

Adaptive CSMA under the SINR Model: Efficient Approximation Algorithms for Throughput and Utility Maximization

Peruru Subrahmanya Swamy, Radha Krishna Ganti, *Member, IEEE*, Krishna Jagannathan, *Member, IEEE*

Abstract—We consider a Carrier Sense Multiple Access (CSMA) based scheduling algorithm for a single-hop wireless network under a realistic Signal-to-interference-plus-noise ratio (SINR) model for the interference. We propose two local optimization based approximation algorithms to efficiently estimate certain attempt rate parameters of CSMA called fugacities. It is known that adaptive CSMA can achieve throughput optimality by sampling feasible schedules from a Gibbs distribution, with appropriate fugacities. Unfortunately, obtaining these optimal fugacities is an NP-hard problem. Further, the existing adaptive CSMA algorithms use a stochastic gradient descent based method, which usually entails an impractically slow (exponential in the size of the network) convergence to the optimal fugacities. To address this issue, we first propose an algorithm to estimate the fugacities, that can support a given set of desired service rates. The convergence rate and the complexity of this algorithm are independent of the network size, and depend only on the neighborhood size of a link. Further, we show that the proposed algorithm corresponds exactly to performing the well-known Bethe approximation to the underlying Gibbs distribution. Then, we propose another local algorithm to estimate the optimal fugacities under a utility maximization framework, and characterize its accuracy. Numerical results indicate that the proposed methods have a good degree of accuracy, and achieve extremely fast convergence to near-optimal fugacities, and often outperform the convergence rate of the stochastic gradient descent by a few orders of magnitude.

Index Terms—CSMA, Gibbs distribution, Bethe approximation, Distributed algorithm, Wireless ad hoc network

I. INTRODUCTION

The problem of link scheduling for maximum throughput has been widely studied, particularly with emphasis on the maximum-weight scheduling framework, developed in [2], [3]. In spite of its throughput maximizing property, maximum-weight scheduling requires centralized control, and necessitates the solution of an NP-hard problem for each scheduling decision. Several works have attempted to modify the maximum-weight algorithm, so as to make it more amenable to simple, distributed implementation [4]–[6]. However, these greedy algorithms do not achieve full throughput.

In a series of recent papers [7]–[9], a class of distributed link scheduling algorithms called adaptive CSMA algorithms have been proposed and proven to be throughput optimal,

i.e., they can support any achievable service rate vector. The central idea behind these algorithms lies in using a reversible Markov chain to sample feasible schedules from a product form distribution called the Gibbs distribution [10, Chapter 7]. Specifically, each link adaptively adjusts its transmission attempt rate (also known as its fugacity) in order to ensure sufficient average service rate.

In order to support a given feasible service rate vector using CSMA, the corresponding fugacities have to be computed. Unfortunately, determining the appropriate fugacities corresponding to the desired service rates is an NP-hard problem [7]. In [7], the optimal fugacities are computed as a solution to an optimization problem (here after referred to as the Gibbsian problem), using a stochastic gradient descent algorithm. Each iteration of the gradient descent requires estimating the average service rates under the current iterate of the fugacities, which in turn entails waiting for the underlying Markov chain to reach steady-state. This ‘mixing time’ of the underlying Markov chain could be very large (exponential in the size of the network), depending on the network load and topology [11], [12]. Therefore, the existing adaptive CSMA algorithms [7] do not provide a practical way to estimate the fugacities, although they effectively support the desired service rates once the optimal fugacities are estimated [13]. The main focus of this paper is in proposing efficient methods to estimate the fugacities, under a realistic SINR (signal-to-interference-plus-noise ratio) model for the interference. Specifically, we consider the following two scenarios under which fugacities are to be computed:

- The service rate requirements of the links are known. The objective is to support these average service rates.
- The service rate requirements are not known, but each link has a utility function of its average service rate. The objective is to maximize the sum utility of the network.

A simple conflict graph based interference model [7]–[9] is widely used in the wireless context due to its simplicity and tractability, although it does not adequately capture the complex nature of the wireless interference [14]. More specifically, a conflict graph based interference model ignores the fact that whether or not two links can transmit simultaneously, depends on the transmission state of the other links and their spatial locations. Some recent papers extend the adaptive CSMA framework to a more realistic interference models like the SINR model [15], [16], and Rayleigh faded channels [17]. However, these papers also essentially employ stochastic gra-

A part of this work [1] has been presented at IEEE Annual Allerton Conference on Communication, Control, and Computing (Allerton) 2015, held at Monticello, IL, USA

P. S. Swamy, R. K. Ganti and K. Jagannathan are with the Department of Electrical Engineering, IIT Madras, Chennai, India 600036. Email: {p.swamy, rganti, krishnaj}@ee.iitm.ac.in

dient descent on a Gibbsian function to estimate the fugacities, and hence suffer from impractically slow convergence rates.

For the conflict graph based interference model assumed in [7], approximate but efficient methods to compute the fugacities have been proposed [18], [19] using a popular variational technique called the Bethe approximation [20]. However, the solutions given in [18], [19], cannot be directly extended to SINR based interference model. This is because the conflict graph based interference model corresponds to a simple pair-wise interaction model [20], while the SINR model involves higher order interactions. The presence of these higher order interactions makes the extension non-trivial. Even for graphical models with higher order interactions, there are well known algorithms like Belief propagation (BP) [20] for solving the Bethe approximation problem. However, in the context of adaptive CSMA under the SINR model, they can directly be used only to estimate the service rates given the fugacities, but not the other way around.

We start with the Gibbsian optimization problem corresponding to the optimal fugacities, and approximate this global problem by decoupling it into local optimization problems at each link. The local problems are identical in structure to the global problem, and are referred to as the local Gibbsian problems. The dimension of the local problem at a link is equal to the size of its immediate neighbourhood and hence typically small, and independent of the network size. Therefore these local Gibbsian problems can be efficiently solved in a scalable fashion. The local solutions are then suitably combined to obtain an approximate solution to the global problem.

We prove that the solution of our local Gibbs optimization method corresponds exactly to the celebrated Bethe approximation [20] to the global Gibbsian optimization problem. The accuracy of the Bethe approximation has been empirically evidenced in various fields [21]. Therefore, it is reasonable to expect a fair degree of accuracy in the context of CSMA as well. In fact, numerical results indicate that in order to obtain the level of accuracy in the fugacities obtained by using our proposed method, the stochastic gradient descent method [7] takes an inordinately long time, often running into tens of millions of time units even for fairly small networks. Therefore, in practical terms, our algorithm will operate with substantially smaller convergence time, compared to the original implementation of adaptive CSMA.

It is worth noting that our algorithm is robust to gradual changes in the desired service rates, as well as to changes in the network topology, since the local solutions can be efficiently re-computed for the new set of requirements. On the other hand, the stochastic gradient descent is likely to take a very long time to converge to its new operating point.

The impractically slow mixing time of the CSMA Markov chain is also known to result in poor delay performance [11], [12]. Recent works like [13], [22] have shown improved delay performance by employing several parallel instances of this Markov chain. However, the result in [13] assumes that the required optimal fugacities can be pre-computed and given to their algorithm. Our local algorithms which efficiently estimate these fugacities can be used in conjunction with the techniques in [13], [22] to obtain a practical CSMA algorithm with good

throughput and delay properties.

The remainder of this paper is organized as follows. In Section II, we describe the SINR based interference model, and review the adaptive CSMA algorithm. In Section III, we introduce the local Gibbsian problems and propose our algorithm for computing the fugacities for a given service rate requirements. Section IV provides a brief review of the Bethe approximation, as relevant to adaptive CSMA. In Section V, we derive our main result which establishes the equivalence between the local Gibbs optimization, and the Bethe approximation. In Section VI, we consider the conflict graph model as a special case of the SINR model and derive closed-form expressions for the local Gibbsian problems. In Section VII, we propose a local algorithm to solve the utility maximization problem and quantify its performance. In Section VIII, we present numerical results to confirm the fast convergence, and Section IX concludes the paper.

II. MODEL AND PRELIMINARIES

We consider a single-hop wireless network and model the links using a bipole model, introduced in [23]. In a bipole model, each transmitter is associated with a receiver on the Euclidean plane. A transmitter and its corresponding receiver are referred to as a link. Let \mathcal{N} denote the set of all the links in the network. Let $|\mathcal{N}| = N$ be the total number of links. Let r_{ii} denote the distance between the transmitter and receiver of link i . For simplicity, we assume¹ that a link distance r_{ii} is much smaller than the distances of the transmitter and the receiver from the other links. With this assumption, we can think of links as points in the Euclidean space \mathbb{R}^2 . Let r_{ji} denote the distance between the links i, j . We assume a time slotted model.

Interference model: We consider the standard path-loss model $\|d\|^{-\alpha}$, $\alpha > 2$, where d is the distance between a receiver and a transmitter, and α is the path loss exponent. Let P_i denote the transmit power of link i . We assume white Gaussian thermal noise at all the receivers with variance w . Let $\mathbf{x}(t) = [x_i(t)]_{i=1}^N$ denote the schedule of the network at time t . Specifically, $x_i(t) = 1$ denotes that the link i is active (transmitting) in time slot t . If there is no ambiguity, we will also use \mathbf{x} to denote $\mathbf{x}(t)$.

Although all the active links in the network can potentially contribute to the interference, the aggregate interference from the transmitters beyond a certain distance can be safely neglected [24], [25]. This approximation is standard in the literature [15] and this distance, referred to as the *close-in* radius is denoted by R_I . Let $\mathcal{N}_i := \{k \mid r_{ik} \leq R_I\}$. For convenience, let link i be also included in the set \mathcal{N}_i . We refer to the links in $\mathcal{N}_i \setminus \{i\}$ as the neighbors of link i . The neighborhood relationship can be represented by an *interference graph* $G(V, E)$. V is the set of links in the network and two links share an edge if they are within a

¹The results in this paper do not require this assumption. This is just to keep the expressions concise.

distance R_I . Then the total interference power at link i is given by

$$I_i(\mathbf{x}) = \sum_{\{j \in \mathcal{N}_i \mid j \neq i, x_j=1\}} P_j r_{ji}^{-\alpha}. \quad (1)$$

Then, the SINR at link i is given by

$$\gamma_i(\mathbf{x}) = \frac{P_i r_{ii}^{-\alpha}}{I_i(\mathbf{x}) + w}. \quad (2)$$

Reception model: We assume that, in each time slot, a single packet of data is transmitted from each active transmitter. If the received SINR at the corresponding receiver exceeds a pre-determined threshold T , i.e., $\gamma_i(\mathbf{x}) \geq T$, the packet is successfully received.

Rate region: A schedule $\mathbf{x} \in \{0, 1\}^{\mathcal{N}}$ is said to be *feasible*, if all the active links in the schedule meet the required SINR constraint, i.e., $\gamma_i(\mathbf{x}) \geq T$, $\forall i$ such that $x_i = 1$. The set of all the feasible schedules is denoted by \mathcal{I} . In our scenario, since each link transmits one data packet whenever it is successful, the long-term service rate of a link is equal to the fraction of time the link is successful. The *rate region* Λ , which is defined as the set of all the possible service rates is given by the convex hull of the feasible schedules in \mathcal{I} . Hence, $\Lambda = \{\sum_{\mathbf{x} \in \mathcal{I}} \alpha_{\mathbf{x}} \mathbf{x} \mid \sum_{\mathbf{x} \in \mathcal{I}} \alpha_{\mathbf{x}} = 1, \alpha_{\mathbf{x}} \geq 0, \mathbf{x} \in \mathcal{I}\}$. If a link scheduling policy can support any rate vector in the rate region, then the scheduling policy is said to be *rate-optimal*.

Adaptive CSMA: We briefly review the adaptive CSMA algorithm [7], [16]. In this algorithm, each link i is associated with a fugacity $\lambda_i > 0$ which defines the underlying Gibbs distribution. In each time slot, a randomly selected link i is allowed to update its schedule $x_i(t)$ based on the information in the previous slot:

- If its SINR is inadequate, i.e., $\gamma_i(\mathbf{x}(t-1)) \leq T$, then $x_i(t) = 0$.
- If $\gamma_i(\mathbf{x}(t-1)) \geq T$, then link i exchanges control messages with its neighbors, to find if they can meet their SINR requirements if link i gets activated. If any of its neighbors cannot meet its requirement, then $x_i(t) = 0$.
- If all the neighbors can meet their SINR requirements even if link i gets activated, then $x_i(t) = 1$ with probability $\frac{\lambda_i}{1+\lambda_i}$, and $x_i(t) = 0$ with probability $\frac{1}{1+\lambda_i}$.

Remarks on Implementation: A challenge in the second step of the above algorithm is to ensure that a newly scheduled link does not alter the SINR requirements of its transmitting neighbours. This problem is typically addressed by introducing a control subslot during which nodes exchange control packets to determine the feasibility of transmissions. More specifically, [16] proposed a control protocol which includes a three-way handshake of control packets: Ready-To-Send (RTS), Clear-To-Send (CTS), and REJECT. In this protocol, if a link is selected to update its status, its transmitter sends an RTS in the control subslot. If any active neighbouring link in the previous schedule $\mathbf{x}(t-1)$ fails to meet the required SINR, the neighbour broadcasts a REJECT signal and disapproves the transmission request. If no REJECT signal is broadcasted, the selected link proceeds to transmit. Detailed descriptions of this handshake protocol can be found in [16, Section III-C].

