ISIT'2013 1569714517

CSMA using the Bethe Approximation for Utility Maximization

Se-Young Yun, Jinwoo Shin, and Yung Yi

***Abstract*—CSMA (Carrier Sense Multiple Access), which resolves contentions over wireless networks in a fully distributed fashion, has recently gained a lot of attentions since it has been proved that appropriate control of CSMA parameters guarantees optimality in terms of system-wide utility. Most algorithms rely on the popular MCMC (Markov Chain Monte Carlo) technique, which enables one to find optimal CSMA parameters through iterative loops of simulation-and-update. However, such a simulation-based approach often becomes a major cause of exponentially slow convergence, being poorly adaptive to flow/topology changes. In this paper, we develop a distributed iterative algorithm which produces approximate solutions with convergence in polynomial time. Our approach is motivated by a scheme in statistical physics, referred to as the Bethe approximation, allowing us to express approximate solutions via a certain non-linear system with polynomial size. We provide numerical results to show that the algorithm produces highly accurate solutions and converges much faster than prior ones.**

I. Introduction

A. Motivation

Recently, it has been proved that CSMA, albeit simple and fully distributed, can achieve high performance in terms of system utility by joint scheduling/congestion controls [1]–[4]. These advances show that even an algorithm with no or little message passing can actually be close to the optimal performance, achieving significant progress in terms of algorithmic complexity over the seminal work of Max-Weight [5] and its descendant researches, each of which often takes a tradeoff point between complexity and performance, see [6], [7]. The main idea underlying the recent CSMA developments is to intelligently control access intensities (i.e., access probability and channel holding time) over links so as to let the resulting long-term link service rate converge to the target rate [8].

However, one of the main drawbacks for such CSMA algorithms is slow convergence, which is practically problematic due to its poor adaptivity to network and flow configuration changes. The root cause of slow convergence stems from the fact that all the above algorithms are based on the MCMC (Markov Chain Monte Carlo) technique, where even for a fixed CSMA intensity, it takes long time, called mixing time, to reach the stationary distribution to observe how the system behaves. Note that the mixing time is typically exponentially large with respect to the number of links [9]. Note that to achieve our ultimate goal, the algorithms require to update the CSMA intensities to converge to the target point, and multiple mixing times are necessary before they converge. For the mixing time issue, there exist algorithms updating CSMA intensities before the system is mixed, e.g., without time-scale separation between the intensity update and the time to get the system state for a given intensity update [3], [4]. However, they are not free from the slow convergence issue since their convergence inherently also requires the mixing property of the underlying network Markov process. In summary, all prior CSMA algorithms suffer from slow convergence explicitly or implicitly. The main goal of this paper is to develop 'mixing-independent' CSMA algorithms to overcome the issue at the marginal cost of performance degradation.

This work was supported by the Center for Integrated Smart Sensors funded by the Ministry of Education, Science and Technology as Global Frontier Project (CISS-2012M3A6A6054195).

S. Yun is with School of Electrical Engineering, KTH, Sweden (e-mail: seyoung@kth.se). Y. Yi is with Department of Electrical Engineering, KAIST, Korea (e-mail: yiyung@kaist.edu). J. Shin is with Mathematical Sciences Department, IBM T. J. Watson Research, USA (e-mail: jshin@us.ibm.com).

B. Background and Contribution

We aim at drastically improving the convergence speed by using techniques in statistical physics (instead of the MCMC based ones) for an utility maximization problem. In order to reach the convergent service rates as the solution of the utility maximization problem, the intermediate target service rates should be iteratively updated toward the optimal rates, from which the transmission intensities are consequently updated. The key contribution of our paper lies in our message passing algorithm's ability to directly compute the required access intensity for given target service rates in a distributed manner, rather than estimation-based approaches in the MCMC technique. To understand how our CSMA problems are related to statistical physics, we first present some backgrounds, followed by our main contributions

The CSMA setting can be naturally understood by a certain Markov random field (MRF) [10], which we call CSMA-MRF, in the domain of physics and probability. In CSMA-MRF, links induce a graph where links are represented by vertices and interfering links generate edges. Access intensities over links correspond to MRF-parameters in CSMA-MRF. Then, the service rate of each link is the marginal distribution of the corresponding vertex in CSMA-MRF. In the area of MRFs, free energy concepts such as 'Gibbs free energy' function and 'Bethe free energy' function defined by the graph and MRF-parameter have been studied to compute marginal probabilities in MRFs. For example, it is known that finding a minimum point of a Bethe function can lead to approximated values for marginal distributions, where its empirical success has been widely evidenced in many areas such as computer vision, artificial intelligence and information theory [11]–[13]. The main benefit of this approach is that zero-gradient (non-linear) equations of a free energy function can provide low-complexity (approximate) consistency conditions between marginal probabilities and MRF-parameters.

