Abstract—Distributed link scheduling algorithms based on carrier sense multiple access and Gibbs sampling are known to achieve throughput optimality, if certain parameters called the fugacities are appropriately chosen. However, the problem of computing these fugacities is NP-hard. Further, the complexity of the existing stochastic gradient descent-based algorithms that compute the exact fugacities scales exponentially with the network size. In this paper, we propose a general framework to estimate the fugacities using regional free energy approximations. In particular, we derive explicit expressions for approximate fugacities corresponding to any feasible service rate vector. We further prove that our approximate fugacities are exact for the class of chordal graphs. A distinguishing feature of our work is that the regional approximations that we propose are tailored to conflict graphs with small cycles, which is a typical characteristic of wireless networks. Numerical results indicate that the proposed methods are quite accurate, and significantly outperform the existing approximation techniques.

Index Terms—Wireless ad hoc network, distributed link scheduling algorithm, CSMA, Gibbs distribution, regional free energy approximations, throughput optimality.

I. INTRODUCTION

INK scheduling algorithms based on CSMA have received renewed attention, since they were recently proven to be throughput optimal [2]–[4], in addition to being amenable to distributed implementation. The central idea lies in sampling the feasible schedules from a product form distribution, using a reversible Markov chain called the Gibbs sampler [5, Ch. 7]. In order to support a feasible service rate vector, certain parameters of the Gibbs distribution called the fugacities are to be appropriately chosen. However, the problem of computing the fugacities for a desired service rate vector is NP-hard [2]. An iterative algorithm based on stochastic approximation is proposed in [2], which asymptotically converges to the exact fugacities. However, the convergence time of this algorithm scales exponentially with the size of the network [6]. Hence, it is not amenable to practical implementation in large networks.

There is a need for effective approximation algorithms to estimate the fugacities. For ease of practical implementation, it is desirable that the complexity of these algorithms does not scale with the network size. In this work, we propose a general framework to estimate the fugacities using regional free energy approximations [7]. Our approximation algorithms are tailored to conflict graphs with small cycles, which is a typical characteristic of wireless networks. We derive explicit expressions for approximate fugacities corresponding to any feasible service rate vector. Since the proposed method does not involve any iterative procedures, it does not suffer from convergence issues. Further, the complexity of our algorithms depends only on the size of the local neighbourhood, and hence does not scale with the network size when the average density of the network is constant. We also prove that our estimated fugacities are exact for the class of chordal graphs.

In the regional approximation framework, the vertices of the conflict graph are divided into smaller subsets called regions, and the Gibbs distribution is approximated using certain locally consistent distributions over these regions. Specifically, the Gibbs free energy, which is a function of the Gibbs distribution, is approximated using a function of these local distributions called the Regional Free Energy (RFE).

In this paper, we propose a general framework to find the approximate fugacities for any given conflict graph, and an arbitrary collection of regions. Our framework involves solving a regional entropy maximization problem, whose solution is then used to find the approximate fugacities. While the approach we propose is applicable to any arbitrary choice of regions, the regional entropy maximization problem may not always be amenable to a distributed solution. Furthermore, the accuracy of the approximation depends strongly on the choice of regions. Therefore, a major challenge in obtaining a practically useful algorithm based on the regional approximation framework lies in judiciously choosing regions to ensure that the resulting CSMA algorithm is both accurate and amenable to distributed implementation.

Spatial network models like the random geometric graphs [8] that are typically used to model the wireless networks contain many short cycles. While the existing approaches based on Belief propagation and the Bethe free energy approximation [9] give considerably good accuracy,
there is a scope to improve the accuracy if we consider regional approximations, which choose regions that explicitly include small cycles. In particular, we consider two choices of regions namely: i) Clique based regions, which improves the error due to 3-cycles, and ii) Clique plus 4-cycle based regions, which improves the error due to both 3, 4-cycles.

Under these choice of regions, we derive a simple closed form solution for the regional maximum entropy problem. In particular, we first prove that the regional maximum entropy problem decouples into local entropy maximization problems over each region. Then, we derive closed form solutions for these local entropy maximization problems. Thus, the approximate fugacities can be explicitly computed without solving any optimization problems. Since we provide explicit formulas for the fugacities, our algorithms are robust to gradual changes in the desired service rates, as well as to changes in the network topology, i.e., one can efficiently re-compute the fugacities for the new topology and service requirements.

We prove that our approximate fugacities are exact for the class of chordal graphs. Further, to understand the performance for other topologies, we consider spatial networks modelled using geometric random graphs, and numerically observe the accuracy of our algorithms. Numerical results indicate that the fugacities obtained by the proposed methods are quite accurate, and significantly outperform the existing approximation techniques [9], [10].

Related Work

An iterative approximation algorithm based on Inverse Generalized Belief Propagation (I-GBP) is proposed in [11]. While the I-GBP considerably improves the performance in the presence of short cycles, it suffers from convergence issues. In particular, it is not always guaranteed to converge, and hence not very reliable [9]. Van Houdt [10] proposed explicit formulas for exact fugacities in chordal graphs. As a property of our proposed clique based approximation, we prove that it results in exact fugacities for chordal graphs. Due to the fact that there is a unique set of (exact) fugacities for a given set of service rates, our clique based approximation formula should essentially boil down to the formula for chordal graphs derived in [10]. This observation is independently proved in [12, Th. 5, Corollary 3]. Further, for non chordal graphs, Van Houdt [10] developed a heuristic approximation called the Local chordal graph approximation. In the numerical analysis, we compare the performance of our algorithms with these local chordal approximations [10].

We remark that a recent independent work [12] has results that are similar to some of the results in our present paper. Specifically, both the papers considered clique based beliefs, and derived approximations for fugacities. In particular, Theorem 3 of this paper is quite similar to [12, Th. 1]. Further, the exactness of Kikuchi approximation is derived for chordal graphs in both the papers. It is also proved in [12, Th. 3], that the Kikuchi approximation is a special case of the size k approximation proposed in [12] with k = n. While there are similarities, there are some notable differences as well, which are described below. The approximation framework, and the formula that we proposed in Theorem 1 and 2 of this paper, are applicable for arbitrary choice of regions and beliefs, while the results in [12] are applicable only when the beliefs are clique based. In addition, we also propose a 4-cycle based approach proposed in our paper which leads to a better accuracy. Van Houdt [12], consider a class of approximations called size k approximations, and propose an iterative formula for counting numbers, which is not considered in our paper.

The rest of the paper is organised as follows. In Section II, we introduce the system model. In Section III, we review the concept of regional free energy approximation. In Section IV, we propose a unified framework to compute the approximate fugacities under arbitrary choice of regions. In Section V, we investigate the accuracy and complexity of different choices of regions. In Section VI, we present our results for clique-based regions. In Section VII, we derive the results corresponding to clique plus 4-cycle based regions. In Section VIII, we present our numerical results, and conclude the paper in Section IX.

II. SYSTEM MODEL AND PROBLEM DESCRIPTION

We consider a single-hop wireless network with N links. We use the letter N to denote the set of all the links in the network. We represent the network using the widely used conflict graph interference model [2], [6]. A conflict graph is an undirected graph \( G(V, E) \), in which each vertex corresponds to a wireless link (Transmitter - Receiver pair), and two vertices share an edge if simultaneous transmissions from the corresponding wireless links result in a collision. For a given link \( i \in V \), the neighbour set \( N_i := \{j : (i, j) \in E\} \) denotes the set of conflicting links.

We consider a slotted time model, and use \( x(t) = [x_i(t)]_{i=1}^N \subseteq \{0, 1\}^N \) to denote the transmission status (or schedule) of the links in the network. Specifically, if a link \( i \) is scheduled to transmit in a given time slot \( t \), then the link is said to be active, and \( x_i(t) \) is set to 1. We assume that an active link can transfer unit data in a given time slot, if none of its conflicting links are active in that slot. We define service rate of a link as the long-term fraction of time that the link is active.

Rate Region: A schedule \( x \) is said to be feasible if no conflicting links are active simultaneously. Hence, the set of feasible schedules is given by \( \mathcal{I} := \{x \in \{0, 1\}^N : x_i + x_j \leq 1, \forall (i, j) \in E\} \). Then the feasible rate region \( \Lambda \), which is the set of all the possible service rates over the links, is the convex hull of \( \mathcal{I} \) given by \( \Lambda := \{\sum_{x \in \mathcal{I}} \alpha x : \sum_{x \in \mathcal{I}} \alpha x = 1, \alpha \geq 0, \forall x \in \mathcal{I}\} \). Next, we describe a basic CSMA algorithm [6].

Basic CSMA: In this algorithm, each link \( i \) is associated with a real-valued parameter \( v_i \in \mathbb{R} \) (referred to as fugacity) which defines how aggressively a link captures the channel. In each time slot, one randomly selected link \( i \) is allowed to update its schedule \( x_i(t) \) based on the information in the previous slot:

- If the channel is sensed busy, i.e., \( \exists j \in N_i \) such that \( x_j(t-1) = 1 \), then \( x_i(t) = 0 \).
- Else, \( x_i(t) = 1 \) with probability \( \frac{\exp(v_i)}{1 + \exp(v_i)} \).

