Distributed learning algorithms for optimal scheduling in wireless networks

A Project Report

submitted by

BELLAM VENKATA PAVAN KUMAR
EE12B010

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, MADRAS.

May 2016

THESIS CERTIFICATE

This is to certify that the thesis entitled **Distributed learning algorithms for optimal scheduling in wireless networks**, submitted by **Bellam Venkata Pavan Kumar (EE12B010)**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Radhakrishna Ganti
Research Guide
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: IIT Madras

Date:10/05/2016

ACKNOWLEDGEMENTS

I would like to thank everyone who helped me with this BTech project. I would like to express my sincere gratefulness and gratitude to my guide and mentor Dr. Radhakrishna Ganti for giving me this opportunity to work with him in this area. He has given me freedom and guidance in all aspects which helped me work freely and explore all possible options. Also I would like to thank Prof.Krishna Jagannathan for his support and guidance through out the project.

I will have to mention the invaluable support from Peruru Subrahmanya Swamy(PHD student under Prof.Ganti and Prof.Krishna) who has been of great support and help in technical and other general areas. I feel that the project would not have been this good without his help and support.

I would also like to thank my friends who have helped me through out the project and also during the 4 year stay at institute.

I cannot end this without thanking my parents and my sister,who have been a constant support,helping me in difficult times and have been an immense support through out my life.

ABSTRACT

KEYWORDS: Wireless ad hoc network, Distributed algorithm, CSMA, Gibbs distribution, Bethe approximation, Kikuchi approximation, Throughput optimality

CSMA algorithms based on Gibbs sampling can achieve throughput optimality. Certain parameters of the Gibbs distribution called the attempt rates are to be estimated to support a given set of service rate requirements. However, the problem of computing these attempt rates is NP-hard. Further, the existing algorithms that estimate the attempt rates suffer from an impractically slow convergence. Inspired by the well-known Kikuchi approximation, we propose a simple distributed algorithm, which obtains closed form estimates for the attempt rates. We also prove that our algorithm is exact for a class of graphs, called the chordal graphs. Numerical results suggest that the proposed algorithm outperforms the existing Bethe approximation based algorithms for spatial networks.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

CSMA	Carrier Sense Multiple Access
I-GBP	Inverse Generalized Belief Propagation
KFE	Kikuchi Free Energy

CHAPTER 1

INTRODUCTION

Recently, Carrier Sense Multiple Access (CSMA) algorithms have received a lot of focus since they are amenable to distributed implementation, and are proved to be throughput optimal, *ie*, they can support any service rate in the rate region [2]. The central idea lies in sampling the feasible schedules from a product form distribution, using a reversible Markov chain called the Gibbs sampler [2]. In order to support a given service rate vector, certain attempt rate parameters of the CSMA algorithm called the fugacities are to be appropriately chosen. However, the problem of computing the fugacities for a given service rate vector is NP-hard [2]. This work proposes an efficient closed form approximation for this problem.

1.1 Related work

A stochastic approximation based iterative algorithm 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 [1]. Hence, it is not amenable to practical implementation in large networks. In [11, 5, 6] the fugacities are approximated using a well-known variational technique called the Bethe approximation [9]. This approximation scheme gives exact solutions when the underlying conflict graph is a tree. However, for spatial networks, which inherently contain short loops, the performance of the Bethe approximation is known to degrade [9].

An iterative approximation algorithm based on inverse generalized belief propagation (I-GBP) is proposed in [3]. While the I-GBP considerably improves the performance in the presence of short loops, it suffers from convergence issues. In particular, it is not always guaranteed to converge, and hence not very reliable [11].

1.2 Our Contribution

We make two contributions in this paper. Firstly, we derive closed form estimates of the fugacities based on the Kikuchi approximation framework [4, 10]. Secondly, we prove that our estimates are exact for a class of graphs called the Chordal graphs. In terms of accuracy, it can be shown that our algorithm gives the same solution as the I-GBP algorithm, and hence retains the good performance in the presence of short loops. Further, unlike the I-GBP, it does not suffer from convergence issues since we provide a closed form solution. We evaluate the performance of our algorithm for spatial networks modeled using geometric random graphs. Numerical results suggest that the proposed algorithm outperforms the existing algorithms based on the Bethe approximation.

We would like to remark that a recent work [8] proposed closed form expressions for the fugacities for chordal graphs. As our algorithm is exact for chordal graphs, our works establishes an interesting connection between the result in [8] and the Kikuchi approximation framework.

CHAPTER 2

SYSTEM MODEL AND PROBLEM DESCRIPTION

We consider a single-hop wireless network with N links, and represent it using the widely used conflict graph interference model [2, 1]. Given a conflict graph $G(V, E)$, each vertex in the graph 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 $\mathcal{N}_i := \{j : (i, j) \in E\}$ denotes the set of links that conflict with it. We consider a slotted time model, and use $\mathbf{x}(t) = [x_i(t)]_{i=1}^N \subseteq \{0, 1\}^N$ 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 there is no collision.

2.1 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 $\mathcal{I} := \{\mathbf{x} \in \{0, 1\}^N \ : \ x_i + x_j \leq 1, \ \forall (i, j) \in E\}$. Then the feasible rate region Λ, which is the set of all the possible service rates over the links, is the convex hull of $\mathcal{I}$ given by $\Lambda := \{\sum_{\mathbf{x} \in \mathcal{I}} \alpha_{\mathbf{x}} \mathbf{x} : \sum_{\mathbf{x} \in \mathcal{I}} \alpha_{\mathbf{x}} = 1, \alpha_{\mathbf{x}} \geq 0, \forall \mathbf{x} \in \mathcal{I}\}$. Next, we describe the basic CSMA algorithm [1].

2.2 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.*, $\exists j \in \mathcal{N}_i$ 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 schedules. Further, the stationary distribution is a product-form Gibbs distribution [1] given by

$$p(\mathbf{x}) = \frac{1}{Z(\mathbf{v})} \exp\left(\sum_{i=1}^{N} x_i v_i\right), \qquad \forall \mathbf{x} \in \mathcal{I} \subseteq \{0,1\}^N, \tag{2.1}$$

where $\mathbf{v} = [v_i]_{i=1}^N$, and $Z(\mathbf{v})$ is the normalization constant. Note that $p(\mathbf{x}) = 0$ if $\mathbf{x}$ is not a feasible schedule. 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 $\mathbf{v}$ are related as follows:

$$s_i = p_i(1) = \sum_{\mathbf{x}\,:\,x_i=1} Z(\mathbf{v})^{-1} \exp\left(\sum_{i=1}^{N} x_i v_i\right), \quad \forall i \in \mathcal{N}, \tag{2.2}$$

where $p_i(1)$ denotes $p(x_i = 1)$.