In this paper, we use Bethe functions to develop a distributed CSMA algorithm, called Bethe Utility Maximization (**BUM**), for the utility maximization problem. We show that it converges in a polynomial number of iterations, which is dramatically faster than prior algorithms based on MCMC. The **BUM** algorithm consists of two phases: the first and second phases aim at computing targeted service rates (i.e., marginal distributions) and corresponding CSMA intensities (i.e., MRF-parameters), respectively. We formulate these computational problems as maximizing (or minimizing) Bethe free energy functions. We show that the Bethe function in the first phase is concave for the popular α-fairness utility functions [14] and develop a distributed gradient algorithm for maximizing it. On the other hand, the Bethe function in the second phase is neither concave or convex, but we show that its local optimum can be computable very efficiently, somewhat surprisingly through one iteration among links. We establish theoretical guarantees for the convergence rate and quantify the output quality of **BUM**. Furthermore, numerical results are provided to compare its performance and that of prior algorithms based on MCMC.

Related works include [15], where the authors studied utility maximization problems using the Belief Propagation (BP) algorithm. BP and Bethe functions are connected as discussed in [11], in that there is an one-to-one correspondence between fixed points of BP and local minima of Bethe functions. However, the BP algorithm often does not converge to the fixed point in some topologies. Our work differs from [15] in that Bethe functions are exploited not to find marginal distributions in CSMA-MRF but to find MRF-parameters given the targeted marginal distributions.

II. System Model

Network model. In a wireless network, each link i, which consists of a transmitter node and a receiver node, shares the wireless medium with its 'neighboring' links that refer to the ones that are interfering with i, i.e., the transmission over i cannot be successful if a transmission in at least one neighboring link occurs simultaneously. We assume that each link has a unit capacity. The interference relationship among links can be represented by a graph $G = (V, E)$, popularly known as the *interference graph*, where links in the wireless network are represented by the set of vertices V, and any two links i, j share an edge $(i, j) \in E$ if their transmissions interfere with each other.

Feasible rate region. We let $\boldsymbol{\sigma}(t) \triangleq [\sigma_i(t) \in \{0,1\} : i \in V]$ [1] denote the *scheduling vector* at time t, i.e., link i is active or transmits packets (if exist) with unit rate at time t if $\sigma_i(t) = 1$ (and does not otherwise). The scheduling vector $\boldsymbol{\sigma}(t)$ is said to be *feasible* if no interfering links are active simultaneously at time t, i.e., $\sigma_i(t) + \sigma_j(t) \le 1, \ \forall (i,j) \in E$. We use $\mathcal{N}(i) \triangleq \{j : (i,j) \in E\}$ to denote the set of the neighboring links of link i, $d(i) \triangleq |\mathcal{N}(i)|$ and $d \triangleq \max_i d(i)$. Then, the set of all feasible schedules $\mathcal{I}(G)$ is given by:

$$\mathcal{I}(G) \triangleq \{\boldsymbol{\sigma} \in \{0,1\}^n : \sigma_i + \sigma_j \le 1, \forall (i,j) \in E\}.$$

The feasible rate region $C(G)$, which is the set of all possible service rates over the links, is simply the convex hull of $\mathcal{I}(G)$, defined as follow:

$$C(G) \triangleq \left\{ \sum_{\boldsymbol{\sigma} \in \mathcal{I}(G)} \alpha_{\boldsymbol{\sigma}} \boldsymbol{\sigma} : \sum_{\boldsymbol{\sigma} \in \mathcal{I}(G)} \alpha_{\boldsymbol{\sigma}} = 1, \, \alpha_{\boldsymbol{\sigma}} \ge 0, \, \forall \boldsymbol{\sigma} \in \mathcal{I}(G) \right\}.$$

CSMA (Carrier Sense Multiple Access). Now we describe a CSMA algorithm which updates the scheduling vector $\boldsymbol{\sigma}(t)$ in a distributed fashion. Initially, the algorithm starts with the null schedule, i.e., $\boldsymbol{\sigma}(0) = \mathbf{0}$. Each link i maintains an independent Poisson clock of unit rate, and when the clock of link i ticks at time t, update its schedule $\sigma_i(t)$ as

- If the medium is sensed busy, i.e., there exists $j \in \mathcal{N}(i)$ such that $\sigma_j(t) = 1$, then $\sigma_i(t^+) = 0$.
- Else, $\sigma_j(t^+) = 1$ with probability $\frac{\exp(r_i)}{\exp(r_i)+1}$ and $\sigma_j(t) = 0$ otherwise.