It can be shown [9, Proposition 1] that the adaptive CSMA algorithm induces a Markov chain on the state space of the schedules $\{0, 1\}^{\mathcal{N}}$. Further, the stationary distribution of the Markov chain, parametrized by the fugacity vector $\lambda = [\lambda_i]_{i=1}^{\mathcal{N}}$, is given by:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{j: x_j=1} \lambda_j \mathbf{1}(\mathbf{x} \in \mathcal{I}), \quad \forall \mathbf{x} \in \{0, 1\}^{\mathcal{N}}, \quad (3)$$

where $\mathbf{1}(\mathbf{x} \in \mathcal{I})$ is an indicator of \mathbf{x} being a feasible schedule, and Z is the normalizing constant. Then, due to the ergodicity of the Markov chain, the long-term service rate of a link i denoted by s_i is equal to the marginal probability that link i is active, i.e., $p_i(x_i = 1)$. Thus, the service rates and the fugacities are related as follows:

$$s_i = p_i(1) = \sum_{\mathbf{x}: x_i=1} Z^{-1} \prod_{j: x_j=1} \lambda_j, \quad \forall i \in \mathcal{N}, \quad (4)$$

where $p_i(1)$ denotes $p_i(x_i = 1)$. The adaptive CSMA algorithm can support any service rate in the rate region provided appropriate fugacities are used for the underlying Gibbs distribution [12, Theorem 5].

If the desired service rates are known, these fugacities can be obtained by solving the system of equations in (4). In [26, Section 3.3], it is shown that solving this system of equations can be posed as a convex optimization problem. The main idea is explained as follows. Given a set of service rates from the rate region, by the definition of rate region, there should exist some distribution $\alpha(\mathbf{x})$ on the state space of feasible schedules \mathcal{I} which would support the required service rates. Note that the definition of rate region does not impose that the distribution $\alpha(\mathbf{x})$ is a Gibbs distribution. In [26, Section 3.3], the authors find a Gibbs distribution that is close to this distribution $\alpha(\mathbf{x})$. It is achieved by minimizing the KL divergence [26, Section 3.3] between $\alpha(\mathbf{x})$ and the family of Gibbs distributions parameterized by the fugacity vectors. Further it has been shown [26, Section 3.3] that the resulting minimization problem is equivalent to the following Gibbsian optimization problem referred to as the global problem.

The global Gibbsian problem:

$$\ln \lambda = \arg \max_{r \in \mathbb{R}^{\mathcal{N}}} G(r), \quad (5)$$

$$\text{where } G(r) := \sum_{k \in \mathcal{N}} s_k r_k - \ln \left(\sum_{y \in \mathcal{I}} \exp \left(\sum_{k \in \mathcal{N}} y_k r_k \right) \right).$$

Here $\{s_i\}_{i \in \mathcal{N}} \in \Lambda$ are the desired service rates.

Remark: To understand that the global Gibbsian problem (5) solves the system of equations in (4), we can simply set $\frac{\partial G(r)}{\partial r_i} = 0$ to obtain

$$s_i - \frac{\sum_{y \in \mathcal{I}: y_i=1} \exp \left(\sum_{k=1}^{\mathcal{N}} y_k r_k^* \right)}{\sum_{y \in \mathcal{I}} \exp \left(\sum_{k=1}^{\mathcal{N}} y_k r_k^* \right)} = 0, \quad \forall i \in \mathcal{N}.$$

Observe that for $\lambda_i = e^{r_i^*}$, the above equations essentially boil down to the desired equations in (4).

A distributed stochastic gradient descent algorithm was proposed in [7] to solve (5). However, estimating the gradient of $G(r)$ in a distributed manner entails the underlying Markov

chain of the CSMA algorithm to converge to steady-state, which takes an impractically long time in general [13].

In the next section, we consider this scenario where the links know their target service rates. We provide an efficient and scalable approximation to this problem, by proposing the following local Gibbsian problems. The solutions of these local problems are appropriately combined to estimate the solution to the global problem.

III. THE LOCAL GIBBSIAN PROBLEMS

We now introduce some definitions required for the description of the local Gibbsian problems.

Local schedule: Let $\mathbf{x}^{(j)} \in \{0, 1\}^{\mathcal{N}_j}$, be the set of variables corresponding to the transmission status of the link j and its neighbors, i.e., $\mathbf{x}^{(j)} := \{x_k \mid k \in \mathcal{N}_j\}$. We refer to $\mathbf{x}^{(j)}$ as the *local schedule* at j . Further, from (1), it can be observed that the SINR of a link depends only on the local schedule. Hence the SINR at a link j can be viewed as a function of $\mathbf{x}^{(j)}$, i.e., $\gamma_j(\mathbf{x}) = \gamma_j(\mathbf{x}^{(j)})$. Also, recall that a schedule \mathbf{x} is said to be feasible, if all the active links in the schedule meet their required SINR threshold. Thus, $\mathbf{1}(\mathbf{x} \in \mathcal{I})$ can be factorized over the local schedule variables $\{\mathbf{x}^{(j)}\}_{j=1}^N$ as $\mathbf{1}(\mathbf{x} \in \mathcal{I}) = \prod_{j: x_j=1} \mathbf{1}(\gamma_j(\mathbf{x}^{(j)}) \geq T)$. Then, (3) can be written as

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{j: x_j=1} \lambda_j \mathbf{1}(\gamma_j(\mathbf{x}^{(j)}) \geq T), \quad \forall \mathbf{x} \in \{0, 1\}^N. \quad (6)$$

Local feasibility: A local schedule $y = [y_k]_{k \in \mathcal{N}_j} \in \{0, 1\}^{\mathcal{N}_j}$ at link j is said to be *feasible*, if either the link j is inactive (i.e., $y_j = 0$), or it is active and meets the required threshold SINR i.e., ($y_j = 1$ and $\mathbf{1}(\gamma_j(y) \geq T)$). The set of all the feasible local schedules at j is denoted by \mathcal{I}_j . It can be observed from (6) that $p(\mathbf{x})$ assigns zero probability to a schedule \mathbf{x} if any of its local schedule is infeasible.

Local service rate vector: Let $\{s_i\}_{i=1}^N$ be the set of service rates of all the links in the network. Then the local service rate vector at link j denoted by $s^{(j)}$ be defined as the set of all the service rates corresponding to link j and its neighbors, i.e., $s^{(j)} := \{s_k \mid k \in \mathcal{N}_j\}$.

Local capacity region: The local capacity region at link j is defined as the convex hull of the local feasible schedules at link j given by

$$\Lambda_j = \left\{ \sum_{z \in \mathcal{I}_j} \alpha_z z \mid \sum_{z \in \mathcal{I}_j} \alpha_z = 1, \alpha_z \geq 0, z \in \mathcal{I}_j \right\}. \quad (7)$$

Now, we define the local Gibbsian problem at link j as follows:

$$\beta_j = \arg \max_{r \in \mathbb{R}^{\mathcal{N}_j}} F(r), \quad (8)$$

where $\beta_j := [\beta_{jk}]_{k \in \mathcal{N}_j}$, and the function $F: \mathbb{R}^{\mathcal{N}_j} \rightarrow \mathbb{R}$ is

$$F(r) := \sum_{k \in \mathcal{N}_j} s_k r_k - \ln \left(\sum_{y \in \mathcal{I}_j} \exp \left(\sum_{k \in \mathcal{N}_j} y_k r_k \right) \right).$$

Observe that, the local problems are structurally similar to the global problem, except that \mathcal{I} in the global problem is replaced by \mathcal{I}_j , and \mathcal{N} is replaced by \mathcal{N}_j . In particular, the dimension of the local problem at link i is just $|\mathcal{N}_i|$. The solutions to

these local problems are referred to as the *local fugacities*. In particular, at each link $j \in \mathcal{N}$, there is a local fugacity vector

$$\beta_j := [\beta_{jk}]_{k \in \mathcal{N}_j}.$$

Local algorithm: Here, we propose a simple and distributed algorithm (Algorithm 1) to solve the local Gibbsian problems and subsequently compute the approximate global fugacities by combining the local solutions using (9). These approximate global fugacities can be directly used in the CSMA algorithm instead of adapting the fugacities using a stochastic gradient descent on the global problem which usually doesn't converge in practical time scales. Each link in the network executes the following algorithm in parallel.

Algorithm 1: Local Gibbsian method at link j

Input: $(s_k, k \in \mathcal{N}_j)$; *Output:* $\tilde{\lambda}_j$.

- 1) Obtain the service rates $(s_k, k \in \mathcal{N}_j)$ from the neighbours.
- 2) Compute the local fugacities $(\beta_{jk}, k \in \mathcal{N}_j)$ by solving the local problem (8) using the Newton's method.
- 3) From each neighbour $k \in \mathcal{N}_j$, obtain the local fugacity β_{kj} .
- 4) Compute the approximate global fugacity $\tilde{\lambda}_j$ as

$$\tilde{\lambda}_j = \left(\frac{1 - s_j}{s_j} \right)^{|\mathcal{N}_j| - 1} \prod_{k \in \mathcal{N}_j} e^{\beta_{kj}}. \quad (9)$$

Information exchange: The algorithm requires only two steps of information exchange with the neighbours. Once in the first step, to obtain the service requirements of the neighbours, and again in the third step to obtain the local fugacities computed at the neighbours. Except for these two information exchanges, the algorithm is fully distributed and can be executed independently at each link.

Computational complexity: The implementation of the Newton's method [27, Section 9.5] in the second step of our algorithm is feasible. This is because the gradient and the Hessian of the local objective function $F(r)$ can be analytically computed since the dimension of the problem is small. The exact expressions for the gradient and Hessian at link j can be computed using certain marginals of the distribution $\hat{b}_j(\mathbf{x}^{(j)}) = Z_j^{-1} \exp(\sum_{k \in \mathcal{N}_j} x_k r_k)$, $\forall \mathbf{x}^{(j)} \in \mathcal{I}_j$, where Z_j is a normalization constant.

For $k \in \mathcal{N}_j$, let $m_k(r)$ represent the probability $\mathbb{P}(x_k = 1)$ under the distribution \hat{b}_j . Similarly, for $i, k \in \mathcal{N}_j$, let $m_{ik}(r)$ denote the probability $\mathbb{P}(x_i = 1, x_k = 1)$ under the same distribution \hat{b}_j . Then, the gradient and Hessian of the function $F(r)$ are given by

$$\begin{aligned} [\nabla(F(r))]_k &= s_k - m_k(r), \quad k \in \mathcal{N}_j, \\ [\nabla^2(F(r))]_{ik} &= \begin{cases} m_{ik}(r) - m_i(r)m_k(r), & i, k \in \mathcal{N}_j, i \neq k \\ m_k(r) - m_k(r)^2, & i = k. \end{cases} \end{aligned}$$

The computation of the gradient and the Hessian requires the information about the local feasible schedules at a link. Specifically, this information is required to compute the normalization constant Z_j . These computations are feasible because the $O(2^{|\mathcal{N}_j|})$ complexity involved in this computation

scales only with the size of the local neighborhood, and is independent of the total size of the network which could be substantially large. In particular, in spatial networks where the neighborhood size does not scale with the network size, our algorithm is order optimal.

In Section V, we prove that the approximate global fugacities $\{\tilde{\lambda}_j\}_{j=1}^N$ obtained using the local Gibbsian method (9) correspond exactly to performing the well known Bethe approximation to the global Gibbsian problem. In the next section, we review the Bethe approximation technique.

IV. REVIEW OF THE BETHE APPROXIMATION

We now introduce some terminology required to describe the Bethe approximation technique [20].

A. Product form distribution

Let S be a finite set, and let $X_i, i = 1, 2, \dots, N$, be random variables each taking values in S . The joint PMF (probability mass function) of the random variables is succinctly denoted as $p(\mathbf{x})$, where $\mathbf{x} = \{x_1, x_2, \dots, x_N\}$. Suppose that $p(\mathbf{x})$ factors into a product of M functions and is given by,

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{j=1}^M f_j(\mathbf{x}^{(j)}), \quad \mathbf{x} \in S^N. \quad (10)$$

The function $f_j(\cdot)$ has arguments $\mathbf{x}^{(j)}$ that are some non-empty subset of $\mathbf{x} = \{x_1, x_2, \dots, x_N\}$. Here Z is a normalization constant. Further, the product form distributions are generally represented using a graph called factor graph [20] that has two sets of nodes namely variable nodes and factor nodes corresponding to the variables and the functions in (10) respectively. An edge is drawn between a variable node and a factor node if the variable is an argument of that factor function.

For the product form distribution considered in (10), we are interested in certain marginal probabilities called the variable marginals and the factor marginals.

