single user is sampling the arms of the bandit.

However, this assumption does not apply in multi-user networks. In the multi-user MAB model, users compete over the arms of *the same* bandit. As a result, they are bound to experience collisions (i.e., multiple users sampling the same arm), unless they employ some form of collision avoidance or coordination mechanism. Collisions in communication networks result in performance degradation, corresponding to reward loss in the MAB model. In order to avoid reward loss, the presence of multiple users must be addressed. We survey several approaches to this issue in Section 1.4.

## 1.3 Extension of the CRN-MAB setting

The novelty introduced in our paper lies in the combination of bandit learning, multiple users, different reward distributions for different users and no direct communication. The combination of these last two demands - different distributions and no direct communication, poses a real challenge.

As explained in detail in Section 2.3 and in Section 2.4, the only thing we can guarantee in terms of network behavior in this setup is stability. In a dynamic, distributed network, stability is of great value. Once a network has reached a stable configuration, users can focus on utilizing its resources, rather than

engaging in coordination or learning efforts; a stable network is more robust and efficient.

Reaching stability is a nontrivial task, since users must learn their channel characteristics while coordinating their actions with the other users, based on very limited observations.

## 1.4 Previous work

We now present several approaches to the CRN-MAB problem, coming from different areas and disciplines.

Our problem may be viewed as an assignment problem, i.e., maximum weight matching in a weighted bipartite graph. Users correspond to agents, channels to tasks, and rewards are simply the complementary of the costs of graph edges. Several papers have been published on the distributed assignment problem, but to the best of our knowledge none of them offers a solution for our problem. The well-known Hungarian method [9] requires full knowledge of the graph (i.e., channel characteristics) and assumes the existence of central control. The Bertsekas auction algorithm [10] frees us from the need for central control, at the cost of direct communication between nodes. The classical Gale-Shapley algorithm [11] solves the problem of finding a stable marriage configuration, but does not take the need to learn into account. Some papers have actually applied it to CRNs, but not in the learning context [12,13]. Another work on distributed stable marriage, that makes use of a variant of the Gale-Shapley algorithm, is [14]. While it is quite foreign to our problem, the potential function defined in the paper is helpful in our analysis. Another noteworthy work in this context is [15]. The authors address the challenge of limiting communication between nodes to a minimum, and propose two communication models. Nevertheless, they allow more communication than we would like, and their formulation does not consider learning. Two additional results that deal with distributed stable marriage offer lower bounds and state that *some* form of information exchange is inevitable when solving such problems [16,17].

The papers closest to ours in spirit are those dealing with multi-user MABs. There has been work on the case of reward distributions that do not vary between users, such as [18] and [19]. The latter introduces an algorithm that is able to cope with a variable number of users. Another paper, that addresses different reward distributions for different users, is [20]. Here, the authors employ the Bertsekas auction algorithm. This approach enables users to reach a reward-maximizing solution, at the price of direct, frequent communication between themselves. We further elaborate on the difference between our approach and the approach of [20] in Section 6.

To this end, we would like to point out that communication between users is undesirable not only because of its price in terms of network resources and time. Once users depend on communication, they are more vulnerable to intentional attacks that may disrupt it, as well as noise bursts that are common in CRNs.

## 2 Model and formulation

We now describe the model, the assumptions accompanying it and our goal.

### 2.1 System and users

We model a communication network with  $K$  channels, servicing  $N$  independent users. Our work is based on the assumption that  $K \geq N$ , which is reasonable since without it, implementing a time division based mechanism is necessary. Once such a mechanism is applied, the assumption that  $K \geq N$  is valid again. Time is slotted and users' clocks are synchronized, also a mild assumption for modern communication systems.

The communication network consists of  $K$  channels, where only one user can transmit over a certain channel during a single time slot. Each transmission yields a reward, which we assume to be stochastic.

The users are a group of  $N$  independent, selfish agents. Their observations are local, consisting only of the history of their actions and rewards. In addition, they do not know the number of users they share a network with. There is no central control managing their use of the network, and they do not have direct communication with each other.

A key characteristic of our model is that the expected reward a channel yields depends not only on the identity of the channel, but also on the identity of the user. Formally, the rewards of the channels are Bernoulli random variables with expected values  $\{\mu_{n,k}\}$ , where  $n \in \{1, \dots, N\}$  and  $k \in \{1, \dots, K\}$ . This property reflects the fact that in real-life users may experience location-based disturbances, manifested in different reward distributions for the same channel.

We model the users' sharing resources through the representation of the communication network by a *single* bandit. This means that two users attempting to access the same channel at the same time, will experience a collision. In our model, the result of a collision is complete loss of communication for that time slot for the colliding users, i.e., zero reward. A user  $n$  that accesses a channel  $k$  alone during a certain time slot will receive a reward drawn i.i.d. from a Bernoulli distribution with expected value  $\mu_{n,k}$ . Throughout the paper, we use the term *configuration* to refer to a mapping of users to channels.

### 2.2 Limited coordination

In an effort to keep our model faithful to real world CRNs, we limit the coordination between users to a minimum. Thus, users can only transmit in a channel of their choice, or sense the spectrum range and receive binary feedback regarding all channels  $\{1, \dots, K\}$  at time  $t$ . A "0" represents no transmission in channel, while "1" stands for the opposite.

## 2.3 Reward maximizing solution

We adopt a system-wide view for characterizing the optimal solution. The optimal configuration must be orthogonal (i.e., no more than one user per channel), in order to avoid collisions and the resulting reward loss. One common approach seeks to maximize the sum of rewards over all users, over time. The assignment of users to channels is chosen accordingly:

$$R^* = \max_{\pi \in \mathcal{C}} \sum_{n=1}^N \mu_{n,\pi(n)},$$

where  $\mathcal{C}$  is the set of all possible permutations of subsets of size  $N$  chosen without replacement from the set  $\{1, \dots, K\}$ .

