A Log-Likelihood Ratio based Generalized Belief Propagation

Alexandru Amaricai
University Politehnica Timisoara
Timisoara, Romania
alexandru.amaricai@cs.upt.ro

Mohsem Bahrami
University of Arizona
Tucson, AZ, USA
bahrami@email.arizona.edu

Bane Vasić
University of Arizona
Tucson, AZ, USA
vasic@ece.arizona.edu

***Abstract*—In this paper, we propose a reduced complexity Generalized Belief Propagation (GBP) that propagates messages in Log-Likelihood Ratio (LLR) domain. The key novelties of the proposed LLR-GBP are: *(i)* reduced fixed point precision for messages instead of computational complex floating point format, *(ii)* operations performed in logarithm domain, thus eliminating the need for multiplications and divisions, *(iii)* usage of message ratios that leads to simple hard decision mechanisms. We demonstrated the validity of LLR-GBP on reconstruction of images passed through binary-input two-dimensional Gaussian channels with memory and affected by additive white Gaussian noise.**

***Index Terms*—Probabilistic inference, graphical models, generalized belief propagation (GBP).**

I. Introduction

Multiple inference problems in computer vision, error-correction coding and artificial intelligence can be reformulated as the computation of marginal probabilities of a joint probability distribution [1]–[3]. Traditional low-complexity approximate algorithms for solving these problems are based on belief propagation (BP) [4], [5] which operate on factor graphs. BP, as an algorithm to compute marginals of functions on a factor graph, has its roots in the broad class of Bayesian inference problems [6]. It is well known that the BP algorithm gives exact inference only on cycle-free graphs (trees). It has been also observed that in some applications BP can provide close approximations to exact marginals on loopy graphs. However, an understanding of the behavior of BP in the latter case is far from complete. Moreover, it is known that BP does not perform well on graphs which contain a large number of short cycles. A new class of message-passing algorithm called generalized belief propagation (GBP) is introduced in [7] to solve the problem of computing marginal probability distributions in factor graphs with short cycles. A powerful conceptual framework for finite-dimensional lattice models is the cluster variation method by Kikuchi [8], [9]. In particular, the algorithm relies on the extension of the cluster variation method, called region graph method proposed by Yedidia *et al.* [7]. The major difference between GBP and BP is that GBP benefits from region-to-region message passing The major difference between GBP and BP is that GBP benefits from region-to-region message passing instead of the node-to-node message passing algorithm of BP. In practice, GBP algorithms can often dramatically outperform BP algorithms in either accuracy or convergence properties [10], [11].

In order to improve throughput and energy consumption characteristics, as well as to obtain real time capabilities, hardware acceleration using dedicated architectures is employed for BP algorithms [12]. However, developing hardware architectures for GBP presents several challenges, due to the fact that the messages propagated among regions are conditional probabilities. These include: *(i)* divisions in message update equations, *(ii)* multiplication in both message and belief update equations, and *(iii)* requirements for very large precision, usually in floating point formats. In this paper, we propose a log-likelihood ratio (LLR) based GBP algorithm to address the hardware implementation issues by relying on only addition based operations (additions, subtractions and comparisons) with messages and beliefs represented in fixed point formats. This is achieved by introducing LLR based representations for messages and beliefs. The LLR representations allow us to devise arithmetic operations in log-likelihood domain for both message and belief update equations. The log-likelihood messages represent the standard approach in a wide range of iterative message-passing algorithms, including Turbo decoding [13], LDPC decoding - both binary [14] and non-binary [15], but far from trivial in inference algorithms such as GBP where messages express complex dependencies among variables. The proposed approach presents the following advantages: *(i)* divisions and multiplications are reduced in logarithm-domain to subtractions and additions; *(ii)* arithmetic operations are performed using fixed point formats, that has reduced complexity with respect to floating point representations; *(iii)* the usage of ratios for decoding and detection problems lead to simple sign based hard decision mechanisms.