Except for the selected link \( i \), all the other links do not update their schedule, i.e., \( x_j(t) = x_j(t-1), \forall j \neq i \). It can be shown
that the above algorithm induces a Markov chain on the state space of feasible schedules. Further, the stationary distribution is a product-form Gibbs distribution [6] given by
\[
p(x) = \frac{1}{Z(v)} \exp \left( \sum_{i=1}^{N} x_i v_i \right), \quad \forall x \in \mathcal{I} \subseteq \{0, 1\}^N, \tag{1}
\]
where \( v = [v_i]_{i=1}^{N} \), and \( Z(v) \) is the normalization 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., \( p(x_i = 1) \).

Thus, the service rates and the fugacity vector \( v \) are related as follows:
\[
s_i = p_i(1) = \sum_{x \in \mathcal{I} : x_i = 1} Z(v)^{-1} \exp \left( \sum_{i=1}^{N} x_i v_i \right), \quad \forall i \in \mathcal{N}, \tag{2}
\]
where \( p_i(1) \) denotes \( p(x_i = 1) \).

Remark: While this paper considers a discrete-time CSMA, the results are applicable for the continuous time model based CSMA [10] as well, since the stationary distribution of the underlying Markov process has the same form.

Problem Description: The CSMA algorithm can support any service rate in the rate region if appropriate fugacities are used for the underlying Gibbs distribution [2]. We consider the scenario where the links always have packets to send, and each link wants to transmit at a certain target service rate. For a given set of target service rates, we address the problem of computing the corresponding fugacity vector. In principle, the fugacities can be obtained by solving the system of equations in (2). Unfortunately, for large networks, solving these equations is highly intractable since it involves computing the normalization constant \( Z(v) \) which has exponentially many terms. In this work, we propose simple, distributed algorithms to efficiently estimate the fugacities. Our solution is inspired by the well known Regional free energy approximation framework [7] which is reviewed in the next section.

III. REVIEW OF REGIONAL APPROXIMATION FRAMEWORK

We introduce the notion of Regional free energy (RFE), which is useful in finding accurate estimates of the marginals of a distribution like \( p(x) \) in (1). We require the following definitions to introduce RFE [7], [14], [15].

Regions and Counting Numbers: For a given conflict graph \( G(V, E) \), let \( \mathcal{R} \subseteq 2^V \) denote some collection of subsets of the vertices \( V \). These subsets are referred to as regions. Further, assume that each region \( r \in \mathcal{R} \) is associated with an integer \( c_r \) called the counting number of that region. A valid set of regions \( \mathcal{R} \), and the corresponding counting numbers \( \{c_r\} \) should satisfy the following two basic rules: a) Each vertex in \( V \) should be covered in at least one of the regions in \( \mathcal{R} \), i.e., \( \bigcup_{r \in \mathcal{R}} r = V \). b) For every vertex, the counting numbers of all the regions containing it, should sum to 1, i.e.,
\[
\sum_{r \in \mathcal{R} : i \in r} c_r = 1, \quad \forall i \in V. \tag{3}
\]

Regional Schedule: The regional schedule at a region \( r \in \mathcal{R} \), denoted by \( x_r \in \{0, 1\}^r \), is defined as the set of variables corresponding to the transmission status of the vertices in that region, i.e., \( x_r := \{x_k \mid k \in r\} \). Let \( \mathcal{I}_r := \{x_r \mid x_i + x_j \leq 1, \forall (i, j) \in E\} \) denote the set of feasible regional schedules. The schedules in \( \mathcal{I}_r \) are said to be locally feasible at the region \( r \).

Regional Distribution and Entropy: If \( b \) denotes the probability mass function of the random variable \( x = [x_i]_{i=1}^{N} \), then the regional distribution \( b_r \) denotes the marginal distribution of \( b \) corresponding to \( x_r \subset x \). In the special case of a region being a singleton set, i.e., \( r = \{i\} \) for some \( i \in V \), then we denote the corresponding marginal distribution as \( b_i \) instead of \( b_{\{i\}} \). The entropy \( H_r(b_r) := -\sum_{x_r} b_r(x_r) \ln b_r(x_r) \) is called the regional entropy at the region \( r \).

Local Consistency: Let \( \{b_r\}_{r \in \mathcal{R}} \) shortly denoted as \( \{b_r\} \), be some set of regional distributions which may not necessarily correspond to valid marginals of any distribution \( b \). If every two regions \( r, q \in \mathcal{R} \) such that \( r \subset q \), satisfy \( \sum_{x_q \cap x_r} b_q(x_q) = b_r(x_r), \forall x_r \), then the set of distributions \( \{b_r\} \) are said to be locally consistent, and are referred to as pseudo marginals.

Now we use these definitions to introduce RFE. Assume that a valid collection of regions \( \mathcal{R} \), and the corresponding counting numbers are given. We assume that all the singleton sets are present in the collection of the regions \( \mathcal{R} \). In case, this assumption is not satisfied, one can simply add those missing singleton sets with a counting number 0. This assumption is useful in proving Theorem 1. Then the RFE for the CSMA distribution, with a fugacity vector \( v \), is defined as follows.

Definition 1 (Regional Free Energy): Let \( v \) be the fugacity vector of CSMA. Then, given a random variable \( x = [x_i]_{i=1}^{N} \) on the space of feasible schedules \( \mathcal{I} \), and its probability distribution \( b \), the RFE denoted by \( F_{\mathcal{R}}(b; v) \) is defined as
\[
F_{\mathcal{R}}(b; v) = F_{\mathcal{R}}(\{b_r\}; v) = U_{\mathcal{R}}(\{b_r\}; v) - H_{\mathcal{R}}(\{b_r\}). \tag{4}
\]
Here the first term, called the average energy, is given by the following weighted expectation
\[
U_{\mathcal{R}}(\{b_r\}; v) = -\sum_{r \in \mathcal{R}} c_r E_{b_r} \left[ \sum_{j \in r} v_j x_j \right]. \tag{5}
\]
The second term \( H_{\mathcal{R}}(\{b_r\}) \), known as the Regional entropy, is an approximation to the actual entropy \( H(x) \), and is given by
\[
H_{\mathcal{R}}(\{b_r\}) = \sum_{r \in \mathcal{R}} c_r H_r(b_r). \tag{6}
\]
The stationary points\(^2\) of the RFE with respect to the regional distributions \( \{b_r\} \), constrained over the set of pseudo marginals, provide accurate estimates [14] of the marginal distributions of \( p(x) \) in (1). In other words, if the set of pseudo marginals \( \{b_i^r\} \) is a stationary point of the RFE \( F_{\mathcal{R}}(\{b_r\}; v) \), then \( \{b_i^r\} \) correspond to the estimates of the marginal distribution of \( p(x) \) over the respective regions. In particular, \( b_i^r(x_i = 1) \) provides an estimate of the service rate \( s_i \) corresponding to the fugacity vector \( v \).

\(^2\)Here, constrained stationary point refers to a stationary point of the Lagrangian function of the RFE that enforces the local consistency contraints.
IV. FUGACITY ESTIMATION USING REGIONAL APPROXIMATION

In this section, we propose a framework to estimate the fugacities for given service rate requirements. To that end, we introduce the notion of Region approximated fugacities (RAF).

Definition 2 (Region Approximated Fugacities): Consider the RFE $F_R \{b_r\} \forall v$ corresponding to some fugacities $\{v_i\}$. If there is a stationary point $\{b_r^*\}$ of the RFE $F_R \{b_r\} \forall v$, such that its corresponding singleton marginals are equal to the desired service rates $\{s_i\}$, i.e., $b_r^*(x_i = 1) = s_i, \; i = 1 \ldots N$, then the fugacities $\{v_i\}$ are said to be Region approximated fugacities for the given service rates $\{s_i\}$.

We propose a two step approach to compute Region approximated fugacities.

1) Find a set of locally consistent regional distributions $\{b_r\}$ with the following two properties:

(P1.) The marginals corresponding to the singleton regions are equal to the desired service rates, i.e., $b_r(x_i = 1) = s_i, \; i = 1 \ldots N$.

(P2.) There exists a set of fugacities $\{v_i\}_{i=1}^N$ for which, the set of regional distributions $\{b_r\}$ is a stationary point of the corresponding RFE $F_R \{b_r\} \forall v$.

2) Explicitly find those fugacities $\{v_i\}_{i=1}^N$ that satisfy P2.

In Theorem 1, we state that obtaining a set of regional distributions $\{b_r\}$ with the properties P1 and P2, boils down to constrained maximization of the regional entropy.