However, reaching such a solution requires frequent information exchange. Assume channel  $k$  is optimal for two different users  $m$  and  $n$ , but  $\mu_{m,k} > \mu_{n,k}$ . To maximize the system-wide reward, user  $n$  must step down and choose a different channel. The lack of central control requires explicit information exchange regarding the values of  $\mu_{m,k}$  and  $\mu_{n,k}$ , for  $m$  and  $n$  to decide which of them should step down. Since the reward estimates are updated as time goes by, such preferences must be communicated repeatedly.

Due to limited information exchange, a reward-maximizing solution cannot be guaranteed in our setup. We therefore focus on convergence to a stable, orthogonal configuration.

## 2.4 Stable marriage solution

Our goal is to develop policies that will lead users to a stable configuration. We employ the notion of stable marriage to formally define stability:

**Definition 1.** *A Stable Marriage Configuration (SMC) is an assignment of users to channels such that no two users would be willing to swap channels, had they known the true values of the expected rewards. Formally, for a pair of users  $n, m$ :*

$$\begin{aligned} S_1 &\triangleq (\mu_{n,a_n} < \mu_{n,a_m}) && \text{user } n \text{ would like to swap} \\ S_2 &\triangleq (\mu_{m,a_m} \leq \mu_{m,a_n}) && \text{user } m \text{ is willing like to swap,} \end{aligned}$$

where  $a_m$  and  $a_n$  are the users' current actions. In an SMC,

$$S_1 \wedge S_2 = 0 \quad \forall n, m.$$

## 2.5 Goal

Given a system with  $K$  channels and  $N$  users, allowing only limited communication as described in Section 2.2, our goal is to reach a configuration that is orthogonal: no two users use the same channel, and an SMC, according to Definition 1.

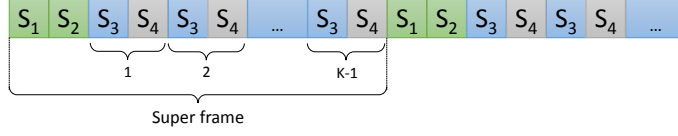


Figure 1: Frame structure

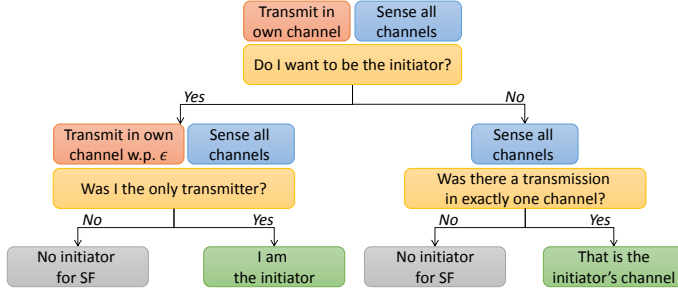


Figure 2: Selection of initiator

### 3 Coordination protocol

Our coordination protocol balances the limitations of Section 2.2 with the users' need for information exchange by introducing a signalling mechanism between pairs of users. At predefined time slots, a user wishing to occupy a channel may transmit in that channel to express her wish. In order to ensure that this signal is received by the user currently occupying the channel, we employ a frame-based protocol. We assume users can transmit and sense at the same time, a reasonable requirement in modern communication systems.

The following explanation is best understood by observing Figure 1. Our protocol divides time into super frames of length  $T_{\text{SF}} = 2 + 2(K - 1)$ . Each super frame begins with a pair of time slots,  $S_1$  and  $S_2$ , during which a single signalling user, the initiator, is coordinated for the entire super frame. The procedure is described in Algorithm 4 and in Figure 2. Next come  $K - 1$  mini-frames of two time slots each, denoted by  $S_3$  and  $S_4$ . Each of these mini-frames corresponds to one channel on the initiator's list of preferred channels. Thus, a single super frame enables one user to go over her entire preference list and signal other users, suggesting they swap channels with her, as explained in Figure 3.

The time slots marked  $S_4$  allow users not participating in the coordinating process during a certain mini-frame to sample their current channel and proceed with the learning-while-transmitting process. Thus, all but two users (initiator and responder) gather a sample during each mini-frame, resulting in at least  $K - 2$  samples for each of the users, except for the initiator, over each super frame.

While this may seem like much coordination, the protocol is very simple

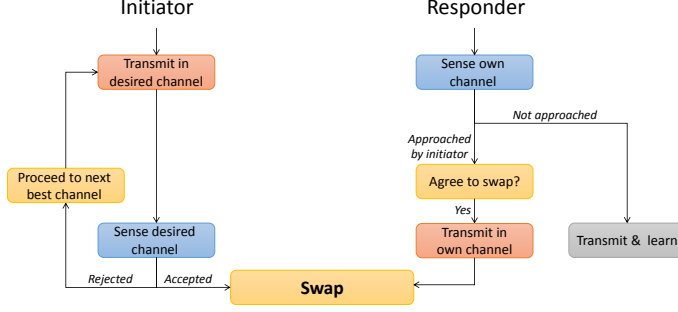


Figure 3: Initiator-responder dynamics

to implement, and is indeed lightweight when compared to other protocols, as further explained in Section 5 and in Section 6.

## 4 The CSM-MAB algorithm

We now turn to a full description of our algorithm, the Coordinated Stable Marriage Multi-Armed Bandit (CSM-MAB) algorithm. We propose a user-level algorithm for a fully distributed system, whose goal is described in Section 2.5. When all users in the network apply CSM-MAB, the assignment of users to channels is guaranteed to be orthogonal, and converges to an SMC.

Our algorithm begins with a start up phase, during which users transmit and sense to detect collisions, in order to reach an initial orthogonal configuration (line 1). This phase follows the lines of the CFL algorithm introduced in [21], and converges quickly. Once an initial orthogonal configuration has been reached, users start executing the CSM-MAB algorithm, described in Figure 4.

At the beginning of each super frame, users execute the **rank.channels** procedure to individually create a list of channels they prefer over their current action (line 4). Channels are assigned values according to their UCB indices, calculated using the well known formula from [5]:

$$I_{n,k}(t) = \hat{\mu}_{n,k} + \sqrt{\frac{2 \ln t}{s_{n,k}}}, \quad (1)$$

where  $\hat{\mu}_{n,k}$  is the empirical mean of the reward acquired by user  $n$  on channel  $k$  up till time  $t$  and  $s_{n,k}$  is the number of times she sampled arm  $k$  up till time  $t$ .