We apply the proposed LLR-GBP for an image reconstruction application, denoising of images affected by a binary-input two-dimensional (2-D) Gaussian channel and additive white Gaussian noise (AWGN). Simulation results show that LLR-GBP with messages and beliefs represented in a 24-bit fixed point format, has similar performance to the floating point implementation. GBP as an image denoising algorithm works on probabilistic graphical model of the 2-D Gaussian channel with AWGN. There are many cycles in the factor graph representation of a 2-D Gaussian channel [16], which invalidates the tree-like assumption used in BP and leads to

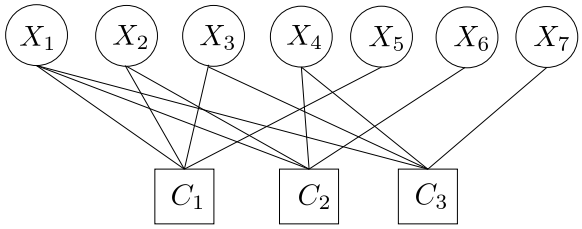

Fig. 1. The factor graph for the joint probability distribution in the Eq. (4) is given. The set of variable nodes $\mathbf{X} = \{X_1, X_2, \ldots, X_7\}$ represents the error patterns and the set of factor nodes $\mathbf{C} = \{C_1, C_2, C_3\}$ verify constraints.

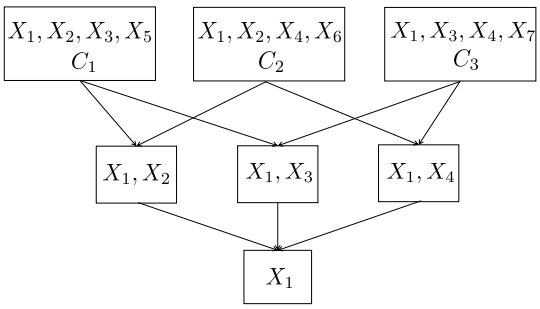

Fig. 2. The region graph associated to the factor graph depicted in Fig. 1

poor performance. In order to show that GBP can address the issues of short cycles in BP related methods, we also compare the performance of our LLR-GBP with JTED [17], that uses fixed point formats, for detection of binary arrays passed through a 2-D intersymbol interference (ISI) channel. JTED can be considered as a sequential tree-reweighted sum-product algorithm [18], where for 2-D detection uses BCJR for computing exact marginals over row and column directions, and incorporates a message passing paradigm along both dimensions in an iterative manner for exchanging extrinsic information. However, this scheme still suffers from the cycles in the underlying graphical model of 2-D ISI channel for passing extrinsic information between row and column detectors. Our simulation results indicate that the reduced complexity LLR-GBP (with 24 bits, 8 bits fractional and 16 bits offset intervals) outperforms JETD with around 2 dB in terms of bit-error rate performance.

The paper is organized as follows. Section II presents the marginalization problem and the probability-domain GBP algorithm; Section III is dedicated to the log likelihood GBP version; simulation results and discussions are presented in Section IV.

II. Marginalization and GBP

A. Problem Formulation

Let $\mathbf{X}$ represent a set of N discrete random variables $\{X_1, X_2, \ldots, X_N\}$ and $\mathbf{x}$ denote an assignment to these variables such that $\mathbf{x} = (x_1, x_2, \ldots, x_N)$. Given a joint distribution $p(\mathbf{X} = \mathbf{x}) = p(X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N)$, the marginal distribution of a subset of variables $\mathbf{x}_S \subset \mathbf{X}$ is the probability distribution of variables $\mathbf{x}_S$ averaging over all information about $\mathbf{x} \setminus \mathbf{x}_S$. This can be calculated by summing $p(x_1, x_2, \ldots, x_N)$ over $\mathbf{x} \setminus \mathbf{x}_S$, i.e,

$$p(\mathbf{x}_S) = \sum_{\mathbf{x} \setminus \mathbf{x}_S} p(x_1, x_2, \ldots, x_N). \tag{1}$$

This process of computing marginal probability distributions can be intractable for large N as it needs to take summation over exponential number of possible assignments of variables. We assume that the given joint probability distribution can be factored into M functions in the following form

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{C_i} f_{C_i}(\mathbf{x}_{C_i}), \tag{2}$$

where the normalization constraint Z, so called the partition function, is given by

$$Z = \sum_{\mathbf{x}} \prod_{C_i} f_{C_i}(\mathbf{x}_{C_i}), \tag{3}$$

C_i is a labeling index for representing the M functions $f_{C_1}, f_{C_2}, \ldots, f_{C_M}$, and the function f_{C_i} is defined over the subset of variables $\mathbf{X}_{C_i} \subset \mathbf{X}$.