Theorem 1: Let $\{b_r^*\}$ be some local optimizer of the optimization problem (7). Then $\{b_r^*\}$ satisfies the properties P1 and P2.

$$\begin{align*}
\arg\max_{\{b_r\}} & \sum_{r \in \mathcal{R}} c_r H_r(b_r), \\
\text{subject to} & \; b_r(x_r) \geq 0, \; x_r \in \mathcal{I}_r; \; \sum_{x_r \in \mathcal{I}_r} b_r(x_r) = 1, \; r \in \mathcal{R}, \\
& \sum_{x_q \in \mathcal{I}_q} b_q(x_q) = b_r(x_r), \; x_r \in \mathcal{I}_r, \; \text{and } r, q \in \mathcal{R} \; \text{s.t. } r \subset q, \\
& b_r(1) = s_i, \; i = 1 \ldots N.
\end{align*}$$

Proof: The property (P1) trivially follows from (10). We provide the proof for property (P2) in Appendix A.

Next, in Theorem 2, we propose a formula to compute the Region approximated fugacities.

Theorem 2: Let $\{b_r^*\}$ be a local optimizer of (7). Then the fugacities

$$\exp(\nu_i) = \prod_{r \in \mathcal{R} \mid i \in r} \left( \frac{b_r^*(x_i^r)}{b_r^*(0)} \right)^{c_r}, \; \forall i \in \mathcal{N},$$

are Region approximated fugacities for the desired service rates $\{s_i\}$. Here $x_i^r$ denotes the argument $x_r$ with $x_j = 1, \; x_j = 0, \forall j \in r \setminus \{i\}$, and $0$ denotes the argument with $x_j = 0, \forall j \in r$.

Proof: Proof is provided in Appendix B.

The results derived in Theorem 1 and 2 provide a unified framework for computing the Region approximated fugacities for arbitrary choice of regions. However, one needs to solve the possibly non-convex maximization problem (7), which is a non-trivial problem in general. Hence, in the next section, we first investigate the effect of different choices of regions on the accuracy and complexity of computing RAF. Then, we propose distributed algorithms for some “useful” choices of regions.

V. CHOICE OF REGIONS

The accuracy of the Regional approximation crucially depends on the choice of the regions. In many applications, it is empirically observed that by appropriately increasing the collection of regions, the accuracy is improved [7]. However, for arbitrary choices of regions, the computation of RAF may not be amenable to a distributed implementation. Hence, the major challenge in using the Regional approximation framework for an algorithm like CSMA, is in choosing the regions that are as large as possible, while retaining the property of distributed implementation. For example, if we consider a special case of Regional approximation called the Bethe approximation [7], simple distributed algorithms are known for the fugacities [16], [17].

A. Bethe Approximation

Under the Bethe approximation framework [7], the collection of regions $\mathcal{R}$ includes only the regions of cardinality one and two. Specifically, those are the regions corresponding to the vertices, and the edges of the conflict graph. Hence, the Bethe approximation is easy to implement. However, its accuracy is known to be good only for tree-like topologies. In particular, the Bethe approximation is not the right choice for typical wireless networks that inherently contain many small cycles. For example, Figure 6 in Section VIII shows typical realization of a random geometric graph that is generally used to model a wireless network. It can be observed that those topologies contain small cycles.

To understand the impact of cycles on the accuracy of the Bethe approximation, we consider a ring topology (see Figure 1), and plot the percentage error due to the Bethe approximation [16, Sec. III] as a function of the cycle size in Figure 2. Here, the Bethe approximation error is defined
as the loss in service rate that occurs due to the approximation used in computing the fugacities (precise definition is provided later). It can be observed from Figure 2, that the significant error is mainly due to cycles of size 3, 4. For cycles of size 5 or more, the error is within 5 percent. Hence, if we can include the cycles of size 3, 4 in the collection of regions, the error can be significantly improved. To that end, we discuss a class of regional approximations called the Kikuchi approximations, which can accommodate the cycles in the collection of regions.

B. Kikuchi Approximation

Kikuchi approximation is a special case of regional approximation framework. The Kikuchi approximation framework imposes certain restrictions on how the regions, and the counting numbers can be selected. Specifically,

1) The collection of regions $\mathcal{R}$ should be closed under intersection.
2) The counting numbers should satisfy

$$\sum_{q \in \mathcal{R} | r \subseteq q} c_q = 1, \quad \forall r \in \mathcal{R}. \quad (12)$$

Remark: It can be observed that the constraints (3) imposed by the general Regional approximation are weaker than the constraints (12) imposed by the Kikuchi approximation. In particular, regional approximation imposes the constraints (12) only on the singleton regions as shown in (3).

Cluster Variation Method (CVM) [7] is a well-known approach to obtain a collection of regions $\mathcal{R}$, that satisfy the properties of the Kikuchi approximation framework. In this approach, we start with an initial set of regions $\mathcal{R}_0$ called the maximal regions. Then $\mathcal{R}$ is obtained by including the set of the maximal regions $\mathcal{R}_0$, and all the regions that can be obtained by taking all the possible intersections of the maximal regions. Further, the counting numbers are computed as follows. First, all the maximal regions are assigned a counting number 1. Then the counting numbers of the other regions are iteratively computed using

$$c_r = 1 - \sum_{q \in \mathcal{R} | r \subseteq q} c_q, \quad \forall r \in \mathcal{R}. \quad (13)$$

In the next section, we use this Kikuchi approximation framework to improve the approximation error incurred due to 3-cycles.

VI. CLIQUE BASED REGIONS TO IMPROVE THE ERROR DUE TO 3-CYCLES

Let $\mathcal{R}_0$ be the set of all the maximal cliques in the conflict graph. Let $\mathcal{R}$ denote the set of regions obtained by including the set of the maximal cliques $\mathcal{R}_0$, and all the regions that can be obtained by taking all the possible intersections of maximal cliques. Note that this choice of regions ensures that the error-inducing 3-cycles (which are nothing but the cliques of size 3), are included in the collection of the regions $\mathcal{R}$. Let the counting numbers be obtained by (13).

A. Explicit Formula for RAF

In Theorem 3, we derive explicit formula for the RAF.

Theorem 3: Let $\mathcal{R}_0$ be the set of maximal cliques in the conflict graph. Let $\mathcal{R}$ denote the collection of regions obtained by performing cluster variation method with $\mathcal{R}_0$ as the initial set of regions. Let the counting numbers be defined by (13). Then, for any given service rate requirements $\{s_i\}$, the Region approximated fugacities are given by

$$\exp(\tilde{v}_i) = s_i \prod_{\{r \in \mathcal{R} | i \in r\}} \left(1 - \sum_{j \in r} s_j\right)^{-c_r}, \quad i \in \mathcal{N}. \quad (14)$$

Proof: Consider the optimization problem (7) in Theorem 1. Now, we use the fact that any region $r \in \mathcal{R}$ is a clique, and show that there is a unique set of $\{b_r(x_r)\}$ that satisfy the feasibility constraints (8) - (10) of the optimization problem (7). Let $\{b_r(x_r)\}$ be feasible for (7). Since, every region $r \in \mathcal{R}$ is a clique, from the definition of local feasibility, no more than one vertex can be simultaneously active in a feasible regional schedule $x_r \in \mathcal{I}_r$. Hence, the set of local feasible schedules for a clique region $r$ is given by

$$\mathcal{I}_r = \{x_r\}_{x_r=1} \cup \{0\}.$$ 

where $x_r^j$ is the schedule with $x_j = 1$, $x_k = 0, \forall k \in r \setminus \{j\}$, and 0 is the schedule with $x_k = 0, \forall k \in r$. Then from the local consistency constraint (9) with $r = \{j\}$ and some $q \supset \{j\}$, we have

$$\sum_{x_q \in \mathcal{I}_q, x_j=1} b_q(x_q) = b_j(1),$$

$$b_q(x_q^j) = b_j(1).$$

Then from (10), and the normalization constraint (8), we conclude that

$$b_r(x_r) = \begin{cases} s_j, & \text{if } x_r = x_r^j, \\ 1 - \sum_{k \in r} s_k, & \text{if } x_r = 0, \end{cases} \quad (15)$$

is the only set of regional distributions $\{b_r(x_r)\}$ that satisfy the feasibility constraints (9) - (10). Hence, it is the optimal solution of (7). Substituting (15) in Theorem 2, we obtain the RAF as follows:

$$\exp(\tilde{v}_i) = \prod_{\{r \in \mathcal{R} | i \in r\}} \left(\frac{s_i - \sum_{j \in r} s_j}{1 - \sum_{j \in r} s_j}\right)^{c_r}, \quad \forall i \in \mathcal{N}. \quad$$

Further, from (3), we have $\sum_{\{r \in \mathcal{R} | i \in r\}} c_r = 1$. Hence, the RAF are given by (14). □

Next, we state a corollary of Theorem 3 which gives the RAF corresponding to the Bethe approximation framework. This result is known due to [16], and we present it for the sake of completeness.