Next, the users coordinate an initiator according to the scheme in Figure 2. Every user who would like to improve upon her current channel presents herself as the initiator with a probability of  $\epsilon = \frac{1}{K}$  (lines 5-11). An agreed initiator for the SF emerges if and only exactly one user raises her flag (the value of  $\epsilon$  is chosen in order to maximize the probability of this occurring). Once a single initiator is agreed upon, all users take note of her current channel, based on

```

1:  $a_n(0) \leftarrow \text{apply\_CFL}(K)$ 
2: for all frames  $t$  do
3:   if  $\text{mod}(t, T_{\text{SF}}) == 1$  then {Beginning of SF}
4:      $list \leftarrow \text{rank\_channels}(a_n(t-1), \hat{\mu}_n, s_n)$ 
5:     if  $list \neq 0$  then {User seeks to change channel}
6:        $flag_n \leftarrow \text{rand}(\text{Bernoulli}, \epsilon)$ 
7:       if  $(flag_n == 1) \wedge (flag_i == 0 \forall i \neq n)$  then
8:          $initiator = n$  {User  $n$  is initiator for this SF}
9:          $pref = 1$  {Initialize swapping preference to 1}
10:      end if
11:    end if
12:  else
13:    if  $(initiator == n) \wedge (pref > 0)$  then { $n$  is the initiator,  $list$  not exhausted yet}
14:       $response \leftarrow \text{propose\_swap}(list(pref))$ 
15:      if  $response == 1$  then {Responder agreed or channel is available}
16:         $a(t) \leftarrow \text{swap}(a_n(t), list(pref))$ 
17:         $pref \leftarrow 0$ 
18:      else
19:         $pref \leftarrow pref + 1$  {Move to next best channel}
20:      end if
21:    end if
22:  end if
23:   $r_n(t) \leftarrow \text{execute\_action}(a_n(t))$ 
24:   $\text{update\_stats}(r_n(t), \hat{\mu}_{n,a_n(t)}, s_{n,a_n(t)})$ 
25: end for
note:  $\hat{\mu}_{n,k}$  is the empirical mean of the reward for user  $n$  on arm  $k$ ;  $s_{n,k}$  is the number of times she has sampled it.

```

Figure 4: The CSM-MAB algorithm



their sensing. They will need this knowledge to decide whether to accept her swapping suggestion.

The initiator proceeds to signal other users, based on her ranking of channels (lines 13-21). Signalling is implemented in **propose\_swap** by transmitting in the initiator's channel of interest. Each responder (i.e., signalled user) checks whether swapping channels with the initiator will improve her situation, based on her own ranking. Once a responder agrees, a swap takes place. No more signalling attempts are made till the end of the SF, and users simply continue sampling their chosen channels. If the responder refuses, the initiator will approach the next-best channel on her list. She will continue the process until she (a) finds a partner that agrees to swap; or (b) exhausts her list of potential swaps. This part of the algorithm is depicted in Figure 3.

## 5 Analysis

We will now show that the CSM-MAB meets the goals defined in Section 2.5. Our main theoretical result is stated in Theorem 1.

**Theorem 1.** *Consider a system with  $K$  channels and  $N$  users, with channel rewards characterized by the matrix  $\boldsymbol{\mu}$ . Applying CSM-MAB (Algorithm 4) by all users will result in convergence to an orthogonal SMC: For all  $\delta > 0$  there exists  $T(\delta)$  such that for all time slots  $t > T$ , the probability of the system's being in an SMC is at least  $1 - \delta$ .*

The proof of Theorem 1 consists of two aspects: orthogonality and stability. The first part is easy to verify.

**Proposition 1.** *The actions of users applying CSM-MAB are orthogonal (i.e., there is at most one user sampling each channel) for all  $t > t_0$  with probability of at least  $1 - \delta_0$ .*

*Proof.* Based on Theorem 1 of [21], the initial configuration reached after running the CFL algorithm is orthogonal with probability 1. The authors provide an upper bound on the distribution of stopping times,  $\tau$ :

$$\mathbb{P}[\tau > k] = \alpha e^{-\gamma k},$$

where  $\alpha$  and  $\gamma$  are some positive constants. The expected stopping time is therefore upper bounded by  $\frac{\alpha e^{-\gamma}}{1 - e^{-\gamma}}$ . Thus, setting  $t_0 \triangleq \frac{2\alpha e^{-\gamma}}{1 - e^{-\gamma}}$ , the probability of not having reached an orthogonal configuration by time  $t_0$  is at most  $\delta_0 \triangleq e^{-2\frac{\alpha e^{-\gamma}}{1 - e^{-\gamma}}}$ . Once the system reaches an orthogonal configuration, a user does not switch to an occupied channel without having coordinated the switch, as defined in Algorithm 4.  $\square$

### 5.1 Stability and potential

Showing that our system converges to a stable solution is more involved. We begin by defining a potential function for the problem. For any user  $n \in$

$\{1, \dots, N\}$ , the potential at time  $t$  is defined as follows:

$$\phi_n(t) \triangleq \sum_{k=1}^K \mathbb{1} \{ \mu_{n,k} > \mu_{n,a_n(t-1)} \}, \quad (2)$$

where  $a_n(t-1)$  is the action taken by user  $n$  in the previous time step. In words, the potential is the number of channels user  $n$  would prefer over her current choice, had she known their true reward distributions. The system-wide potential is the sum of potentials over all users:

$$\Phi(t) \triangleq \sum_{n=1}^N \phi_n(t) \quad (3)$$

An illustration of the potential appears in Tables 1 and 2.

Table 1: Table of users' channel rankings (first row represents best channel, last row represents worst). Cells highlighted in yellow and underline represent user's current choice.

	$U_1$	$U_2$	$U_3$
<b>1</b>	1	2	<u>4</u>
<b>2</b>	2	<u>1</u>	1
<b>3</b>	4	3	2
<b>4</b>	<u>3</u>	4	3

Table 2: User potentials corresponding to the configuration in Table 1.