2.3 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 know their target service rates, and address the problem of computing the corresponding fugacity vector. In principle, the fugacities can be obtained by solving the system of equations in (2.2). Unfortunately, for large networks, solving these equations is highly intractable since it involves computing the normalization constant $Z(\mathbf{v})$ which has exponentially many terms. In this work, we provide a simple, distributed algorithm to efficiently estimate the fugacities.

Our solution is inspired by the well known Kikuchi approximation frame work [9] which is useful in estimating the marginals of a product distribution. It is worth nothing that Kikuchi approximation framework is generally useful in computing the service rates for given fugacities. The novelty of our work lies in using this framework to solve the reverse problem of computing the fugacities for a given service rate vector.

CHAPTER 3

DISTRIBUTED ALGORITHM FOR FUGACITIES

In this section, we propose a distributed algorithm to estimate the fugacities. Each link in the network can independently execute the algorithm once it obtains the a) target service rates of the neighbours, b) the local neighbourhood topology, *i.e.*, which of its neighbours share an edge among themselves in the conflict graph.

There are mainly two steps in the algorithm at a link i. The first step involves computing the maximal cliques[1], and their intersections in which the link i is part of. The next step is to compute its fugacity by using the formula (3.2). We introduce some notations before we present the algorithm.

3.1 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 3.1, then $\mathcal{R}_0^2 = \{\{1, 2\}, \{2, 8, 7\}, \{2, 3, 7\}\}$. Similarly $\mathcal{R}_0^3 = \{\{2, 3, 7\}, \{3, 5, 6, 7\}, \{3, 4\}\}$. Using the information about the local topology, any standard algorithm like [7] 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 constant $c_r = 1$.

[1]Note that, this overhead of computing the maximal cliques is involved in the I-GBP algorithm [3] too.

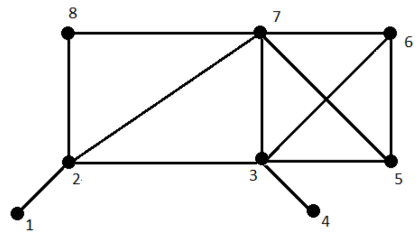

Figure 3.1: Example of a small conflict graph consisting of 8 labeled nodes .

3.2 Algorithm

Distributed algorithm at link i

Input: $\mathcal{R}_0^i$, service rates $\{s_j\}_{j\in\mathcal{N}_i}$;   *Output:* fugacity $\tilde{v}_i$.

1. Initialize $l = 0$.
2. Consider the set of all the intersections of the cliques in $\mathcal{R}_l^i$ to obtain $\mathcal{R}_{l+1}^i := \{q_1 \cap q_2 \mid q_1, q_2 \in \mathcal{R}_l^i, q_1 \neq q_2\}$. If there are no intersections, *i.e.*, $\mathcal{R}_{l+1} = \Phi$, go to Step 6; Else continue.
3. From the set $\mathcal{R}_{l+1}^i$, discard the 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
$$c_r = 1 - \sum_{q\in\mathcal{S}(r)} c_q, \tag{3.1}$$
where $\mathcal{S}(r) = \{q \in \cup_{k\leq l}\mathcal{R}_k^i \mid r \subset q\}$ 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 $\mathcal{R}^i := \cup_k \mathcal{R}_k^i$ denote the collection of all the regions computed above. Then the fugacity is computed as
$$\exp(\tilde{v}_i) = s_i \prod_{r\in\mathcal{R}^i} \Big(1 - \sum_{j\in r} s_j\Big)^{-c_r}. \tag{3.2}$$

3.3 Complexity:

The worst case complexity incurred by a link to compute the corresponding maximal cliques $\mathcal{R}_0^i$ is $O(3^{d/3})$, where d is the the maximum degree of the graph [7]. Hence, for spatial networks, where the degree of the graph does not scale with the network size, our algorithm computes the fugacities with $O(1)$ complexity.

3.4 Information exchange:

The information exchange required for our algorithm is very limited, since the algorithm is fully distributed except for obtaining the local topology information and neighbours service rates.

3.5 Simple example:

Let us consider the conflict graph shown in Figure 3.1, and compute the fugacity for the link 2 using the proposed algorithm. Then the set of maximal cliques containing the vertex 2, will be $\mathcal{R}_0^2 = \{\{1,2\},\{2,8,7\},\{2,3,7\}\}$. 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 $\{2,7\}$. Hence $\mathcal{R}_1^2 = \{\{2,7\}\}$. From (3.1), it can be easily observed that for the set $\{2,7\}$, $c_r = -1$, since it has two super sets namely $q_1 = \{2,8,7\},\ q_2 = \{2,3,7\}$ with $c_{q_1} = 1, c_{q_2} = 1$. As there are no further intersections to be taken in $\mathcal{R}_1^2$, the following expression gives the fugacity $\exp(\tilde{v}_2)$:

$$\frac{s_2(1-s_2-s_7)}{(1-s_1-s_2)(1-s_2-s_8-s_7)(1-s_2-s_3-s_7)}.$$

Our claim is that the fugacities computed in (3.2) are good estimates of the exact fugacities. We justify our claim using the Kikuchi approximation framework.

CHAPTER 4

REVIEW OF KIKUCHI APPROXIMATION

In this section, we introduce the notion of Kikuchi free energy (KFE), which is useful in finding accurate estimates of the marginals of a distribution like $p(x)$ in (2.1). We require the following definitions to introduce KFE [9], [4], [10].

4.1 Regions and Counting numbers:

For a given conflict graph $G(V, E)$, let $\mathcal{R} \subset 2^V$ 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.*, $\{r \in \mathcal{R} | i \in r\} \neq \Phi$, b) For every vertex, the counting numbers of all the regions containing it, should sum to 1, *i.e.*, $\sum_{\{r\in\mathcal{R}|i\in r\}} c_r = 1, \forall i \in V$.