In above, $r_i > 0$ is called the *transmission intensity* (or simply *intensity*) of link i. The schedule $\sigma_i(t)$ of link i remains unchanged while its clock does not tick.

Under the algorithm, the scheduling process $\{\boldsymbol{\sigma}(t) : t \ge 0\}$ becomes a time reversible Markov process. It is easy to check that its stationary distribution for given $\boldsymbol{r} = [r_i]$ becomes:

$$\pi^{\boldsymbol{r}} = [\pi^{\boldsymbol{r}}_{\boldsymbol{\sigma}} : \boldsymbol{\sigma} \in \mathcal{I}(G)] \text{ where } \pi^{\boldsymbol{r}}_{\boldsymbol{\sigma}} = \frac{\exp\left(\sum_{i \in V} \sigma_i r_i\right)}{\sum_{\boldsymbol{\rho} \in \mathcal{I}(G)} \exp\left(\sum_{i \in V} \rho_i r_i\right)}. \quad (1)$$

In other words, the stationary distribution is expressed as a product form of transmission intensities over links. Then, due to the ergodicity of Markov process $\{\boldsymbol{\sigma}(t)\}$, the long-term service rate of link i is a function of transmission intensity $\boldsymbol{r}$, which is the sum of all stationary probabilities of the schedules where i is active. We denote by $s_i(\boldsymbol{r})$ the service rate of link i, which is

$$s_i(\boldsymbol{r}) = \sum_{\substack{\boldsymbol{\sigma} \in \mathcal{I}(G) \\ : \sigma_i = 1}} \pi^{\boldsymbol{r}}_{\boldsymbol{\sigma}} = \frac{\sum_{\boldsymbol{\sigma} \in \mathcal{I}(G) : \sigma_i = 1} \exp(\sum_{i \in V} \sigma_i r_i)}{\sum_{\boldsymbol{\sigma}' \in \mathcal{I}(G)} \exp(\sum_{i \in V} \sigma'_i r_i)}. \quad (2)$$

For the service rate, link i has utility function $U_i(s_i(\boldsymbol{r}))$ and our main goal is that

$$\textbf{(OPT)} \qquad \max_{\boldsymbol{r}} \quad \sum_{i \in V} U_i(s_i(\boldsymbol{r})). \quad (3)$$

III. Preliminaries: Free Energies for CSMA

Free energy functions. We introduce the free energy functions for CSMA Markov processes for transmission intensity $\boldsymbol{r}$.

Definition 3.1 (Gibbs and Bethe Free Energy): Given a random variable $\boldsymbol{\sigma} = [\sigma_i]$ on space $\mathcal{I}(G)$ and its probability distribution ν, *Gibbs* free energy (GFE) and *Bethe* free energy (BFE) functions denoted by $F_G(\nu; \boldsymbol{r})$ and $F_B(\nu; \boldsymbol{r})$ are defined as:

$$F_G(\nu; \boldsymbol{r}) = \mathcal{E}(\nu; \boldsymbol{r}) - H_G(\nu), \ F_B(\nu; \boldsymbol{r}) = \mathcal{E}(\nu; \boldsymbol{r}) - H_B(\nu),$$

where $\mathcal{E}(\nu; \boldsymbol{r}) = -E[\boldsymbol{r} \cdot \boldsymbol{\sigma}]$, $H_G(\nu) = H(\boldsymbol{\sigma})$, and

$$H_B(\nu) \ = \ \sum_{i \in V} H(\sigma_i) - \sum_{(i,j) \in E} I(\sigma_i; \sigma_j).$$

In above, E, H, and I are the expected value, standard entropy, and mutual information, respectively. BFE can be thought as an approximate function of GFE,[2] where H_B is called the

[1] Let $[x_i : i \in V]$ denote the vector whose i-th element is x_i. For notational convenience, instead of $[x_i : i \in V]$, we use $[x_i]$ in the remaining of this paper.

[2] $F_B(\nu; \boldsymbol{r}) = F_G(\nu; \boldsymbol{r})$ if the interference graph G is a tree.

'Bethe' entropy. We note that in general the energy term $\mathcal{E}(\nu; \boldsymbol{r})$ can have a (different) form other than $-E[\boldsymbol{r} \cdot \boldsymbol{\sigma}]$.

How free energy meets CSMA. The following theorem is a direct adaptation of the known results in literature (cf. [16]).