B. Factor Graph

Graphical models provide an intuitive framework for representing interacting sets of variables and functions. Using the factor graph formalism [3], the factorization in Eq. (2) can be described by a bipartite graph $G = (\mathbf{X} \cup \mathbf{F}, \mathbf{E})$ with two types of nodes, namely variable nodes $\mathbf{X}$ and factor nodes $\mathbf{F}$, and a set of edges $\mathbf{E}$ - Fig. 1. Variables $X_i \in \mathbf{X}$ are symbolized by variable nodes; functions F_{C_j} are symbolized by factor nodes; and the dependence of a constraint on a variable is symbolized by an edge joining the two. We denote the variable nodes by circle nodes and the functions by square nodes, where the edge $\left(X_i, F_{C_j}\right)$ between the factor node F_{C_j} and the variable node X_i included in $\mathbf{E}$ if and only if $X_i \in \mathbf{X}_{C_j}$. The set of variable nodes connected to the factor node F_{C_j} is denoted by $\mathcal{N}_{C_j}$ and similarly the set of factor nodes connected to the variable node X_i is denoted by $\mathcal{N}_{X_i}$.As an example, a factor graph corresponding to the following joint distribution

$$p(x_1, x_2, x_3, \ldots, x_7) = \frac{1}{Z} f_{C_1}(x_1, x_2, x_3, x_5)\, f_{C_2}(x_1, x_2, x_4, x_6) \\ f_{C_3}(x_1, x_3, x_4, x_7), \tag{4}$$

is depicted in Fig. 1.

C. Region Graph

The GBP algorithm is employed to provide an approximate solution to the problem of minimizing the Gibbs free energy [7], [19]. This algorithm is empirically observed to provide marginal probability estimates, close to true marginal probabilities [20], [21]. The main characteristic of GBP is

represented by the fact that messages are passed between clusters of variable nodes, along the region graph.

A region graph consists of clusters (regions) of variable and factor nodes, and can be constructed from a given factor graph as we explain next. A region graph initially is formed by clustering every factor node and its neighboring variables nodes into a region, which is called a basic (ancestor) region, so that every ancestor region contains only one factor node $f_{C_j} \in \mathbf{F}$. Then, the cluster variation method [7] is applied to establish the remaining of the region graph. We construct the remaining regions by taking the intersection of the basic regions and their intersections - as shown in Fig. 2. The set of all regions in the region graph is denoted by $\mathcal{R}$. For every region $R \in \mathcal{R}$, we denote the set of variable nodes in the region R by $\mathbf{X}_R$ and the state of these variables by $\mathbf{x}_R$. Let $b(\mathbf{x}_R)$ and $p(\mathbf{x}_R)$ be the belief and the probability of $\mathbf{x}_R$. Furthermore, $\mathcal{P}(R)$ and $\mathcal{D}(R)$ denote, respectively, the parent and descendant regions of region R, and $\mathcal{E}(R) = R \cup \mathcal{D}(R)$

D. Message and Belief Update Equations

An iteration means updating all messages between regions and the beliefs of every region. We set the initial messages and beliefs to be uniform equal to 1. For every region $R \in \mathcal{R}$, the messages from its parent regions $P \in \mathcal{P}_R$ at iteration k is given by

$$m_{P\to R}^{(k)}(\mathbf{x}_R) \propto \frac{\sum\limits_{\mathbf{x}_{P\setminus R}} \prod\limits_{F_{C_j}\in F_{P\setminus R}} f_{C_j}(\mathbf{x}_{C_j}) \prod\limits_{(I,J)\in N(P,R)} m_{I\to J}^{(k-1)}(\mathbf{x}_J)}{\prod\limits_{(I,J)\in D(P,R)} m_{I\to J}^{(k-1)}(\mathbf{x}_J)}, \tag{5}$$

where $N(P,R)$ is the set of all connected pairs of regions, (I,J) such that $J \in \mathcal{E}(P)\backslash\mathcal{E}(R)$ while $I \notin \mathcal{E}(P)$. $D(P,R)$ is the set of all connected pairs of regions (I,J) such that $J \in \mathcal{E}(R)$, while $I \in \mathcal{E}(P)\backslash\mathcal{E}(R)$. $F_{P\setminus R}$ is the set of factor nodes in the region $P\backslash R$.