Corollary 3: The Region approximated fugacities under the Bethe approximation framework are given by

$$\exp(\tilde{v}_i) = s_i (1 - s_i)^{\mathcal{N}_i - 1} \prod_{j \in \mathcal{N}_i} (1 - s_i - s_j), \quad i \in \mathcal{N}, \quad (16)$$

where the set $\mathcal{N}_i$ denotes the neighbours of the vertex $i$ in the conflict graph.

Proof: Proof trivially follows from (14). □
B. Distributed Algorithm

We now propose a distributed algorithm to estimate the fugacities using cluster variation method with cliques as the maximal regions. Each link in the network can independently execute the algorithm once it obtains the a) target service rates of the neighbours, b) the local one-hop neighbourhood topology (See Figures 3, 4 for an example).

There are mainly two steps in the algorithm at a link \( i \). The first step involves computing the maximal cliques, and their intersections in which the link \( i \) is part of. The next step is to compute its fugacity by using the formula (14). We introduce some notations before we present the algorithm.

Notation: Let \( \mathcal{R}_0^i \) be the collection of maximal cliques of the conflict graph \( G(V,E) \) in which the vertex \( i \) is part of. For example, if we consider the graph in Figure 3, then \( \mathcal{R}_0^i = \{ \{1,2\}, \{2,8,7\}, \{2,3,7\} \} \). Similarly \( \mathcal{R}_1^i = \{ \{2,3,7\}, \{3,5,6,7\}, \{3,4\} \} \). Using the information about the local topology, any standard algorithm like [18] can be used for finding this set of maximal cliques \( \mathcal{R}_0^i \). For every maximal clique \( r \in \mathcal{R}_0^i \), let us associate a counting number \( c_r = 1 \).

Example: Let us consider the conflict graph shown in Figure 3, and compute the fugacity for the vertex 2 using the proposed algorithm. The local topology, and the set of regions at vertex 2 are shown in Figures 4, 5. The set of maximal cliques containing the vertex 2 is \( \mathcal{R}_0^2 = \{ \{1,2\}, \{2,8,7\}, \{2,3,7\} \} \). Considering their intersections we get \( \mathcal{R}_1^2 = \{ \{2,7\}, \{2\} \} \). However, we discard the set \{2\} since it is a proper subset of another region \{2,7\} in the same level. Hence \( \mathcal{R}_1^2 = \{ \{2,7\} \} \). Now, \( \mathcal{R}_2^2 \) is obtained by intersecting \{2,7\} with regions in the previous level \( \mathcal{R}_1^2 \). The only new region we obtain is the singleton set \{2\}, i.e., \( \mathcal{R}_2^2 = \{ \{2\} \} \).

Next, we compute the counting numbers of these regions. As defined earlier, all the maximal cliques \( \mathcal{R}_0^i \) will be given a counting number 1. Further from (17), it can be easily observed that for the set \( r = \{2,7\} \), \( c_r = -1 \), since it has two super sets namely \( q_1 = \{2,8,7\}, q_2 = \{2,3,7\} \) with \( c_{q_1} = 1, c_{q_2} = 1 \). Similarly, since every region in \( \mathcal{R}_0^2 \cup \mathcal{R}_1^2 \) is a super set of the region \( r = \{2\} \), the counting number of

### Algorithm 1 Clique Based Distributed Algorithm at Link \( i \)

**Input:** \( \mathcal{R}_0^i \), service rates \( \{s_j\}_{j \in N_i} \)

**Output:** fugacity \( \hat{\nu}_i \).

1. Consider a variable \( l \) called level, and initialize \( l = 0 \).
2. Obtain \( \mathcal{R}_{l+1}^i \) by intersecting the cliques in level \( l \), with the cliques in levels less than or equal to \( l \), i.e.,

\[
\mathcal{R}_{l+1}^i := \{ q_1 \cap q_2 \mid q_1 \in \mathcal{R}_l^i, q_2 \in \bigcup_{k \leq l} \mathcal{R}_k^i, q_1 \neq q_2 \}.
\]

If there are no intersections, i.e., \( \mathcal{R}_{l+1}^i = \emptyset \), go to Step 6; Else continue.
3. From the set \( \mathcal{R}_{l+1}^i \), discard\(^4\) the following:
   - (i) Cliques which are already present in a previous level, i.e., discard \( r \in \mathcal{R}_{l+1}^i \) if \( r \in \bigcup_{k \leq l} \mathcal{R}_k^i \).
   - (ii) Cliques which are proper subsets of some other cliques in \( \mathcal{R}_{l+1}^i \), i.e., discard \( r \in \mathcal{R}_{l+1}^i \) if there exist any other set \( q \in \mathcal{R}_{l+1}^i \) such that \( r \subset q \).
4. For each clique \( r \in \mathcal{R}_{l+1}^i \), compute

\[
c_r = 1 - \sum_{q \in S(r)} c_q,
\]

where \( S(r) = \{ q \in \bigcup_{k \leq l} \mathcal{R}_k^i \mid r \subset q \} \) is the set of cliques which are super sets of a given set \( r \).
5. Increment \( l \) by 1, and go to step 2.
6. Let \( \mathcal{R}_l^i := \bigcup_{k} \mathcal{R}_k^i \) denote the collection of all the regions computed above. Then the fugacity is computed as

\[
\exp(\hat{\nu}_i) = s_i \prod_{r \in \mathcal{R}_l^i} \left( 1 - \sum_{j \in r} s_j \right)^{-c_r}.
\]

\( r = \{2\} \) is \( c_r = 1 - c_{\{2,7\}} - c_{\{1,2\}} - c_{\{2,8,7\}} - c_{\{2,3,7\}} = -1 \). Hence, the following expression gives the fugacity \( \exp(\hat{\nu}_2) \):

\[
s_2(1 - s_2 - s_7)(1 - s_2) - s_1 s_2 s_7 (1 - s_2 - s_3 - s_7).
\]

Next, we prove that the Region approximated fugacities computed using the clique based approach are exact for a class of graphs called the chordal graphs.

C. Exactness of Clique Based Approach

It is known that the Bethe approximation is exact for tree graphs. In the Kikuchi approximation framework, we have considered a larger collection of regions by including the maximal cliques of the graph. Hence, one may expect the accuracy to improve. Indeed, we confirm this intuition by proving that the clique based Kikuchi approximation is exact for a wider class of graphs called the chordal graphs (Trees are a special case of chordal graphs).

**Definition 4 (Chordal Graph):** A graph is said to be chordal if all cycles of four or more vertices have a chord. Here, a chord refers to an edge that is not part of the cycle but connects two vertices of the cycle.

\(^4\)These regions are discarded because, it can be shown that [15], the counting numbers of such regions will be zero, and moreover they do not impose any additional constraints. Hence, we do not gain anything by adding such regions except that they increase the complexity of subsequent operations.
Theorem 5: Let the conflict graph be a complete graph. Then, the fugacities that exactly support the desired service rates \( \{ s_i \} \) are given by

\[
\exp(v_i) = \frac{s_i}{1 - \sum_{j \in N^c} s_j}, \quad \forall i \in N.
\]

Proof: In a complete graph, the only maximal clique constitutes the whole network. Hence, \( r = N \) is the only region in the collection of regions \( R \). Further, due to (13), the counting number \( c_r = 1 \). Hence, from Theorem 3, the result follows. \( \square \)

VII. IMPROVING THE ERROR DUE TO 3-CYCLES AND 4-CYCLES

In Section VI, we have considered clique based regions to improve the error due to 3-cycles. In this section, we focus on improving the error due to 4-cycles. Specifically, in this 4-cycle based approach, our collection of regions at a vertex \( i \) includes

- All the 4-cycle regions that include vertex \( i \).
- All the cliques that include vertex \( i \).

Further, the counting numbers are computed using (13).

Remark: In this paper, we only deal with 4-cycles that do not have a chord. Whenever, we use the term 4-cycle, we mean a 4-cycle without a chord.

Observe that the above collection of regions does not fall under the standard CVF framework, since our collection of regions may not include all the intersections of the 4-cycle regions. The advantage of proposing this collection of regions is that it results in a distributed solution for fugacities. In particular, as we will prove in Theorem 5, the regional entropy maximization problem (7) which is possibly a non-convex problem, decouples into local entropy maximization problems at each region. Further, these local entropy maximization problems have a closed form solution as stated in Theorem 6. We now formally state these results.