Theorem 3.1: The stationary distribution $\pi^{\boldsymbol{r}}$ in (1) of the CSMA Markov process with intensity $\boldsymbol{r}$ is the unique minimizer of $F_G(\nu; \boldsymbol{r})$, i.e., $\pi^{\boldsymbol{r}} = \arg\min_\nu F_G(\nu; \boldsymbol{r})$.

Theorem 3.1 provides a variational characterization of $\pi^{\boldsymbol{r}}$ (and thus the service rate vector $[s_i(\boldsymbol{r})]$). Since BFE approximates GFE, the (non-rigorous) statistical physics method suggests that a (local) minimizer or zero-gradient point (if exists) of $F_B(\nu; \boldsymbol{r})$ can approximate $\pi^{\boldsymbol{r}}$ (and $[s_i(\boldsymbol{r})]$). The main advantage of studying BFE (instead of GFE) is that BFE depends only on the first-order marginal probabilities of joint distribution ν, i.e., its domain complexity is significantly smaller than that of GFE.

Specifically, by letting $\boldsymbol{y} = [y_i]$ and $y_i = E[\sigma_i]$, which is the service rate of link i, one can obtain the following expression:

$$F_B(\nu; \boldsymbol{r}) = -\sum_{i \in V} y_i r_i - \sum_{i \in V} \Big[(d(i)-1)(1-y_i)\log(1-y_i) - y_i \log y_i\Big] + \sum_{(i,j)\in E} (1 - y_i - y_j)\log(1 - y_i - y_j). \quad (4)$$

Namely, $F_B(\nu; \boldsymbol{r})$ is represented by service rate (or marginal probability) vector $\boldsymbol{y}$. Thus, without loss of generality, we redefine BFE as a function of $\boldsymbol{y}$ as following: $F_B(\boldsymbol{y}; \boldsymbol{r}) = \mathcal{E}(\boldsymbol{y}; \boldsymbol{r}) - H_B(\boldsymbol{y})$, where $\mathcal{E}(\boldsymbol{y}; \boldsymbol{r}) = -\sum_{i\in V} y_i r_i$ and $H_B(\boldsymbol{y})$ includes the other terms in (4). The underlying domain D_B of F_B is

$$D_B \;=\; \{\boldsymbol{y} : y_i \geq 0, y_i + y_j \leq 1, \text{ for all } (i,j) \in E\}. \quad (5)$$

Hence, a (local) minimizer or zero gradient point $\boldsymbol{y}$ of $F_B(\boldsymbol{y}; \boldsymbol{r})$ under the domain D_B provides a candidate to approximate $[s_i(\boldsymbol{r})]$, i.e., $y_i \approx s_i(\boldsymbol{r})$. It is known [11] that the popular Belief Propagation (BP) algorithm for estimating marginal distributions in MRFs can find such a point $\boldsymbol{y}$ if it converges. To summarize, the advantage of studying BFE instead of GFE is that finding service rates (or marginal distribution) reduces to solving a certain non-linear system $\nabla F_B(\boldsymbol{y}; \boldsymbol{r}) = 0$ or $\nabla \Lambda(\boldsymbol{y}, \cdot) = 0$, where Λ is the Lagrange function of $F_B(\boldsymbol{y}; \boldsymbol{r})$. Furthermore, one can prove that there always exists a solution to $\nabla F_B(\boldsymbol{y}; \boldsymbol{r}) = 0$ using the Brouwer fixed-point theorem.

In general, the service rates estimated by BFE do not coincide with the exact service rates. We formally define the error for this Bethe approach as the maximum difference between the estimated rate and the exact service rate across all links.

Definition 3.2 (Bethe Error): For a given transmission intensity $\boldsymbol{r}$, the Bethe error e_B is defined by:

$$e_B(\boldsymbol{r}) = \max_{\boldsymbol{y}: \nabla F_B(\boldsymbol{y};\boldsymbol{r})=0} \max_{i\in V} |y_i - s_i(\boldsymbol{r})|.$$

The empirical success of BP and the known connection between BFE and BP explain that the Bethe error is usually 'small'.

IV. Utility Maximization

In this section, we present an approximation algorithm for the network utility maximization problem (3). To design a distributed algorithm finding transmission intensity $\boldsymbol{r}$ for (3), the approaches in literature [1], [3], [4], instead, considers the following variant of (3): for $\beta > 0$,

$$\max_{\boldsymbol{r}} \quad \beta \cdot \sum_{i\in V} U_i(s_i(\boldsymbol{r})) + H_G(\pi^{\boldsymbol{r}}). \quad (6)$$

The proposed algorithms [1], [3], [4] converge to the transmission intensities $\boldsymbol{r}$ which is the solution to (6). Since the entropy term $H_G(\pi^{\boldsymbol{r}})$ is bounded above and below, a solution to (6) can provide an approximate solution to (3) if β is large.