The belief update equation for every region $R \in \mathcal{R}$ at iteration $k \geq 1$ is given by

$$b_R^{(k)}(\mathbf{x}_R) \propto \prod_{F_{C_j}\in F_R} f_{C_j}(\mathbf{x}_{C_j}) \left(\prod_{P\in\mathcal{P}(R)} m_{P\to R}^{(k)}(\mathbf{x}_R) \right) \times \left(\prod_{D\in\mathcal{D}(R)} \prod_{P'\in\mathcal{P}(D)\setminus\mathcal{E}(R)} m_{P'\to D}^{(k)}(\mathbf{x}_D) \right), \tag{6}$$

where A_R is the set of factor nodes in region R and $f_{C_j}(\mathbf{x}_{C_j})$'s are their functions, and $\mathcal{P}(D)\backslash\mathcal{E}(R)$ is the set of all regions that are parents of region D except for R and descendants of R. In order to help convergence and avoid the overshooting problem [7] for the belief updates, every message after each iteration is a convex combination of the old and the updated message such that

$$m_{P\to R}^{(k)}(\mathbf{x}_R) = \omega^{(k)} m_{P\to R}^{(k)}(\mathbf{x}_R) + (1-\omega^{(k)}) m_{P\to R}^{(k-1)}(\mathbf{x}_R), \tag{7}$$

where $0 \leq \omega^{(k)} \leq 1$ is the weight or damping factor. There is no concrete theory on choices of damping factor so it needs to be verified for each specific application.

E. Related Work

Several approaches to improve the computational parameters - processing time and memory requirements - of GBP have been proposed in [22]–[24]. These optimization techniques rely on two approaches: *(i)* reducing the number of arithmetic operations, by employing techniques such as result caching, conversion of a grid search into a linear search problem, or hierarchical state-space reduction [22], [23], and *(ii)* reducing the complexity of arithmetic operations for message and belief update equations, by performing them in logarithm-domain [24]. The latter targets elimination of divisions and multiplications, using only addition based operations.

The proposed optimization target complexity reduction in the message and belief updates, targeted mainly for decoding and detection problems, performing the operations in logarithm-domain. With respect to [24], our main contributions are: *(i)* development of a ratio based version - defined by equations (10), (11), (13), and (14); the logarithm is applied on the ratio based GBP; *(ii)* utilization of fixed point formats, instead of the more computationally complex floating point format.

III. Log-Likelihood Ratio Based GBP Algorithm

Similar to the log-likelihood versions of BP [13], [14], as a first step to reduce the complexity of GBP, we define ratios for messages and beliefs. The ratio of beliefs for the region $R \in \mathcal{R}$ at iteration k is defined by

$$\beta_R^{(k)}(\mathbf{x}_R) = \frac{b_R^{(k)}(\mathbf{x}_R)}{b_R^{(k)}(\mathbf{x}_R^{\text{ref}})}, \tag{8}$$

where $\mathbf{x}_R^{\text{ref}}$ represents the reference state for the ratio-domain, and $b_R^{(k)}(\mathbf{x}_R^{\text{ref}})$ is the belief corresponding to this event. Similarly, the ratio of messages coming to the region R from its parent regions $P \in \mathcal{P}_R$ at iteration k is determined by

$$\lambda_{P\to R}^{(k)}(\mathbf{x}_R) = \frac{m_{P\to R}^{(k)}(\mathbf{x}_R)}{m_{P\to R}^{(k)}(x_R^{\text{ref}})}, \tag{9}$$

where $m_{P\to R}^{(k)}(x_R^{\text{ref}})$ is the probability that the parent region $P \in \mathcal{P}_R$, at iteration k, sends a message to the region R that the state of its variables is the reference state. We have considered the all-one state (the state that all variables have value 1) as the reference state in our implementation.

Using the ratio of messages, the message update equation at iteration k becomes

$$\lambda_{P\to R}^{(k)}(\mathbf{x}_R) = \frac{\sum\limits_{\mathbf{x}_{P\setminus R}} \prod\limits_{F_{C_j}\in F_{P\setminus R}} \phi_{C_j}(\mathbf{x}_{C_j}) \prod\limits_{(I,J)\in N(P,R)} \lambda_{I\to J}^{(k-1)}(\mathbf{x}_J)}{\left(\prod\limits_{(I,J)\in D(P,R)} \lambda_{I\to J}^{(k-1)}(\mathbf{x}_J) \right) c_{P\to R}^{(k)}}, \tag{10}$$

where $\phi_{C_j}(\mathbf{x}_{C_j})$ is the ratio of constraint and $c^{(k)}_{P\to R}$ is the correction factor which ensures $\lambda^{(k)}_{P\to R}(\mathbf{x}^{\text{ref}}_R)=1$. The ratio of constraint is defined by