Theorem 5: Let the set of all the 4-cycles, and all the cliques be denoted by \( \mathcal{R}_{4C}, \mathcal{R}_{CI} \) respectively. Let \( R = \mathcal{R}_{4C} \cup \mathcal{R}_{CI} \) be the collection of regions. Let \( \{ b_r^* \} \) be the solution of the regional entropy maximization problem (7). For the clique regions \( r \in \mathcal{R}_{CI} \), \( b_r^* \) is given by (15). For the 4-cycle regions \( r \in \mathcal{R}_{4C} \), \( b_r^* \) is given by the following optimization problem:

\[
\begin{align*}
\arg \max_{b_r} & \quad H_r(b_r), \\
\text{subject to} & \quad b_r(x_r) \geq 0, \quad x_r \in I_r; \quad \sum_{x_r \in I_r} b_r(x_r) = 1, \\
& \quad \sum_{x_r \in I_r, x_i=1} b_r(x_r) = s_i, \quad i \in r.
\end{align*}
\]

Proof: As obtained in the proof of Theorem 3, the regional distributions of the set of cliques \( \{ b_r^* \}_{r \in \mathcal{R}_{4C}} \) are explicitly determined by the service rates \( \{ s_i \} \) as shown in (15). Hence, in the objective of (7), the entropy terms corresponding to the cliques regions are completely determined by the service rates. In other words, the entropy terms corresponding to the 4-cycles are the only terms that are to be considered while maximizing the objective (7). Further, since no clique or another 4-cycle can be a super set of a 4-cycle, from (13), it can be observed that \( c_r = 1 \) for all the 4-cycle regions \( r \in \mathcal{R}_{4C} \). Hence, the optimization problem (7) can be effectively reduced to an optimization problem over \( \{ b_r^* \}_{r \in \mathcal{R}_{4C}} \) as follows:

\[
\arg \max_{\{ b_r^* \}_{r \in \mathcal{R}_{4C}}} \sum_{r \in \mathcal{R}_{4C}} H_r(b_r),
\]

s.t. \( b_r(x_r) \geq 0, \ x_r \in I_r; \sum_{x_r \in I_r} b_r(x_r) = 1, \ r \in \mathcal{R}_{4C}, \)

\[
\sum_{x_q \in I_q, x_r} b_q(x_q) = b_r^*(x_r), \quad x_r \in I_r, \ r \in \mathcal{R}_{CI}, \ q \in \mathcal{R}_{4C}, \ s.t. \ r \subset q.
\]
where \( x_i \) denote the argument with \( x_i = 1, x_j = 0, \forall j \in r \setminus \{i\} \), and \( 0 \) denote the argument with \( x_j = 0, \forall j \in r \).

**Proof:** Proof follows from equations (7) and (8) of [19]. \( \square \)

**Remark:** In Theorem 6, we obtained a closed form expression for the regional distributions corresponding to 4-cycles. We already got closed form expressions for the regional distributions corresponding to cliques in (15). Hence, the RAF (11) under our 4-cycle based approach can be explicitly computed without solving any optimization problems.

To demonstrate our 4-cycle based approach, we consider the widely studied grid topology [20], and derive the RAF explicitly. A 10 \( \times \) 10 grid topology is shown in Figure 8.

**Corollary 6:** Let \( s_i = s, \forall i \in N \) be the service rate requirements for a grid topology. Then the RAF under the 4-cycle based method is given by

\[
\exp(\hat{v}_i) = \begin{cases} 
\frac{(-1 + 4s + \sqrt{1 - 4s + 8s^2})^4}{16(1 - s)s^3}, & \text{if } |N_i| = 4, \\
\frac{(-1 + 4s + \sqrt{1 - 4s + 8s^2})^2}{4s(1 - 2s)}, & \text{if } |N_i| = 3, \\
\frac{2 - 4s}{-1 + 4s + \sqrt{1 - 4s + 8s^2}}, & \text{if } |N_i| = 2.
\end{cases}
\]

**Proof:** We prove the result for a vertex of degree 4. Other cases can be obtained similarly. Consider a vertex \( i \) with degree 4 (for example, vertex 12 in Figure 8). Then the vertex 12 belongs to four 4-cycle regions namely \{2, 3, 12, 13\}, \{12, 13, 22, 23\}, \{1, 2, 11, 12\}, \{11, 12, 21, 22\}. Each of these 4-cycle regions will have a counting number (13) equal to 1. Further, the intersection of these 4-cycle regions results in 4 edge based regions namely \{11, 2\}, \{12, 11\}, \{12, 13\}, \{12, 22\}. Since each edge region has two super sets in the form of 4-cycles, the counting number (13) of each of these edges is equal to \(-1\). Further, the intersection of the edges result in the singleton set containing the considered vertex 12. Since, all the four edges, and the four 4-cycles are super sets for this singleton vertex, the counting number of this singleton set computed using (13) is equal to 1. Now, from Theorems 5, 6, we know the expressions for the optimal regional distributions \( b_{r_i} \) of all the regions. Substituting these expressions in the formula for the RAF (11) derived in Theorem 2 gives the required result. \( \square \)

**Remark on Complexity:** While the proposed approximation schemes are relatively more accurate than the existing Bethe approximation and the local chordal approximations, it comes at the cost of increased complexity. In particular, the complexity of different approximation schemes are as follows. Let \( \Delta \) be the maximum degree of the conflict graph. Then for the Bethe approximation, the complexity of computing the fugacity at a vertex is \( O(\Delta) \). In the local chordal approximation [10], one should extract a maximal chordal subgraph in the local neighbourhood graph, and hence incurs a worst case complexity of \( O(\Delta^3) \). In our clique based approximation scheme, a vertex should compute all the clique regions that it is part of, and compute their counting numbers. This has a worst case complexity of \( O(2^\Delta) \). Further, in the 4-cycle plus clique based approximation, there is an additional complexity of \( O(\Delta^3) \) to compute the 4-cycle regions. A recent independent work [12] optimizes the clique based approximation by proposing a recursive formula (17) of [12], whose complexity is \( O(2^{2\Delta}) \). We remark that this recursive implementation has a relatively better complexity than our current implementation of our Algorithm 1, that incurs \( O(2^{2\Delta}) \) if implemented in the naive way without any optimizations. Since the complexity of these algorithms just depends on the size of the neighbourhood, the complexity does not scale with the network size, when the average density of the network is constant.

**VIII. Numerical Results**

We evaluate the performance of our algorithms for several network topologies, namely the chordal graph, grid topology and some random geometric graphs. We compare the accuracy of our algorithms with the existing Bethe approximation [16], and the Local chordal approximation [10].

**Approximation Error:** We consider symmetric service rate requirements (i.e., the target service rate is same for all the links). We define the approximation error as the average deviation from the target service rate normalized with the target service rate. In particular, for a given target service rate \( s \), the percentage error is given by

\[
\text{Percentage error} = \frac{100}{N} \sum_{i=1}^{N} \frac{|s - s_i|}{s},
\]

where, \( s^a = [s_i]_{i=1}^{N} \) are the service rates obtained by using the approximated fugacities \( \{v_i\}_{i=1}^{N} \). This error is estimated by measuring the empirical service rates obtained by using the approximated fugacities in the CSMA algorithm, which is run for \( 10^7 \) time-slots.

For a given target service rate, we define load as the ratio of the target service rate to the maximum supportable service rate. Since finding the maximum supportable service rate is computationally difficult for large networks, we use \( 1/(\text{size of maximal clique}) \) to approximate it, which is an
upper bound for the same. We vary the load by increasing the target service rate, and plot the percentage error.

**A. Random Graphs**

First, we consider a 100-link random topology shown in Figure 6. It is generated by randomly placing 100 vertices in a square of side length 6. Any two vertices within unit distance are connected by an edge. The maximum, minimum and average degrees are 13, 4 and 7.7 respectively. The maximum size of a clique in this graph is 8. In Figure 7, we also consider a 200-link random graph with maximum, minimum and average degrees given by 10, 1 and 5.3 respectively. The maximum size of a clique in this graph is 6.

We plotted the corresponding approximation errors in Figures 9 and 10. It can be observed that our algorithms perform significantly better than the Bethe approximation [16] and the Local chordal approximation [10] in terms of accuracy.

**In particular, for the 100-link topology, at a load of 0.95, the Bethe approximation has an error of about 12 percent, and the local chordal approximation [10] has an error of about 8 percent. In contrast, both our algorithms incur less than 2 percent error. Further, for the 200-link topology, at a load of 0.95, the Bethe approximation has an error of about 10 percent, and the local chordal approximation has an error of about 4 percent. In contrast, our clique based and 4-cycle based algorithms have an error of about 1.8 and 1.1 percent respectively. We remark that this improvement comes at the cost of additional complexity, whose details are discussed earlier in Section VII.**

**Average Error for Random Geometric Graphs:** To present the average behavior of the error, we considered 50 random geometric graphs of size 100 on a two-dimensional square of side length 6. Any two vertices within unit distance are connected by an edge. The observed average degree of these graphs is between 7 to 8. The observed maximum clique size of these graphs is between 7 and 11.

The required service rates are set to a load of 0.9. We have presented the average error obtained across these topologies. As observed from Table I, our algorithms have a better error performance. Specifically, the Bethe approximation and local chordal approximation has an error of 11.09 and 3.88 percent respectively. Our clique based algorithm has an average error of 2.15 percent, and the 4-cycle based algorithm has an average error of 1.98 percent.

