

The Mutual Benefits of Primary-Secondary User Cooperation in Wireless Cognitive Networks

Evangelia Matskani*, Nestor Chatzidiamantis*, Leonidas Georgiadis*, Iordanis Koutsopoulos† and Leandros Tassioulas‡

*Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Greece, Emails:{matskani, nestoras, leonid}@auth.gr

†Department of Computer Science, Athens University of Economics and Business, and Center of Research and Technology, Greece, Email: jordan@aub.gr

‡Department of Computer and Communication Engineering, University of Thessaly, Greece, E-mail:leandros@uth.gr

Abstract—In cognitive radio networks, secondary users (SUs) may cooperate with the primary user (PU) in order to obtain more transmission opportunities and thus maximize their throughput. The synergy consists in the following: the SU opts to cooperate by using its own transmit power to improve the probability of successful transmission of the PU. By increasing the probability of successful packet transmission for the PU, the SU essentially increases the service rate of the PU queue and thus, for given packet arrival rate, it increases the chances that it will be empty, and the channel will be free to use. Due to power limitations however, SUs have to take intelligent decisions on whether to cooperate or not and at which power level. Cooperation policies in this framework require the solution of a constrained Markov decision problem with infinite state space. In our work, we restrict attention to the class of stationary policies that take randomized decisions of an SU activation and its transmit power in every time slot based only on spectrum sensing. The proposed class of policies is shown to achieve the same set of SU rates as the more general policies, while significantly enlarging the stability region of the PU queue. Finally, a lightweight distributed protocol based on the proposed class of policies is presented, which is amenable to implementation in realistic scenarios.

I. INTRODUCTION

Cognitive radio networks (CRNs) have received considerable attention due to their potential for improving spectral efficiency [1]. The main idea behind CRNs is to allow unlicensed users, also known as *secondary users* (SUs), to identify spatially or temporally available spectrum, and transmit opportunistically, thus gaining access to the underutilized shared spectrum while maintaining limited interference to the licensed user, also known as *primary user* (PU).

Recently, the concept of cooperation between PU and SUs in CRNs emerged, as a means for providing benefits for both types of users. These benefits stem from the fact that, by exploiting the transmit power resources of SUs towards improving the effective transmission rate of the PU, the chances that the PU queue will be empty are increased, and

hence the PU channel is free to use more often. From an information theoretic perspective, cooperation between SUs and PUs at the physical layer has been investigated in many works (see [2] and references therein). Queuing theoretic aspects and spectrum leasing strategies for cooperative CRNs have been investigated in [3]–[6]. A protocol where a SU relays the PU packets that have not been correctly received by their destination, was suggested and investigated in terms of SU stable throughput in [4], while similar protocols were suggested and compared in [5], considering various physical layer relaying strategies.

In this work we study optimal cooperative PU-SUs transmission control algorithms with the objective to maximize a function of the transmission rates of the SUs, while guaranteeing unobstructed packet transmission for the PU, and stability of its queue. SUs have limited transmit power resources, therefore intelligent cooperation decisions must be taken. This is the main idea behind the work in [6], where a dynamic decision policy for the SUs' activities (i.e., whether to relay PU transmissions and at which power level) is suggested, aiming at maximizing SUs' throughput utility subject to SUs' average power constraints. The proposed policy is proved to be optimal, however, its basic requirement is that the PU packet arrival rates must be lower than a threshold value, which guarantees that the PU queue is stable even when SUs never cooperate. This regime places significant restrictions on the achievable PU stability region, since the sustainable arrival rates of PUs may be much larger than this threshold value.

In this paper we investigate transmission policies for cooperative CRNs that can be applied even when PU transmission rates are above the threshold set by [6], while still permitting the SUs to utilize the channel for their own transmissions. Since the SU decision options and success probabilities are different during the idle and busy PU periods, while the PU queue size is in turn affected by the cooperation decisions, such policies require in general the solution of a non-trivial constrained Markov decision problem with infinite state space, where the state is the size of the PU queue. Moreover, the implementation of such policies requires in general knowledge of the PU queue size [7].

The main contribution of this work is the introduction of a class of stationary policies which take random decisions on SU activities in every time-slot based only on the PU channel

The work of E. Matskani, N. Chatzidiamantis, L. Georgiadis and L. Tassioulas has been co-financed by the European Union (European Social Fund) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the NSRF-Research Funding Program: Thales. Investing in knowledge society through the European Social Fund.

I. Koutsopoulos acknowledges the support of ERC08-RECITAL project, co-financed by Greece and the European Union (European Social Fund) through the Operational Program "Education and Lifelong Learning"-NCRF 2007-2013.

spectrum sensing result, i.e., PU channel busy or idle. The benefits of our approach are as follows. First, our approach is proven to achieve the same set of SU rates as the more general policies in which (i) decision may depend on the PU queue size, or (ii) a SU packet may be transmitted instead of a PU packet, when the PU queue is non-empty. Hence, the policies in the restricted class are sufficient for optimality with respect to any utility function. Second, compared to [6], the proposed class of policies allows for a significantly larger range of PU traffic arrival rates for which the PU queue is stable, which, even more interestingly, still allows the SUs to utilize slots that are unused by the PU, in order to transmit their own traffic. In addition, we design a distributed algorithm for determining the appropriate parameters of the proposed policy when the objective is to maximize a concave SU utility function. This version offers a robust alternative to the centralized implementation and distributes the computational burden across the SUs without loss in performance.

II. SYSTEM MODEL

We consider a system with one PU and multiple SUs. The PU is the licensed owner of the channel and transmits whenever it has data to send. On the other hand, the SUs do not have any licensed spectrum and seek transmission opportunities on the PU channel. We assume that one of the SUs can cooperate with the PU by allocating some of its power resources in order to improve the success probability of PU transmissions. SU cooperation may be realized with various techniques that span one or more communication layers. For example, the SU may relay PU traffic (e.g. through decode-and-forward, or amplify-and-forward). Alternatively, this aid by the SU can be provided by means of link layer techniques, such as retransmission of the overheard PU packet by the SU. The model that will be described in the sequel is transparent to all these techniques, which are abstracted out in terms of the SU consumed transmit power resources.