$$\phi_{C_j}(\mathbf{x}_{C_j})=\frac{f_{C_j}(\mathbf{x}_{C_j})}{f_{C_j}(\mathbf{x}^{\text{ref}}_{C_j})}, \tag{11}$$

where $f_{C_j}(\mathbf{x}^{\text{ref}}_{C_j})$ is value of function at the constraint C_j when the state of their variables, $\mathbf{x}_{C_j}$, is the reference state. The correction factor for messages from a parent region P to the region R is given by

$$c^{(k)}_{P\to R}=\sum_{\mathbf{x}_{P\setminus R}}\prod_{F_{C_j}\in F_{P\setminus R}}\phi_{C_j}(\mathbf{x}^{\text{ref}}_{C_j})\prod_{(I,J)\in N(P,R)}\lambda^{(k-1)}_{I\to J}(\mathbf{x}^{\text{ref}}_J). \tag{12}$$

Furthermore, Eq. (7) becomes

$$\begin{aligned}\lambda^{(k)}_{P\to R}(\mathbf{x}_R)=&\lambda^{(k)}_{P\to R}(\mathbf{x}_R)\times\frac{1}{1+\frac{1-\omega^{(k)}}{\omega^{(k)}}\times\frac{\sigma^{(k)}}{\sigma^{(k-1)}}}+\\&\lambda^{(k-1)}_{P\to R}(\mathbf{x}_R)\times\frac{1}{1+\frac{\omega^{(k)}}{1-\omega^{(k)}}\times\frac{\sigma^{(k-1)}}{\sigma^{(k)}}},\end{aligned} \tag{13}$$

where $\sigma^{(k)}=\sum_{\mathbf{x}_R}\lambda^{(k)}_{P\to R}(\mathbf{x}_R)$. The update of $\sigma^{(k)}$ is performed as follows

$$\sigma^{(k)}=\omega^{(k)}\sigma^{(k)}+(1-\omega^{(k)})\sigma^{(k-1)}. \tag{14}$$

The belief ratio update equation is similar to Eq. (6) except messages and functions are replaced with the ratio of them.

Applying the logarithm, the multiplications in both belief and message update equations are reduced to additions, while the division in the message update equation becomes a subtraction. The message update equation (Eq. (10)) turns into

$$\begin{aligned}\Lambda^{(k)}_{P\to R}(\mathbf{x}_R)=&\underset{\mathbf{x}_{P\setminus R}}{\diamond}\Big(\sum_{F_{C_j}\in F_{P\setminus R}}\Phi_{C_j}(\mathbf{x}_{C_j})\sum_{(I,J)\in N(P,R)}\Lambda^{(k-1)}_{I\to J}(\mathbf{x}_J)\Big)\\&-\sum_{(I,J)\in D(P,R)}\Lambda^{(k-1)}_{I\to J}(\mathbf{x}_J)-C^{(k)}_{P\to R},\end{aligned} \tag{15}$$

where $\Lambda^{(k)}_{P\to R}$, Φ_{C_j} and $C^{(k)}_{P\to R}$, respectively, defined as the logarithm of $\lambda^{(k)}_{P\to R}$, ϕ_{C_j} and $c^{(k)}_{P\to R}$, $\diamond(.)$ indicates the approximation used for computing the logarithm of the sum, $(\log(\sum))$ which is explained in the following.

Considering two positive real numbers $\lambda_1,\lambda_2\in\mathbb{R}$, we have

$$\begin{aligned}\diamond(\lambda_1,\lambda_2)&=\log(\lambda_1+\lambda_2)=\log(\max(\lambda_1,\lambda_2)+\min(\lambda_1,\lambda_2)),\\&=\log(\max(\lambda_1,\lambda_2))+\log\left(1+\frac{\min(\lambda_1,\lambda_2)}{\max(\lambda_1,\lambda_2)}\right).\end{aligned}$$

We denote the term $\frac{\min(\lambda_1,\lambda_2)}{\max(\lambda_1,\lambda_2)}$ by η. According to the above equation, computation of $\log(\sum)$ is reduced to a maximum and computation of $\log(1+\eta)$. As $\lambda_1,\lambda_2>0$, $0<\eta\leq 1$, and therefore $0<\log(1+\eta)\leq\log(2)$, we use the following method for approximating the term $\log(1+\eta)$. We first split the $(0,1)$ interval into k equal intervals as follows $(0,l_1),[l_1,l_2),...,[l_{k-1},1)$, where $l_i=\frac{1}{i\times k}$ and $i\leq k$. η is approximated with l_i, if $l_i\leq\eta<l_{i+1}$. In this method, we only need to perform k comparisons among η and l_i's. In the logarithm-domain, the terms $\log(l_i)$ and $\log(1+l_i)$ are constant and can be computed offline for a fixed number of intervals, k. A larger k allows better approximation at the expense of higher complexity.