$\phi_1$	$\phi_2$	$\phi_3$
3	1	0

In terms of potential, a configuration is an SMC if no two users can swap channels and decrease their potential by doing so. We note that a stable configuration does not necessarily correspond to zero system-wide potential, since not all users might be able to achieve zero potential simultaneously, depending on network parameters. Also, a system may have several stable configurations, each characterized by a different potential. Nevertheless, observing a system's potential does provide an indication regarding stability: once a system reaches a stable configuration, its potential will no longer change.

We prove convergence to an SMC by using the potential function, considering three aspects:

1. The maximal potential of a system with  $K$  channels and  $N$  users is finite and equal to  $N(K-1)$ .
2. The potential  $\Phi(t)$  is monotonously non-increasing with high probability.

3. Until an SMC is reached, changes in potential are bound to happen within finite time.

We formalize and prove these statements in the sequel.

Since users' decisions are guided by UCB indices, while stability is examined with respect to true reward distributions, users do not always update their choice of channels in a way that matches the ground truth. Thus, the system potential may occasionally increase, due to users' exploration or inaccurate statistics. In our proof we show that despite this, users ultimately converge to a stable configuration.

## 5.2 Proof of Theorem 1

We begin by ensuring the monotonicity of the potential.

**Lemma 1.** *For all times  $t$  for which  $t > \frac{16K}{\Delta_{\min}^2} \ln t$ , if a change in potential occurs, it is a decrease, with probability of at least  $1 - 2t^{-4}$ .*

$\Delta_{\min}$  is a distribution dependent constant. In the appendix we derive an upper bound on the minimal time for which the condition above holds:

$$t_{\min} \leq \frac{M - 1 - \sqrt{(M - 1)^2 - 4M}}{2}, \quad (4)$$

where  $M \triangleq \frac{16K}{\Delta_{\min}^2}$ . This bound will enable us to use  $t_{\min}$  in the proof.

Next, we introduce a lemma that concerns the ability of a single user to reach the position of the initiator.

**Lemma 2.** *If  $\phi_n(t) > 0$  for some user  $n$ , then her probability of becoming the next initiator is at least  $\epsilon(1 - \epsilon)^{N-1}$ .*

Using Lemma 2, we show another result:

**Lemma 3.** *If the system is not in an SMC at some time  $t$ , then a change in the potential will occur within no more than  $t'(\delta_1)$  time slots with probability of at least  $1 - \delta_1$ .*

The exact dependency of  $t'$  on  $\delta_1$  appears in the appendix, as do the proofs of all lemmas.

The probability of the system's reaching an SMC within  $\tau \triangleq t'N(K - 1)$  time slots after time  $t_{\min}$  is at least

$$P_{\text{SMC}} \triangleq \left[ (1 - \delta_1)(1 - 2t_{\min})^{-4} \right]^{N(K-1)}.$$

We model the convergence to an SMC using a Markov chain. Let  $S_t$  denote the state of the system at time  $t$ :

$$S_t = \begin{cases} 0 & \text{if in SMC,} \\ 1 & \text{else.} \end{cases}$$

The following holds for the chain's transition probability:

$$\mathbb{P}[S_{t+\tau} = 1 | S_{t_{\min}} = 0] \geq P_{\text{SMC}},$$

and also

$$\mathbb{P}[S_T = 0 | S_{t_{\min}} = 0] \leq (1 - P_{\text{SMC}})^{\lfloor \frac{T-t_{\min}}{\tau} \rfloor}, \quad \forall T > t_{\min} + \tau.$$

Defining  $\delta \triangleq (1 - P_{\text{SMC}})^{\lfloor \frac{T-t_{\min}}{\tau} \rfloor}$  completes the proof, and inverting yields

$$T = t_{\min} + \tau \frac{\ln \delta}{\ln(1 - P_{\text{SMC}})}.$$

Our next result quantifies the time devoted to signalling.

**Proposition 2.** *In every super-frame  $(K-1)(N-2)$  learning samples are gathered by all users combined. During this period  $4K$  signalling and sensing actions are performed by all users combined, so the signalling to learning ratio is*

$$L \triangleq \frac{4K}{(K-1)(N-2)}.$$

Clearly, the effort the users put into coordination is most effective when the number of users is close to the number of channels. This is a result of the frames' length being dictated by the number of channels rather than the number of users, in order for the user-level algorithm to be independent of the number of users.

## 6 Experiments

To demonstrate the merits of our algorithm, we implement a simulation of a distributed multi-user communication network. The users in our network are synchronized, and time is slotted.

In this network, users cannot communicate with each other directly. However, they can sense the entire frequency range (i.e., listen to all channels). They may also transmit over a channel of their choice, updating this choice each time slot.

A user  $n$  transmitting over a channel  $k$  receives a binary reward, drawn i.i.d. from a Bernoulli distribution with parameter  $\mu_{n,k}$ . This can be viewed as a form of the classic binary symmetric channel. As far as the different values of the reward parameters go, we ran experiments in two different modes:

1. random: the  $\mu_{n,k}$ 's are drawn uniformly and independently from the interval  $[0, 1]$ .

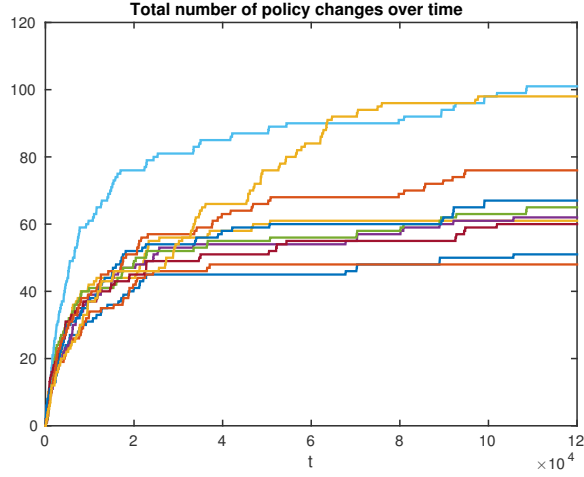


Figure 5: Changes in users' choice of channels, single realization

2. real-world: users are divided into clusters, and each cluster has a preferred group of channels. This represents a scenario in which users sharing a cluster are geographically close, and experience an interference in part of the frequency range. In real-world wireless communication systems, an agent that does not belong to the network but is transmitting in its vicinity will often cause a similar phenomenon.