4.2 Regional schedule:

The regional schedule at a region $r \in \mathcal{R}$, denoted by $\mathbf{x}_r \in \{0, 1\}^r$, is defined as the set of variables corresponding to the transmission status of the vertices in that region, *i.e.*, $\mathbf{x}_r := \{x_k \mid k \in r\}$.

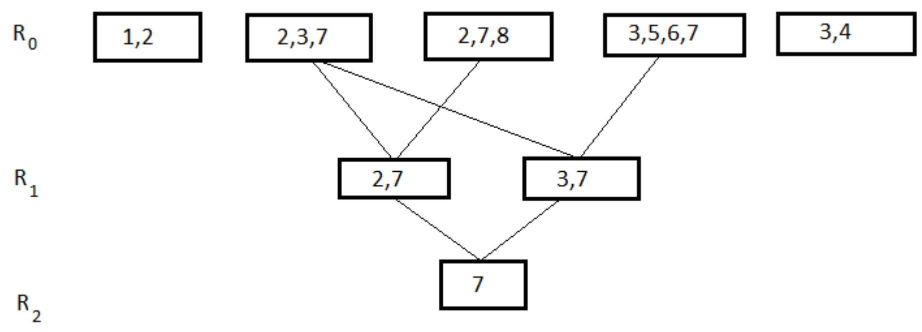

Figure 4.1: Region graph containing regions for conflict graph in Figure 3.1.

4.3 Regional distribution:

If b denotes the probability mass function of the random variable $\mathbf{x} = [x_i]_{i=1}^{N}$, then the regional distribution b_r denotes the marginal distribution of b corresponding to $\mathbf{x}_r \subset \mathbf{x}$. In the special case of a region being a singleton set, *i.e.*, $r = \{i\}$ for some i, then we denote the corresponding marginal distribution as b_i instead of $b_{\{i\}}$.

4.4 Local consistency:

Let $\{b_r\}$ be some set of distributions which may not necessarily come from a distribution b. If every two regions $r, q \in \mathcal{R}$ such that $r \subset q$, satisfy $\sum_{x_q \setminus x_r} b_q(x_q) = b_r(x_r), \forall x_r$, then the set of distributions $\{b_r\}$ are said to be locally consistent, and are referred to as pseudo marginals.

4.5 Region graph

The set of regions $\mathcal{R}$ constitute the vertices of the region graph as shown in Figure 4.1, and two regions $r \subset q \in \mathcal{R}$ share an edge if there does not exist any other region $s \in \mathcal{R}$ such that $r \subset s \subset q$. This condition for edges is useful in removing some redundant equations.

4.6 Kikuchi Free Energy

Assume that a valid collection of regions $\mathcal{R}$, and the corresponding counting numbers are given.[1] Then the KFE for the CSMA distribution, with a fugacity vector $\mathbf{v}$, is defined as follows.

Definition 1. *(Kikuchi Free Energy) Let* $\mathbf{v}$ *be the fugacity vector of CSMA. Then, given a random variable* $\boldsymbol{x} = [x_i]_{i=1}^N$ *on the space of feasible schedules* $\mathcal{I}$*, and its probability distribution* b*, the KFE denoted by* $F_K(b;\boldsymbol{v})$ *is defined as*

$$F_K(b;\boldsymbol{v}) = F_K(\{b_r\};\boldsymbol{v}) = U_K(\{b_r\};\boldsymbol{v}) - H_K(\{b_r\}), \tag{4.1}$$

where the first term, called the average energy, is given by the following weighted expectation

$$U_K(\{b_r\};\boldsymbol{v}) = -\sum_{r\in\mathcal{R}} c_r \mathbb{E}_{b_r}\Big[\sum_{j\in r} v_j x_j\Big],$$

and the term $H_K(\{b_r\})$*, known as the* Kikuchi entropy*, is an approximation to the actual*

[1]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.

entropy $H(\boldsymbol{x})$, and is given by

$$H_K(\{b_r\}) = -\sum_{r\in\mathcal{R}} c_r \left(\sum_{\boldsymbol{x}_r} b_r(\boldsymbol{x}_r) \log b_r(\boldsymbol{x}_r) \right). \tag{4.2}$$

The stationary points of the KFE with respect to the regional distributions $\{b_r\}$, constrained over the set of psuedo marginals, provide accurate estimates [9] of the marginal distributions of $p(\mathbf{x})$ in (2.1). In other words, if the set of psuedo marginals $\{b_r^*\}$ is a stationary point of the KFE $F_K(\{b_r\}; \mathbf{v})$, then they correspond to the estimates of the marginal distribution of $p(x)$ over the respective regions. In particular, $b_i^*(x_i = 1)$ provides an estimate of the service rate s_i corresponding to the fugacity vector $\mathbf{v}$.

CHAPTER 5

MAIN RESULTS

In this section, we state our main theorem which explains the connection between our distributed algorithm for the fugacities, and the Kikuchi approximation. Recall that $\mathcal{R}^i$ is the set of cliques computed in the distributed algorithm at link i. Then, it can be easily verified that the set of regions given by $\mathcal{R} := \cup_{i=1}^{N} \mathcal{R}^i$, with $\{c_r\}$ computed in (3.1), constitute a valid collection of Kikuchi regions[1], and counting numbers. Further, the set of local consistency conditions can be captured using a region graph with the cliques in $\mathcal{R}$ as its vertices. For example, Figure 4.1 shows the region graph corresponding to the conflict graph considered in Figure 3.1.

The non trivial step of our algorithm is the formula (3.2) proposed for the fugacities. The following theorem justifies the proposed formula. If CSMA algorithm employs the set of fugacities $\{\tilde{v}_i\}_{i=1}^{N}$ estimated in (3.2), then the service rates obtained by the CSMA, are close to the desired service rates. This is because, the desired service rates correspond to stationary point of the KFE defined by $\{\tilde{v}_i\}_{i=1}^{N}$ as stated below.