Furthermore, after sensing the PU channel, SUs decide on which SU will cooperate and at which power level (if the PU channel is busy), or which SU will transmit and at which power level (if the PU channel is idle). In what follows we describe the parameters of the system model under consideration as well as the available controls.

A. System Model Parameters

We consider the time-slotted model, where time slot $t = 0, 1, \dots$ corresponds to time interval $[t, t + 1)$; t and $t + 1$ are called the “beginning” and “end” of slot t respectively. The PU queue receives new packets in each time slot t according to an i.i.d. arrival process $A_p(t)$ with mean rate λ_p packets/slot. We assume that the SUs are infinitely backlogged so that they have always packets to transmit.

We denote by \mathcal{S} the set of SUs. Each SU $s \in \mathcal{S}$ may use one of I_s power levels, $P_s(i)$, $i = 1, \dots, I_s$, where $P_s(i) < P_s(i + 1)$, to either transmit its own data or to assist the PU. To simplify the description that follows, we set $P_s(0) = 0$. At each time slot, only a single packet transmission can take

place. Furthermore, when transmission of packets from the PU takes place, at most one of the SUs can cooperate. There is a constraint on the long-term average power \hat{P}_s consumed by each SU $s \in \mathcal{S}$. Hence, for every $s \in \mathcal{S}$, if $i(t)$ is the power level used by s at slot t , it must hold,

$$\limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^t \mathbb{E}[P_s(i(\tau))] \leq \hat{P}_s, \quad i(\tau) \in \mathcal{I}_s^0, \quad (1)$$

where $\mathbb{E}[\cdot]$ denotes expectation, $\mathcal{I}_s = \{1, 2, \dots, I_s\}$ and $\mathcal{I}_s^0 = \mathcal{I}_s \cup \{0\}$.

We assume an erasure channel model, i.e., that each transmission (by the PU or one of the SUs) is either received correctly or erased.

- When SU s transmits one of its own packets with power level $i \in \mathcal{I}_s^0$, the probability of success is $r_s(i)$, where $r_s(0) = 0$, i.e. the success probability is zero if no power is used for transmission.
- When SU s cooperates with the PU using power level i , the success probability of the PU transmitted packet is $r_p(s, i)$. If $i = 0$, the SU “cooperates” with zero transmission power, hence in effect no cooperation takes place; therefore it is natural to assume that $r_p(s, 0) = r_p(0) \geq 0$ for all $s \in \mathcal{S}$, where $r_p(0)$ denotes the probability of successful packet transmission by the PU when the SUs do not cooperate. In addition, we assume that $r_p(s, i) \leq r_p(s, i + 1)$, i.e., the probability of successful reception is a non-decreasing function of transmission power.

B. Available Controls

In the beginning of time slot t there are various control options, depending of the status of the primary queue $Q_p(t)$. In case $Q_p(t) > 0$ (namely, the PU channel is busy), then the available controls are:

- A packet from the PU queue is transmitted, and transmission of SU packets is excluded. We refer to this constraint as *PU priority constraint*.
- A SU $s(t)$ is selected for cooperation with the PU in order to assist the transmission of the PU packet.
- A power level $i(t) \in \mathcal{I}_s^0$ is selected, so that $s(t)$ cooperates with the PU using power level $P_{s(t)}(i(t))$. When $i(t) = 0$ no cooperation takes place.

On the other hand, when $Q_p(t) = 0$ (namely, the PU channel is idle), the available controls are the following:

- A SU $s(t)$ is selected to transmit its own packet.
- A power level $i(t) \in \mathcal{I}_s^0$ is selected, so that $s(t)$ transmits its own packets using power level $P_{s(t)}(i(t))$. If $i(t) = 0$, no transmission takes place in slot t .

C. Admissible Policies, Rate Region, Performance Objective

A control policy is called *admissible* if the following *Policy Constraints* are satisfied:

- The policy obeys the PU priority constraint.
- The PU queue must be mean-rate stable, i.e., the output long-term average rate of the PU queue should be equal to its input rate [8].

- The power constraints of (1) are satisfied.

Under an admissible policy, each SU $s \in \mathcal{S}$ obtains a long-term average transmission rate $\bar{r}_s = \liminf_{t \rightarrow \infty} \frac{\sum_{\tau=0}^{t-1} \mathbb{E}[r_s(P_s(i(\tau)))]}{t}$, where $i(t)$ is the power level at which s transmits in slot t . In the sequel, we denote by $\bar{\mathbf{r}}$ the vector of the long-term average transmission rates of SUs, i.e., $\bar{\mathbf{r}} \triangleq \{\bar{r}_s\}_{s \in \mathcal{S}}$. The *achievable rate region* for the problem under consideration is defined as the set of vectors of SU rates $\bar{\mathbf{r}}$ that can be obtained by all admissible policies.

Optimization objective: The selection of an admissible policy depends on the particular optimization objective, which is expressed as a function of the vector of achievable long-term average SU transmission rates $\bar{\mathbf{r}}$. The optimization objective is the maximization of the utility function $f(\bar{\mathbf{r}})$, where $\bar{\mathbf{r}}$ belongs to the rate region. In the simplest case, $f(\cdot)$ is a linear function of $\bar{\mathbf{r}}$, however, fairness considerations may require $f(\cdot)$ to be a nonlinear (usually separable) function of $\bar{\mathbf{r}}$ [9].

The PU queue size $Q_p(t)$ can be seen as the state of a constrained Markov Decision Process problem [7], where the constraints are imposed by the policy constraints described above. Let \mathcal{C}_1 be the class of admissible policies of this Markov Decision Process. This class contains policies that are based on past history actions and includes the class of randomized stationary policies of the following form:

- When $Q_p(t) = m$, $m > 0$, select a SU s to cooperate with the PU at power level i with a certain probability that depends on m .
- When $Q_p(t) = 0$, select a SU s to transmit its own packets at power level i with a certain probability.