We present results obtained in an experiment with  $K = 12$  channels and  $N = 10$  users. The users are divided into two clusters. Users 1-5 belong to one cluster, and experience an interference in the frequency range of channels 7-12. Users 6-10, on the other hand, experience similar performance over the entire frequency range. Experiments last  $T = 120000$  time slots, and results are averaged over 50 repetitions.

We begin by examining the cumulative number of policy changes per user over time, plotted in Figure 5 and in Figure 6. Since our goal is stability, we would like the number of policy changes to be small, and indeed the rate of changes decreases significantly over time. Another observation, demonstrated by the two figures, is that different users have different patterns, depending on the realization but more importantly on the difficulty of their problem: users that have small differences between channels will need more samples in order to tell them apart, and will therefore experience more policy changes.

Our next result examines the convergence to different SMCs over several repetitions of one setup. In this case, the set of SMCs consists of 305 configurations. Naturally, the size of this set depends on the number of users,  $N$ , the number of channels,  $K$ , and also on the specific realization of the  $\mu_{n,k}$ 's. Figure 7 shows that the periods of time users spend in unstable configurations decrease as the experiment advances, and users move between different SMCs,

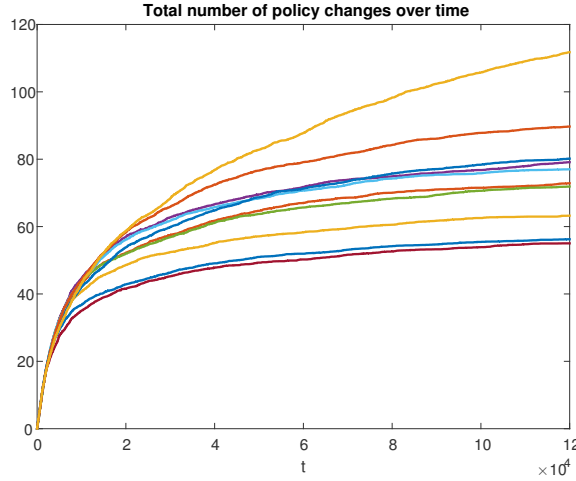


Figure 6: Changes in users’ choice of channels, empirical average

depending on the realization.

To complement our proof, we provide a visualization of the system potential over time, averaged over several repetitions, in Figure 8. As shown in the proof, the potential decays on average. The shaded area around the plot represents the variance over iterations, which also decays over time. As explained in Section 5.1, the potential does not necessarily decay to zero, but rather to a constant value that represents the potential of the SMC.

Our last result examines the reward acquired by users employing the CSM-MAB algorithm. While our theoretical guarantees focus on stability, the algorithm incorporates reward maximization implicitly by using UCB indices to rank channels. However, as explained in Section 2.3, reaching a reward-optimal configuration cannot be guaranteed with the limited form of communication we allow. In Figure 9 we compare the cumulative system-wide reward of two algorithms: our CSM-MAB and the dUCB4 algorithm, introduced in [20]. As explained in Section 1.4, dUCB4 incorporates an auction algorithm in order to achieve an orthogonal reward maximizing configuration.

The price of reward maximization is, clearly, communication, which our scheme attempts to bring to a minimum. In order to implement the auction algorithm required by dUCB4, users must have distinct id’s and knowledge of the number of users. This rather technical requirement hinders the ability of the algorithm to deal with a variable number of users. Our algorithm naturally extends to a scenario in which users arrive and leave at random times, that is quite likely in the context of CRNs. In addition, auction algorithms inherently rely on the good will of users, and are therefore more vulnerable to malicious agents (e.g., agents that report false high bids for attractive channels).

The results in Figure 9 demonstrate the tradeoff between communication

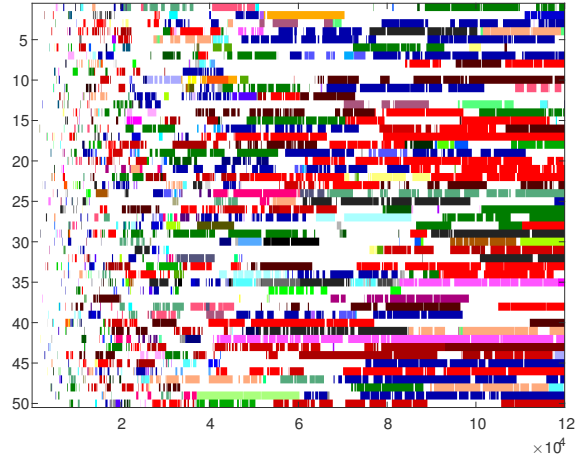


Figure 7: Convergence to SMC for different realizations: horizontal axis shows time, vertical axis shows numbering of realizations. White pixels represent unstable configurations, other colors correspond to different SMCs. As time goes by, longer stretches of time are spent in SMCs.

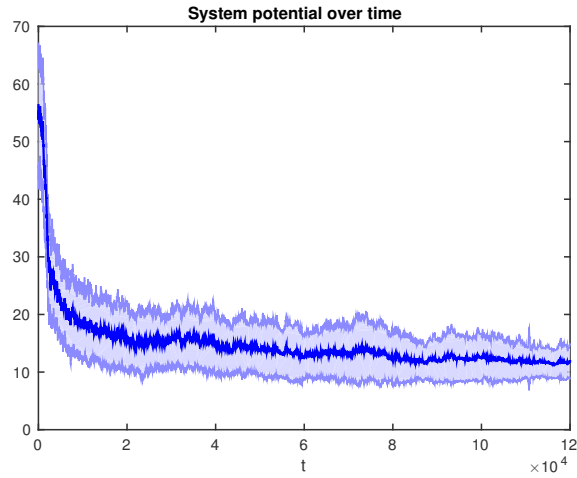


Figure 8: Decay of system potential over time, averaged over 50 repetitions. The shaded area represents variance.