Theorem 1. *Let $\tilde{\mathbf{v}}$ be a set of fugacities that define the Kikuchi free energy $F_K(\{b_r\}; \tilde{\mathbf{v}})$. A locally consistent set of pseudo marginals $\{b_r\}$ will be a stationary point of the KFE if and only if $\sum_{j \in r} b_j(1) < 1, \forall r \in \mathcal{R}$, and*

$$\exp(\tilde{v}_i) = b_i(1) \prod_{r \in \mathcal{R}^i} \Big(1 - \sum_{j \in r} b_j(1)\Big)^{-c_r}, \ \forall i, \tag{5.1}$$

[1] In fact, the regions computed in this paper are inspired by the cluster variation method [10], which is a standard algorithm to generate valid regions.

where $b_i(x_i), b_j(x_j)$ are the regional distributions corresponding to the singleton region sets $\{i\}, \{j\}$ respectively.

Proof. Recall that a schedule $\mathbf{x} = \{x_i\}$ is feasible iff no two adjacent links of the conflict graph transmit simultaneously. Since all our regions are cliques of the conflict graph, for any feasible distribution that is locally consistent, we have

$$b_r(x_r) = \begin{cases} b_j(1), & \text{if } x_j = 1, x_k = 0, \forall k \in r \setminus \{j\}, \\ 1 - \sum_{j\in r} b_j(1), & \text{if } x_j = 0, \forall j \in r, \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

An interesting implication of the above observation is, the KFE $F_k(\{b_r\}; \tilde{\mathbf{v}})$ in (4.1), which is in general a function of all the pseudo marginals, can now be expressed as a function of the singleton marginals alone. Specifically, the average energy is

$$\begin{aligned} U_K(\{b_r\}; \tilde{\mathbf{v}}) &= -\sum_{r\in\mathcal{R}} c_r \mathbb{E}_{b_r}\Big[\sum_{j\in r} \tilde{v}_j x_j\Big], \\ &= -\sum_{r\in\mathcal{R}} c_r \Big(\sum_{j\in r} \tilde{v}_j b_j(1)\Big), \\ &= -\sum_{j=1}^{N} \Big(\sum_{\{r \mid j\in r\}} c_r\Big) \tilde{v}_j b_j(1) = -\sum_{j=1}^{N} \tilde{v}_j b_j(1). \end{aligned}$$

Similarly, the term corresponding to a region r in the Kikuchi entropy $H_K(\{b_r\})$ defined in (4.2) can be expressed as

$$\begin{aligned} &\sum_{x_r} b_r(x_r) \log b_r(x_r) \\ &= \Big(1 - \sum_{j\in r} b_j(1)\Big) \log \Big(1 - \sum_{j\in r} b_j(1)\Big) + \sum_{j\in r} b_j(1) \log b_j(1). \end{aligned}$$

Using this expression, the gradient of the Kikuchi free energy $F_K(\{b_j(1)\}; \tilde{\mathbf{v}})$ can be computed as

$$\frac{\partial F_K}{\partial b_i(1)} = -\tilde{v}_i + \sum_{\{r|i\in r\}} c_r \Big(\log b_i(1) - \log \big(1 - \sum_{j\in r} b_j(1)\big)\Big).$$

By using the fact that $\sum_{\{r\in\mathcal{R}|i\in r\}} c_r = 1$, and setting the gradient to zero, the proof is complete. □

Next, we state a corollary of Theorem 1 which derives the fugacities that correspond to the Bethe approximation framework used in [11]. Specifically, if the collection of regions $\mathcal{R}$ is formed using only the edges and the vertices of the conflict graph, it results in the Bethe approximation framework.

Corollary 2. *For a given set of required service rates, the estimate of the fugacities $\{v_i^B\}$ obtained using the Bethe approximation framework is given by*

$$\exp(v_i^B) = \frac{s_i(1-s_i)^{d_i-1}}{\prod_{j\in\mathcal{N}_i}(1-s_i-s_j)},\ \forall i,$$

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

Proof. The proof follows by observing a set of valid counting numbers for the regions. Specifically, the counting number of the regions corresponding to each edge in the conflict graph can be set to 1, and the counting number of a region corresponding to a vertex can be set to $1 - d_i$. Then using (5.1), the proof is completed. □

5.1 Accuracy of the Kikuchi approximation

The accuracy of the Kikuchi approximation crucially depends on the choice of the regions. As the collection of regions becomes larger, the accuracy will improve at the cost of increased complexity. Hence, the major challenge in using this Kikuchi 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 the collection of regions include only the edges and the vertices of the conflict graph, the resulting Bethe approximation is exact for tree graphs. As we have considered larger collection of regions by including the maximal cliques of the graph, the accuracy is expected to improve. Indeed, we confirm this intuition by proving that our approximation algorithm is exact for a class of graphs called the chordal graphs, while the Bethe approximation is exact only for trees (which constitute a special case of chordal graphs).

Definition 3. (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.*

Theorem 2. *If the conflict graph is chordal, the formula proposed in* (3.2) *gives the exact fugacities that correspond to the desired service rates, i.e., if we marginalize the* $p(x)$ *corresponding to those fugacities, we obtain the required service rates.*

Proof. The main idea of the proof is based on a framework called the *hypertree reparameterization* (HR) proposed in [9]. Let us introduce the notion of *junction tree* that is required for the HR framework. 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 for any two maximal regions $r, q \in \mathcal{R}_0$, the elements in $r \cap q$ are part of any maximal region on the unique path from r to q in T.

A collection of regions $\mathcal{R}$ is said to satisfy the junction tree decomposition property if it is possible to construct a junction tree with the maximal regions of $\mathcal{R}$.

Given a collection of regions $\mathcal{R}$ which satisfies the junction tree decomposition property, and includes all the intersections of its maximal regions, the HR framework [9] factorizes[2] a product form distribution $p(\mathbf{x})$ in terms of its exact regional distributions as $p(\mathbf{x}) = \prod_{r \in \mathcal{R}} p_r(x_r)^{c_r}$.