Consider a subclass of the policies in \mathcal{C}_1 , denoted by \mathcal{C}_0 , which consists of policies whose decisions are based solely on whether the PU queue is zero or not. In each time slot t , a policy in \mathcal{C}_0 acts as follows:

- When $Q_p(t) > 0$, select a SU s to cooperate at power level i with a probability $q(s, i|b)$.
- When $Q_p(t) = 0$, select a SU s to transmit its own data at power level i with probability $q(s, i|e)$.

Next, we consider the extended class of policies \mathcal{C}_2 which follow the policy constraints with the exception the PU priority constraint, i.e., the SUs are allowed to transmit their own traffic even when the PU queue is nonempty. In this case, the available controls at the beginning of each time slot are of the form (u, s, i) , $u \in \{1, 0\}$, $s \in \mathcal{S}$, $i \in \mathcal{I}_s^0$, where

- Control $(1, s, i)$, dictates transmission of PU traffic and assigns SU s at power level i to cooperate with the PU.
- Control $(0, s, i)$, dictates transmission of SU traffic, and selects SU s to transmit at power level i .

Since policies in \mathcal{C}_2 do not impose the PU priority constraint, and they may include even non-stationary policies, it follows that $\mathcal{C}_0 \subseteq \mathcal{C}_1 \subseteq \mathcal{C}_2$. Hence, the corresponding achievable rate regions \mathcal{R}_0 , \mathcal{R}_1 , \mathcal{R}_2 , satisfying the policy constraints under the classes of policies \mathcal{C}_0 , \mathcal{C}_1 , \mathcal{C}_2 , satisfy $\mathcal{R}_0 \subseteq \mathcal{R}_1 \subseteq \mathcal{R}_2$.

It might seem at first glance that a policy in class \mathcal{C}_0 with a restricted control space will lead to suboptimal performance.

However, this is not the case. In the next section we show that $\mathcal{R}_2 \subseteq \mathcal{R}_0$, thus reaching the interesting conclusion that $\mathcal{R}_0 = \mathcal{R}_1 = \mathcal{R}_2$. Hence, under any optimization objective, *it suffices to restrict attention to policies in \mathcal{C}_0 even if one has the freedom of not adhering to the PU priority constraint.*

III. CHARACTERIZATION OF ACHIEVABLE RATE REGIONS

In this section we substantiate our previous claim. Towards this end, we determine first the achievable rate region of policies in \mathcal{C}_0 , namely \mathcal{R}_0 , in subsection (III-A), as well as the achievable rate region of policies in \mathcal{C}_2 , namely \mathcal{R}_2 , in subsection (III-B). We then prove in subsection (III-B) that \mathcal{R}_0 coincides with \mathcal{R}_2 .

A. Achievable Rate Region of Policies in Class \mathcal{C}_0

For a given policy π in class \mathcal{C}_0 , the average packet service rate of the PU queue is given by $\bar{r}_p = \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} r_p(s, i) q(s, i|b)$. Standard results from queuing theory show that the stability region of the PU queue under π , that is, the closure of the set of PU arrival rates λ_p for which the PU queue is mean-rate stable [8], is the set of arrival rates that fall in the interval $[0, \bar{r}_p]$. Assume that $\lambda_p \in [0, \bar{r}_p)$ (so that the PU queue is stable) and let q_b be the steady state probability that the PU queue is busy under π . Viewing the transmitter at the PU as a queuing system holding 0 (if the PU queue is empty) or 1 packet (i.e., the packet whose transmission is attempted if the PU queue is non-empty) and applying Little's formula to this system, we have

$$q_b = \Pr \{\text{PU queue is non-empty}\} = \frac{\lambda_p}{\bar{r}_p}. \quad (2)$$

Hence, the steady state probability that the PU queue is empty is $q_e = 1 - q_b$. Due to the imposed PU priority constraint, SUs may transmit their own data only when the PU queue is empty. Hence, the average service rate of SU s traffic is

$$\bar{r}_s = \left(\sum_{i \in \mathcal{I}_s} r_s(i) q(s, i|e) \right) q_e. \quad (3)$$

The average power consumption of SU $s \in \mathcal{S}$ is

$$\bar{P}_s = q_e \sum_{i \in \mathcal{I}_s} P_s(i) q(s, i|e) + q_b \sum_{i \in \mathcal{I}_s} P_s(i) q(s, i|b), \quad (4)$$

and since $\pi \in \mathcal{C}_0$, it satisfies the power constraints (1), i.e., $\bar{P}_s \leq \hat{P}_s$, $s \in \mathcal{S}$. The discussion above shows that the constraints that need to be satisfied by the set of probabilities $\{q_b, q(s, i|b), q(s, i|e), q_e\}$, $s \in \mathcal{S}$, according to (1), (2), are given by (5)-(10) at the top of the next page.

Conversely, given the set of probabilities $\{q_b, q(s, i|b), q(s, i|e), q_e\}_{s \in \mathcal{S}, i \in \mathcal{I}_s^0}$ that satisfy the constraints (5)-(10), with $q_b < 1$, an admissible policy in \mathcal{C}_0 can be defined. Hence, the performance space of these policies is the set of $\bar{\mathbf{r}}$ defined by (3) where the set of probabilities $\{q_b, q(s, i|b), q(s, i|e), q_e\}_{s \in \mathcal{S}, i \in \mathcal{I}_s^0}$ satisfy the constraints (5)-(10).

$$q_b \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} r_p(s, i) q(s, i | b) = \lambda_p \quad (5)$$

$$q_e \sum_{i \in \mathcal{I}_s} P_s(i) q(s, i | e) + q_b \sum_{i \in \mathcal{I}_s} P_s(i) q(s, i | b) \leq \hat{P}_s, \quad s \in \mathcal{S} \quad (6)$$

$$q_b + q_e = 1 \quad (7)$$

$$\sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} q(s, i | b) = 1 \quad (8)$$

$$\sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} q(s, i | e) = 1 \quad (9)$$

$$q_b \geq 0, q_e \geq 0, q(s, i | b) \geq 0, q(s, i | e) \geq 0, \quad s \in \mathcal{S}, i \in \mathcal{I}_s^0 \quad (10)$$