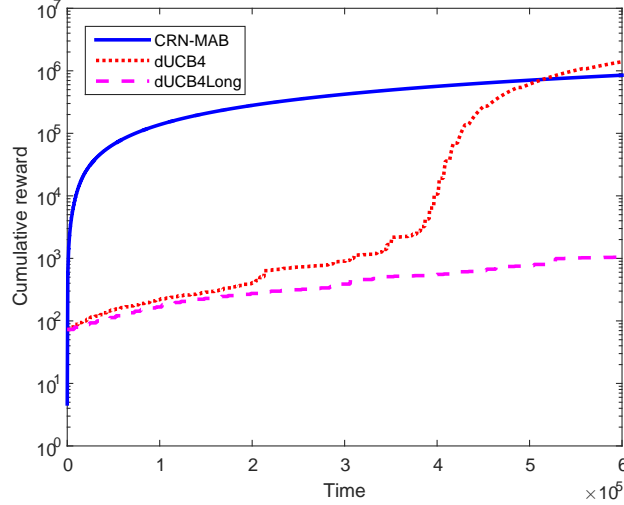


Figure 9: Cumulative system-wide reward over time, for different algorithms

and reward maximization: the time dUCB4 invests in auctioning is quite dominant. The two variants of the algorithm differ in the accuracy of the auctioning algorithm. The “dUCB4” variant (dotted red) uses 32 bits to encode variables, while the “dUCB4Long” variant (dashed magenta) uses 64 bits. Because of auctioning, it takes the algorithm a long time to turn its focus to reward maximization. In the high-accuracy case, the users exhaust all their time auctioning. In the low-accuracy case, they only begin acquiring rewards towards the end of the experiment. In real-world networks, with constantly changing conditions, such a long start-up phase is difficult to overlook. For the sake of example, let us examine an average 802.11n WLAN network, with a nominal frame size of 2000 bits and typical bit rate of 25 megabits per second. The  $4 \cdot 10^5$  time slots it takes dUCB4 to start acquiring rewards are translated into a period of  $\frac{4 \cdot 10^5 \cdot 2000}{25 \cdot 10^6} = 32\text{sec}$ . This start-up phase doubles to over one minute when 64 bit accuracy is used for the auction algorithm. Of course, lighter schemes than the 802.11 can be used, but these numbers clearly demonstrate the potentially crippling overhead brought on by communication.

We note that when  $N$  is strictly less than  $K$ , our algorithm often reaches the reward optimal configuration, or a configuration very similar in reward values. Therefore, the variance of the cumulative reward is very small. Our intuitive explanation is that when  $N < K$  users have a certain degree of freedom, increasing their chances of landing in the optimal configuration.

Despite reaching a configuration that is very close to optimal in the presented simulations, our algorithm acquires reward at a slower rate than dUCB4, due to the constant ratio of coordination and exploitation. Decreasing the amount of time devoted to coordination may considerably increase the reward, at the



cost of impairing the algorithm’s ability to handle a variable number of users. We plan to address this issue in detail in the future.

## 7 Discussion

We present an extension of the multi-user MAB problem, for the case of different reward distributions between the users, with limited information exchange. Using a specialized signalling method, our algorithm enables multiple users to learn network characteristics and converge to an orthogonal configuration that is also a stable marriage. We provide a theoretical analysis of our algorithm’s performance, based on the notion of system potential. Finally, we present the results of an experimental setup and examine different aspects of our approach’s performance, including a comparison to the dUCB4 algorithm of [20]. As explained in Section 6 in further detail, the main difference between the algorithms is the way they strike a balance between minimizing communication and maximizing the reward. We argue that our algorithm is better suited for real world problems.

In the future we intend to extend our work to a dynamic scenario, both in terms of channel characteristics and number of users. The latter should be straightforward due to the minimal inter-dependency of users, while the former will require some adjustment of the learning algorithm. Another interesting variant, applicable to networks with a fixed number of users, alters the ratio between coordination and exploitation as time goes by, to enable better use of network resources.

## References

- [1] J. Mitola and G. Maguire, “Cognitive radio: making software radios more personal,” *Personal Communications, IEEE*, 1999.
- [2] I. Akyildiz, L. Won-Yeol, M. Vuran, and S. Mohanty, “A survey on spectrum management in cognitive radio networks,” *Communications Magazine, IEEE*, vol. 46, no. 4, pp. 40–48, April 2008.
- [3] S. Haykin, “Cognitive radio: brain-empowered wireless communications,” *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 2, pp. 201 – 220, February 2005.
- [4] W. Jouini, D. Ernst, C. Moy, and J. Palicot, “Multi-armed bandit based policies for cognitive radio’s decision making issues,” in *Signals, Circuits and Systems (SCS), 2009 3rd International Conference on*. IEEE, 2010, pp. 1–6.
- [5] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” *Machine learning*, vol. 47, no. 2, 2002.