Variable node marginals: The marginal probability distribution $p_i(x_i)$ corresponding to a variable node i , is obtained by summing $p(\mathbf{x})$ over the variables corresponding to all other variable nodes, i.e., $p_i(x_i) = \sum_{\mathbf{x} \setminus x_i} p(\mathbf{x})$, $x_i \in S$, $i = 1, \dots, N$.

Factor node marginals: Corresponding to each factor node, $j = 1, \dots, M$, the marginal probability function $\hat{p}_j(\mathbf{x}^{(j)})$, is obtained by summing $p(\mathbf{x})$ over all the variables in $\mathbf{x} \setminus \mathbf{x}^{(j)}$. Let N_j denote the number of arguments in f_j , i.e., $N_j = |\mathbf{x}^{(j)}|$. Then, $\hat{p}_j(\mathbf{x}^{(j)}) = \sum_{\mathbf{x} \setminus \mathbf{x}^{(j)}} p(\mathbf{x})$, $\mathbf{x}^{(j)} \in S^{N_j}$.

Let $\mathbf{p}_v := \{p_i\}_{i=1}^N$, $\mathbf{p}_f := \{\hat{p}_j\}_{j=1}^M$ denote the collection of all the variable node marginals and the factor node marginals corresponding to the distribution $p(\mathbf{x})$ respectively.

The computation of these marginal probability distributions requires the computation of the normalization constant Z , which is an NP-hard problem [20]. Next, we discuss the notions of Gibbs free energy (GFE) and the Bethe free energy (BFE) which provide a variational characterization of the normalization constant and the marginal distributions.

B. Gibbs free energy

Consider a probability distribution $p(\mathbf{x})$ of the form (10) for which we are interested in finding the normalization constant Z , and the marginals. Let $b(\mathbf{x})$ be some distribution on S^N . Here, $p(\mathbf{x})$ is referred to as the *true* distribution, and $b(\mathbf{x})$ is referred to as the *trial* distribution as it will be used to estimate the true distribution $p(\mathbf{x})$. We now introduce the notion of energy function which is required to define the Gibbs free energy [20]. For a given distribution $p(\mathbf{x})$ in the form of (10), the energy function $E(\mathbf{x}) : S^N \rightarrow \mathbb{R}$ is defined as $E(\mathbf{x}) = -\sum_{j=1}^M \ln f_j(\mathbf{x}^{(j)})$. Observe that the energy function $E(\mathbf{x})$ completely specifies the distribution $p(\mathbf{x})$ in (10), as $p(\mathbf{x}) = \frac{1}{Z} e^{-E(\mathbf{x})}$. Using the definition of the energy function, the GFE can be defined as follows.

Definition 1. Consider an energy function $E(\mathbf{x})$ which corresponds to a true distribution $p(\mathbf{x})$. Let $b(\mathbf{x})$ be a trial distribution. Then the Gibbs free energy $F_G(b)$ is defined as $F_G(b) = U_G(b) - H_G(b)$, where the term $U_G(b) = \sum_{\mathbf{x} \in S^N} b(\mathbf{x}) E(\mathbf{x})$ is called the average energy, and $H_G(b) = -\sum_{\mathbf{x} \in S^N} b(\mathbf{x}) \ln b(\mathbf{x})$ is the entropy of the distribution $b(\mathbf{x})$.

The GFE provides a variational characterization of the normalization constant Z of $p(\mathbf{x})$ as $-\ln Z = \min_b F_G(b)$. Further, if the minimum of $F_G(b)$ is achieved at b^* , then $p(\mathbf{x}) = b^*(\mathbf{x})$, $\forall \mathbf{x} \in S^N$.

Here, the optimization is over all the possible distributions on S^N . However, as N becomes large, this procedure is intractable, as the optimization variables take exponentially large memory to store. Moreover, this method only computes the partition function, but doesn't explicitly compute the marginals. We hence take recourse to the Bethe approximation, which is an approximate, but a more practical technique to estimate the marginals explicitly. In particular, the Bethe approximation does two approximations: First, the GFE $F_G(b)$ is approximated using the BFE $F_B(b)$ defined in (11). Second, the optimization of the BFE is performed over a restricted set of distributions, which will be described in (13).

C. Bethe approximation

1) *Bethe free energy (BFE):* We first define the Bethe entropy of a distribution $b(\mathbf{x})$, which is a function of the factor and variable marginals of $b(\mathbf{x})$. The Bethe entropy is given by

$$H_B(\mathbf{b}_f, \mathbf{b}_v) = \sum_{j=1}^M \hat{H}_j(\hat{b}_j) - \sum_{i=1}^N (d_i - 1) H_i(b_i).$$

Here, d_i is the degree of the variable node i in the factor graph, and $\hat{H}_j(\hat{b}_j) = -\sum_{y \in S^{N_j}} \hat{b}_j(y) \ln \hat{b}_j(y)$, $H_i(b_i) = -\sum_{y \in S} b_i(y) \ln b_i(y)$ denote the entropies of the factor marginal $\hat{b}_j(\mathbf{x}^{(j)})$, and the variable marginal $b_i(x_i)$ respectively.

Definition 2. Consider a true distribution $p(\mathbf{x})$ of the form (10). Let $b(\mathbf{x})$ be a trial distribution with factor and variable marginals given by $\mathbf{b}_f = \{\hat{b}_j\}_{j=1}^M$ and $\mathbf{b}_v = \{b_i\}_{i=1}^N$. Then, the Bethe free energy $F_B(b)$ corresponding to the true distribution $p(\mathbf{x})$ is defined as

$$F_B(b) = F_B(\mathbf{b}_f, \mathbf{b}_v) = U_B(\mathbf{b}_f, \mathbf{b}_v) - H_B(\mathbf{b}_f, \mathbf{b}_v), \quad (11)$$

$$\text{where } U_B(\mathbf{b}_f, \mathbf{b}_v) = - \sum_{j=1}^M \sum_{\mathbf{x}^{(j)} \in S^{N_j}} \hat{b}_j(\mathbf{x}^{(j)}) \ln f_j(\mathbf{x}^{(j)}). \quad (12)$$

2) *Bethe optimization (BO)*: The following optimization problem is referred to as the Bethe optimization.

$$\begin{aligned} \arg \min_{\mathbf{b}_f, \mathbf{b}_v} F_B(\mathbf{b}_f, \mathbf{b}_v), \text{ subject to} \quad (13) \\ b_i(x_i) \geq 0, \quad i = 1, \dots, N, \quad x_i \in S, \\ \sum_{x_i} b_i(x_i) = 1, \quad i = 1, \dots, N, \\ \hat{b}_j(\mathbf{x}^{(j)}) \geq 0, \quad j = 1, \dots, M, \quad \mathbf{x}^{(j)} \in S^{N_j}, \\ \sum_{\mathbf{x}^{(j)}} \hat{b}_j(\mathbf{x}^{(j)}) = 1, \quad j = 1, \dots, M, \\ \sum_{\mathbf{x}^{(j)} \setminus \{x_i\}} \hat{b}_j(\mathbf{x}^{(j)}) = b_i(x_i), \quad j = 1, \dots, M, \quad i \in \mathcal{N}_j, \quad x_i \in S, \end{aligned}$$

where \mathcal{N}_j is the set of indices of all the variable nodes associated with the factor node j .

Note: A feasible collection $(\mathbf{b}_f, \mathbf{b}_v)$ of the Bethe optimization problem does not necessarily represent the marginals of any coherent joint distribution over S^N , and are therefore referred to as *pseudo-marginals* [20]. In spite of performing the optimization over a relaxed set, the solution of the Bethe optimization $(\mathbf{b}_f^*, \mathbf{b}_v^*)$ exactly coincides with the marginals of the true distribution $(\mathbf{p}_f^*, \mathbf{p}_v^*)$, if the underlying factor graph is a tree. Further, for factor graphs with loops, the solution leads to very good estimates of the true marginals [20], [21]. In many applications, solving for the global minimizer of (13) is not feasible because of its non-convexity. Hence, a stationary point of the BFE satisfying the constraints in (13) is considered as a good estimate of the true marginals [20], [21].

D. Bethe approximation for CSMA

We observe that the stationary distribution of the CSMA Markov chain $p(\mathbf{x})$ given by (6) is in product form (10) with the factor functions $\{f_j\}_{j=1}^N$ defined as

$$f_j(\mathbf{x}^{(j)}) = \begin{cases} \lambda_j, & \text{if } x_j = 1, \gamma_j(\mathbf{x}^{(j)}) \geq T, \\ 0, & \text{if } x_j = 1, \gamma_j(\mathbf{x}^{(j)}) < T, \\ 1, & \text{if } x_j = 0. \end{cases} \quad (14)$$

where $\mathbf{x}^{(j)}$ is the local schedule of the link j defined earlier.

By the definition of local feasibility, $p(\mathbf{x})$ assigns zero probability to a schedule \mathbf{x} , if it is locally infeasible at any link. Hence, the factor node marginals $\mathbf{p}_f = \{\hat{p}_j\}_{j=1}^N$ have to satisfy $\hat{p}_j(y) = 0, \quad \forall y \notin \mathcal{I}_j, \forall j \in \mathcal{N}$.

BFE for CSMA: We shall consider the product form representation of the CSMA stationary distribution $p(\mathbf{x})$ in (6) and compute the BFE. As defined in (11), the BFE for a given trial distribution $b(\mathbf{x})$ has two terms $U_B(\mathbf{b}_f, \mathbf{b}_v)$, $H_B(\mathbf{b}_f, \mathbf{b}_v)$. Now we compute the first term $U_B(\mathbf{b}_f, \mathbf{b}_v)$. Specifically, we show that the term $U_B(\mathbf{b}_f, \mathbf{b}_v)$ which is in general a function of the factor marginals \mathbf{b}_f , reduces to a function of just the variable marginals \mathbf{b}_v , subject to the feasibility conditions stated in the following lemma.

Lemma 1. *Let $(\mathbf{b}_f, \mathbf{b}_v)$ be a set of feasible factor and variable marginals of the Bethe optimization problem (13). If the factor marginals of a trial distribution \mathbf{b}_f , assign zero probabilities to all the infeasible local schedules, i.e.,*

$$\hat{b}_j(y) = 0, \quad \forall y \notin \mathcal{I}_j, \quad \forall j \in \mathcal{N}, \quad (15)$$

then $U_B(\mathbf{b}_f, \mathbf{b}_v)$ reduces to $U_B(\mathbf{b}_f, \mathbf{b}_v) = U_B(\mathbf{b}_v) = -\sum_{i=1}^N b_i(1) \ln \lambda_i$. Moreover, if \mathbf{b}_f does not satisfy (15), $U_B(\mathbf{b}_f, \mathbf{b}_v) = \infty$, which in turn implies $F_B(\mathbf{b}_f, \mathbf{b}_v) = \infty$.

Proof. Recalling the definition of local feasibility (introduced in Section III), we obtain

$$\mathcal{I}_j = \{\mathbf{x}^{(j)} \mid x_j = 0\} \cup \{\mathbf{x}^{(j)} \mid x_j = 1, \gamma_j(\mathbf{x}^{(j)}) \geq T\}.$$

Consider the inner summation of $U_B(\mathbf{b}_f, \mathbf{b}_v)$ in (12), and expand it into two terms as follows:

$$\begin{aligned} & \sum_{\mathbf{x}^{(j)} \in \{0,1\}^{N_j}} \hat{b}_j(\mathbf{x}^{(j)}) \ln f_j(\mathbf{x}^{(j)}) \\ &= \sum_{\{\mathbf{x}^{(j)} \in \mathcal{I}_j\}} \hat{b}_j(\mathbf{x}^{(j)}) \ln f_j(\mathbf{x}^{(j)}) + \sum_{\{\mathbf{x}^{(j)} \notin \mathcal{I}_j\}} \hat{b}_j(\mathbf{x}^{(j)}) \ln f_j(\mathbf{x}^{(j)}). \end{aligned} \quad (16)$$

Now, we evaluate the first term of (16).

$$\begin{aligned} & \sum_{\{\mathbf{x}^{(j)} \in \mathcal{I}_j\}} \hat{b}_j(\mathbf{x}^{(j)}) \ln f_j(\mathbf{x}^{(j)}) \stackrel{(a)}{=} \\ & \sum_{\{\mathbf{x}^{(j)} \mid x_j=0\}} \hat{b}_j(\mathbf{x}^{(j)}) \ln f_j(\mathbf{x}^{(j)}) + \sum_{\{\mathbf{x}^{(j)} \mid x_j=1, \gamma_j(\mathbf{x}^{(j)}) \geq T\}} \hat{b}_j(\mathbf{x}^{(j)}) \ln f_j(\mathbf{x}^{(j)}), \\ & \stackrel{(b)}{=} \sum_{\{\mathbf{x}^{(j)} \mid x_j=0\}} \hat{b}_j(\mathbf{x}^{(j)}) \ln 1 + \sum_{\{\mathbf{x}^{(j)} \mid x_j=1, \gamma_j(\mathbf{x}^{(j)}) \geq T\}} \hat{b}_j(\mathbf{x}^{(j)}) \ln \lambda_j, \\ &= \ln \lambda_j \sum_{\{\mathbf{x}^{(j)} \mid x_j=1, \gamma_j(\mathbf{x}^{(j)}) \geq T\}} \hat{b}_j(\mathbf{x}^{(j)}). \end{aligned} \quad (17)$$

where (a) follows from the definition of local feasibility, (b) follows from the definition of factors in (14).