While the constraints of (5)-(10) are nonlinear with respect to parameters $\{q_b, q(s, i | b), q(s, i | e), q_e\}$, they can be easily transformed into linear ones through the transformation $q(b, s, i) = q_b q(s, i | b)$, $q(e, s, i) = q_e q(s, i | e)$. Note that $q(b, s, i)$ is the probability that the PU is busy *and* SU s is selected for cooperation at power level i , while $q(e, s, i)$ is the probability that the PU is idle *and* SU s packets are transmitted in a slot at power level i . The new variables $q(b, s, i)$, $q(e, s, i)$ satisfy the following constraints.

$$\sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} r_p(s, i) q(b, s, i) = \lambda_p \quad (11)$$

$$\sum_{i \in \mathcal{I}_s} P_s(i) q(e, s, i) + \sum_{i \in \mathcal{I}_s} P_s(i) q(b, s, i) \leq \hat{P}_s, \quad s \in \mathcal{S} \quad (12)$$

$$\sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} q(e, s, i) + \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} q(b, s, i) = 1 \quad (13)$$

$$q(e, s, i) \geq 0, q(b, s, i) \geq 0, \quad s \in \mathcal{S}, i \in \mathcal{I}_s^0. \quad (14)$$

In addition, the achievable rate of each SU $s \in \mathcal{S}$, given by (3), can be rewritten as

$$\bar{r}_s = \sum_{i \in \mathcal{I}_s} r_s(i) q(e, s, i). \quad (15)$$

Using in effect the inverse transformation it can be easily seen that any vector $\bar{\mathbf{r}}$ satisfying (11)-(15) also satisfies (3) and (5)-(10). Hence, the achievable rate region of policies in class \mathcal{C}_0 is characterized by (11)-(15). Next we show that this region coincides with the achievable rate region of policies in \mathcal{C}_2 .

B. Achievable Rate Region of Policies in Class \mathcal{C}_2

Contrary to the available controls when the PU priority constraint is imposed, the set of available controls for policies in \mathcal{C}_2 do not obey the PU priority constraint (thus, a slot may be allocated to PU packet transmission, even if the PU queue is empty). Hence, this class of policies falls in the framework of policies studied in [8] and its achievable rate region can be characterized again by the achievable rate region of stationary policies. In the latter framework, a stationary policy selects

at the beginning of each time slot the control (u, s, i) with probability $p(u, s, i)$. Under such a policy, the probability of successful transmission of SU s packets is

$$\bar{r}_s = \sum_{i \in \mathcal{I}_s} r_s(i) p(0, s, i), \quad (16)$$

while, the probability of successful transmission of PU packets is

$$\bar{r}_p = \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} r_p(s, i) p(1, s, i), \quad (17)$$

and stability of the PU queue requires that

$$\bar{r}_p \geq \lambda_p. \quad (18)$$

Also, the average power constraint requirement implies that

$$\sum_{i \in \mathcal{I}_s} P_s(i) p(0, s, i) + \sum_{i \in \mathcal{I}_s} P_s(i) p(1, s, i) \leq \hat{P}_s, \quad s \in \mathcal{S}. \quad (19)$$

Finally, since $p(u, s, i)$ are probabilities, we must have

$$\sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} p(0, s, i) + \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} p(1, s, i) = 1 \quad (20)$$

$$p(0, s, i) \geq 0, p(1, s, i) \geq 0, \quad s \in \mathcal{S}, i \in \mathcal{I}_s^0. \quad (21)$$

Constraints (18)-(21) together with (16) define the achievable rate region \mathcal{R}_2 of policies in \mathcal{C}_2 . The similarity of these constraints compared to (11)-(15) should be noted. From a math perspective, the only difference is that there is equality in (11), as opposed to inequality in (18). However, there is difference in the interpretation of these probabilities. Specifically,

- $q(b, s, i)$ is the probability that PU channel is busy *and* SU s is selected for cooperation at power level i , while $p(1, s, i)$ is the probability that SU s is selected for cooperation at power level i and dictating PU transmission as well.
- $q(e, s, i)$ is the probability that PU is idle *and* secondary user s packets are transmitted in a slot at power level i , while $p(0, s, i)$ is the probability of selecting secondary user s packet for transmission at power level i , while PU does not transmit.

It becomes clear from the discussion above that $\mathcal{R}_0 \subseteq \mathcal{R}_2$. The next theorem shows that $\mathcal{R}_2 = \mathcal{R}_0$.

Theorem 1. *It holds $\mathcal{R}_2 \subseteq \mathcal{R}_0$, hence $\mathcal{R}_0 = \mathcal{R}_1 = \mathcal{R}_2$.*

Proof: Please refer to the Appendix. ■

IV. DISTRIBUTED IMPLEMENTATION

In this section, we focus on approaches that do not rely on central coordination and are based on policies in class \mathcal{C}_0 that achieve the following objective, OPT1:

$$\begin{aligned} & \text{maximize} && \sum_{s \in \mathcal{S}} f_s(\bar{r}_s) && (22) \\ & \text{subject to} && (11), (12), (13), (14), (15) \end{aligned}$$

Functions $\{f_s(\cdot)\}_{s \in \mathcal{S}}$ are usually selected so that certain fairness criteria for SU rate allocation are satisfied, and they are assumed to be concave with respect to \bar{r}_s . Thus, since $\bar{r}_s, s \in \mathcal{S}$, is a linear function of variables $\left\{ \{q(e, s, i)\}_{i \in \mathcal{I}_s^0}, \{q(b, s, i)\}_{i \in \mathcal{I}_s^0} \right\}$, $f_s(\bar{r}_s)$ is also a concave function of these variables. Hence, OPT1 is a convex optimization problem and can be solved efficiently via interior point methods.

In an operational environment where parameters may change with time, problem OPT1 will have to be solved whenever significant changes to such parameters occur. A centralized solution requires a single node to be responsible for gathering instantaneous parameter values, for the solution of OPT1 and for determining the appropriate scheduling of packet transmissions. While such a solution may be acceptable in certain environments, it creates a “single point of failure”. Moreover the central node must be continually informing the SUs as to which one will cooperate or transmit in each time slot and at which power level. There may also be a scalability issue with this approach since the number of variables is of the order $2|\mathcal{S}|I$, where I is the maximum number of power levels of SU nodes ($\sum_{i \in \mathcal{S}} |\mathcal{I}_s^0|$ parameters $\{q(b, s, i)\}_{s \in \mathcal{S}, i \in \mathcal{I}_s^0}$ plus $\sum_{i \in \mathcal{S}} |\mathcal{I}_s^0|$ parameters $\{q(e, s, i)\}_{s \in \mathcal{S}, i \in \mathcal{I}_s^0}$). Hence, depending on the computing power and memory availability at the central node, solving problem OPT1 in a centralized location may become prohibitive for larger number of SUs.