**B. Chordal Graph and Grid Graph**

As proved in Theorem 4, our algorithm is exact if the underlying conflict graph is chordal. This result is verified using several chordal graphs. For the grid topology, we consider a $10 \times 10$ grid graph shown in Figure 8. The corresponding
approximation error is plotted in Figure 11. Since, there are no cliques other than the trivial edges, all the algorithms except the 4-cycle based algorithm give the same solution as the Bethe approximation. As seen from Figure 11, the 4-cycle based algorithm has a better accuracy among all the other algorithms. For example, at a load of 0.7, all the algorithms except the 4-cycle algorithm incur an error of about 30 percent, while our 4-cycle based algorithm has an error of about 10 percent.

C. Approximation Error as a Function of Network Size

We increase the network size, and observe how the error behaves. We observed that there are two possible behaviors depending on whether the density of the network is kept constant or increased, while scaling the network size. In particular, if the network size is increased by increasing the density of the graph, then the approximation error increases. However, the approximation error does not seem to scale with the network size, when the density is kept constant. We consider random geometric graphs of different sizes 25, 75 and 100 under the two settings given below, and plot the average error obtained at a fixed load of 0.9, and over 20 realizations of the random graphs of each size. The plots are shown in Figures 13 and 14. The details of the simulation setting is as follows:

Increasing density: We consider a fixed two-dimensional square of side length 6, and randomly place the vertices in it. Any two vertices within unit distance are connected by an edge. The average degree of the realized graphs are in between 6 to 7 for all the three network sizes.

Constant density: We keep the density of the graph constant by proportionately increasing the area of the two-dimensional square with the network size. In particular, we consider a square of side length 3, $\sqrt{3}$ and 6 for network sizes of 25, 75 and 100 respectively. Any two vertices within unit distance are connected by an edge. The average degree of the realized graphs is in between 6 to 7 for all the three network sizes.

D. Comparison With SGD

Simulation Setting: We run the SGD algorithm [2], with various step sizes and plot the percentage error as a function of the iteration number of the SGD algorithm in Figure 12. In each iteration, to estimate the gradient, the CSMA algorithm is run for 10^7 time-slots. We considered two constant step sizes of 0.1, 0.01, and a variable step size algorithm with step size of $1/t$ for the iteration number $t$. We consider the 100-link random topology shown in Figure 6, and compare the performance of our algorithms with the SGD based algorithm, at load of 0.9. As seen from from Figure 12, after 400 iterations (i.e., $400 \times 10^7$ time-slots of CSMA algorithm), the SGD has about 4 percent error in the best case (i.e., step size=0.1). In comparison, as shown in Figure 9, our approximation algorithms incur less than 2 percent error.

Remark: Although we do not propose a utility maximization algorithm for the regional free energy approximation framework, one can use the “BUM” algorithm in [16] to compute the target service rates that maximize the utility. Then our approximation method can be used to find the fugacities for those target service rates.

IX. CONCLUSIONS

The problem of computing the optimal fugacities for Gibbs sampling based CSMA algorithms is NP-hard. In this work, we proposed a general framework to estimate the
fugacities using the regional free energy approximations. Further, we proposed two useful choices of regions, and derived explicit expressions for the approximate fugacities. We further proved that our approximate fugacities are exact for the class of chordal graphs. A distinguishing feature of our work is that the regional approximations that we proposed are tailored to conflict graphs with small cycles, which is a typical characteristic of wireless networks. Numerical results indicate that the proposed methods are quite accurate, and significantly outperform the existing approximation techniques.

**Appendix**

**A. Proof of Theorem 1**

The proof outline is as follows:

1) We define a Lagrangian function and characterize the stationary points of the RFE.
2) We derive Lagrangian for the optimization problem (7).
3) We compare these two Lagrangian functions to prove the property (P2).

**Remark:** In the proof, \( r \subset q \) denotes that \( r \) is a strict subset of \( q \). The same is applicable for superset.

Let \( \{b_r(x_r)\} \) be a set of locally consistent regional distributions. Then the average energy (5) can be computed as

\[
U_R(\{b_r\}; \mathbf{v}) = -\sum_{r \in R} c_r \mathbb{E}_{b_r} \left[ \sum_{x \in r} v_j x_j \right],
\]

Further, to enforce the constraint \( \sum_{x_r \in I_r} b_r(x_r) = 1 \) for the regional distribution at a region \( r \), we use the Lagrange multiplier \( \gamma_r \). The resulting Lagrangian is

\[
\mathcal{L}(\{b_r\}; \{\lambda_{q,r}(x_r)\}; \{\gamma_r\})
= F_R(\{b_r\}; \mathbf{v}) + \sum_{r \in R} \gamma_r \left( \sum_{x_r \in I_r} b_r(x_r) - 1 \right)
+ \sum_{\{r,q \in R, r \subset q\}} \lambda_{q,r}(x_r) \left( \sum_{x_q \in I_q} b_q(x_q) - b_r(x_r) \right).
\]

(29)

By setting the partial derivative of \( \mathcal{L}(\{b_r\}, \{\lambda_{q,r}(x_r)\}, \{\gamma_r\}) \) with respect to \( \{b_r(x_r)\} \) to zero, we obtain the conditions for stationary points (30)-(32). Specifically, setting the partial derivatives of \( \mathcal{L}(\{b_r\}, \{\lambda_{q,r}(x_r)\}, \{\gamma_r\}) \) with respect to the singleton distributions \( b_i(1) \), and \( b_i(0) \) to zero, gives

\[
-c_r (1 + \ln b_r(x_r)) + \gamma_r - \sum_{\{q \in R \mid \{i\} \subset q\}} \lambda_{q,r}(x_r) = 0,
\]

\[
\forall x_r \in I_r, \ r \in R',
\]

(32)

where \( R' \) denote the set of all the regions \( r \in R \) except the singleton regions.

Now, we consider the optimization problem (7), and define the corresponding Lagrangian. Firstly, in the optimization problem (7), observe that the constraints (10) related to the singleton distributions \( \{b_i\} \) can be absorbed into the local consistency constraints (9). In particular, the constraints

\[
\sum_{x_i \setminus \{x_i\}} b_i(x_q) = b_i(x_i), \ x_i \in \{0, 1\}, \text{ and } q \in R \text{ s.t. } \{i\} \subset q,
\]

\[
b_i(1) = s_i; \ b_i(1) + b_i(0) = 1,
\]

(33)

can be absorbed into the local consistency constraints (33)-(34) as follows:

\[
\sum_{x_q \setminus x_i = 1} b_q(x_q) = s_i, \ q \in R \text{ s.t. } \{i\} \subset q,
\]

(33)

\[
\sum_{x_q \setminus x_i = 0} b_q(x_q) = 1 - s_i, \ q \in R \text{ s.t. } \{i\} \subset q.
\]

(34)

After this modification of the constraints, the optimization problem (7) will not contain the variables corresponding to \( \{b_i\} \). In other words, it contains only \( \{b_r\}_{r \in R'}, \) where \( R' \) denotes the set of regions which includes all the regions \( r \in R \) except for the singleton regions. Note that when we simply
write \( \{b_r\} \), it refers to \( \{b_r\}_{r \in \mathbb{R}} \), i.e., the set of all the regional distributions including the singleton distributions.

Now, we define the Lagrangian \( \mathcal{G} \) for the optimization problem (7). While enforcing the constraints, we replace one of its constraints, namely the constraint (10) with the equivalent modified constraints (33)-(34).

\[
\mathcal{G}(\{b_r\}_{r \in \mathbb{R}'}, \{\lambda_{qr}(x_r)\}, \{\gamma_r\}_{r \in \mathbb{R}'}) = \sum_{r \in \mathbb{R}'} c_r \sum_{x_r \in \mathbb{I}_r} b_r(x_r) \ln b_r(x_r) + \sum_{r \in \mathbb{R}'} \gamma_r \left( \sum_{x_r \in \mathbb{I}_r} b_r(x_r) - 1 \right)
+ \sum_{\{r, q \in \mathbb{R}' \mid r < q\}} \sum_{x_q \in \mathbb{I}_r} \lambda_{qr}(x_r) \left( \sum_{x_q \in \mathbb{I}_r} b_q(x_q) - b_r(x_r) \right)
+ \sum_{q \in \mathbb{R}'}(i \mid \{i\} \subset q) \lambda_{q}(1) \left( \sum_{x_{i} : x_{i} = 1} b_q(x_q) - s_i \right)
+ \sum_{q \in \mathbb{R}'(i \mid \{i\} \subset q)} \lambda_{q} \left( \sum_{x_{i} : x_{i} = 0} b_q(x_q) - (1 - s_i) \right). \tag{35}
\]

Here, the Lagrange multipliers \( \{\lambda_{qr}(x_r)\} \) enforce the local consistency constraints (9), and \( \{\gamma_r\}_{r \in \mathbb{R}'} \) enforce the normalization constraints. For some \( r \in \mathbb{R}' \), setting the partial derivative of \( \mathcal{G}(\{b_r\}_{r \in \mathbb{R}'}, \{\lambda_{qr}(x_r)\}, \{\gamma_r\}_{r \in \mathbb{R}'}) \) with respect to \( b_r(x_r) \) to zero, we obtain

\[
c_r (1 + \ln b_r(x_r)) + \gamma_r - \sum_{q \in \mathbb{R}' \mid r < q} \lambda_{qr}(x_r)
+ \sum_{p \in \mathbb{R}' \mid p < r} \lambda_{rp}(x_p) + \sum_{i \mid \{i\} \subset r} \lambda_{ri}(x_i) = 0, \quad r \in \mathbb{R}'. \tag{36}
\]

Observe that the sum of the last two summations in (36) is equal to \( \sum_{p \in \mathbb{R}' \mid p < r} \lambda_{rp}(x_p) \). Hence, (36) is same as (32).