We now show that the second term of (16) evaluates to zero. If $\mathbf{x}^{(j)} \notin \mathcal{I}_j$, then from (14) and the assumption (15), we have $f_j(\mathbf{x}^{(j)}) = 0$ and $\hat{b}_j(\mathbf{x}^{(j)}) = 0$ respectively. Hence the summation corresponding to $\{\mathbf{x}^{(j)} \notin \mathcal{I}_j\}$ in (16) is equal to zero.

Now, we use the fact that $(\mathbf{b}_f, \mathbf{b}_v)$ is feasible for the Bethe optimization problem (13). Specifically, we use the last equality constraint in (13) for $i = j$ case (recall from the definition of \mathcal{N}_j , that the link j is also included in the set \mathcal{N}_j) to obtain

$$\begin{aligned} b_j(1) &= \sum_{\{\mathbf{x}^{(j)} \mid x_j=1\}} \hat{b}_j(\mathbf{x}^{(j)}), \\ &= \sum_{\{\mathbf{x}^{(j)} \mid x_j=1, \gamma_j(\mathbf{x}^{(j)}) \geq T\}} \hat{b}_j(\mathbf{x}^{(j)}) + \sum_{\{\mathbf{x}^{(j)} \mid x_j=1, \gamma_j(\mathbf{x}^{(j)}) < T\}} \hat{b}_j(\mathbf{x}^{(j)}), \\ & \stackrel{(15)}{=} \sum_{\{\mathbf{x}^{(j)} \mid x_j=1, \gamma_j(\mathbf{x}^{(j)}) \geq T\}} \hat{b}_j(\mathbf{x}^{(j)}) + 0. \end{aligned} \quad (18)$$

Substituting (17), (18) in (16), we obtain

$$\sum_{\mathbf{x}^{(j)} \in \{0,1\}^{N_j}} \hat{b}_j(\mathbf{x}^{(j)}) \ln f_j(\mathbf{x}^{(j)}) = b_j(1) \ln \lambda_j.$$

Using the above equation in the definition of $U_B(\mathbf{b}_f, \mathbf{b}_v)$ in (12) completes the proof of this Lemma. \square

Using Lemma 1 and (11), the BFE for any $(\mathbf{b}_f, \mathbf{b}_v)$ that satisfies the feasibility constraints in (13), (15) is given by

$$F_B(\mathbf{b}_f, \mathbf{b}_v) = F_B\left(\{\hat{b}_j\}_{j=1}^N, \{b_i\}_{i=1}^N\right), \quad (19)$$

$$= \sum_{i=1}^N -b_i(1) \ln \lambda_i - \hat{H}_i(\hat{b}_i) + (d_i - 1)H_i(b_i),$$

where d_i is degree of variable node i in the factor graph. We refer $F_B(\mathbf{b}_f, \mathbf{b}_v)$ (19) as the BFE corresponding to the set of fugacities $\{\lambda_i\}_{i=1}^N$. It essentially means that $F_B(\mathbf{b}_f, \mathbf{b}_v)$ is the BFE corresponding to $p(\mathbf{x})$ in (6).

Remark: Since we are only interested in the Bethe optimization problem (13), throughout this paper, we implicitly assume that the factor and variable marginals $(\mathbf{b}_f, \mathbf{b}_v)$ satisfy the feasibility conditions in (13), (15), and use the BFE $F_B(\mathbf{b}_f, \mathbf{b}_v)$ expression obtained in (19).

V. EQUIVALENCE OF LOCAL GIBBSIAN METHOD AND THE BETHE APPROXIMATION

We first prove certain important structural properties of the BFE, which we then use to establish an equivalence between the local Gibbsian method (Algorithm 1) and the Bethe approximation. In particular, the above equivalence hinges on establishing the following two important properties, which are formalized subsequently in Lemmas 2 through 4.

- At a stationary point of the BFE, the relation between the factor and the variable marginals is captured by the local Gibbsian problem.
- The factor and variable marginals corresponding to a stationary point of the BFE uniquely determine the fugacities.

Remark: The above properties are derived for the Bethe approximation under the SINR model considered in this paper. The Bethe approximation under general settings may not satisfy these properties.

A. Characterization of the stationary points of the BFE

In this subsection, we characterize the stationary points of the Bethe free energy $F_B(\mathbf{b}_f, \mathbf{b}_v)$ (19) in Lemmas 2 and 4. The following Lemma asserts that the stationary points of $F_B(\mathbf{b}_f, \mathbf{b}_v)$ satisfy a maximum entropy property.

Lemma 2. *Let $F_B(\mathbf{b}_f, \mathbf{b}_v)$ (19) denote the BFE corresponding to the fugacities $\{\lambda_i\}_{i=1}^N$. Let $(\mathbf{b}_f^*, \mathbf{b}_v^*)$ be a stationary point of $F_B(\mathbf{b}_f, \mathbf{b}_v)$ that satisfies the feasibility constraints in (13), (15). Then, for each $i \in \mathcal{N}$, the factor marginal \hat{b}_i^* is related to its corresponding variable marginals $\{b_j^*\}_{j \in \mathcal{N}_i}$, through the following constrained entropy maximization problem, parametrized by the variable marginals $\{b_j^*\}_{j \in \mathcal{N}_i}$:*

$$\hat{b}_i^* = \arg \max_{\hat{b}_i} \hat{H}_i(\hat{b}_i), \text{ subject to} \quad (20)$$

$$\hat{b}_i(\mathbf{x}^{(i)}) \geq 0, \quad \forall \mathbf{x}^{(i)} \in \mathcal{I}_i; \quad \sum_{\mathbf{x}^{(i)} \in \mathcal{I}_i} \hat{b}_i(\mathbf{x}^{(i)}) = 1,$$

$$\sum_{\mathbf{x}^{(i)} \setminus \{x_j\}} \hat{b}_i(\mathbf{x}^{(i)}) = b_j^*(x_j), \quad \forall j \in \mathcal{N}_i, \quad x_j = 1. \quad (21)$$

Proof. Proof is provided in Appendix X-A. \square

Further, it can be shown [26, Section 3.5] that there is a unique solution to the maximum entropy problem (20). Specifically, the variable marginals $\{b_j^*\}_{j \in \mathcal{N}_i}$ uniquely characterize the corresponding factor marginal \hat{b}_i^* , through the local Gibbsian problem (22); this is formalized in Lemma 3.

Lemma 3. *Consider the following local Gibbsian problem defined by the variable marginals $\{b_j^*\}_{j \in \mathcal{N}_i}$:*

$$v_i = \arg \max_{r \in \mathbb{R}^{\mathcal{N}_i}} \sum_{k \in \mathcal{N}_i} b_k^*(1) r_k - \ln \left(\sum_{y \in \mathcal{I}_i} \exp \left(\sum_{k \in \mathcal{N}_i} y_k r_k \right) \right). \quad (22)$$

Then the corresponding factor marginal (optimal solution of (20)) is given by

$$\hat{b}_i^*(\mathbf{x}^{(i)}) = \frac{1}{Z_i} \exp \left(\sum_{k \in \mathcal{N}_i} x_k v_{ik} \right), \quad \forall \mathbf{x}^{(i)} \in \mathcal{I}_i, \quad (23)$$

where $v_i = [v_{ik}]_{k \in \mathcal{N}_i}$ is the solution of (22), Z_i is a normalization constant.

Proof. Proof follows by considering the dual problem of (20). The proof can be found in [26, Section 3.5]. \square

Next, in Lemma 4, we prove that the factor and variable marginals at a stationary point of the BFE uniquely characterize the fugacities.

Lemma 4. *Let $F_B(\mathbf{b}_f, \mathbf{b}_v)$ (19) denote the BFE corresponding to the fugacities $\{\lambda_i\}_{i=1}^N$. Let $(\mathbf{b}_f^*, \mathbf{b}_v^*)$ be a stationary point of the $F_B(\mathbf{b}_f, \mathbf{b}_v)$. Then the fugacities are uniquely determined by the factor and variables marginals $(\mathbf{b}_f^*, \mathbf{b}_v^*)$ as follows:*

$$\lambda_i = \left(\frac{1 - b_i^*(1)}{b_i^*(1)} \right)^{d_i - 1} \prod_{j \in \mathcal{N}_i} e^{v_{ji}}, \quad \forall i \in \mathcal{N}, \quad (24)$$

where v_{ji} is an element of the vector $v_j = [v_{jk}]_{k \in \mathcal{N}_j}$, that characterizes the factor marginal \hat{b}_j^* of some $j \in \mathcal{N}_i$.

Proof. Proof is provided in Appendix X-B. \square

In Lemmas 2, 4 we have derived the necessary conditions that a stationary point of the BFE should satisfy. In Appendix X-C, we show that the conditions in these two Lemmas together constitute a sufficient condition for a stationary point of the BFE.

B. Local Gibbsian method gives the Bethe approximated fugacities

We are now in a position to state a key result of this paper, which asserts that the approximated global fugacities (9) obtained by solving the local Gibbsian problems (8) correspond exactly to the Bethe approximated fugacities. We formalize this result through Definition 3, Theorem 1.

Definition 3. *(Bethe approximated fugacities) For a given set of service rates $\{s_i\}_{i=1}^N$, a set of fugacities $\{\lambda_i\}_{i=1}^N$ are said to be Bethe approximated fugacities if the following holds: “Consider the BFE $F_B(\mathbf{b}_f, \mathbf{b}_v)$ corresponding to the fugacities $\{\lambda_i\}_{i=1}^N$. Then there should exist a stationary point*

$(\mathbf{b}_f^*, \mathbf{b}_v^*)$ point of the BFE such that the corresponding variable marginals are equal to the given service rates, i.e., $b_i^*(1) = s_i, \forall i$ ”.

Theorem 1. Let $\{s_i\}_{i=1}^N$ be the desired service rates.

1. Then the global fugacities $\{\tilde{\lambda}_i\}_{i=1}^N$ estimated in (9) are the Bethe approximated fugacities for $\{s_i\}_{i=1}^N$.
2. For a given set of service rates, the Bethe approximated fugacities are unique.

Proof. Consider the BFE (25) corresponding to the fugacities $\{\tilde{\lambda}_i\}_{i=1}^N$. We show there exists a stationary point $(\{\hat{b}_j\}, \{\underline{b}_j\})$ of the BFE, given by (26)-(27), such that the corresponding variable marginals $\{\underline{b}_j(1)\}$ (26) are equal to the desired service rates $\{s_i\}_{i=1}^N$.

$$F_B(\mathbf{b}_f, \mathbf{b}_v) = \sum_{i=1}^N -b_i(1) \ln \tilde{\lambda}_i - \hat{H}_i(\hat{b}_i) + (d_i - 1)H_i(b_i), \quad (25)$$

$$\underline{b}_j(1) = s_j, \underline{b}_j(0) = 1 - s_j, \quad \forall j, \quad (26)$$

$$\hat{b}_j(\mathbf{x}^{(j)}) = \frac{1}{Z_j} \exp\left(\sum_{k \in \mathcal{N}_j} x_k \beta_{jk}\right), \quad \forall \mathbf{x}^{(j)} \in \mathcal{I}_j, \forall j, \quad (27)$$

where $\beta_j = [\beta_{jk}]_{k \in \mathcal{N}_j}$ is the local fugacity vector at link j obtained from (8), and Z_j is the corresponding normalization constant.

Now we use the structural properties of the BFE derived in Lemmas 2 through 4 to complete the proof. In particular, recall from (8) that the local fugacity vector $\beta_j = [\beta_{jk}]_{k \in \mathcal{N}_j}$ is obtained by solving the local Gibbsian problem (8), which is same as (22). Hence, due to Lemma 3, it is clear that the set of marginals $(\{\hat{b}_j\}, \{\underline{b}_j\})$ in (26)-(27) satisfy the maximum entropy property stated in Lemma 2.

Next, due to the definition (9) of the global fugacities $\{\tilde{\lambda}_i\}_{i=1}^N$, the variable and factor marginals defined in (26)-(27) immediately satisfy the condition (24) in Lemma 4.

Therefore, $(\{\hat{b}_j\}, \{\underline{b}_j\})$ given in (26)-(27) is a stationary point of the BFE (25). Further, note that the corresponding variable marginals $\{\underline{b}_j(1)\}$ (26) at this stationary point are equal to the desired service rates $\{s_i\}_{i=1}^N$. Hence, the fugacities $\{\tilde{\lambda}_i\}_{i=1}^N$ are the Bethe approximated fugacities for $\{s_i\}_{i=1}^N$.