Now, we use the fact that maximal cliques of the chordal graphs satisfy the junction tree decomposition property [9]. Hence, the distribution $p(\mathbf{x})$ defined in (2.1) can be factorized in terms of the collection of cliques $\mathcal{R}$ computed earlier:

$$\frac{1}{Z(v)} \exp\Big(\sum_j v_j x_j\Big) = \prod_{r \in \mathcal{R}} p_r(x_r)^{c_r}, \ \forall x \in \mathcal{I}. \tag{5.3}$$

Substituting $\mathbf{x} = [x_j]_{j=1}^N$ with $x_j = 0$ for all $j \in \mathcal{N}$ in (5.3), along with the observations made in (A.1), we obtain

$$Z(v)^{-1} = \prod_{r \in \mathcal{R}} \Big(1 - \sum_{j \in r} p_j(1)\Big)^{c_r}. \tag{5.4}$$

Similarly, substituting $\mathbf{x} = [x_i]_{i=1}^N$ with $x_i = 1$, $x_j = 0$ for all $j \in \mathcal{N} \setminus \{i\}$ in (5.3), and using (A.1), we obtain

$$\frac{\exp(v_i)}{Z(v)} = \prod_{q \in \mathcal{R}^i} (p_i(1))^{c_q} \prod_{r \in \mathcal{R} \setminus \mathcal{R}^i} \Big(1 - \sum_{j \in r} p_j(1)\Big)^{c_r}. \tag{5.5}$$

[2]This factorization is proposed in [9, page 101] using the terminology of mobius function. It can be easily argued that the factorization that we use here is equivalent to that in [9].

From (5.4), (5.5), and the fact that $\sum_{q\in\mathcal{R}^i} c_q = 1$, we obtain

$$\exp(v_i) = p_i(1) \prod_{r\in\mathcal{R}^i} \Big(1 - \sum_{j\in r} p_j(1)\Big)^{-c_r}.$$

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 (3.2) follows the same relation, and hence our algorithm is exact for chordal graphs. □

CHAPTER 6

NUMERICAL RESULTS

6.1 Simulation Setting:

We generate random geometric graphs of size 20 on a two dimensional square of length three as shown in Figure 6.1 . Two vertices are connected by an edge if they are within a distance of 0.8. We present the results for random network topologies shown in Figure 6.2, 6.4,6.6 and 6.8. We consider symmetric service rate requirements for all the links, and execute the proposed distributed algorithm to compute the fugacities.

6.2 Approximation error:

We define the approximation error $e(s^t)$, as the maximum deviation from the required service rate, among the links whose service requirements are not met. In particular, for a given target service rate vector $s^t = [s_i^t]_{i=1}^N$, $e(s^t) = \max_i (s_i^t - s_i^a)$, where, $s^a = [s_i^a]_{i=1}^N$ are the service rates supported by using the approximated fugacities $\{\tilde{v}\}_{i=1}^N$. We vary the load (the fraction of the maximum permissible service rate) of the network by increasing the required service rates, and plot the percentage error in Figure **??**. We compared the accuracy of our algorithm with the existing Bethe approximation based algorithm [11].

For random graph in Figure 6.2 containing many big loops,from simulation results in 6.3 we observe that Kikuchi approximation gives less error than Bethe approximation. Also for random graph in Figure 6.4 containing few loops of sizes between 3 to 5,from

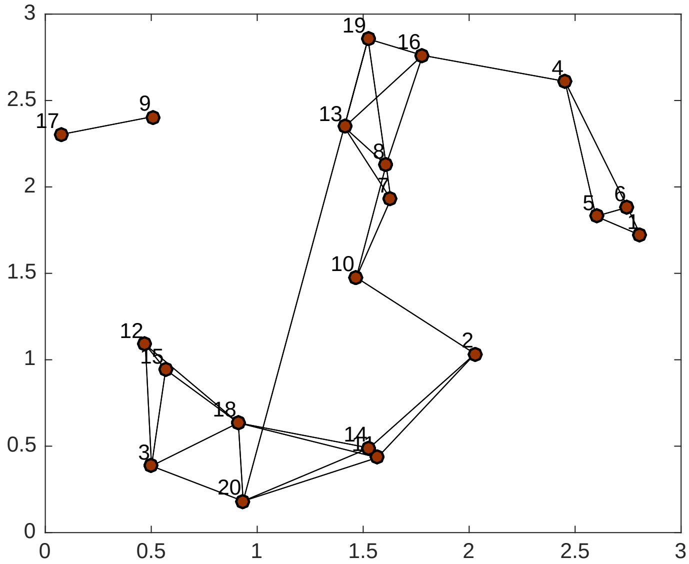

Figure 6.1: Example of randomly generated 20 node graph

simulation results in 6.5 we observe that Kikuchi approximation gives less error(5 percent) than Bethe approximation(40 percent). Similarly for random graph in Figure 6.6 containing random loops,from simulation results in 6.7 we observe that Kikuchi approximation gives relatively less error than Bethe approximation. As proved in Theorem 2, our algorithm is exact if the underlying conflict graph is chordal. This result is verified using a randomly generated chordal graph shown in Figure 6.8. As shown in Figure 6.9, the Bethe approximation based algorithm incurs an error of up to 27 percent as the network operates at the maximum capacity, while our Kikuchi approximation based algorithm is exact. These numerical simulations have to extended to many sets of graphs of different types and it has been observed than in all cases Kikuchi performs better than Bethe approximation i.e Kikuchi approximation gives less error compared to Bethe approximation.