In this section, we derive a solution to OPT1 in a distributed fashion. The main features of this approach are the following.

a) The PU involvement in the algorithm is only to announce its arrival rate λ_p at the beginning of the algorithm - no further participation is required.

b) A SU node does not need to know the parameters (i.e., $r_s(i), r_p(s, i), i \in \mathcal{I}_s$) of other SU nodes.

c) The distributed solution requires each SU node $s \in \mathcal{S}$ to solve optimization problems with I_s variables, hence the computational complexity per node does not increase with the number of SU nodes.

d) Two messages are broadcasted by each SU node per iteration. The number of iterations for convergence depends on the number of SU nodes, but this is tolerable for the algorithm execution in a real setting, as long as the system parameters do not change very rapidly.

e) Once convergence of the algorithm is reached, the SUs need only observe the state of the PU channel (busy or idle); they can decide autonomously which SU node is scheduled to either cooperate with the PU or to transmit its own traffic, without the need of a scheduler, or the exchange of control messages.

We assume that there is a separate low-rate channel which is used by the SUs for control message exchanges [10]. In particular we assume that control messages may be broadcasted among the SUs, either because the low-rate channel is broadcast in nature, or through the establishment of Broadcast Trees that usually are employed in ad-hoc networks [11].

A distributed solution to problem OPT1 may be obtained through dual decomposition method, which decomposes the global problem into $|\mathcal{S}|$ parallel subproblems, each involving only local variables and parameters of node s . However, this method can be very slow in terms of convergence [12]. Indeed, the dual decomposition based algorithm that we initially applied failed to converge within a tolerable number of iterations. Among all alternatives we tried, the best algorithm in terms of convergence was the one built upon the Alternating Direction Method of Multipliers (ADMoM). ADMoM has superior convergence properties over the dual ascent method [12]–[14]. To apply ADMoM to OPT1, we first turned the average power inequality constraints (12) into equalities, by introducing auxiliary variables $\{y_s\}_{s \in \mathcal{S}}$, where y_s is associated with the respective s^{th} constraint, and is positive-valued. Also, for notational simplicity, we equivalently rewrite problem OPT1 as OPT2 given by

$$\text{minimize} \quad -\sum_{s \in \mathcal{S}} f_s(\mathbf{x}_s) \quad (23)$$

$$\text{u.c.} \quad \sum_{s \in \mathcal{S}} g_{1s}(\mathbf{z}_s) = \lambda_p \quad (24)$$

$$h_s(\mathbf{x}_s, \mathbf{z}_s, y_s) = \hat{P}_s, s \in \mathcal{S} \quad (25)$$

$$\sum_{s \in \mathcal{S}} g_{2s}(\mathbf{x}_s) + \sum_{s \in \mathcal{S}} g_{2s}(\mathbf{z}_s) = 1 \quad (26)$$

$$\mathbf{x}_s \geq 0, \mathbf{z}_s \geq 0, y_s \geq 0, s \in \mathcal{S} \quad (27)$$

where $\mathbf{x}_s \triangleq \{q(e, s, i)\}_{i \in \mathcal{I}_s^0}$, $\mathbf{z}_s \triangleq \{q(b, s, i)\}_{i \in \mathcal{I}_s^0}$, and $h_s(\mathbf{x}_s, \mathbf{z}_s, y_s) \triangleq \sum_{i \in \mathcal{I}_s} P_s(i)q(e, s, i) + \sum_{i \in \mathcal{I}_s} P_s(i)q(b, s, i) + y_s, s \in \mathcal{S}$.

Let ν and ξ denote the dual variables associated with the constraints of (24) and (26) respectively, and μ_s the dual variable associated with the s^{th} constraint of (25). Then, the augmented Lagrange function corresponding to OPT2 used by ADMoM, parametrized by the penalty parameter $\rho > 0$, is given by (28) at the top of the next page [12], [13]. The augmented Lagrangian is the standard Lagrange function of OPT2, given by $\sum_{s \in \mathcal{S}} L_s - \nu \lambda_p - \xi$, plus the penalty quadratic term multiplied by $\frac{\rho}{2}$. The penalty parameter $\rho > 0$ is the step size used for the dual variables updates and plays a key role for the convergence of ADMoM [12]–[14].

Computational complexity: The optimization steps and variables updates that need to be carried out at each SU node $s \in \mathcal{S}$, according to ADMoM, are given by (29)–(34), at the next page, where k denotes the iteration index. Note that

$$L_p = \sum_{s \in \mathcal{S}} L_s - \nu \lambda_p - \xi + \frac{\rho}{2} \left\{ \left(\sum_{s \in \mathcal{S}} g_{1s}(z_s) - \lambda_p \right)^2 + \sum_{s \in \mathcal{S}} \left(h_s(\mathbf{x}_s, z_s, y_s) - \hat{P}_s \right)^2 + \left(\sum_{s \in \mathcal{S}} g_{2s}(\mathbf{x}_s) + \sum_{s \in \mathcal{S}} g_{2s}(z_s) - 1 \right)^2 \right\},$$

$$L_s \triangleq -f_s(\mathbf{x}_s) + \nu g_{1s}(z_s) + \mu_s \left(h_s(\mathbf{x}_s, z_s, y_s) - \hat{P}_s \right) + \xi g_{2s}(\mathbf{x}_s) + \xi g_{2s}(z_s), \quad s \in \mathcal{S}. \quad (28)$$

$$\mathbf{x}_s^{k+1} = \arg \min_{\mathbf{x}_s \geq 0} L_s(\mathbf{x}_s, z_s^k, y_s^k, v^k, \xi^k, \mu_s^k) + \frac{\rho}{2} \left(h_s(\mathbf{x}_s, z_s^k, y_s^k) - \hat{P}_s \right)^2 \quad (29)$$