Next, we prove the uniqueness of the Bethe approximated fugacities. For the given service rates $\{s_i\}_{i=1}^N$, let us assume that two sets of fugacities $\{\lambda_i^1\}_{i=1}^N, \{\lambda_i^2\}_{i=1}^N$ satisfy the definition of the Bethe approximated fugacities. Let $F_B^1(\mathbf{b}_f, \mathbf{b}_v), F_B^2(\mathbf{b}_f, \mathbf{b}_v)$ be the BFE functions corresponding to the fugacities $\{\lambda_i^1\}_{i=1}^N, \{\lambda_i^2\}_{i=1}^N$ respectively. Then there should exist stationary points $(\{\hat{b}_j^1\}, \{\underline{b}_j^1\}), (\{\hat{b}_j^2\}, \{\underline{b}_j^2\})$ of their corresponding BFE functions such that the variables marginals are equal to the service rates, i.e., $\{\underline{b}_j^1(1)\} = \{\underline{b}_j^2(1)\} = \{s_j\}$. Since the variable marginals are same, due to Lemmas 2, 3, the factor marginals corresponding to these stationary points should be the same. In other words, i.e., $(\{\hat{b}_j^1\}, \{\underline{b}_j^1\}) = (\{\hat{b}_j^2\}, \{\underline{b}_j^2\})$. Then (24) from Lemma 4 asserts that $\{\lambda_i^1\}_{i=1}^N = \{\lambda_i^2\}_{i=1}^N$. Hence, for a given set of service rates, the Bethe approximated fugacities are unique. \square

VI. SPECIAL CASE - CONFLICT GRAPH MODEL

Conflict graph model [19] is a special case of the SINR interference model. In the conflict graph model, two links cannot transmit simultaneously, if one link is within the interference range of the other link. Under the conflict graph model, we derive simple closed form expressions for the local fugacities (8).

Theorem 2. For the conflict graph model, the local fugacities i.e., solution of the local Gibbsian problem (8) at a link i is given by

$$e^{\beta_{ij}} = \begin{cases} s_i (1 - s_i)^{|\mathcal{N}_i| - 2} \prod_{k \in \mathcal{N}_i \setminus \{i\}} (1 - s_i - s_k)^{-1}, & \text{if } j = i, \\ s_j (1 - s_i - s_j)^{-1}, & \text{if } j \in \mathcal{N}_i \setminus \{i\}. \end{cases} \quad (28)$$

Proof. Proof is provided in Appendix X-D. \square

Comparison with the results in [19]:

The fugacities for the conflict graph model have been derived in [19]. Here, we derive the global fugacities using our approach and compare to the results in [19].

Corollary 4. For the conflict graph model, the global fugacity at a link i is

$$\tilde{\lambda}_i = \frac{s_i (1 - s_i)^{2|\mathcal{N}_i| - 3}}{\prod_{k \in \mathcal{N}_i \setminus \{i\}} (1 - s_i - s_k)^2}.$$

Proof. We obtain this result by substituting the local fugacities obtained in (28) in the expression proposed for global fugacities (9). \square

The expression for the global fugacities proposed in [19] is

$$\tilde{\lambda}_i = \frac{s_i (1 - s_i)^{|\mathcal{N}_i| - 2}}{\prod_{k \in \mathcal{N}_i \setminus \{i\}} (1 - s_i - s_k)},$$

which is different from the expression derived in Corollary 4. In essence, these expressions are different because, the factorizations considered in the two cases are different. Specifically, it is well known that for a given product form distribution, the Bethe approximation technique could lead to different results, corresponding to different factorizations of the product form distribution [28, Chapter 2, Page 10].

To elucidate this point, we consider the conflict graph interference model, and present two natural factorizations that lead to the same CSMA distribution (3). In the conflict graph model, a schedule \mathbf{x} is said to be feasible if it is an independent set of the underlying graph $G(V, E)$. Using this observation, the term $\mathbf{1}(\mathbf{x} \text{ is feasible})$ in (3) can be factorized in two ways as given below.

1. *Edge-centric factorization:* This factorization ensures that for each edge, not more than one of its end vertices are active.

$$\mathbf{1}(\mathbf{x} \text{ is feasible}) = \prod_{(i,j) \in E} \mathbf{1}(x_i x_j = 0).$$

2. *Vertex-centric factorization:* In this factorization, whenever a vertex is active, it ensures that all its neighbours are inactive.

$$\mathbf{1}(\mathbf{x} \text{ is feasible}) = \prod_{j \in V} f_j(\mathbf{x}^{(j)}),$$

where $\mathbf{x}^{(j)}$ is the set of activation status of node j and its neighbours, and f_j is given by

$$f_j(\mathbf{x}^{(j)}) = \begin{cases} 1, & \text{if } x_j = 0, \\ 1, & \text{if } x_j = 1 \text{ and } \{x_k\}_{k \in \mathcal{N}_j \setminus \{j\}} = 0, \\ 0, & \text{otherwise.} \end{cases}$$

In [19], the authors use the edge-centric factorization, which cannot be directly used to capture the SINR model. Hence we use a more general factorization (14), which reduces to the vertex-centric factorization for the conflict graph case.

Remark: The existing theory on the Bethe approximation is not sufficient to conclude or prove that one of the factorizations is always better than the other. However, the following can be said under some special cases.

- For the conflict graph model, the formula derived in [19] is provably exact if the conflict graph is a tree. Hence, if there is prior knowledge that the conflict graph has tree topology, the formula in [19] should be used.
- From the simulations that we have conducted, we have the following observation for topologies with loops. For the formula proposed in [19], we observed that the achieved service rate is less than the target service rate even for small load. However, for the formula obtained using our approach, the achieved service rate is more than the target service rate at smaller loads. Hence, if we have prior knowledge that we are operating in the low load regime, empirical evidence suggests that it is beneficial to use the approach proposed in our paper.

VII. UTILITY MAXIMIZATION

In this section, we consider the utility maximization problem and provide an approximation algorithm to solve the problem in a distributed manner. The problem is defined as follows. Suppose each link i in the network is associated with a concave utility function of its service rate $U_i : [0, 1] \rightarrow \mathbb{R}_+$. Our objective is to find the service rates that maximize the system wide utility, *i.e.*,

$$\max_{y \in \Lambda} \sum_{i=1}^N U_i(y_i), \quad (29)$$

and subsequently compute the global fugacities that correspond to these optimal service rates.

In [7], an iterative algorithm to update the global fugacities is proposed. However, each iteration of the algorithm requires an underlying slowly mixing Markov chain to reach steady state. Hence it suffers from impractically slow convergence to the optimal fugacities. To address this issue, we propose an iterative algorithm which updates the local fugacities instead of directly updating the global fugacities. These local fugacity updates are computationally simple and do not require any Markov chain to convergence. These local fugacities are then used to obtain the approximate global fugacities using (9) proposed in Section III.

Each link in the network executes *Algorithm 2* in parallel. The algorithm involves solving an one dimensional optimization problem (30) related to the original optimization problem

(29). Here $\theta > 0$, is a parameter of the algorithm that can be tuned. The solution of this optimization problem $s_j(t)$ is used in the subsequent steps of the algorithm to update the local fugacities (31) and the global fugacities. We will later show that the update equation (31) is inspired by a subgradient descent algorithm for a related optimization problem. The term $\alpha(t)$ in (31) corresponds to the step-size of the subgradient descent algorithm. Any standard step-sizes that satisfy the convergence criteria for a subgradient descent method can be used [29, Chapter 2]. A typical example is $\alpha(t) = \frac{1}{t}$.

Algorithm 2: Local utility maximization at link j

- 1) At $t = 0$, initialize $\beta_{jk}(t) = 0, \forall k \in \mathcal{N}_j$.

Computing global fugacities:

- 2) From each neighbour $k \in \mathcal{N}_j$, obtain the local fugacity $\beta_{kj}(t)$.
- 3) Compute $s_j(t)$ as

$$s_j(t) = \arg \max_{q \in [0,1]} \theta U_j(q) - q \sum_{k \in \mathcal{N}_j} \beta_{kj}(t). \quad (30)$$

- 4) Compute the approximate global fugacity $\tilde{\lambda}_j(t)$ from local fugacities ($\beta_{kj}(t), k \in \mathcal{N}_j$) and $s_j(t)$ using (9).

Updating the local fugacities:

- 5) Consider the distribution

$$\hat{b}_j(\mathbf{x}^{(j)}; t) = Z_j^{-1} \exp \left(\sum_{k \in \mathcal{N}_j} x_k \beta_{jk}(t) \right), \forall \mathbf{x}^{(j)} \in \mathcal{I}_j,$$

and for each $k \in \mathcal{N}_j$, let $m_{jk}(t) := \sum_{\mathbf{x}^{(j)}: x_k=1} \hat{b}_j(\mathbf{x}^{(j)}; t)$ represent the marginal probability corresponding to x_k under this distribution.

- 6) For each $k \in \mathcal{N}_j$, obtain $s_k(t)$ computed at the neighbour k , and update the local fugacity at j using

$$\beta_{jk}(t+1) = \beta_{jk}(t) + \alpha(t) (s_k(t) - m_{jk}(t)). \quad (31)$$

Complexity: The computational complexity of the above algorithm is $O(2^{|\mathcal{N}_j|})$, which depends only on the size of the local neighbourhood, and is independent of the total size of the network.

Accuracy: The accuracy of the above algorithm can be understood by splitting the error into two parts. The first part is the error involved in estimating the optimal service rates in (29). The second part is the error involved in estimating the global fugacities corresponding to these service rates. We characterize the first part of the error in Theorem 3, which states that the gap to the optimum utility is $O(\frac{1}{\theta})$. In other words, the local algorithm provides a good estimate of the optimal service rates when θ is large. The second part of the error depends only on the accuracy of the Bethe approximation. The Bethe error is evidenced to be reasonably small in many applications [21].

Theorem 3. In Algorithm 2, the service rates $s(t) = [s_j(t)]_{j=1}^N$ (30) converge to some $\underline{s} = [s_j]_{j=1}^N \in [0, 1]^N$, such that the limit \underline{s} satisfies

$$\sum_{j=1}^N U_j(\underline{s}_j) \geq \max_{y \in \Lambda} \sum_{j=1}^N U_j(y_j) - \frac{\sum_j \log |\mathcal{I}_j|}{\theta}. \quad (32)$$

Proof. We prove this theorem in the following steps:

1. First, we define a new optimization problem (34) that is related to the original utility maximization problem (29).
2. Then, we prove that the proposed local utility maximization algorithm corresponds to a subgradient descent algorithm for the new optimization problem (34).
3. Finally, we complete the proof by showing that the solution of the new optimization problem (34) satisfies the inequality (32) stated in this theorem.

Step 1: Firstly, we relax the constraints of (29) by replacing the actual capacity region Λ , with the Bethe approximated capacity region defined below:

$$\Lambda_B := \{y \in [0, 1]^N \mid y^{(j)} \in \Lambda_j, \forall j \in \mathcal{N}\}. \quad (33)$$

From Section III, recall that $y^{(j)} = [y_k]_{k \in \mathcal{N}_j}$ is the local service rate vector, and Λ_j defined in (7) is the local capacity region which is nothing but the convex hull of the local feasible schedules \mathcal{I}_j . In other words, a local service rate vector $y^{(j)}$ belongs to Λ_j , if and only if there exists a distribution \hat{b}_j on the local feasible schedules \mathcal{I}_j , that supports the service rates $y^{(j)}$, i.e., $y_k = \sum_{\{\mathbf{x}^{(j)} \in \mathcal{I}_j \mid x_k=1\}} \hat{b}_j(\mathbf{x}^{(j)})$, $k \in \mathcal{N}_j$. As this definition of Λ_B imposes only local feasibility of service rates, it is easy to argue that the Bethe capacity region is a relaxation of the actual capacity region, i.e., $\Lambda \subseteq \Lambda_B$.

Secondly, we scale the objective function (29) by a factor θ , and add local entropy terms. The resulting new optimization problem is given by

$$\begin{aligned} \max_{y, \{\hat{b}_j\}} \quad & \theta \sum_{j=1}^N U_j(y_j) + \sum_{j=1}^N H(\hat{b}_j), \quad \text{subject to} \quad (34) \\ & y \in [0, 1]^N; \quad \hat{b}_j(\mathbf{x}^{(j)}) \geq 0, \quad \mathbf{x}^{(j)} \in \mathcal{I}_j, \quad j = 1 \dots N; \\ & \sum_{\mathbf{x}^{(j)} \in \mathcal{I}_j} \hat{b}_j(\mathbf{x}^{(j)}) = 1, \quad j = 1 \dots N; \\ & y_k = \sum_{\{\mathbf{x}^{(j)} \in \mathcal{I}_j \mid x_k=1\}} \hat{b}_j(\mathbf{x}^{(j)}), \quad k \in \mathcal{N}_j, \quad j = 1 \dots N. \quad (35) \end{aligned}$$

Here, the constraint set is specified by explicitly expanding the definition of Λ_j involved in description of Λ_B (33). For a large θ , this new objective function (34) closely approximates the original objective (29) since the entropy is bounded from above and below. The advantage of defining this new optimization problem is that it is amenable to a distributed solution [7]. In particular, our local algorithm solves this problem in a distributed fashion.