In other words, we have

\[
\frac{\partial \mathcal{L}(\{b_r\}, \{\lambda_{qr}(x_r)\}, \{\gamma_r\})}{\partial b_r(x_r)} = \frac{\partial \mathcal{G}(\{b_r\}_{r \in \mathbb{R}'}, \{\lambda_{qr}(x_r)\}, \{\gamma_r\}_{r \in \mathbb{R}'})}{\partial b_r(x_r)}
\quad \forall x_r \in \mathbb{I}_r, \quad r \in \mathbb{R}', \tag{37}
\]

where \( \mathcal{L}(\{b_r\}, \{\lambda_{qr}(x_r)\}, \{\gamma_r\}) \) (29) is the Lagrangian corresponding to the RFE.

Now, let \( \{b_r^*\}_{r \in \mathbb{R}'} \) correspond to a local optimal point of the optimization problem (7). Further define

\[
b_r^*(1) = s_i, \quad \forall i \in \mathcal{N},
b_r^*(0) = 1 - s_i, \quad \forall i \in \mathcal{N},
\]

and extend \( \{b_r^*\}_{r \in \mathbb{R'}} \) to \( \{b_r^*\} \). Now, we want to show that \( \{b_r^*\} \) is a stationary point of the RFE (28) for some fugacities. For that, we have to find some fugacities \( \{\tilde{v}_i\} \), and Lagrange multipliers \( \{\lambda_{qr}(x_r)\}, \{\gamma_r\} \) such that \( \{b_r^*\}, \{\lambda_{qr}(x_r)\}, \{\gamma_r\} \) satisfies the stationarity conditions (30) - (32) of the RFE.

Using the fact that \( \{b_r^*\}_{r \in \mathbb{R}'} \) is a stationary point (local optimizer) of the optimization problem (7), we are guaranteed to have Lagrange multipliers \( \{\lambda_{qr}(x_r)\}, \{\gamma_r\}_{r \in \mathbb{R}'} \) such that \( \{\lambda_{qr}(x_r)\}, \{\gamma_r\}_{r \in \mathbb{R}'} \) satisfies the stationarity conditions (36) of the Lagrangian \( \mathcal{G}(\{b_r\}_{r \in \mathbb{R}'}, \{\lambda_{qr}(x_r)\}, \{\gamma_r\}_{r \in \mathbb{R}'}) \). Further due to the equality of the stationary conditions shown in (37), we conclude that \( \{b_r^*\}_{r \in \mathbb{R}'}, \{\lambda_{qr}(x_r)\}, \{\gamma_r\}_{r \in \mathbb{R}'} \) satisfies (32).

Now, inspired by (30), (31), we define

\[
\gamma^*_i = \sum_{q \in \mathbb{R}' \mid \{i\} \subset q} \lambda^*_q(0) - c_i (1 + \ln b^*_i(0)), \quad i \in \mathcal{N}, \tag{38}
\]

\[
\tilde{v}_i = c_i (1 + \ln b^*_i(1)) + \gamma^*_i - \sum_{q \in \mathbb{R}' \mid \{i\} \subset q} \lambda^*_q(1), \quad i \in \mathcal{N}. \tag{39}
\]

Hence, in the light of (37), and the definition of \( \gamma^*_i, \tilde{v}_i \) (38) - (39), it follows that \( \{b_r^*\}, \{\lambda_{qr}(x_r)\}, \{\gamma_r\} \) satisfies all the conditions (30) - (32) required for stationarity of \( \mathcal{L}(\{b_r\}, \{\lambda_{qr}(x_r)\}, \{\gamma_r\}) \) (29). Hence, \( \{b_r^*\} \) is a stationary point of the RFE \( F_{\mathbb{R}}(\{b_r\} ; \tilde{v}) \) (28) for the fugacities \( \{\tilde{v}_i\} \) defined in (39). Therefore, \( \{b_r^*\} \) satisfies the property (P2) defined in Section IV.

**B. Proof of Theorem 2**

Consider the stationarity conditions (30) - (32) of the Lagrangian \( \mathcal{L}(\{b_r\}, \{\lambda_{qr}(x_r)\}, \{\gamma_r\}) \) (29). Subtracting (31) from (30) gives

\[
c_i \ln \left( \frac{b_i(1)}{b_i(0)} \right) = v_i - \sum_{q \in \mathbb{R}' \mid \{i\} \subset q} (\lambda_q(0) - \lambda_q(1)). \tag{40}
\]

Let \( r \supset \{i\} \) be some non-singleton region containing \( i \). Let \( x^i_r \) denote the argument \( x \) such that \( x_i = 1, x_j = 0, \forall j \in r \setminus \{i\} \). Similarly for any \( r \in \mathbb{R} \), let \( 0 \) denote arguments \( x \) such that \( x_i = 0, \forall j \in r \). Then by setting the derivative of \( \mathcal{L}(\{b_r\}, \{\lambda_{qr}(x_r)\}, \{\gamma_r\}) \) given in (29) with respect to \( b_r(x^i_r) \) to zero, we obtain

\[
c_r (1 + \ln b_r(x^i_r)) + \gamma_r - \sum_{q \in \mathbb{R}' \mid r < q} \lambda_{qr}(x^i_r)
+ \sum_{p \in \mathbb{R}' \mid p < r, i \in p} \lambda_{rp}(x^i_p) + \sum_{p \in \mathbb{R}' \mid p < r, i \in p} \lambda_{ri}(x^i_i) = 0. \tag{41}
\]

Similarly, setting the derivative of \( \mathcal{L}(\{b_r\}, \{\lambda_{qr}(x_r)\}, \{\gamma_r\}) \) given in (29) with respect to \( b_r(0) \) to zero, we obtain

\[
c_r (1 + \ln b_r(0)) + \gamma_r - \sum_{q \in \mathbb{R}' \mid r < q} \lambda_{qr}(0)
+ \sum_{p \in \mathbb{R}' \mid p < r, i \in p} \lambda_{rp}(0) + \sum_{p \in \mathbb{R}' \mid p < r, i \in p} \lambda_{ri}(0) = 0. \tag{42}
\]

Now, subtracting (42) from (41) we obtain

\[
c_r \ln \left( \frac{b_r(x^i_r)}{b_r(0)} \right) = - \sum_{q \in \mathbb{R}' \mid r < q} (\lambda_{qr}(0) - \lambda_{qr}(x^i_r))
- \sum_{p \in \mathbb{R}' \mid p < r, i \in p} (\lambda_{rp}(x^i_p) - \lambda_{rp}(0)). \tag{43}
\]

For shorthand notation, for a given \( i \in V \), let us define

\[
\beta_{qr} := (\lambda_{qr}(0) - \lambda_{qr}(x^i_r)), \quad \forall q, r \ni i.
\]
Then (43) can be written as
\[
c_r \ln \left( \frac{b_r(x_r^i)}{b_r(0)} \right) = \sum_{\{q \in \mathcal{R} | i \in q\}} \left( 1(r \supset q) \beta_{rq} - 1(r \subset q) \beta_{qr} \right), \quad \forall r \supset \{i\}. \tag{44}
\]

Next, let us consider the expression \( \sum_{\{r \in \mathcal{R} | i \in r\}} c_r \ln \left( \frac{b_r(x_r^i)}{b_r(0)} \right) \) and split it as
\[
\sum_{\{r \in \mathcal{R} | i \in r\}} c_r \ln \left( \frac{b_r(x_r^i)}{b_r(0)} \right) = c_i \ln \left( \frac{b_i(1)}{b_i(0)} \right) + \sum_{\{r \in \mathcal{R} | r \supset \{i\}\}} c_r \ln \left( \frac{b_r(x_r^i)}{b_r(0)} \right). \tag{46}
\]

If we substitute (40) and (45) in (46), we obtain
\[
\sum_{\{r \in \mathcal{R} | i \in r\}} c_r \ln \left( \frac{b_r(x_r^i)}{b_r(0)} \right) = v_i - \sum_{\{q \in \mathcal{R} | i \in q\}} \beta_{qi} + \sum_{\{r \in \mathcal{R} | r \supset \{i\}\}} \sum_{\{q \in \mathcal{R} | i \in q\}} \left( 1(r \supset q) \beta_{rq} - 1(r \subset q) \beta_{qr} \right),
\]
\[
= v_i + \sum_{\{q \in \mathcal{R} | i \in q\}} \left( 1\{i\} \supset q \beta_{iq} - 1\{i\} \subset q \beta_{qi} \right) + \sum_{\{r \in \mathcal{R} | r \supset \{i\}\}} \sum_{\{q \in \mathcal{R} | i \in q\}} \left( 1(r \supset q) \beta_{rq} - 1(r \subset q) \beta_{qr} \right),
\]
\[
= v_i + \sum_{\{r \in \mathcal{R} | i \in r\}} \sum_{\{q \in \mathcal{R} | i \in q\}} \left( 1(r \supset q) \beta_{rq} - 1(r \subset q) \beta_{qr} \right), \tag{47}
\]

Here, the equality (a) is obtained by simply rewriting the term \(- \sum_{\{q \in \mathcal{R} | i \in q\}} \beta_{qi}\) into the equivalent form
\[
\sum_{\{q \in \mathcal{R} | i \in q\}} \left( 1\{i\} \supset q \beta_{iq} - 1\{i\} \subset q \beta_{qi} \right). \tag{48}
\]

Further, the equality (b) is obtained by observing that all the terms other than \(v_i\) cancel out, and evaluate to zero.