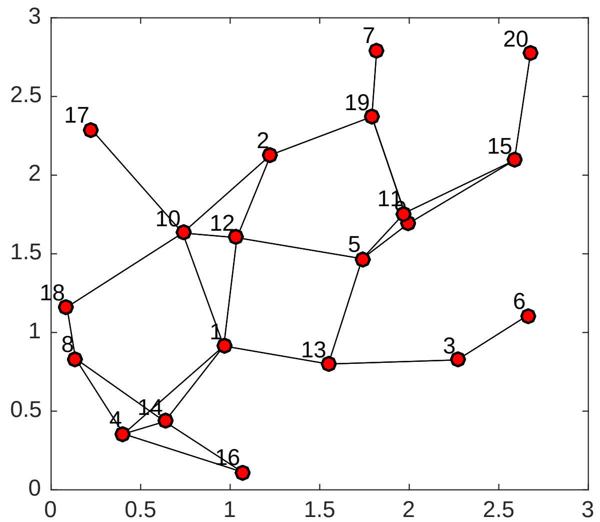

Figure 6.2: Randomly generated graph with 20 nodes and loops of sizes 3,4 and 5

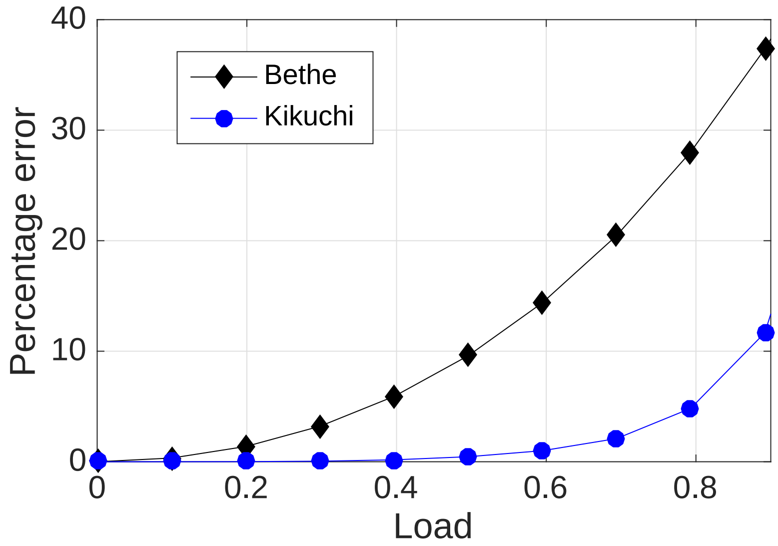

Figure 6.3: Simulation results indicating less error for Kikuchi compared to Bethe approximation

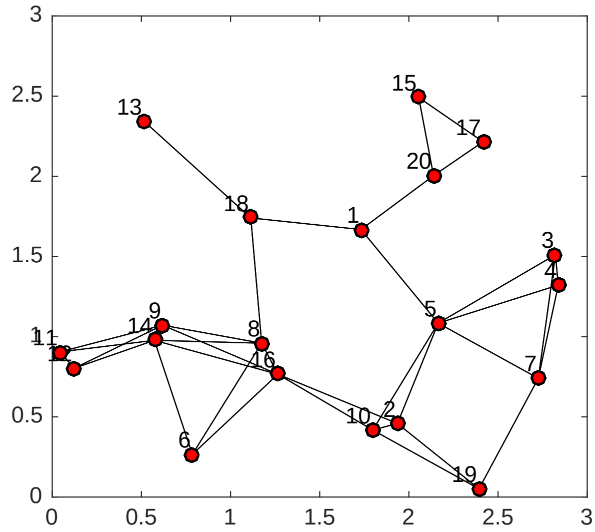

Figure 6.4: Randomly generated graph with 20 nodes and few loops

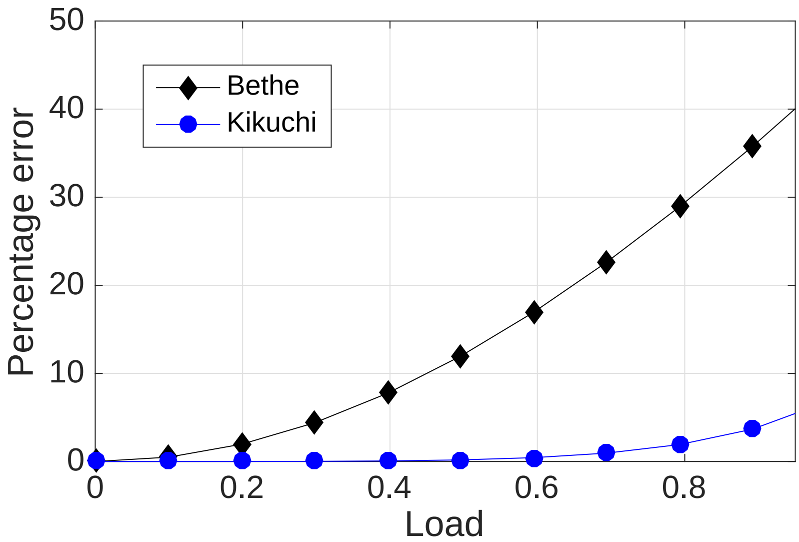

Figure 6.5: Simulation results indicating less error for Kikuchi compared to Bethe approximation

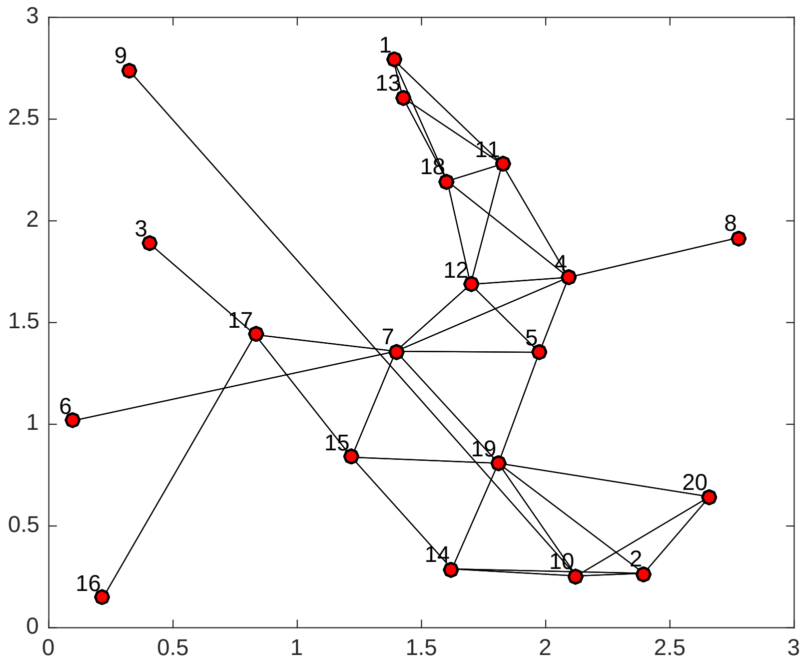

Figure 6.6: Randomly generated graph with 20 nodes

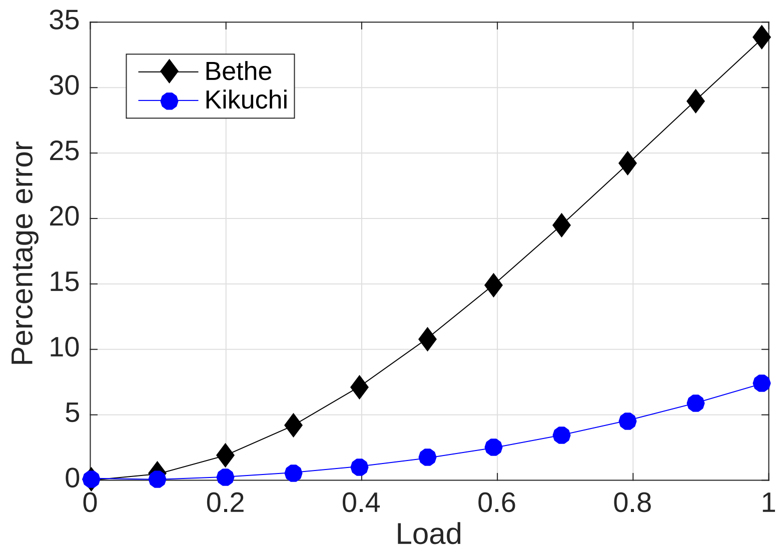

Figure 6.7: Simulation results indicating less error for Kikuchi compared to Bethe approximation