Step 2: In the following Lemma, we show that Algorithm 2 solves the new optimization problem (34).

Lemma 5. Let $(\underline{s}, \{\hat{b}_j\})$ be the solution of the optimization problem (34). Then the service rates $\{s_j(t)\}$ (30) in Algorithm 2 converge to the limit $\underline{s} = \{s_j\}$.

Proof. The outline of the proof is to show Algorithm 2 corresponds to the dual subgradient method for (34). The proof is provided in Appendix X-E. \square

Step 3: We now show that the solution of the optimization problem (34) achieves the performance guarantee claimed in (32). In other words, we show that $\underline{s} = \{s_j\}$ satisfies (32).

Recollect that we have added the local entropy terms to the actual objective function to obtain a new problem (34). Here, we shall evaluate the effect of these entropy terms on the optimal utility. To that end, let us consider $(\underline{s}, \{\hat{b}_j\})$, the solution to the maximization problem (34). Let us also consider the following optimization problem which does not have the local entropy terms. $s^* := \arg \max_{y \in \Lambda_B} \sum_j U_j(y_j)$. Note that s^* could be different from \underline{s} . Also, from the definition of Λ_B , for every $y \in \Lambda_B$ there exists some $\{\hat{b}_j\}$ such that $(y, \{\hat{b}_j\})$ is feasible for the problem (34). Hence for $s^* \in \Lambda_B$, there exists some $\{\hat{b}_j^*\}$ such that $(s^*, \{\hat{b}_j^*\})$ is feasible for (34). Then,

$$\begin{aligned} \theta \sum_j U_j(s_j^*) & \leq \theta \sum_j U_j(s_j^*) + \sum_j H(\hat{b}_j^*), \\ & \stackrel{(a)}{\leq} \theta \sum_j U_j(\underline{s}_j) + \sum_j H(\hat{b}_j), \\ & \stackrel{(b)}{\leq} \theta \sum_j U_j(\underline{s}_j) + \sum_j \log |\mathcal{I}_j|, \quad (36) \end{aligned}$$

where (a) follows from the fact that $(\underline{s}, \{\hat{b}_j\})$ maximizes (34), (b) follows from the fact that entropy $H(\hat{b}_j) \leq \log |\mathcal{I}_j|$. Next, using the fact that $\Lambda \subseteq \Lambda_B$, we have

$$\max_{y \in \Lambda} \sum_j U_j(y_j) \leq \max_{y \in \Lambda_B} \sum_j U_j(y_j) = \sum_j U_j(s_j^*). \quad (37)$$

Using (37) in (36) completes the proof of Theorem 3. \square

We conclude this section by contrasting our utility maximization algorithm with a prior work [7] which is also a subgradient descent based algorithm. The key difference between these works is in the complexity involved in computing the subgradients. Specifically, in our algorithm, the complexity of computing the subgradient is independent of the total size of the network. On the other hand, it could be exponentially large in the network size, for the algorithm given in [7].

VIII. NUMERICAL RESULTS

In this section, we present simulations to evaluate the performance of our algorithms. We consider three topologies namely random graphs, complete graph, and grid graph.

A. Bethe error

Simulation setting for random graphs: We generate a spatial random network by uniformly placing the transmitter nodes on a two dimensional square plane of length 8. Each transmitter is associated with a receiver at a distance of 0.5 in a random direction. The path loss exponent α is set to 3, the close-in radius R_I is set to 2.4, and the threshold SINR is set to 15 dB. The transmit power of the links is set to 1. We consider

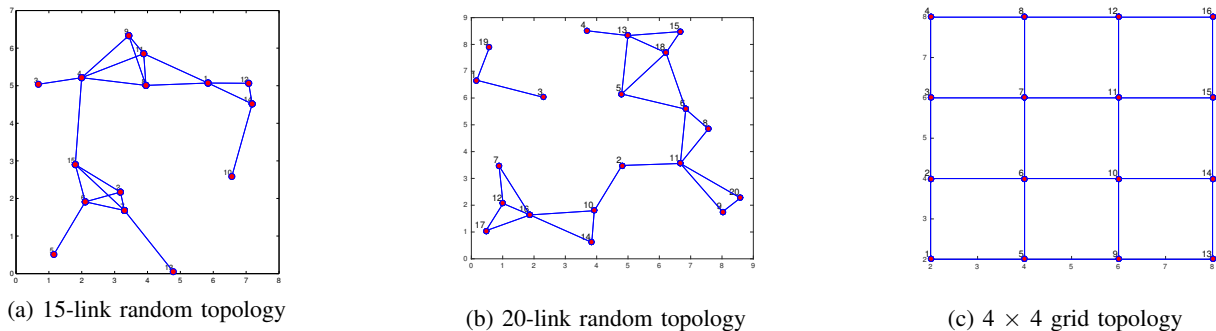
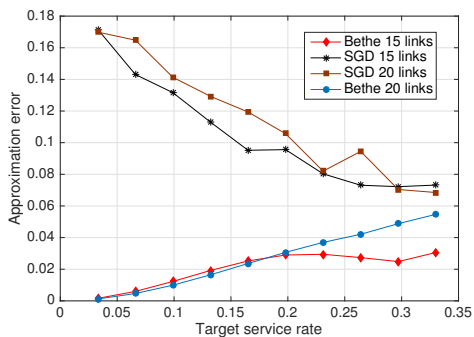
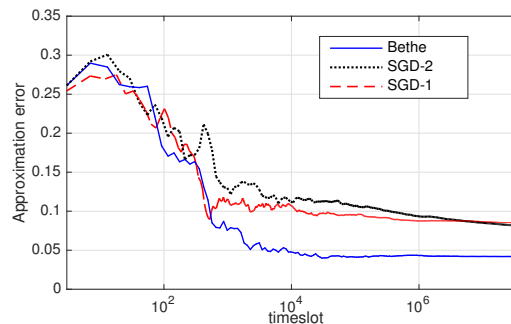


Fig. 1: Illustration of some interference graphs used: Each vertex represents a link in the network. An edge is present if two links are in the interference range.



(a) Error as a function of target service rate for the random topologies



(b) Error as a function of time for 20-link random graph

Fig. 2: Comparison of errors due to SGD and our Bethe approximation based algorithms for random topologies in Figure 1

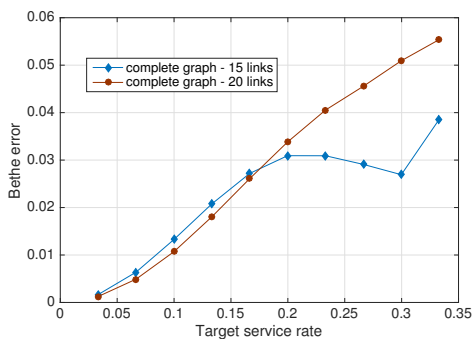


Fig. 3: Bethe error for complete graph topology

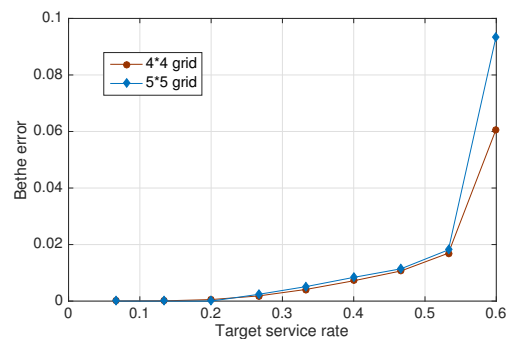


Fig. 4: Bethe error for grid topology

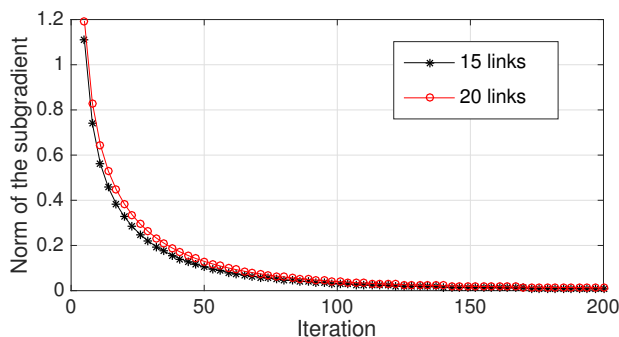


Fig. 5: Convergence of local utility maximization algorithm

equal service rate requirements for all the links. For a given target service rate vector $s^t = [s_i^t]_{i=1}^N$, we define the Bethe approximation error as

$$e(s^t) = \frac{\sum_{i=1}^N |s_i^t - s_i^a|}{N},$$

where $s^a = [s_i^a]_{i=1}^N$ are the service rates that can be supported by the approximated fugacities $\{\tilde{\lambda}_i\}_{i=1}^N$.

Bethe error as function of target service rate: We generated two random networks of sizes 15 and 20 links (shown in Figure 1a, 1b). We compared the error of our approximation algorithm with the residual error of the SGD algorithm [7] after running the SGD for 10^8 time-slots of the CSMA algorithm. In Figure 2a, we plot the approximation error as a

function of the target service rate. Further, for the considered SINR constraints, we numerically observed that for both the random topologies in Figure 1, CSMA can support till a service rate of 0.33 for all the links. Hence, we varied the target service rate from 0 to 0.33. It can be observed from Figure 2a, that for practical time-scales, our algorithm results in better accuracy than the SGD for all the target service rates.

Error as a function of time: In Figure 2b, for a fixed target service rate of 0.25, we plot the approximation error as a function of time, by using the time-averaged service rates observed from CSMA algorithm. Here, our local Gibbs algorithm computes the Bethe approximated fugacities, and uses these static fugacities (*i.e.*, they are not adapted during the algorithm) in the CSMA algorithm. The SGD algorithm starts with some initial fugacities, and adapts the fugacities by observing the corresponding service rates. We simulated two versions of the SGD algorithm (SGD-1, SGD-2) proposed in [7], [30], whose details are as follows: The update rule of SGD algorithm has two functions to be chosen, namely update interval $T(j)$, and step size $\alpha(j)$, for the j^{th} iteration of the gradient descent. The update rule for SGD-1 [7, Section II-D] is given by $\alpha(j) = \frac{1}{(j+2)\log(j+2)}$, $T(j) = j + 2$. The update rule for SGD-2 [30, Scheduling Algorithm 1] is given by $\alpha(j) = \frac{1}{j}$, $T(j) = \exp(\sqrt{j})$.

Although the SGD algorithm will eventually converge to the exact fugacities, from Figure 2b, it can be observed that for practical time-scales of the order of 10^8 time slots, the residual error is rather large compared to our approximation algorithm. This is because, the CSMA Markov chain has to mix only one time for the Bethe approximation based approach, as the fugacities are static. However, in the SGD based approach, for every update in the fugacities, the Markov chain tries to mix to a new steady state distribution.

Complete graph and Grid graph: Here, we consider two complete graph topologies with sizes 15 and 20, and two grid topologies of sizes 16 and 25. A 4×4 grid topology is illustrated in Figure 1c. The range of the target service rates is chosen by numerically observing the maximum supportable service rates for the respective topologies. We plot the Bethe error for these two topologies as a function of the target service rates in Figure 3, 4. As seen from the plots, the error is considerably small for both the topologies.

B. Convergence of Utility maximization algorithm

Here, we consider the case of computing the fugacities for the proportional fairness utility setting, *i.e.*, $U_i(s_i) = \log s_i$ for all the links. We update the fugacities using Algorithm 2 proposed in Section VII. In Figure 5, we plot the norm of the subgradient corresponding to the maximization problem (34), which indicates the convergence of the algorithm. The convergence is plotted for the two random topologies of size 15 and 20. It can be observed that the proposed algorithm converges within 200 iterations.

IX. CONCLUSIONS

We considered the adaptive CSMA algorithm under the SINR interference model, which is known to be throughput

optimal. Under this model, we first proposed a distributed algorithm, namely the local Gibbsian method to efficiently estimate the fugacities, for a given service rate requirements. The convergence rate and the complexity of the proposed algorithm depend only on the maximum size of a link's neighbourhood. We proved that our approximation corresponds exactly to performing the well known Bethe approximation to the global Gibbsian problem. We also proposed an approximation algorithm to estimate the fugacities under a utility maximization framework. Our numerical results indicate that the proposed approximation algorithms can lead to a good degree of accuracy, and improve the convergence time by a few orders of magnitude, compared to the existing stochastic gradient descent methods.

ACKNOWLEDGEMENT

The authors express their sincere gratitude to Dr. Pascal Vontobel for his valuable comments on this work.