In our approach, Bethe entropy $H_B(\boldsymbol{y})$ replaces the Gibbs entropy $H_G(\pi^{\boldsymbol{r}})$ in (6), which is the following optimization problem:

$$\max_{\boldsymbol{y}\in D_B} \quad K_B(\boldsymbol{y}) = \beta \cdot \sum_{i\in V} U_i(y_i) + H_B(\boldsymbol{y}) \quad (7)$$

where the Bethe entropy allows to replace the term $s_i(\boldsymbol{r})$ by a new variable y_i, and the domain constraint D_B given by (5) is necessary to evaluate $H_B(\boldsymbol{y})$. Once (7) is solved, one has to recover $\boldsymbol{r}$ from $\boldsymbol{y}$ such that $s_i(\boldsymbol{r}) = y_i$. To summarize, our algorithm for utility maximization consists of two phases:

1. Run a (distributed) gradient algorithm solving (7) and obtain $\boldsymbol{y}$.
2. Compute a transmission intensity $\boldsymbol{r}$ for the target service rate vector $\boldsymbol{y}$ such that $\nabla F_B(\boldsymbol{y}; \boldsymbol{r}) = 0$.

The algorithm is formally described in the following:

Bethe Utility Maximization: BUM

- Initially, set $t = 1$ and $y_i(1) = 1/4,\ i \in V$. [3]
- Intensity-update based on $\boldsymbol{y}$.
  Obtain $(y_j, j \in \mathcal{N}(i))$ through message passing with the neighbors, and set transmission intensity $r_i(t)$ of link i for time t :
  $$r_i(t) = \log\left(\frac{y_i(t)\big(1-y_i(t)\big)^{d(i)-1}}{\prod_{j\in\mathcal{N}(i)}(1 - y_i(t) - y_j(t))}\right). \quad (8)$$
- $\boldsymbol{y}$-update based on time-varying gradient projection.
  $y_i(t+1)$ is updated for time $t+1$ at each link i:
  $$y_i(t+1) = \left[y_i(t) + \frac{1}{\sqrt{t}} \frac{\partial K_B}{\partial y_i}\bigg|_{\boldsymbol{y}(t)} \right]_*,$$
  where the projection $[\cdot]_*$ is defined as
  $$[x]_* = \begin{cases} c_1(t) & \text{if } x < c_1(t) \\ 1 - c_2(t) & \text{if } x > 1 - c_2(t) \\ x & \text{otherwise} \end{cases},$$
  $$c_1(t) = (100 \cdot \log(e+t))^{-1} \quad \text{and}$$
  $$c_2(t) = (1 - y_i(t) + \max_{j\in\mathcal{N}(i)} y_j(t))/2 + t^{-1/4}/10.$$

---

[3]The initial point can be any feasible point in D_B. The point, $y_i = 1/4$ for all i, is just a feasible point.

Intensity Update. As discussed in Section III, for given transmission intensity $\boldsymbol{r}$, approximate service rates can be obtained by the Bethe free energy function $F_B(\boldsymbol{y};\boldsymbol{r})$ by finding the service rate $\boldsymbol{y}$ such that $\nabla F_B(\boldsymbol{y};\boldsymbol{r}) = 0$. Motivated by it, we propose the intensity update algorithm: for given target service rate $\boldsymbol{y}(t)$, find the transmission intensity $\boldsymbol{r}(t)$ such that $\nabla F_B\big(\boldsymbol{y}(t);\boldsymbol{r}(t)\big) = 0$. It is noteworthy that in other applications, BFE is exploited to estimate the marginal distribution of each vertex. The intensity update algorithm by (8) uses BFE in an opposite way, which is one of main novelties of this work. The following theorem verifies that the intensity update achieve the zero gradient of BFE at the target rate $\boldsymbol{y}(t)$ with one iteration of message passing among links.