$$+ \frac{\rho}{2} \left(\sum_{m=1}^{s-1} g_{2m}(\mathbf{x}_m^{k+1}) + \sum_{m=s+1}^{|\mathcal{S}|} g_{2m}(\mathbf{x}_m^k) + g_{2s}(\mathbf{x}_s) + \sum_{s \in \mathcal{S}} g_{2s}(z_s^k) - 1 \right)^2$$

$$z_s^{k+1} = \arg \min_{z_s \geq 0} L_s(\mathbf{x}_s^{k+1}, z_s, y_s^k, v^k, \xi^k, \mu_s^k) + \frac{\rho}{2} \left(h_s(\mathbf{x}_s^{k+1}, z_s, y_s^k) - \hat{P}_s \right)^2 \quad (30)$$

$$+ \frac{\rho}{2} \left(\sum_{m=1}^{s-1} g_{1m}(z_m^{k+1}) + \sum_{m=s+1}^{|\mathcal{S}|} g_{1m}(z_m^k) + g_{1s}(z_s) - \lambda_p \right)^2$$

$$+ \frac{\rho}{2} \left(\sum_{s \in \mathcal{S}} g_{2s}(\mathbf{x}_s^{k+1}) + \sum_{m=1}^{s-1} g_{2m}(z_m^{k+1}) + \sum_{m=s+1}^{|\mathcal{S}|} g_{2m}(z_m^k) + g_{2s}(z_s) - 1 \right)^2$$

$$y_s^{k+1} = \arg \min_{y_s \geq 0} L_s(\mathbf{x}_s^{k+1}, z_s^{k+1}, y_s, v^k, \xi^k, \mu_s^k) + \frac{\rho}{2} \left(h_s(\mathbf{x}_s^{k+1}, z_s^{k+1}, y_s) - \hat{P}_s \right)^2 \quad (31)$$

$$\xi^{k+1} = \xi^k + \rho \left(\sum_{s \in \mathcal{S}} g_{2s}(\mathbf{x}_s^{k+1}) + \sum_{s \in \mathcal{S}} g_{2s}(z_s^{k+1}) - 1 \right) \quad (32)$$

$$v^{k+1} = v^k + \rho \left(\sum_{s \in \mathcal{S}} g_{1s}(z_s^{k+1}) - \lambda_p \right) \quad (33)$$

$$\mu_s^{k+1} = \mu_s^k + \rho \left(h_s(\mathbf{x}_s^{k+1}, z_s^{k+1}, y_s^{k+1}) - \hat{P}_s \right) \quad (34)$$

the computational burden is distributed across SU nodes; the computational complexity at each node depends primarily on the two quadratic optimization problems in (29) and (30), each of which has I_s variables, and can be efficiently solved via interior point methods, or standard methods such as Newton Method. All the following steps involve a single variable and are straightforward.

Communication overhead: Each node s , in order to perform the steps in (29) and (30), needs to know information concerning the updated local variables of other nodes. This can be accomplished through message broadcasts by each SU node via the control channel in the following manner. The nodes update their local variables and broadcast the messages required sequentially, in a prespecified order. Specifically, for the step in (29), each node $s \in \mathcal{S}$ updates its primal variable \mathbf{x}_s^{k+1} and broadcasts message $g_{2s}(\mathbf{x}_s^{k+1})$. Similarly, for the step in (30), each SU node updates its variable z_s^{k+1} and broadcasts $g_{1s}(z_s^{k+1})$ and $g_{2s}(z_s^{k+1})$ in one message, according to the prespecified order. Steps dictated by (31)-(34), for each node s , require only its local variables and information that is already acquired by s from the previous message broadcasts

and thus can be implemented in parallel by all nodes. Each iteration of the distributed algorithm consists of one round of these update steps by all $|\mathcal{S}|$ nodes. Consequently, the communication overhead of the algorithm is $2|\mathcal{S}|$ message broadcasts per iteration.

Convergence: For the convergence of the algorithm in decentralized manner, each SU keeps track of a local metric and determines local convergence with respect to it, within a prespecified accuracy. This local metric for each node $s \in \mathcal{S}$ may be the successive differences of its local objective function under optimization, i.e., $f_s(\mathbf{x}_s)$. Once this local metric drops under the prespecified accuracy, local convergence is declared, and node s announces it via the control channel. As soon as all SU nodes reach convergence, the algorithm terminates.

Real-time implementation: We assume that the PU broadcasts its arrival rate λ_p at the beginning of the algorithm. Once convergence of the algorithm for a given λ_p is reached, all SUs have knowledge of the sums of probabilities $g_{2s}(\mathbf{x}_s^{opt})$, $g_{2s}(z_s^{opt})$, $\forall s \in \mathcal{S}$. Thus, if the SUs use the same randomization algorithm and common seed, as long as they

observe the state of the PU channel, they can all independently produce the same result as to who SU is scheduled to cooperate with the PU or transmit its own data in every time slot. Then, the scheduled SU determines its power level for its transmission based on its own probability parameters. The system evolves without the need for further coordination among network nodes.

The algorithm runs again only when some of the parameters of the operational environment change significantly. Thus, when the arrival rate changes within a prespecified percentage of its previous value, the PU informs the SUs about the new value of λ_p . Also, in case wireless channel gains change for some SU within a certain percentage, the corresponding SU may announce the rerun of the algorithm. In such cases the algorithm can adapt to changes in the operational environment; the problem is not solved from scratch, but the algorithm is initialized at the optimal point of the previous system state. This speeds up its convergence and reduces the overall communication overhead, as reported in the simulation results that follow.

V. SIMULATION AND NUMERICAL RESULTS

In this section we present simulation results that validate the theoretical analysis. We consider as objective optimization function $f(\cdot)$ the sum of transmission rates of the SUs, i.e., $f(\bar{\mathbf{r}}) = \sum_{s \in \mathcal{S}} \bar{r}_s$. We first investigate the performance of an optimal policy in class \mathcal{C}_0 in comparison with an optimal dynamic policy from the class \mathcal{C}_2 , constructed through the Lyapunov optimization techniques [8], in terms of SU throughput and average PU queue size. We also present the performance of the transmission algorithm presented in [6]. Finally, we examine the convergence of the distributed algorithm, as well as its ability to adapt to changing parameters.