X. APPENDIX

A. Proof of Lemma 2

Let us consider the Bethe optimization for the BFE defined in (19):

$$\arg \min_{\{\hat{b}_i\}_{i=1}^N, \{b_i\}_{i=1}^N} F_B \left(\{\hat{b}_i\}_{i=1}^N, \{b_i\}_{i=1}^N \right), \text{ subject to} \quad (38)$$

$$\sum_{\mathbf{x}^{(i)} \setminus \{x_j\}} \hat{b}_i(\mathbf{x}^{(i)}) = b_j(x_j), \quad i \in \mathcal{N}, j \in \mathcal{N}_i, x_j = 1, \quad (39)$$

$$\sum_{\mathbf{x}^{(i)} \in \mathcal{I}_i} \hat{b}_i(\mathbf{x}^{(i)}) = 1, \quad i = 1 \dots N, \quad (40)$$

$$b_i(1) + b_i(0) = 1, \quad i = 1 \dots N. \quad (41)$$

As all the constraints of (38) are linear, they can be eliminated by suitable variable transformation to get an equivalent unconstrained problem [27, chapter 10]. Let us now look at the optimization variables in (38). If we consider the distribution $b_i(\cdot)$, there are 2 optimization variables associated with it, namely $b_i(0)$ and $b_i(1)$. If we consider the distribution $\hat{b}_i(\cdot)$, there is an optimization variable corresponding to each argument $\rho \in \mathcal{I}_i$, *i.e.*, there are $|\mathcal{I}_i|$ variables associated with it, namely $\{\hat{b}_i(\rho)\}_{\rho \in \mathcal{I}_i}$. For convenience, we split the arguments $\rho \in \mathcal{I}_i$ into two sets. We use A to denote the set of arguments in which there is at most one non-zero element, *i.e.*, $A = \{\rho = [\rho_j]_{j \in \mathcal{N}_i} \in \mathcal{I}_i \mid \sum_{j \in \mathcal{N}_i} \rho_j \leq 1\}$, and use B to denote the set of all the other arguments in \mathcal{I}_i . Later, we use the definition of these two sets, to propose some variable transformations that eliminate the equality constraints.

Example: Before we proceed further, let us consider an example which will be used to illustrate the variable transformations. Consider an interference graph with 3-node line topology as shown in Figure 6. For the ease of illustration, let us assume that the SINR constraints are such that $(1, 1, 1)$ is the only infeasible local schedule. In other words, $\mathcal{I}_1 = \{0, 1\}^2$, $\mathcal{I}_2 = \{0, 1\}^3 \setminus (1, 1, 1)$, $\mathcal{I}_3 = \{0, 1\}^2$.

Then for this example, let us consider node 2, and split the arguments of $b_2(\rho)$ into sets A and B . As defined earlier,

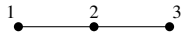


Fig. 6: Illustration of a 3-node interference graph.

Variable marginals: $b_1(x_1)$ $b_2(x_2)$ $b_3(x_3)$

$$\begin{array}{l|l|l} b_1(0) = 1 - y_1 & b_2(0) = 1 - y_2 & b_3(0) = 1 - y_3 \\ b_1(1) = y_1 & b_2(1) = y_2 & b_3(1) = y_3 \end{array}$$

Factor marginals: $\hat{b}_1(x_1, x_2)$ $\hat{b}_2(x_1, x_2, x_3)$ $\hat{b}_3(x_2, x_3)$

$$\begin{array}{l|l|l} \hat{b}_1(0, 0) = 1 - y_1 - y_2 + z_1(1) & \hat{b}_2(0, 0, 0) = 1 - \sum_{i=1}^3 y_i + \sum_{k=1}^3 z_2(k) & \hat{b}_3(0, 0) = 1 - y_2 - y_3 + z_3(1) \\ \hat{b}_1(0, 1) = y_2 - z_1(1) & \hat{b}_2(0, 0, 1) = y_3 - z_2(1) - z_2(2) & \hat{b}_3(0, 1) = y_3 - z_3(1) \\ \hat{b}_1(1, 0) = y_1 - z_1(1) & \hat{b}_2(0, 1, 0) = y_2 - z_2(1) - z_2(3) & \hat{b}_3(1, 0) = y_2 - z_3(1) \\ \hat{b}_1(1, 1) = z_1(1) & \hat{b}_2(0, 1, 1) = z_2(1) & \hat{b}_3(1, 1) = z_3(1) \\ & \hat{b}_2(1, 0, 0) = y_1 - z_2(2) - z_2(3) & \\ & \hat{b}_2(1, 0, 1) = z_2(2) & \\ & \hat{b}_2(1, 1, 0) = z_2(3) & \\ & \hat{b}_2(1, 1, 1) = 0 & \end{array}$$

TABLE I: Illustration of the variable transformations used in the proof of Lemma 2, for the 3-node topology given in Figure 6

the set of arguments which contain at most one non-zero element is given by $A = \{(0, 0, 0), (1, 0, 0), (0, 1, 0), (0, 0, 1)\}$. The other set B containing feasible arguments with more than one non-zero element is given by $B = \{(0, 1, 1), (1, 0, 1), (1, 1, 0)\}$. Similarly, if we consider node 2, the corresponding sets will be $A = \{(0, 0), (0, 1), (1, 0)\}$ and $B = \{(1, 1)\}$.

Variable transformations: Let us now perform the following variable transformations to eliminate the constraint equations, and thereby obtain an equivalent unconstrained optimization problem. For each $i \in \mathcal{N}$,

- (i) Replace the variable $b_i(1)$ by a new variable y_i . Hence, by expressing $b_i(0) = 1 - y_i$, we can eliminate the constraint $b_i(0) + b_i(1) = 1$ in (41).
- (ii) Let a new variable $\mathbf{z}_i = [z_i(k)]_{k=1}^{|B|}$ (which is a vector of length $|B|$) replace the optimization variables corresponding to the arguments in B . For example, if we consider the node 1, the set $B = \{(1, 1)\}$ has only one element. We use $\mathbf{z}_1 = (z_1(1))$, a vector of length one to replace $\hat{b}_1(1, 1)$. Similarly if we consider node 2, its set $B = \{(0, 1, 1), (1, 0, 1), (1, 1, 0)\}$ has 3 elements. We use the vector $\mathbf{z}_2 = (z_2(1), z_2(2), z_2(3))$ to replace $\{\hat{b}_2(0, 1, 1), \hat{b}_2(1, 0, 1), \hat{b}_2(1, 1, 0)\}$ respectively. This transformation is shown in Table I.
- (iii) If an argument $\rho \in A$ has the non-zero element in j th position (i.e., $\rho_j = 1$), then its corresponding variable $\hat{b}_i(\rho)$ can be expressed only in terms of the vector \mathbf{z}_i and variable y_j . This transformation is to replace the local consistency condition in (39). For example, if we consider the 3-node graph, the argument of the probability $\hat{b}_2(1, 0, 0)$ has its non zero element at position $j = 1$. Now consider the local consistency constraint (39) for $i = 2, j = 1$ to obtain $\hat{b}_2(1, 0, 0) + \hat{b}_2(1, 0, 1) + \hat{b}_2(1, 1, 0) = b_1(1)$. Then to capture the above constraint, the transformation $\hat{b}_2(1, 0, 0) = y_1 - z_2(2) - z_2(3)$ can be used. Thus $\hat{b}_2(1, 0, 0)$ is expressed only in terms of the elements of the vector \mathbf{z}_2 , and variable y_1 . It is shown in Table I.
- (iv) The variable $\hat{b}_i(\rho)$ corresponding to $\rho = (0, 0, \dots, 0)$ can be expressed in terms of the vector \mathbf{z}_i and $(y_j, j \in \mathcal{N}_i)$.

This is done to eliminate (40). For example, at node 1, the constraint $\hat{b}_1(0, 0) + \hat{b}_1(0, 1) + \hat{b}_1(1, 0) + \hat{b}_1(1, 1) = 1$ can be eliminated by expressing $\hat{b}_1(0, 0)$ in terms of the vector \mathbf{z}_1 and (y_1, y_2) as shown in Table I.

By following the above steps, the distribution $\hat{b}_i(\mathbf{x}^{(i)})$ can be expressed as a linear function of $(\mathbf{z}_i, \{y_j\}_{j \in \mathcal{N}_i})$. Upon this transformation, the free energy $F_B(\mathbf{b}_f, \mathbf{b}_v)$ (19) is expressed in terms of the new variables as

$$\begin{aligned} F_B(\{\mathbf{z}_i\}_{i \in \mathcal{N}}, \{y_i\}_{i \in \mathcal{N}}) & \quad (42) \\ & = \sum_{i=1}^N \left[-(\ln \lambda_i) y_i - \hat{H}_i(\mathbf{z}_i, \{y_j\}_{j \in \mathcal{N}_i}) + (d_i - 1) H_i(y_i) \right], \end{aligned}$$

and the equivalent unconstrained optimization problem is to minimize $F_B(\{\mathbf{z}_i\}_{i \in \mathcal{N}}, \{y_i\}_{i \in \mathcal{N}})$ over the new variables. Technically, there should be additional inequality constraints on these new variables to impose positivity constraints described in (13). However, it can be shown that any stationary point of the BFE given in (19) implicitly satisfies those positivity constraints [28, Remark 4.1 in Page 85].

Now suppose $(\{\mathbf{z}_i^*\}_{i \in \mathcal{N}}, \{y_i^*\}_{i \in \mathcal{N}})$ is a stationary point of (42). Then the gradient of $F_B(\{\mathbf{z}_i\}_{i \in \mathcal{N}}, \{y_i\}_{i \in \mathcal{N}})$ at that stationary point should be zero. In particular, if we take the partial derivative of $F_B(\{\mathbf{z}_i\}_{i \in \mathcal{N}}, \{y_i\}_{i \in \mathcal{N}})$ (42) with respect to the elements of the vector $\mathbf{z}_i = \{z_i(k)\}_{k=1}^{|B|}$, all the terms in (42) other than the term corresponding to \hat{H}_i vanishes. Therefore, setting $\frac{\partial F_B(\{\mathbf{z}_i\}_{i \in \mathcal{N}}, \{y_i\}_{i \in \mathcal{N}})}{\partial z_i(k)} = 0$, we have for $k = 1$ to $|B|$,

$$\left. \frac{\partial \hat{H}_i(\mathbf{z}_i, \{y_j\}_{j \in \mathcal{N}_i})}{\partial z_i(k)} \right|_{(\mathbf{z}_i, \{y_j\}_{j \in \mathcal{N}_i}) = (\mathbf{z}_i^*, \{y_j^*\}_{j \in \mathcal{N}_i})} = 0. \quad (43)$$

Next, we consider the maximum entropy property (20) stated in Lemma 2, and argue that it essentially boils down to the above system of equations (43). Firstly, since the constraints of (20) are same as the constraints required for the Bethe optimization problem (38), they can be eliminated using the variable transformations used in this proof. From (21), it can be observed that all the variable marginals are fixed at

$\{\hat{b}_j^*\}_{j \in \mathcal{N}_i}$, and the optimization (20) is done only over the factor marginals \hat{b}_i . In terms of the transformed variables, it essentially boils down to maximizing the entropy $\hat{H}_i(\mathbf{z}_i, \{y_j\}_{j \in \mathcal{N}_i})$ subject to fixing the variable marginals at $\{y_j^*\}_{j \in \mathcal{N}_i}$, which is captured by (43). This observation essentially asserts that the factor marginals \hat{b}_i^* , and the variable marginals $\{b_j^*\}_{j \in \mathcal{N}_i}$ corresponding to a stationary point $(\mathbf{b}_f^*, \mathbf{b}_v^*)$ are related by the maximum entropy problem defined in (20).

B. Proof of Lemma 4

This proof is a continuation of the proof of Lemma 2. By interpreting the factor and variable marginals $(\mathbf{b}_f, \mathbf{b}_v)$ as linear functions of the variables $(\{\mathbf{z}_i\}_{i \in \mathcal{N}}, \{y_i\}_{i \in \mathcal{N}})$, and setting the partial derivative of $F_B(\{\mathbf{z}_i\}_{i \in \mathcal{N}}, \{y_i\}_{i \in \mathcal{N}})$ in (19) with y_i to zero, we obtain

$$\ln \lambda_i = \frac{\partial \left[(d_i - 1)H_i(b_i) - \sum_{j \in \mathcal{N}_i} \hat{H}_j(\hat{b}_j) \right]}{\partial y_i} \Bigg|_{(b_i, \{\hat{b}_j\}_{j \in \mathcal{N}_i}) = (b_i^*, \{\hat{b}_j^*\}_{j \in \mathcal{N}_i})} \quad (44)$$

Observe the following from steps (ii)-(iv) of the variable transformation in the proof of Lemma 2: The distribution $\hat{b}_j(\rho)$ depends on y_i , if only if the argument is either the all zero pattern, *i.e.*, $(0, 0, \dots, 0)$ or if $\rho_i = 1$, is the only non zero element in that argument. Let us denote this argument as $e^{(i)} := (0, \dots, 0, 1, 0, \dots, 0)$, where 1 is in the i^{th} position. Hence, only two terms of the entropy $\hat{H}_j(\hat{b}_j)$ depend on y_i . (See Table I. For example, only two terms of \hat{b}_2 , namely $\hat{b}_2(1, 0, 0)$ and $\hat{b}_2(0, 0, 0)$ depend on y_1 .)