Theorem 4.1: $\nabla F_B(\boldsymbol{y};\boldsymbol{r}) = 0$ if and only if for all $i \in V$

$$r_i = \log\left(\frac{y_i\big(1-y_i\big)^{d(i)-1}}{\prod_{j\in\mathcal{N}(i)}(1-y_i-y_j)}\right).$$

From the form (4) of F_B, it is easy to prove Theorem 4.1.

y-update. In the $\boldsymbol{y}$-update phase, each link i updates y_i in a distributed manner based on a gradient-projection method. However, our projection $[\cdot]_*$ is far from a classical projection, where our projection varies over time (see $c_1(t)$ and $c_2(t)$), which our algorithm's convergence and distributed operation critically relies on. We delay the discussion on why and how our special projection contributes to the theoretical performance guarantee of **BUM**, and first present its feasibility of distributed operation. Note that the gradient $\frac{\partial K_B}{\partial y_i}$ in the $\boldsymbol{y}$-update phase is:

$$\begin{aligned}\left.\frac{\partial K_B}{\partial y_i}\right|_{\boldsymbol{y}(t)} &= \beta \cdot U_i'(y_i(t)) - (d(i)-1)\log(1-y_i(t)) \\ &\quad - \log y_i(t) + \sum_{j\in\mathcal{N}(i)} \log(1-y_i(t)-y_j(t)), \qquad (9)\end{aligned}$$

Indeed, this gradient can be easily obtained by the link i via local message passing only with its neighbors.

Performance guarantee. We now establish the theoretical performance guarantee of **BUM** for the popular class of α-fair utility functions [14], i.e., $U_i(x) = \begin{cases}\log x & \text{if } \alpha = 1\\ \frac{x^{1-\alpha}}{1-\alpha} & \text{otherwise}\end{cases}$. The parameter α represents the degree of fairness for the throughput allocation: when $\alpha = 0$, the total link throughput is maximized; $\alpha = 1$ gives the proportional fair allocation when $\alpha \to \infty$, it corresponds to the max-min fairness.

Let $\boldsymbol{y}^* = \arg\max_{\boldsymbol{y}\in D_B} K_B(\boldsymbol{y})$. Theorem 4.2 shows that, for any given α, with sufficiently large β, $K_B(\boldsymbol{y}(t))$ by **BUM** always converges to $K_B(\boldsymbol{y}^*)$ in polynomially large enough time T, where the distance between $K_B(\boldsymbol{y}(t))$ and $K_B(\boldsymbol{y}^*)$ is less than $O\left(\frac{n\log T}{\sqrt{T}}\right)$.

Theorem 4.2: Let μ be a probability distribution on $\{1,\ldots,T\}$, such that $\mu(t) = \frac{t^{-1/2}}{\sum_{s=1}^{T} s^{-1/2}}$ for $t \in \{1,\ldots,T\}$. Then, for $\beta > 2d/\alpha$,

$$E\left[K_B(\boldsymbol{y}^\star) - K_B(\boldsymbol{y}(t))\right] = O\left(\frac{n\log T}{\sqrt{T}}\right), \qquad (10)$$

where the expectations are taken over the distribution μ.

Proof: Please see our technical report [17]. ∎

We give a sketch of the proof. First, we show that, for $\beta > 2d/\alpha$, $K_B(\boldsymbol{y})$ is concave, which might allow to use known convex optimization tools solving (7), such as the interior-point method, the Newton's method, the ellipsoid method, etc. However, these algorithms are not easy to implement in a distributed manner, and it is still far from being clear whether such a simple distributed gradient algorithm as used in **BUM** can solve (7) (in a polynomial number of iterations) since the optimization is 'constrained', i.e., $y_i \geq 0$ and $y_i + y_j \leq 1$ for $(i,j) \in E$. Thus, we carefully design the dynamic projection $[\cdot]_*$, where $\log(t)$ and $t^{1/4}$ enforce $\boldsymbol{y}(t)$ to be strictly inside of D_B. For large enough t, we show that $c_1(t) < y_i(t) < 1 - c_2(t)$, i.e., the algorithm does not hit the 'boundary' of $[\cdot]_*$ anymore, which means that **BUM** acts like a gradient algorithm in 'unconstrained' optimization.

We note that for $\beta > 2d/\alpha$, $\boldsymbol{y}(t)$ always converges to the unique $\boldsymbol{y}^*$, because K_B is a (strictly) concave function. The following theorem bounds the gap between the achieved utility of **BUM** and the maximum utility.

Theorem 4.3: The transmission intensity

$$\boldsymbol{r}^* := \left[\log\left(\frac{y_i^*\big(1-y_i^*\big)^{d(i)-1}}{\prod_{j\in\mathcal{N}(i)}(1-y_i^*-y_j^*)}\right)\right]$$

satisfies

$$\max_{\boldsymbol{r}} \sum_{i\in V} U_i(s_i(\boldsymbol{r})) - \sum_{i\in V} U_i\Big(s_i(\boldsymbol{r}^*)\Big) \leq \sum_{i\in V}\frac{e_B(\boldsymbol{r}^*)}{s_i(\boldsymbol{r}^*)^\alpha} + \frac{n\log 2}{\beta}.$$

Proof: Please see our technical report [17]. ∎

As we mentioned earlier, the Bethe error $e_B(\boldsymbol{r}^*)$ is small[4] empirically in many applications [11]–[13], and then the remaining error term is negligible for large β.