Several simulation experiments on various setups have been conducted in order to evaluate the performance of the proposed class of policies. Due to space limitations, we provide here some indicative simulation results on a particular experimental setup. We consider 5 SUs and a set of 5 available transmit power levels, common for all SUs, $\mathcal{I}_s^0 = \{0, 1, 2, 3, 4\}$, where $\mathbf{P}_s = (0, 0.25, 0.5, 0.75, 1)$, for all $s \in \mathcal{S}$. We also assume $\mathbf{r}_p(s) = (0.4, 0.5, 0.6, 0.7, 0.8)$ and $\mathbf{r}_s = (0, 0.3, 0.5, 0.8, 1)$, for all $s \in \mathcal{S}$. Finally, the average power constraint is $\bar{P}_s = 0.15$, for all SUs.

For such a scenario, the performance of the system is depicted by Figs. 1-2 in terms of $f(\bar{\mathbf{r}})$ and average backlog of PU queue, respectively. It can be seen in Fig. 1 that, as expected, the sum rate achieved by SUs that employ an optimal policy from the restricted class of policies \mathcal{C}_0 is identical to the sum rate achieved under the optimal policy in \mathcal{C}_2 . As can be observed by Fig. 2, the average backlog of the PU queue remains very low under the optimal policy in \mathcal{C}_0 . On the contrary, the dynamic policy from \mathcal{C}_2 induces large sizes to PU queue even for small arrival rates. Furthermore, when compared to the control algorithm presented in [6], the class \mathcal{C}_0 of policies extends the range of λ_p that can be supported by the system, providing mutual benefits to both PU and SUs out

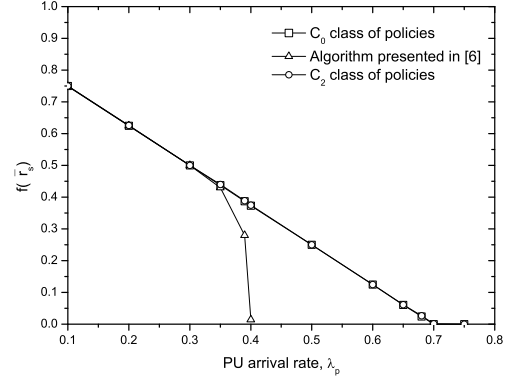


Fig. 1. The sum of the SUs rates.

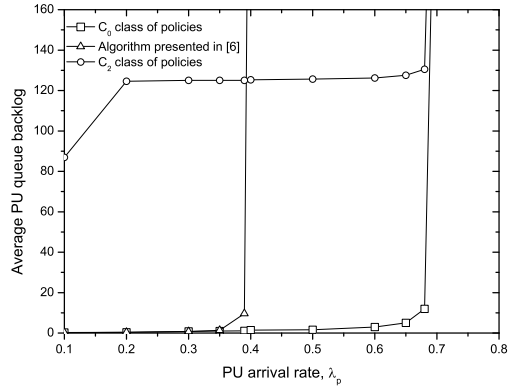


Fig. 2. The average backlog of the PU queue.

of their cooperation. In particular, transmission rates higher than the PU queue service rate without SU cooperation can be supported for the PU through the class of policies \mathcal{C}_0 , while transmission opportunities are provided to SUs to transmit their own data. It should be noted that the policy in [6] was shown to be optimal for $\lambda_p < 0.4$, and this is confirmed in Fig. 1, where it is seen that all three policies achieve the same sum-rate for $\lambda_p < 0.4$. However, as can be seen by this experiment, the policy in [6] renders the PU queue unstable for $\lambda_p > 0.4$ and reduces the SU sum rates to zero. The reason is the following. In [6], decisions are taken at the end of busy periods of the PU queue. If $\lambda_p > 0.4$, whenever a decision of not cooperating is taken, there is a nonzero probability that the primary queue never becomes empty, and hence there is no possibility for the SUs to take corrective actions.

We also conducted experiments regarding convergence of the distributed algorithm on the same setup. Due to lack of space, we report a summary of the results of these simulations. We used as initial values $\{q(e, s, i)^0\}_{i \in \mathcal{I}_s^0} = 0.01, \forall s \in \mathcal{S}$, $\{q(b, s, i)^0\}_{i \in \mathcal{I}_s^0} = 0.03, \forall s \in \mathcal{S}$, $\{\mu_s^0\}_{s \in \mathcal{S}} = 1$, $\xi^0 = 1$, $\nu^0 = 1$, as well as the stepsize parameter $\rho = 0.1$ and accuracy for convergence equal to $\epsilon = 10^{-5}$. Varying the PU rate from 0.2 to 0.7, the number of iterations required for convergence was found to vary from 263 to 74. Furthermore,

the distributed algorithm reached the same objective value as its centralized counterpart in all these experiments. We also conducted an experiment where the PU rate changed from 0.5 to some new value in the range 0.35 to 0.7, and as initial values we used the optimal values obtained for $\lambda_p^0 = 0.5$. The number of iterations required for convergence in this case was considerably reduced, and ranged from 44 to 16.

VI. CONCLUSIONS

In this work we put forward novel PU-SU cooperation policies for cognitive radio networks that orchestrate a PU and co-existing SUs in a wireless channel. The major contribution to the state of the art is that, although the proposed policies require only the sensing of the state of PU channel (busy or empty) for their realization, they: a) achieve substantial augmentation of stability region of the PU queue, and b) can obtain any long-term SU rates achievable by policies for which the restriction of always giving priority to PU traffic is removed. A distributed version of the algorithm is also presented. There exist several directions for future study. First, there is the issue of how to design the system when SUs are not backlogged and packets arrive randomly to them instead. Second, in this work, we assumed that channel sensing is error free; imperfect sensing introduces several new features and alters the structure of the problem. Finally, the issue of designing a dynamic online version of this algorithm is open.