Using this observation in (44), and simplifying gives us
$$\lambda_i = \left(\frac{(b_i^*(0))^{d_i-1} \prod_{j \in \mathcal{N}_i} \hat{b}_j^*(e^{(i)})}{(b_i^*(1))^{d_i-1} \prod_{j \in \mathcal{N}_i} \hat{b}_j^*(0,0,0,\dots,0)} \right).$$
 Then we conclude the proof of this lemma by applying (23) from Lemma 3, which gives us
$$\lambda_i = \left(\frac{b_i^*(0)}{b_i^*(1)} \right)^{d_i-1} \prod_{j \in \mathcal{N}_i} e^{v_{ji}}.$$

C. Sufficient condition for a stationary point of the BFE

The condition (43) in Lemma 2 is obtained when the partial derivatives of $F_B(\{\mathbf{z}_i\}_{i \in \mathcal{N}}, \{y_i\}_{i \in \mathcal{N}})$ (42) with respect to the elements of the variables $\{\mathbf{z}_i\}$ are set to zero. Similarly, the condition (44) in Lemma 4 are obtained when the partial derivatives of $F_B(\{\mathbf{z}_i\}_{i \in \mathcal{N}}, \{y_i\}_{i \in \mathcal{N}})$ with respect to $\{y_i\}$ are set to zero. Hence, the properties derived in Lemmas 2 and 4 together constitute a sufficient condition for a stationary point of the BFE.

D. Proof of Theorem 2

As discussed in Section II, the global Gibbsian problem (5) essentially solves a system of equations given in (4). From that analogy, it suffices to show that $\beta_i = [\beta_{ij}]_{j \in \mathcal{N}_i}$, the solution of the local Gibbsian optimization problem (8) at a link i , is consistent with the following system of equations:

$$s_j = \sum_{\mathbf{x}^{(i)} \in \mathcal{I}_i : x_j=1} \frac{1}{Z_i} \left(\prod_{k \in \mathcal{N}_i : x_k=1} e^{\beta_{ik}} \right), \quad \forall j \in \mathcal{N}_i, \quad (45)$$

$$\text{where } Z_i = \sum_{\mathbf{x}^{(i)} \in \mathcal{I}_i} \left(\prod_{k \in \mathcal{N}_i : x_k=1} e^{\beta_{ik}} \right). \quad (46)$$

For the conflict graph model, the above equations can be simplified as follows. In the conflict graph model, a link is active if and only if all its neighbours are inactive. Hence, there is only one local feasible schedule $\mathbf{x}^{(i)} \in \mathcal{I}_i$ in which link i is active, namely $\mathbf{x}^{(i)} = (0, \dots, 0, 1, 0, \dots, 0)$ where 1 is in the i^{th} position. Using this observation in (45) with $j = i$, we obtain

$$s_i = \frac{1}{Z_i} e^{\beta_{ii}}. \quad (47)$$

Recall the definition of local feasibility from Section III. Any local schedule $\mathbf{x}^{(i)} \in \mathcal{I}_i$ at a link i , is feasible if that link i is inactive. In other words, if $x_i = 0$ in a local schedule $\mathbf{x}^{(i)} \in \{0, 1\}^{\mathcal{N}_i}$, all the $2^{|\mathcal{N}_i|-1}$ combinations of its neighbours activations are allowed. This implies that for any $j \in \mathcal{N}_i \setminus \{i\}$, the set $\{\mathbf{x}^{(i)} \in \mathcal{I}_i : x_j = 1\}$ has all the $2^{|\mathcal{N}_i|-1}$ possible schedules. Using this observation in (45) gives us

$$s_j = \frac{1}{Z_i} e^{\beta_{ij}} \prod_{k \in \mathcal{N}_i \setminus \{i, j\}} (1 + e^{\beta_{ik}}), \quad j \in \mathcal{N}_i \setminus \{i\}. \quad (48)$$

Similarly, the normalization constant Z_i (46) can be simplified to

$$Z_i = e^{\beta_{ii}} + \prod_{j \in \mathcal{N}_i \setminus \{i\}} (1 + e^{\beta_{ij}}). \quad (49)$$

Note that (47)-(49) characterize the relation between the local fugacities, and the service rates under the conflict graph model. Hence, it sufficient to prove that the local fugacities (28) stated in Theorem 2, are consistent with the set of equations (47)-(49). This step can be verified by simply substituting the local fugacity expressions (28) in (47)-(49).

E. Proof of Lemma 5

The outline of the proof is to show that Algorithm 2 corresponds to the dual subgradient method for (34). First, we compute the Lagrangian, and the dual problem for (34). Considering the equality constraints in (35), the partial Lagrangian of (34) is given by

$$L(y, \{\hat{b}_j\}; \beta) = \theta \sum_j U_j(y_j) + \sum_j H(\hat{b}_j) + \sum_{jk} \beta_{jk} \left(\sum_{\mathbf{x}^{(j)} : x_k=1} \hat{b}_j(\mathbf{x}^{(j)}) - y_k \right). \quad (50)$$

Here $\beta := \{\beta_{jk}\}$ is a short hand notation for the dual variables $\{\beta_{jk}, j = 1 \dots N, k \in \mathcal{N}_j\}$. These dual variables have an interpretation of local fugacities. Hence we abuse the notation by using the same notation for both of them. The dual function is given by $D(\beta) = \sup L(y, \{\hat{b}_j\}; \beta)$ over $y \in [0, 1]^{\mathcal{N}}$ and the distributions $\{\hat{b}_j\}$ on the local schedules. We require the following result (Lemma 6) for completing the proof.

Lemma 6. *The primal and dual solutions of the optimization problem (34) satisfy strong duality. Further, the primal and dual variables are related as follows. For a given dual*

value $\beta = \{\beta_{jk}\}$, the Lagrangian $L(y, \{\hat{b}_j\}; \beta)$ attains its supremum at primal values given by

$$y_j(\beta) = \arg \max_{q \in [0,1]} \theta U_j(q) - q \sum_{k \in \mathcal{N}_j} \beta_{jk}(t), \forall j, \quad (51)$$

$$\hat{b}_j(\mathbf{x}^{(j)}; \beta) = Z_j^{-1} \exp \left(\sum_{k \in \mathcal{N}_j} x_k \beta_{jk}(t) \right), \forall \mathbf{x}^{(j)} \in \mathcal{I}_j, j \in \mathcal{N}. \quad (52)$$

Proof. The utility functions $\{U_i\}$ are concave, and the entropy is strictly concave. Hence, in (34) we are maximizing a strictly concave function with affine constraints. Further, since the rate region Λ_B is non empty, there always exist some $\{b_j\}$ and y such that they are feasible for (34). Hence, the Slater's condition for convex problems with affine constraints [27, Page 226] implies strong duality.

For a given dual variable $\beta = \{\beta_{jk}\}$, let $y_j(\beta)$, $\hat{b}_j(\mathbf{x}^{(j)}; \beta)$ be the corresponding primal variables that maximize the Lagrangian. Then from the structure of the Lagrangian (50), it follows that

$$y_j(\beta) = \arg \max_{q \in [0,1]} \theta U_j(q) - q \sum_{k \in \mathcal{N}_j} \beta_{jk}(t), \forall j.$$

Next, consider the following partial derivative of the Lagrangian (50) to obtain

$$\frac{\partial L(y, \{\hat{b}_j\}; \beta)}{\partial \hat{b}_j(\mathbf{x}^{(j)})} = -\ln \hat{b}_j(\mathbf{x}^{(j)}) - 1 + \sum_{k \in \mathcal{N}_j: x_k=1} \beta_{jk}.$$

Hence, the optimal value should satisfy

$$\hat{b}_j(\mathbf{x}^{(j)}; \beta) \propto \exp \left(\sum_{k \in \mathcal{N}_j} x_k \beta_{jk}(t) \right), \forall \mathbf{x}^{(j)} \in \mathcal{I}_j, \forall j. \quad \square$$

The subgradient for a given dual variable is equal to the residual error in the corresponding primal constraints (See [29, Chapter 2] for details). Specifically, the subgradient at β denoted by $g(\beta) := \{g_{jk}(\beta)\}$ is given by

$$g_{jk}(\beta) = \left(\sum_{\mathbf{x}^{(j)}: x_k=1} \hat{b}_j(\mathbf{x}^{(j)}; \beta) \right) - y_k(\beta).$$

Hence the update rule (31) essentially corresponds to a dual subgradient method for (34). This completes the proof of Lemma 5.

REFERENCES

- [1] P. S. Swamy, R. K. Ganti, and K. Jagannathan, "Adaptive CSMA under the SINR model: Fast convergence through local Gibbs optimization," in *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*. IEEE, 2015.
- [2] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *Automatic Control, IEEE Transactions on*, vol. 37, no. 12, pp. 1936–1948, 1992.
- [3] —, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Transactions on Information Theory*, vol. 39, no. 2, pp. 466–478, 1993.
- [4] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar, "Throughput and fairness guarantees through maximal scheduling in wireless networks," *IEEE Transactions on Information Theory*, vol. 54, no. 2, p. 572, 2008.
- [5] A. Dimakis and J. Walrand, "Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid limits," *Advances in Applied Probability*, pp. 505–521, 2006.
- [6] X. Wu, R. Srikant, and J. Perkins, "Queue-length stability of maximal greedy schedules in wireless networks," in *Proceedings of Information Theory and Applications Inaugural Workshop, 2006*, pp. 6–10.
- [7] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 18, no. 3, pp. 960–972, 2010.
- [8] S. Rajagopalan, D. Shah, and J. Shin, "Network adiabatic theorem: an efficient randomized protocol for contention resolution," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 1. ACM, 2009, pp. 133–144.
- [9] J. Ni, B. Tan, and R. Srikant, "Q-CSMA: Queue-length-based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks," *Networking, IEEE/ACM Transactions on*, vol. 20, no. 3, pp. 825–836, 2012.
- [10] P. Bremaud, *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer, 1999, vol. 31.
- [11] D. Shah, D. N. Tse, and J. N. Tsitsiklis, "Hardness of low delay network scheduling," *Information Theory, IEEE Transactions on*, vol. 57, no. 12, pp. 7810–7817, 2011.
- [12] L. Jiang, M. Leconte, J. Ni, R. Srikant, and J. Walrand, "Fast mixing of parallel Glauber dynamics and low-delay CSMA scheduling," *Information Theory, IEEE Transactions on*, vol. 58, no. 10, pp. 6541–6555, 2012.
- [13] D. Lee, D. Yun, J. Shin, Y. Yi, and S.-Y. Yun, "Provable per-link delay-optimal CSMA for general wireless network topology," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 2535–2543.
- [14] M. M. Halldorsson and T. Tonoyan, "How well can graphs represent wireless interference?" *arXiv preprint arXiv:1411.1263*, 2014.
- [15] D. Qian, D. Zheng, J. Zhang, and N. Shroff, "CSMA-based distributed scheduling in multi-hop MIMO networks under SINR model," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.
- [16] J.-G. Choi, C. Joo, J. Zhang, and N. B. Shroff, "Distributed link scheduling under SINR model in multihop wireless networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 4, pp. 1204–1217, 2014.
- [17] P. S. Swamy, R. K. Ganti, and K. Jagannathan, "Spatial CSMA: A distributed scheduling algorithm for the SIR model with time-varying channels," in *Communications (NCC), 2015 Twenty First National Conference on*. IEEE, 2015, pp. 1–6.
- [18] C. H. Kai and S. C. Liew, "Applications of belief propagation in CSMA wireless networks," *Networking, IEEE/ACM Transactions on*, vol. 20, no. 4, pp. 1276–1289, 2012.
- [19] S.-Y. Yun, J. Shin, and Y. Yi, "CSMA using the Bethe approximation for utility maximization," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 206–210.
- [20] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *Information Theory, IEEE Transactions on*, vol. 51, no. 7, pp. 2282–2312, 2005.
- [21] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 467–475.
- [22] J. Kwak, C.-H. Lee, and D. Y. Eun, "Exploiting the past to reduce delay in csma scheduling: a high-order markov chain approach," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 1. ACM, 2013, pp. 353–354.
- [23] F. Baccelli and C. Singh, "Adaptive spatial Aloha, fairness and stochastic geometry," in *Modeling & Optimization in Mobile, Ad Hoc & Wireless Networks (WiOpt), 2013 11th International Symposium on*. IEEE, 2013, pp. 7–14.
- [24] G. Brar, D. M. Blough, and P. Santi, "Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking*. ACM, 2006, pp. 2–13.
- [25] P. C. Pinto and M. Z. Win, "Communication in a Poisson field of interferers," in *Information Sciences and Systems, 2006 40th Annual Conference on*. IEEE, 2006, pp. 432–437.
- [26] L. Jiang and J. Walrand, "Scheduling and congestion control for wireless and processing networks," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–156, 2010.
- [27] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2009.

- [28] M. J. Wainwright and M. I. Jordan, “Graphical models, exponential families, and variational inference,” *Foundations and Trends® in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008.
- [29] N. Z. Shor, *Minimization methods for non-differentiable functions*. Springer Science & Business Media, 2012, vol. 3.
- [30] L. Jiang, D. Shah, J. Shin, and J. Walrand, “Distributed random access algorithm: Scheduling and congestion control,” *arXiv preprint arXiv:0907.1266*, 2009.