Hence, from (47), we can conclude that if \(b_r^*\) is a stationary point of \(F_{\mathcal{R}}(\{b_r\}; v)\), which is locally consistent, we have
\[
\exp(v_i) = \prod_{\{r \in \mathcal{R} | i \in r\}} \left( \frac{b_r^*(x_r^i)}{b_r^*(0)} \right)^{c_r}, \quad \forall i \in \mathcal{N}.
\]

C. Proof of Theorem 4

We now introduce the notion of a junction tree which is required for this proof.

Definition 7 (Junction Tree): Let \(\mathcal{R}\) denote a given collection of regions, and \(\mathcal{R}_0\) denote the maximal regions of \(\mathcal{R}\). A junction tree \(T = (\mathcal{R}_0, \mathcal{E})\) is a tree in which the nodes correspond to the maximal regions, and the edges are such that they satisfy the following Running intersection property: For any two maximal regions \(r, q \in \mathcal{R}_0\), the elements in \(r \cap q\)

should be part of all the maximal regions on the unique path from \(r\) to \(q\) in the tree \(T\).

It is a known fact that, if we consider a chordal graph with maximal cliques as the collection of regions, then we can construct a junction tree [14, p. 30]. The definition of Junction tree is illustrated with the example of a chordal graph in Figures 15, 16. In Figure 16, the top tree is a valid junction tree, since every pair of regions satisfy the running intersection property. For example, consider the pair of regions \(\{1, 4, 5\}\), \(\{2, 3, 5\}\). The region \(\{1, 2, 5\}\) which is in the path joining them, contains their intersection \(\{5\}\). Similarly, it is easy to observe that the figure in the bottom is not a valid junction tree. For example, consider the pair of regions \(\{1, 2, 5\}, \{2, 3, 5\}\). The region \(\{1, 4, 5\}\) which is in the path joining them, does not contain their intersection \(\{2, 5\}\), and hence violates the running intersection property.

Next, we require the following result from [14, p. 101]: “Consider a collection of regions \(\mathcal{R}\) that is closed under intersection. Assume that there exists a junction tree for the collection \(\mathcal{R}\). Then, it is known that a product form distribution \(p(x)\) of the form given in (1), can be factorized in terms of its exact regional distributions as \(p(x) = \prod_{r \in \mathcal{R}} p_r(x_r)^{c_r}\).”

As discussed earlier, there exists a junction tree representation for the maximal cliques of a chordal graph [14, p. 30]. Hence, using the above result, we can factorize \(p(x)\) defined in (1) in terms of the collection of cliques \(\mathcal{R}\) as follows:
\[
p(x) = \frac{1}{Z(v)} \exp \left( \sum_j w_j x_j \right) = \prod_{r \in \mathcal{R}} p_r(x_r)^{c_r}, \quad \forall x \in \mathcal{I}. \tag{48}
\]

Substituting \(x = 0 \) (i.e., \(x_j = 0\) for all \(j \in \mathcal{N}\)) in (48) gives
\[
Z(v)^{-1} = \prod_{r \in \mathcal{R}} p_r(0)^{c_r}. \tag{49}
\]

Now using the observations made in (15), we obtain
\[
Z(v)^{-1} = \prod_{r \in \mathcal{R}} \left( 1 - \sum_{j \in \mathcal{E}} p_j(1) \right)^{c_r}. \tag{49}
\]

5This factorization is presented in [14, p. 101], and is referred to as Hyper-tree based Re-parametrization. The result is presented using the terminology of mobius function. It can be easily argued that the factorization that we use here is equivalent to that in [14].
Similarly, substituting $x = [x_i]_{i=1}^N$ with $x_i = 1$, $x_j = 0$ for all $j \in N \setminus \{i\}$ in (48), and using (15), we obtain
\[
\exp(v_i) = \frac{\prod_{q \in R_i} (p_{i}(1))^q \prod_{r \in R \setminus R_i} (1 - \sum_{j \in r} p_{j}(1))^{cr}}{Z(v)},
\]
(50)
where $R_i = \{ r \in R | i \in r \}$ is the set of regions in which vertex $i$ is involved.

Now, dividing (50) with (49), we obtain
\[
\exp(v_i) = \frac{\prod_{q \in R_i} (p_{i}(1))^q \prod_{r \in R \setminus R_i} (1 - \sum_{j \in r} p_{j}(1))^{cr}}{\prod_{q \in R_i} (1 - \sum_{j \in r} p_{j}(1))^{cr}}.
\]
Further, from (3), we know that $\sum_{q \in R_i} c_q = 1$. Hence,\[\exp(v_i) = p_{i}(1) \prod_{q \in R_i} (1 - \sum_{j \in r} p_{j}(1))^{cr} .\]

Since the service rate $s_i$ is to be obtained as the marginal $p_{i}(1)$, the exact fugacities and the corresponding service rates are related by the above equation. It can observed that our estimate of fugacities (18) follows the same relation, and hence our algorithm is exact for chordal graphs.

REFERENCES


Peruru Subrahmanya Swamy received the B.Tech. degree in electronics and communication engineering from Sastra University, Thanjavur, India, in 2011. He was a Project Associate with the Analog Devices DSP Learning Center, IIT Madras, Chennai, India, from 2011 to 2012. He was a Graduate Student with the Department of Electrical Engineering, IIT Madras. He is currently a Senior Research Engineer with the Data Science Team, Pramati Technologies, Chennai. His research interests lie in communication networks, optimization, and probabilistic graphical models.

Venkata Pavan Kumar Bellam received the bachelor’s degree in electrical engineering from IIT Madras, Madras, and the master’s degree in computer engineering from Virginia Tech. He also received the Graduate Certificate in data analytics from Virginia Tech. He was with Samsung R&D, Securitii, and Entray over the years. He is currently with Microsoft, Redmond, WA, USA, as a Machine Learning Software Engineer. His research interests lie in machine learning, artificial intelligence, optimization, and communication networks.

Radha Krishna Ganti (S’00–M’10) received the B.Tech. and M.Tech. degrees in electrical engineering from IIT Madras, Chennai, India, and the master’s degree in applied mathematics and the Ph.D. degree in electrical engineering from the University of Notre Dame in 2009. His doctoral work focused on the spatial analysis of interference networks using tools from stochastic geometry. He was a Post-Doctoral Researcher with the Wireless Networking and Communications Group, UT Austin, from 2009 to 2011. He is currently an Assistant Professor with IIT Madras. He has coauthored the monograph Interference in Large Wireless Networks (NOW Publishers, 2008). He received the 2014 IEEE Stephen O. Rice Prize, the 2014 IEEE Leonard G. Abraham Prize, and the 2015 IEEE Communications Society Young Author Best Paper Award.

Krishna Jagannathan received the B.Tech. degree in electrical engineering from IIT Madras, in 2004, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT) in 2006 and 2010, respectively. From 2010 to 2011, he was a Visiting Post-Doctoral Scholar in computing and mathematical sciences with Caltech and an off-campus Post-Doctoral Fellow at MIT. Since 2011, he has been with the Department of Electrical Engineering, IIT Madras, where he is currently an Associate Professor. He was a Consultant with the Mathematical Sciences Research Center, Bell Laboratories, Murray Hill, NJ, USA, in 2005, and an Engineering Intern with Qualcomm, Campbell, CA, USA, in 2007. His research interests lie in the stochastic modeling and analysis of communication networks, transportation networks, network control, and queueing theory. He was a recipient of the Best Paper Award at WiOpt 2013 and the Young Faculty Recognition Award for excellence in Teaching and Research at IIT Madras in 2014. He serves on the Editorial Board of the IEEE/ACM TRANSACTIONS ON NETWORKING and Performance Evaluation.