APPENDIX

Proof: Let $\bar{\mathbf{r}} \in \mathcal{R}_2$. We will show that $\bar{\mathbf{r}} \in \mathcal{R}_0$, which proves *Theorem 1*.

If $\lambda_p = \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} r_p(s, i) p(1, s, i)$, then clearly $\bar{\mathbf{r}} \in \mathcal{R}_0$. Assume next that $\lambda_p < \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} r_p(s, i) p(1, s, i)$. We distinguish the following cases:

Case 1. $\lambda_p \geq r_p(0) p(1)$, where $p(1) \triangleq \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} p(1, s, i)$ denotes the total probability that PU transmits, summed over all SU s and transmit power levels. Note that since

$$r_p(0) p(1) \leq \lambda_p < \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s^0} r_p(s, i) p(1, s, i), \quad (35)$$

for each λ_p in the interval above, there exists a parameter α , with $0 \leq \alpha < 1$, such that it holds

$$\lambda_p = \alpha \left(\sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}_s} r_p(s, i) p(1, s, i) \right) + (1 - \alpha) r_p(0) p(1). \quad (36)$$

We define now the new set of parameters $q(b, s, i)$ and $q(e, s, i)$ by setting $q(e, s, i) = p(0, s, i)$ for all $s \in \mathcal{S}$ and $i \in \mathcal{I}_s^0$ and

$$q(b, s, i) = \begin{cases} \alpha p(1, s, i) & \text{if } i \in \mathcal{I}_s \\ \alpha p(1, s, 0) + (1 - \alpha) p(1, s) & \text{if } i = 0, \end{cases} \quad (37)$$

for all $s \in \mathcal{S}$, where $p(1, s) \triangleq \sum_{j \in \mathcal{I}_s^0} p(1, s, j)$. Since

$0 \leq \alpha < 1$, parameters $q(e, s, i)$ and $q(b, s, i)$, for all $s \in \mathcal{S}$ and $i \in \mathcal{I}_s^0$, are non-negative. Furthermore, note that $\sum_{i \in \mathcal{I}_s^0} q(b, s, i) = \sum_{i \in \mathcal{I}_s^0} p(1, s, i)$. Hence the new set of

parameters satisfies (13). Also, since $P_s(0) = 0$, after some algebraic manipulations, it can be seen that the new set of parameters satisfy (19). Finally, due to (36), it follows that (11) is satisfied. Hence the new set of parameters satisfy (11)-(14). Also since the SU rates computed according to (15) (where $q(e, s, i) = p(0, s, i)$ for all $s \in \mathcal{S}$ and $i \in \mathcal{I}_s^0$) are the same as those given by (16), it follows that $\bar{\mathbf{r}} \in \mathcal{R}_0$.

Case 2. $\lambda_p < r_p(0) p(1)$.

Define the new set of parameters as follows

$$q(b, s, i) = \begin{cases} 0 & \text{if } i \in \mathcal{I}_s \\ \frac{\lambda_p}{r_p(0) p(1)} p(1, s) & \text{if } i = 0, \end{cases} \quad (38)$$

$$q(e, s, i) = \begin{cases} p(0, s, i) & \text{if } i \in \mathcal{I}_s \\ \beta \sum_{i \in \mathcal{I}_s^0} p(0, s, i) + p(0, s, 0) & \text{if } i = 0, \end{cases} \quad (39)$$

for all $s \in \mathcal{S}$, where $\beta = \frac{1 - \frac{\lambda_p}{r_p(0)}}{1 - p(1)} - 1$. Since $\lambda_p < r_p(0) p(1)$, and $p(1) \leq 1$, it follows that $\beta > 0$, hence, all the defined parameters are non-negative. Also, due to (20), (13) is satisfied. Next, it can be easily shown that (11) is satisfied. Furthermore, due to (19), (12) is also satisfied. Finally, since $P_s(0) = 0$, it follows that the SU rates computed according to (15) (where $q(e, s, i)$ are selected according to (39) for all $s \in \mathcal{S}$ and $i \in \mathcal{I}_s^0$) are the same as those given by (16). Hence we conclude that $\bar{\mathbf{r}} \in \mathcal{R}_0$. ■

REFERENCES

- [1] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.
- [2] A. Goldsmith, S. A. Jafar, I. Maric, and S. Srinivasa, "Breaking spectrum gridlock with cognitive radios: An information theoretic perspective," in *Proc. IEEE*, vol. 97, Jan 2009, pp. 894–914.
- [3] O. Simeone, I. Stanojev, S. Savazzi, Y. Bar-Ness, U. Spagnolini, and R. Pickholtz, "Spectrum leasing to cooperating secondary ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 26, pp. 203–213, Jan 2008.
- [4] O. Simeone, Y. Bar-Ness, and U. Spagnolini, "Stable throughput of cognitive radios with and without relaying capability," *IEEE Trans. Commun.*, vol. 55, pp. 2351–2360, Jan 2007.
- [5] I. Krikidis, J. Laneman, J. Thompson, and S. McLaughlin, "Protocol design and throughput analysis for multi-user cognitive cooperative systems," *IEEE Trans. Wireless Commun.*, vol. 8, pp. 4740–4751, Jan 2009.
- [6] R. Urgaonkar and M. Neely, "Opportunistic cooperation in cognitive femtocell networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 607–616, April 2012.
- [7] E. Altman, *Constrained Markov Decision Processes*. Chapman & Hall/CRC, 1999.
- [8] M. J. Neely, *Stochastic Network Optimization with Application to Communication & Queueing Systems*. Morgan & Claypool, Aug. 2010.
- [9] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 556–567, 2000.
- [10] B. F. Lo, "A survey of common control channel design in cognitive radio networks," *Ph. Commun.*, vol. 4, no. 1, pp. 26–39, 2011.
- [11] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-efficient broadcast and multicast trees in wireless networks," *Mob. Networks and Appl.*, vol. 7, pp. 481–492, 2002.
- [12] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. 2nd ed. Belmont, MA: Athena Scientific, 1996.
- [13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [14] D. Bertsekas and N. J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. 2nd ed. Belmont, MA: Athena Scientific, 1999.