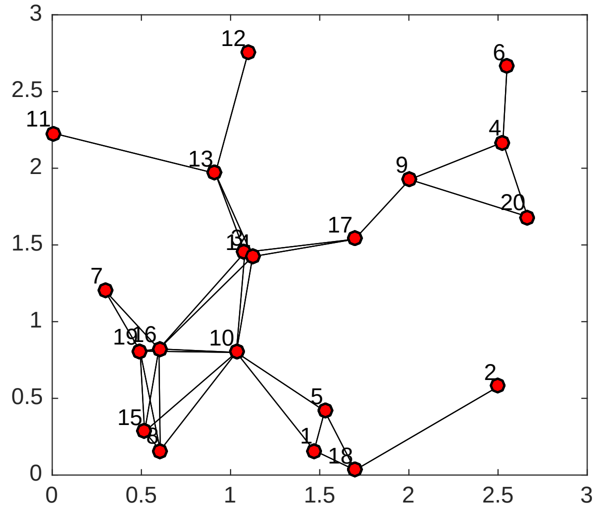

Figure 6.8: Randomly generated Chordal graph with 20 nodes

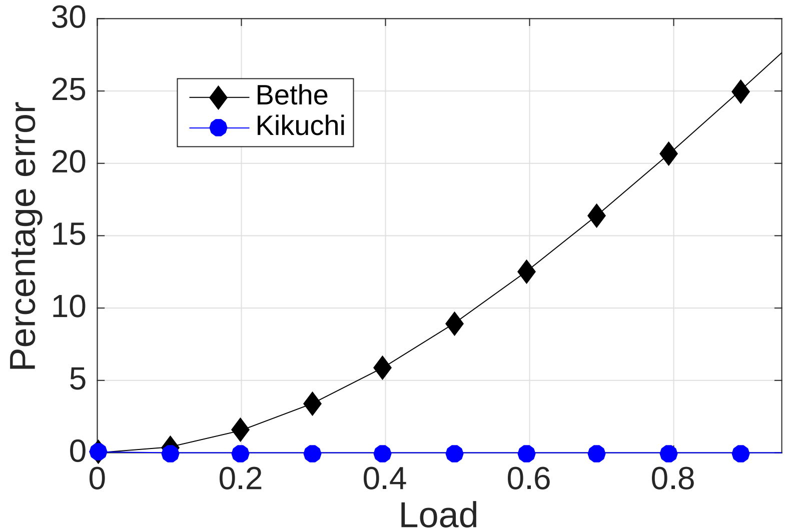

Figure 6.9: Simulation results indicating exactness of Kikuchi for Chordal class of graphs

CHAPTER 7

CONCLUSIONS

We addressed the problem of computing the CSMA attempt rates to support a desired service rate vector. We proposed a simple distributed algorithm, which obtains closed form estimates for the attempt rates. We also proved that our algorithm is exact for a class of graphs, called the chordal graphs. Our algorithm is based on the well-known Kikuchi approximation framework. Numerical results suggest that, for spatial networks, which naturally contain many short loops, our algorithm outperforms the existing Bethe approximation based algorithms.

APPENDIX A

CLIQUE SPECIAL PROPERTY

Clique is a class of graphs where every node in graph is connected to every other node in graph. This class of graphs have an unique advantage for local fugacities computation since local fugacitiy expression turns out to be a simple formula.

Since every node is connected to every other node in graph,to avoid conflicts at any point only one of the node should be 'ON' or there can be a case where none of them are 'ON'. This makes the feasible region of cliques to be only one node 'ON' and no node 'ON' conditions. Now, to satisfy service rate expression, for all the one node 'ON' scenarios we have

$$b_r(x_r) = b_j(1), \text{if } x_j = 1, x_k = 0, \forall k \in r \setminus \{j\}$$

where schedule x_r is schedule where only one node is 'ON'.

All other two or more node 'ON' scenarios are non-feasible thus giving

$$b_r(x_r) = 0$$

Thus the only scenario of no node 'ON' takes value of

$$b_r(x_r) = 1 - \sum_{j \in r} b_j(1), \text{if } x_j = 0, \forall j \in r$$

to ensure sum of all local fugacities i.e sum of probabilities of all possible scenarios leads to one.

Thus,local fugacities for clique region turns out to be a formula given as,

$$b_r(x_r) = \begin{cases} b_j(1), & \text{if } x_j = 1, x_k = 0, \forall k \in r \setminus \{j\}, \\ 1 - \sum_{j \in r} b_j(1), & \text{if } x_j = 0, \forall j \in r, \\ 0, & \text{otherwise.} \end{cases} \tag{A.1}$$

APPENDIX B

FUTURE WORK

The work done till now focuses on considering cliques as regions. This approach leads to less error than Bethe approximation thus standing as better approximation. This method of considering cliques as regions covers edge regions as Bethe does and also in addition takes in cliques as regions thus eliminating loops of size 3(which are cliques) leading to better accuracy. Based on the fact that loops in a graph are reason for error in Bethe approximation we have to try and cover all such loops for better accuracy. In accordance with this idea, a concept of generalized region based approximation is being worked on where not just cliques but also other regions are considered thus making it even more accurate than clique based Kikuchi approximation. For instance, consider an extended version of clique based approximation where we consider loops of size 4 along with cliques as regions. This method eliminates loops of size 3 and 4 and since any loops of greater size do not have much effect this approximation works very well and gives very less error compared to clique based Kikuchi approximation. But the trick here is that since fugacity/service rate at node depends on local fugacities of regions that node is involved in, we need an easy approach to compute local fugacities of regions we consider. This extended method proposes use of cliques and 4-loops as regions. But here,local fugacities of clique region is easy to compute(as stated in Appendix A) and also local fugacities of nodes in 4-loop reduce down to a simple closed from expression thus making computation of fugacities easier and the method on a whole to be more accurate than all existing methods. On similar grounds, further work is being done to see if any other regions can be included which can lead to further improvement in accuracy while still maintaining the property that computation of local fugacitites is easy.