IV. Simulation Results

A. Image Denoising Application

In order to compare the performance of the proposed LLR based approach for GBP with the probability-domain floating point version, we use GBP for an image denoising application for reconstruction of images affected by 2-D Gaussian channels and independent noise, such as AWGN.

We assume that in all our experiments the size of Gaussian kernel is 3×3. Let us denote the binary representation of an input image by an array $\mathbf{x}=[x_{i,j}]$, the kernel of Gaussian filters by $\mathbf{H}$, and the distorted version of input image by an array $\mathbf{y}=[y_{i,j}]$. We are interested in finding the most likely input samples $\hat{x}_{i,j}$ from $\mathbf{y}$. The (i,j)-th output sample, $y_{i,j}$, is the binary input affected by the 2-D Gaussian channel and is given by

$$y_{i,j}=\mathbf{H}\mathbf{x}\,[i,j]+n\,[i,j]\,, \tag{16}$$

where

$$\mathbf{x}\,[i,j]=\begin{bmatrix}x_{i-1,j-1}&x_{i-1,j}&x_{i-1,j+1}\\x_{i,j-1}&x_{i,j}&x_{i,j+1}\\x_{i+1,j-1}&x_{i+1,j}&x_{i+1,j+1}\end{bmatrix} \tag{17}$$

and $\mathbf{H}$ is represented the considered 3×3 Gaussian kernel, and $n\,[i,j]$ is a sample from a zero-mean and σ^2-variance Gaussian distribution. The variance σ^2 is defined as a function of signal-to-noise ratio (SNR) so that

$$\sigma=||\mathbf{H}||\times 10^{-\text{SNR}/20}, \tag{18}$$

where SNR is given in db and $||.||$ denotes the l_2-norm.

The problem is to find the most likely input bits $\{x_{i,j}\}$ from $\mathbf{y}$ that maximizes $p\,(x_{i,j}|\mathbf{y})$, for a fixed SNR value. The problem of maximizing these probabilities is reduced to computing

$$p(x_{i,j}|\mathbf{y})\propto\sum_{\mathbf{x}\setminus x_{i,j}}\prod_{i,j}\exp\left(\frac{(y_{i,j}-\mathbf{H}\mathbf{x}\,[i,j])^2}{2\sigma^2}\right). \tag{19}$$

The probabilities $\{p(x_{i,j}|\mathbf{y})\}$ are called *a posteriori* probabilities (APPs). Computing APPs is a hard problem as it requires to taking sum over exponential number of variables. We use the logarithmic likelihood ratio version of GBP for estimating APPs. The performane loss shows that the algorithm suffers from dependencies of messages and existense of cycles in the underlying graphical model for exchanging extrinsic information between row and column BCJR detector.

We have applied GBP in both probability-domain, with messages and beliefs represented using 64-bits IEEE754 double precision floating point format, and in logarithm-domain using 24-bit fixed point format, with 4 and 8 bits for fractional part and with 4 and 16 offset constants in the approximation of $\log(\sum)$, for a SNR range of the AWGN noise from 0 to 5 db. The considered Gaussian kernel corresponds to a zero mean

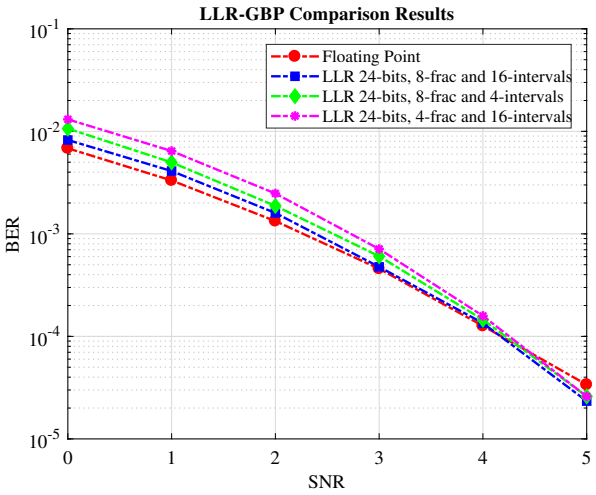

Fig. 3. Detection performance curves of GBP for 64-bit double precision format, 24-bit fixed point LLR.