- [6] P. Auer and R. Ortner, “UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem,” *Periodica Mathematica Hungarica*, vol. 61, no. 1, pp. 55–65, 2010.
- [7] A. Garivier and O. Cappé, “The KL-UCB algorithm for bounded stochastic bandits and beyond,” in *Conference On Learning Theory*, 2011, pp. 359–376.
- [8] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire, “The nonstochastic multiarmed bandit problem,” *SIAM Journal on Computing*, 2002.
- [9] H. Kuhn, “The hungarian method for the assignment problem,” *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- [10] D. Bertsekas, “The auction algorithm: A distributed relaxation method for the assignment problem,” *Annals of operations research*, 1988.
- [11] D. Gale and L. Shapley, “College admissions and the stability of marriage,” *American mathematical monthly*, pp. 9–15, 1962.
- [12] K. Cohen, A. Leshem, and E. Zehavi, “Game theoretic aspects of the multi-channel aloha protocol in cognitive radio networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 31, 2013.
- [13] A. Leshem, E. Zehavi, and Y. Yaffe, “Multichannel opportunistic carrier sensing for stable channel access control in cognitive radio systems,” *Selected Areas in Communications, IEEE Journal on*, vol. 30, 2012.
- [14] P. Floréen, P. Kaski, V. Polishchuk, and J. Suomela, “Almost stable matchings by truncating the gale–shapley algorithm,” *Algorithmica*, vol. 58, no. 1, pp. 102–118, 2010.
- [15] N. Amira, R. Giladi, and Z. Lotker, “Distributed weighted stable marriage problem,” in *Structural Information and Communication Complexity*. Springer, 2010, pp. 29–40.
- [16] Y. Gonczarowski and N. Nisan, “A stable marriage requires communication,” *arXiv preprint arXiv:1405.7709*, 2014.
- [17] A. Kipnis and B. Patt-Shamir, “A note on distributed stable matching,” in *IEEE International Conference on Distributed Computing Systems*, 2009.
- [18] A. Anandkumar, N. Michael, A. Tang, and A. Swami, “Distributed algorithms for learning and cognitive medium access with logarithmic regret,” *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 4, pp. 731–745, 2011.
- [19] O. Avner and S. Mannor, “Concurrent bandits and cognitive radio networks,” in *European Conference on Machine Learning*, 2014.

- [20] D. Kalathil, N. Nayyar, and R. Jain, “Decentralized learning for multiplayer multiarmed bandits,” *IEEE Transactions on Information Theory*, vol. 60, no. 4, pp. 2331–2345, April 2014.
- [21] D. Leith, P. Clifford, V. Badarla, and D. Malone, “WLAN channel selection without communication,” *Computer Networks*, 2012.

# Appendices

## A Proof of Lemma 1

We would like to show that for all values of  $t$  for which  $t > \alpha \ln t$ , the probability that the potential decreases every time it changes is at least  $1 - 4t^{-4}$ , where  $\alpha = \frac{32K}{\Delta_{\min}^2}$ .

Given that a change in potential occurs at time  $t$ , it is guaranteed to result in a potential decrease if it benefits both users. This will happen if both users’ indices, that guide their decisions, are accurate w.r.t the true distribution.

Since we condition on a change in potential,

$$\mathbb{P}[\Phi_{\text{Dec}}] = 1 - \mathbb{P}[\Phi_{\text{Inc}}]$$

Let us upper bound  $\mathbb{P}[\Phi_{\text{Inc}}]$ . For a user  $n$  switching from arm  $j$  to arm  $i$  at time  $t$ ,  $\mu_{n,i} < \mu_{n,j}$ ,

$$\mathbb{P}[\Phi_{\text{Inc}}] = \mathbb{P}[I_{n,i}(t) \geq I_{n,j}(t) \cap \mu_{n,i} < \mu_{n,j}],$$

where  $I_{n,i}(t)$  is user  $n$ ’s UCB index of arm  $i$  at time  $t$ , defined in (1). Following the proof of Theorem 1 of [5],

$$\begin{aligned} \mathbb{P}[\Phi_{\text{Inc}}] &= \mathbb{P}[\hat{\mu}_{n,i}(t) + c_{t,s_{n,i}} \geq \hat{\mu}_{n,j}(t) + c_{t,s_{n,j}} \cap \mu_{n,i} < \mu_{n,j}] \\ &\leq 2t^{-4}, \end{aligned}$$

provided that

$$s_{n,i} \geq \frac{8 \ln t}{\Delta_{i,j}^2(n)}, \tag{5}$$

where  $s_{n,i}$  is the number of times user  $n$  sampled arm  $i$  up till time  $t$  and  $\Delta_{i,j}(n) \triangleq \mu_{n,i} - \mu_{n,j}$ . If (5) does not hold, then the UCB index “misleads” user  $n$ , causing her to mistakenly favor arm  $i$ , despite its lower expected reward. Switching from arm  $j$  to arm  $i$  will result in an increase in potential. However, once she acquires another sample of arm  $i$ , its index will decrease. In the meantime, the index of arm  $j$  will increase due to the passing time, and the indices will ultimately reflect the correct preference, resulting in a potential decrease.

The extreme value for (5), i.e., the largest number of required samples, corresponds to the minimal value of  $\Delta_{i,j}(n)$ . Let us define:

$$\Delta_n \triangleq \min_{\substack{i,j \in \{1,\dots,K\} \\ i \neq j}} [\mu_{n,i} - \mu_{n,j}]$$

$$\Delta_{\min} \triangleq \min_{n \in \{1,\dots,N\}} \Delta_n$$

Thus, when all arms have been sampled at least

$$s_{\min} \triangleq \frac{8 \ln t}{\Delta_{\min}^2} \quad (6)$$

times, the probability of an increase in potential is very small.

In order to allow for the coordination protocol, users do not gather informative samples in every time slot. Instead, they gather at least  $K - 2$  samples in each super frame, whose length is  $T_{\text{SF}} = 2 + 2(K - 1) = 2K$ .

Therefore, taking into account the fact that the sampling condition in (6) must apply for all arms, the condition on  $t$  is

$$t > K \frac{T_{\text{SF}}}{K - 2} s_{\min} = \frac{16K^2}{(K - 2) \Delta_{\min}^2} \ln t > \frac{16K}{\Delta_{\min}^2} \ln t. \quad (7)$$

For all times  $t$  for which (7) holds, if a change in potential occurs, it is a decrease, with probability of at least  $1 - 2t^{-4}$ .

When we apply this lemma we will use a quantity  $t_{\min}$ , an upper bound on the minimal  $t$  for which (7) holds. Introducing a well-known lower bound on the logarithmic function:

$$\ln x \geq \frac{x - 1}{x + 1} \quad \forall x > 1.$$

We use this lower bound together with (7):

$$t_{\min} = \frac{16K}{\Delta_{\min}^2} \ln t_{\min} \geq \frac{16K}{\Delta_{\min}^2} \frac{t_{\min} - 1}{t_{\min} + 1}.$$

Denoting  $M \triangleq \frac{16K}{\Delta_{\min}^2}$ , we continue:

$$t_{\min} \geq M \frac{t_{\min} - 1}{t_{\min} + 1}$$

$$t_{\min}^2 + (1 - M) t_{\min} + M \geq 0.$$

Our conclusion is that  $t_{\min} \leq \frac{M-1-\sqrt{(M-1)^2-4M}}{2}$ . Since this expression is finite, we may now use it in our proof.