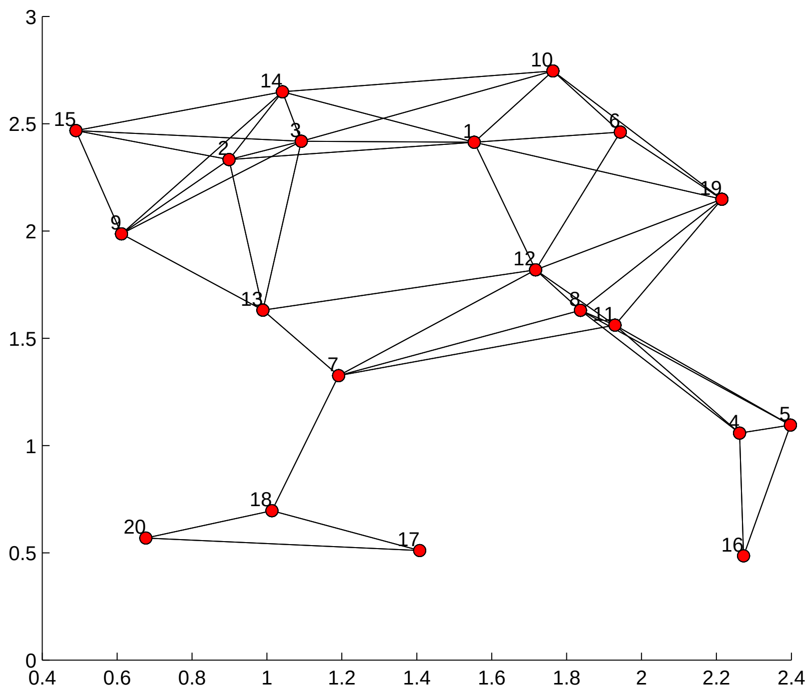

Figure B.1: Randomly generated graph with 20 nodes and lots of loops

Correspondingly,a simulation of this new method is done on a randomly generated 20 node graph containing lots of loops as shown in Figure B.1. We can observe from simulation results in Figure B.2 that loop region based approximation method is more accurate than both Bethe and also clique based Kikuchi approximation thus giving a smaller error than all existing approximations.

Further work and simulations are being done in search of best generalized region based approximation method.

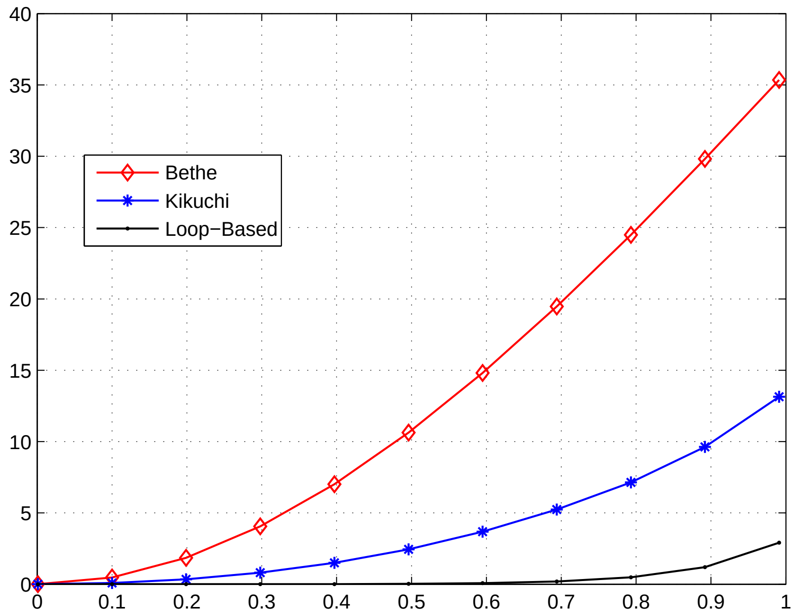

Figure B.2: Simulation results indicating less error for loop based approximation compared to Bethe and clique based KIkuchi approximations

Publications

1. Peruru Swamy, Venkata Pavan Kumar Bellam, Radha Krishna Ganti and Krishna P Jagannathan. Efficient CSMA Based on Kikuchi Approximation. 2016 International Conference on Signal Processing and Communications (SPCOM 2016)

REFERENCES

[1] L. Jiang, M. Leconte, J. Ni, R. Srikant, and J. Walrand, "Fast mixing of parallel Glauber dynamics and low-delay CSMA scheduling," *Information Theory, IEEE Transactions on*, vol. 58, no. 10, pp. 6541–6555, 2012.

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

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

[4] P. Pakzad and V. Anantharam, "Estimation and marginalization using the kikuchi approximation methods," *Neural Computation*, vol. 17, no. 8, pp. 1836–1873, 2005.

[5] P. S. Swamy, R. K. Ganti, and K. Jagannathan, "Adaptive CSMA under the SINR model: Fast convergence through local Gibbs otpimization," in *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*. IEEE, 2015.

[6] ——, "Adaptive CSMA under the SINR model: Efficient approximation algorithms for throughput and utility maximization," *arXiv preprint arXiv:1601.06065*, 2016.

[7] E. Tomita, A. Tanaka, and H. Takahashi, "The worst-case time complexity for generating all maximal cliques," in *Computing and Combinatorics*. Springer, 2004, pp. 161–170.

[8] B. Van Houdt, "Explicit back-off rates for achieving target throughputs in csma/ca networks," *arXiv preprint arXiv:1602.08290*, 2016.

[9] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends® in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008.

[10] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *Information Theory, IEEE Transactions on*, vol. 51, no. 7, pp. 2282–2312, 2005.

[11] S.-Y. Yun, J. Shin, and Y. Yi, "CSMA using the Bethe approximation for utility maximization," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 206–210.