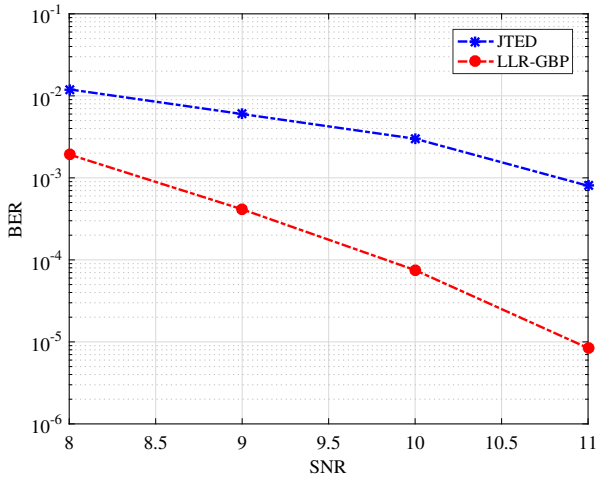

Fig. 4. Comparison results between the proposed LLR-GBP (24-bit: 8 bits fractional and 16 bits offset intervals) and JTED.

and a Results are plotted in Fig. 3. Fig. 3 indicates that the proposed LLR version has similar performance with respect to the floating point implementation, with a slight decrease in performance for low SNR regions (0-3 dB), and a slight increase in performance for higher SNR (5 dB). Reducing the number of bits associated with the fractional part will lead to a performance decrease. Furthermore, reducing the number of offset intervals in $\log(\sum)$ approximation will also impact the performance of the GBP. It is worth noted that reducing the number of bits associated to the fractional part does not lead to reduced computational complexity, while reducing the number of offset intervals in the $\log(\sum)$ approximation will lead to reduced number of performed arithmetic operations (reduced number of comparisons with constants).

B. Comparison Results with JTED

In this subsection, we present the comparision results between the 24-bit fixed point LLR-GBP, with 8 bits for fractional and 16 offset intervals, and JTED proposed in [17] for detection of 2-D binary arrays passed through a 2-D ISI channel. The JTED method uses BCJR detectors [25], which give exact APPs for 1-D case, in row and column directions allowing the message passing along both dimensions in an iterative manner. The considered ISI channel has been defined by

$$\mathbf{H} = \begin{bmatrix} 0.0625 & 0.25 & 0.0625 \\ 0.25 & 1 & 0.25 \\ 0.0625 & 0.25 & 0.0625 \end{bmatrix}. \quad (20)$$

We should note that, due to the computational complexity of the considered formulation of the GBP algorithm for detection, the maximum size of an input binary array can be 32×32. For this, we have performed simulations on random 2-D binary arrays of size 32×32 for LLR-GBP, with respect to 64×64 random binary arrays for JTED [17]. Simulation results, presented in Fig. 4, indicate that the proposed LLR-GBP provides an almost 2 dB improvement in bit-error rate performance comparing with JTED.

V. Conclusions

In this paper, we propose a LLR version in order to reduce both the computational complexity and the storage requirements for GBP. From a computational perspective, the main advantages of the proposed approach are:

1) arithmetic operations are performed in fixed point formats rather the computationally complex floating point formats,
2) multiplications in the belief and message update rules are reduced to additions,
3) divisions in the message update rules are reduced to subtractions, and
4) signed based hard-decision extraction mechanism for single variable regions, as is the case in the vast majority of detection problems.

Regarding the approximation of the logarithm of the addition, our approach employs a maximum computation, as well as comparisons with a number of offline computed constants. Therefore, the proposed LLR version of GBP employs only fixed point addition based operations - addition, subtraction and comparisons - that makes it suitable for hardware acceleration on FPGA devices.

Simulation results performed for an image reconstruction application indicate that for 24-bit fixed point formats, a slight degradation in performance in low SNR regions (SNR 0 to 3) is obtained with respect to the 64-bit floating point probabilistic GBP. However, this slight degradation will come with improved storage requirements for the LLR version, with more than 2.5x reduction is storage for LLR based version. Reducing the number of fractional bits, as well as the number of offset constants used in the approximation of $\log(\sum)$, will reduce the detection performance in the low SNR regions.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation under grants ECCS-1500170 and SaTC-1813401.