V. Comparison with Prior Approach

A. Prior Approaches

In [1], [3], gradient based algorithms solve (6). In this section, we denote by **JW** and **EJW** (the names are used in [4]) the algorithms in [1] and [3], respectively. Technically, the algorithms take the dual problem of (6) where transmission intensity r_i is Lagrangian multiplier and $U'^{-1}\left(\frac{r_i(t)}{\beta}\right) - s_i(\boldsymbol{r}(t))$ is the gradient of the dual problem (6) for r_i. Thus, transmission intensities are commonly described as the following distributed iterative procedure:

$$r_i(t+1) = r_i(t) + \alpha_i(t)\left(U'^{-1}\Big(\frac{r_i(t)}{\beta}\Big) - s_i(\boldsymbol{r}(t))\right), \qquad (11)$$

where $\alpha_i(t) > 0$ is the step size of link i. In both schemes, $\alpha_i(t) = 1/t$ which guarantees the convergence of $r_i(t)$. However, to update $r_i(t+1)$ as per (11), $s_i(\boldsymbol{r}(t))$ is hard to compute. For the issue, a empirical service rate $\hat{s}_i(t)$ has been used instead of $s_i(\boldsymbol{r}(t))$.

The authors in [1] take a large and increasing length of intervals (i.e., $r_i(t)$ is fixed during each interval) so that $s_i(\boldsymbol{r}(t))$ can be estimated well by its empirical estimation $\hat{s}_i(t)$ at the end of each interval. On the other hand, the authors in [3], with a fixed length of intervals (which does not have to be

[4]In particular, $e_B(\boldsymbol{r}^*) = 0$ if the interference graph is a tree.

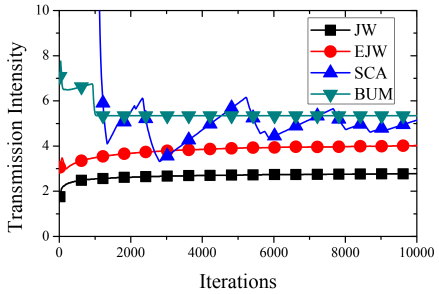

Fig. 1. Intensity traces for the star interference graph with 5 links (i.e., one link interferes with other 4 links which do not interfere with each other). Note that **EJW** and **JW** are increasing until 10000 slots.

very large), use the empirical estimation $\hat{s}_i(t)$. By stochastic approximation, with sufficiently large T,

$$\lim_{t\to\infty} r_i(t+T)-r_i(t) = \sum_{j=t}^{t+T} \alpha(j)\left(U'^{-1}\left(\frac{r_i(j)}{\beta}\right) - s_i(\boldsymbol{r}(j))\right).$$

Both approaches, however, suffer from slow convergence: the updating interval should be extremely large in [1] and $\alpha_i(t)$ should be extremely small in [3] for $\hat{s}_i(t) \approx s_i(\boldsymbol{r}(t))$.

In [4], the authors propose an algorithm called Simulated Steepest Ascent (**SCA**) algorithm converging to the same point with the above two algorithms, where the algorithm is not a gradient based approach but a steepest ascent based algorithm. In SCA scheme, at each iteration t, link i sets transmission intensity as $r_i(t) = \beta U'(\frac{1}{t}\sum_{j=1}^{t}\hat{s}_i(j))$. Then, $\pi_{\boldsymbol{\sigma}}^{\boldsymbol{r}}$ is maximized at $\sigma^* := \arg\max_{\sigma\in\mathcal{I}(G)} \sum_{i\in V} \sigma_i U'\left(\frac{1}{t}\sum_{j=1}^{t}\hat{s}_i(j)\right)$, which is the exact steepest ascent direction. As the steepest ascent algorithms converge to the optimal service rates in many applications, the SCA algorithm makes the service rates converge to the optimal rates quickly, compared to the gradient based algorithms. To guarantee the convergence, however, SCA algorithm may still have to spend extremely large iterations because schedules are stochastically selected over time.

B. Numerical Results

In this section, we provide numerical results in that **BUM** are compared with three prior algorithms introduced in Section V-A regarding to their convergence rates and their achieved network utilities, under the setup $\alpha = 1$ and $\beta = 1$.