## B Proof of Lemma 2

The probability of a specific user becoming the initiator when there are  $\ell$  interested users is

$$\begin{aligned} P_s(\epsilon, \ell) &\triangleq \mathbb{P}[\text{specific initiator} | \ell \text{ interested}] \\ &= \epsilon(1 - \epsilon)^{(\ell-1)} \quad \forall \ell \in \{1, \dots, N\}. \end{aligned}$$

The probability is minimized when all  $N$  users would like to become the initiator, yielding the bound  $\epsilon(1 - \epsilon)^{N-1}$ .

## C Proof of Lemma 3

If the system has not reached an SMC, then according to Definition 1, the conditions  $S_1$ ,  $S_2$  hold for at least one pair of users  $n, m$ .

According to the definition of the CSM-MAB algorithm, if  $S_1$  holds, then user  $n$  will add the channel user  $m$  is sampling to her list of preferred channels with a probability of at least  $1 - \delta$ . Following arguments similar to those presented in the proof of Lemma 1,  $\delta < 2t^{-4}$ . If  $S_2$  holds, user  $m$  will accept user  $n$ 's swap proposal, assuming her statistics are accurate. This, once again, happens with a probability of at least  $1 - \delta$ . Once users  $n$  and  $m$  swap channels, the potential will change.

In the worst case (i.e., largest  $t'$ ), user  $m$ 's channel will be the last channel on user  $n$ 's list, and all users higher on the list will decline user  $n$ 's swap proposals. If user  $n$  approaches a different user (whose channel is ranked higher than  $m$ 's), and that user agrees to swap, the potential will also change.

What is left to prove is that the time it shall take user  $n$  to receive the privilege of being initiator is finite. Once  $n$  is appointed the initiator, it will take no more than  $K - 1$  mini-frames, i.e.,  $2(K - 1)$  time slots, until she approaches user  $m$  and a swap takes place.

There are two different cases - if  $n, m$  are the the only unstable pair, then they will be the only ones interested in becoming the initiators. Furthermore, if only one of them is dissatisfied, then there will only be one user interested in initiating. In the notation of Lemma 2, this corresponds to  $\ell = 2$  or  $\ell = 1$ , respectively. The probability of exactly one of them becoming the initiator is  $P_{1,2} = \min\{\epsilon, 2\epsilon(1 - \epsilon)\}$ .

If there are additional unstable pairs, there will be more nominees for initiating. However, not all super frames necessarily result in a decrease in potential - if the initiator only targets channels occupied by "satisfied" users, all her attempts will be rejected. Therefore, we need to address the worst case scenario, in which all  $N$  users attempt to initiate, but only one of them is in a position that will actually result in a swap. Based on Lemma 2, the probability of that user emerging as the single initiator is at least  $\epsilon(1 - \epsilon)^{N-1}$ , for a single super frame. This probability is smaller than  $P_{1,2}$  for all  $\epsilon, N$ , and is therefore the lower bound for the probability of a single initiator with actual capacity for a decrease in potential.

The number of SFs in a time interval of length  $t'$  is  $C = \left\lfloor \frac{t'}{T_{\text{SF}}} \right\rfloor$ . The probability that a single initiator with actual capacity for a decrease in potential *does not* emerge in a certain SF is less than  $1 - \epsilon(1 - \epsilon)^{N-1}$ , and the probability that a single initiator does not emerge in the interval is less than  $P_C \triangleq \left(1 - \epsilon(1 - \epsilon)^{N-1}\right)^C$ . As  $t' \rightarrow \infty$ , so does  $C$ , and  $P_C$  decays to zero.

Binding the two aspects of this lemma together, we have that the probability of a single initiator with actual capacity for coordinating a switch emerging in an interval of length  $t'$  is at least  $1 - P_C$ . The probability of a swap between users whose actions do not correspond to a stable configuration is at least  $(1 - 2t^{-4})^2$ . The combined result: if the system is not in an SMC at time  $t$ , then a change in the potential will occur within no more than  $t'$  time slots with probability of at least  $(1 - P_C)(1 - 2t^{-4})^2$ , where  $P_C \triangleq \left(1 - \epsilon(1 - \epsilon)^{N-1}\right)^{\left\lfloor \frac{t'}{T_{\text{SF}}} \right\rfloor}$ .

Let us re-write the result for the sake of clarity: if the system is not in an SMC at time  $t$ , then a change in the potential will occur within no more than  $t'(\delta_1)$  time slots with probability of at least  $1 - \delta_1$ . Developing the previous expression for the probability of a change in potential:

$$\begin{aligned} (1 - P_C)(1 - 2t^{-4})^2 &= (1 - P_C)(1 - 4t^{-4} + 4t^{-8}) \\ &\geq (1 - P_C)(1 - 4t^{-4}) \\ &= 1 - P_C - 4t^{-4} + 4P_Ct^{-4} \\ &\geq 1 - P_C - 4t_{\min}^{-4}. \end{aligned}$$

From now on, we denote  $\delta_1 = P_C + 4t_{\min}^{-4}$ . Using this, we can derive an expression for  $t'(\delta_1)$ :

$$\begin{aligned} P_C &= \delta_1 - 4t_{\min}^{-4} \\ \left(1 - \epsilon(1 - \epsilon)^{N-1}\right)^{\left\lfloor \frac{t'}{T_{\text{SF}}} \right\rfloor} &= \delta_1 - 4t_{\min}^{-4} \\ \frac{t'}{T_{\text{SF}}} \ln \left(1 - \epsilon(1 - \epsilon)^{N-1}\right) &= \ln(\delta_1 - 4t_{\min}^{-4}) \\ t' &= T_{\text{SF}} \frac{\ln(\delta_1 - 4t_{\min}^{-4})}{\ln \left(1 - \epsilon(1 - \epsilon)^{N-1}\right)}. \end{aligned}$$