REFERENCES

[1] M. Jordan, *Learning in Graphical Models*. MIT Press, 1999.

[2] M. Jordan and C. Bishop, *An Introduction to Graphical Models*. draft, 2000.

[3] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498 –519, Feb. 2001.

[4] J. Pearl, *Probablisitic Reasoning in Intelligent Systems*. San Francisco, CA: Kaufmann, 1988.

[5] R. G. Gallager, "Low density parity check codes," Ph.D. dissertation, Cambridge, MA, 1963.

[6] B. J. Frey, *Graphical models for machine learning and digital communication*. Cambridge, MA, USA: MIT Press, 1998.

[7] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inform. Theory*, vol. 51, pp. 2282–2312, July 2005.

[8] R. Kikuchi, "A Theory of Cooperative Phenomena," *Physical Review Online Archive (Prola)*, vol. 81, no. 6, p. 988, Mar. 1951.

[9] T. Morita, *Foundations and applications of cluster variation method and path probability method*. Publication Office, Progress of Theoretical Physics, 1994.

[10] S. Shamai, L. H. Ozarow, and A. D. Wyner, "Information rates for a discrete-time gaussian channel with intersymbol interference and stationary inputs," *IEEE Trans. on Inf. Theory*, vol. 37, no. 6, pp. 1527–1539, Nov 1991.

[11] G. Sabato and M. Molkaraie, "Generalized belief propagation for the noiseless capacity and information rates of run-length limited constraints," *IEEE Trans. Commun.*, vol. 60, no. 3, pp. 669–675, Mar. 2012.

[12] C. Liang, C. Cheng, Y. Lai, L. Chen, and H. H. Chen, "Hardware-efficient belief propagation," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 21, no. 5, pp. 525–537, May 2011.

[13] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain," in *Proc. Int. Conf. on Commun.*, vol. 2, Jun 1995, pp. 1009–1013.

[14] M. P. C. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. on Commun.*, vol. 47, no. 5, pp. 673–680, May 1999.

[15] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary ldpc codes over GF(q)," *IEEE Trans. on Commun.*, vol. 55, no. 4, pp. 633–643, April 2007.

[16] O. Shental, N. Shental, S. Shamai, I. Kanter, A. J. Weiss, and Y. Weiss, "Discrete-input two-dimensional gaussian channels with memory: Estimation and information rates via graphical models and statistical mechanics," *IEEE Trans. on Inf. Theory*, vol. 54, no. 4, pp. 1500–1513, April 2008.

[17] Y. Chen and S. G. Srinivasa, "Joint self-iterating equalization and detection for two-dimensional intersymbol-interference channels," *IEEE Trans. on Commun.*, vol. 61, no. 8, pp. 3219–3230, August 2013.

[18] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, pp. 1–305, Nov. 2008.

[19] P. Pakzad and V. Anantharam, "Kikuchi approximation method for joint decoding of LDPC codes and partial-response channels." *IEEE Trans. on Commun.*, vol. 54, no. 7, pp. 1149–1153, 2006.

[20] C. K. Matcha, S. Roy, M. Bahrami, B. Vasić, and S. G. Srinivasa, "2D LDPC codes and joint detection and decoding for two-dimensional magnetic recording," *IEEE Trans. on Magn.*, vol. 54, no. 2, pp. 1–11, Feb. 2018.

[21] C. Matcha, M. Bahrami, S. Roy, S. Srinivasa, and B. Vasić, "Generalized belief propagation based TDMR detector and decoder," in *Proc. IEEE Int. Symp. Inf. Theory*, July 2016.

[22] K. Petersen, J. Fehr, H. Burkhardt, and G. Rigoll, "Fast generalized belief propagation for map estimation on 2d and 3d grid-like markov random fields," in *Pattern Recognition, DAGM 2008*. Springer Berlin Heidelberg, 2008.

[23] S. Chen and Z. Wang, "Acceleration strategies in generalized belief propagation," *IEEE Trans. on Industrial Informatics*, vol. 8, no. 1, pp. 41–48, Feb 2012.

[24] A. D. Shigyo and K. Ishibashi, "QR-decomposed generalized belief propagation with smart message reduction for low-complexity mimo signal detection," in *2017 Asia-Pacific Signal and Inf. Processing Assoc. Annual Summit and Conf. (APSIPA ASC)*, Dec 2017, pp. 1795–1799.

[25] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284 – 287, Mar. 1974.