Convergence rate. Fig. 1 shows the evolution of transmission intensities over time for a star interference graph. Note that star graphs are tree and, in tree graphs, all of the algorithms have to converge to the same point because $e_B(\boldsymbol{y}) = 0$. We observe that **BUM** converges within 1000 iterations, whereas the other algorithms do not converge even until 10000 iterations.

Network utility. As we stated in Theorem 4.3, **BUM** generates error due to the Bethe approximation on intensity update. However, the error is not significant in our test scenarios. By numerical analysis, we get the network utility when **BUM** is used:-19.9 (for a 5×5 grid interference graph) and -8.1 (for a complete interference graph links). The utility is close to that from the conventional algorithms based on MCMC: -20.6 (for a 5×5 grid interference graph) and -8.05 (for a complete interference graph with 5 links). For the star graph with 5 links, all of the algorithms converge to -3.3. We found that all of the algorithms achieve similar utilities, while **BUM** converges much faster than prior algorithms.

VI. Conclusions

Recently, utility optimal CSMA algorithms are proposed. The simple and distributed MAC protocol can achieve optimality in aggregate utility with just local parameter control. In these algorithms, links iteratively update transmission intensities by their own empirical service rates. However, their convergence speed is often slow because of the stochastic behavior of scheduling. In this paper, we firstly connect Bethe Free Energy (BFE) with CSMA so as to dramatically reduce the convergence speed. With BFE, transmission intensity is represented as a function of target service rates and we propose an utility-maximizing algorithm **BUM** based on the intensity update algorithm using BFE. Since the algorithm does not use empirical values, **BUM** probably converges in polynomial time, where such a guarantee cannot be achievable via prior known schemes.

References

[1] L. Jiang, D. Shah, J. Shin, and J. Walrand, “Distributed random access algorithm: Scheduling and congestion control,” *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 6182 –6207, 2010.

[2] L. Jiang and J. Walrand, “A distributed CSMA algorithm for throughput and utility maximization in wireless networks,” *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 960 –972, 2010.

[3] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and H. V. Poor, “Towards utility-optimal random access without message passing,” *Wiley Journal of Wireless Communications and Mobile Computing*, vol. 10, no. 1, pp. 115–128, 2010.

[4] N. Hegde and A. Proutiere, “Simulation-based optimization algorithms with applications to dynamic spectrum access,” in *Proceedings of CISS*, 2012.

[5] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks,” *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1949, 1992.

[6] X. Lin, N. B. Shroff, and R. Srikant, “A tutorial on cross-layer design in wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 1452–1463, 2006.

[7] Y. Yi and M. Chiang, “Stochastic network utility maximisationa tribute to Kelly’s paper published in this journal a decade ago,” *European Transactions on Telecommunications*, vol. 19, no. 4, pp. 421–442, 2008.

[8] S.-Y. Yun, Y. Yi, J. Shin, and D. Y. Eun, “Optimal CSMA: a survey,” in *Proceedings of ICCS*, 2012.

[9] L. Jiang, M. Leconte, J. Ni, R. Srikant, and J. Walrand, “Fast mixing of parallel Glauber dynamics and low-delay CSMA scheduling,” in *Proceedings of infocom*, 2011.

[10] R. Kindermann, J. Snell, and A. M. Society, *Markov random fields and their applications*, ser. Contemporary mathematics. American Mathematical Society, 1980. [Online]. Available: http://books.google.com/books?id=NeVQAAAAMAAJ

[11] J. S. Yedidia, W. T. Freeman, and Y. Weiss, “Constructing free energy approximations and generalized belief propagation algorithms,” *IEEE Transactions on Information Theory*, vol. 51, pp. 2282–2312, 2005.

[12] J. G. David Forney, “Codes on graphs: News and views,” in *Conference on Information Sciences and Systems*, 2001.

[13] Y. W. K. P. Murphy and M. Jordan, “Loopy belief propagation for approximate inference: an empirical study,” in *In Proceedings of Uncertainty in Artificial Intelligence*, 1999.

[14] J. Mo and J. Walrand, “Fair end-to-end window-based congestion control,” *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556–567, 2000.

[15] C. H. Kai and S. C. Liew, “Applications of belief propagation in CSMA wireless networks,” *IEEE/ACM Transactions on Networking*, vol. 20, no. 4, pp. 1276–1289, 2012.

[16] H. Georgii, *Gibbs Measures and Phase Transitions*, ser. De Gruyter Studies in Mathematics. W. de Gruyter, 1988, no. V. 9. [Online]. Available: http://books.google.com/books?id=3YdI0yww12QC

[17] S. Yun, J. Shin, and Y. Yi, “CSMA using statistical physics toward throughput and utility optimal CSMA,” *ArXiv e-prints*, 2013.