Author(s): Abdi, Younes; Ristaniemi, Tapani

Title: Optimization of Linearized Belief Propagation for Distributed Detection

Optimization of Linearized Belief Propagation for Distributed Detection

Younes Abdi, *Member, IEEE,* and Tapani Ristaniemi, *Senior Member, IEEE*

Abstract—In this paper, we investigate distributed inference schemes, over binary-valued Markov random fields, which are realized by the belief propagation (BP) algorithm. We first show that a decision variable obtained by the BP algorithm in a network of distributed agents can be approximated by a linear fusion of all the local log-likelihood ratios. The proposed approach clarifies how the BP algorithm works, simplifies the statistical analysis of its behavior, and enables us to develop a performance optimization framework for the BP-based distributed inference systems. Next, we propose a blind learning-adaptation scheme to optimize the system performance when there is no information available a priori describing the statistical behavior of the wireless environment concerned. In addition, we propose a blind threshold adaptation method to guarantee a certain performance level in a BP-based distributed detection system. To clarify the points discussed, we design a novel linear-BP-based distributed spectrum sensing scheme for cognitive radio networks and illustrate the performance improvement obtained, over an existing BP-based detection method, via computer simulations.

Index Terms—Statistical inference, distributed systems, belief-propagation algorithm, linear data-fusion, Markov random fields, spectrum sensing, blind signal processing.

I. INTRODUCTION

MANY statistical inference problems of practical importance are solved by calculating marginal distribution functions of some random variables represented within a factor graph [1]. These marginals can be well approximated with low computational complexity by using the so-called belief propagation (BP), a.k.a., the sum-product algorithm [2]–[5]. This algorithm works as a message-passing mechanism within a network of interconnected nodes structured according to the factor graph, which encodes Bayesian or Markovian interdependencies between the random variables concerned.

There exist certain obstacles in the application of the BP algorithm to concrete problems [2]. We know that if the BP converges, it is not clear in general whether the results are good approximations of the exact marginals. Even sometimes the BP does not converge and in these cases gives no approximations at all. Due to the nonlinearity of the message update rule in the BP algorithm, analyzing the statistical behavior of the messages (and beliefs) is certainly a challenge. The lack of understanding of the behavior of the BP algorithm makes the optimization of a BP-based inference system difficult. As explained in detail in Section II, this message-passing algorithm works based on factorizing the joint probability distribution of random variables, observed in the network, whose statistical behavior is described by a factor graph. However, fitting a proper factor graph into a network behavior

Y. Abdi and T. Ristaniemi are with the Faculty of Information Technology, University of Jyväskylä, P. O. Box 35, FIN-40014, Jyväskylä, Finland, Tel. +358 40 7214 218 (e-mail:younes.abdi@jyu.fi, tapani.ristaniemi@jyu.fi).

is not an easy task in general. In [6], the major challenges in learning a graphical model from a number of statistically independent data samples are explained.

Even though several works in distributed inference fit a factor graph to the network statistics, they often use heuristic techniques, which do not take into account, within a standard optimization framework, the impact of the parameters of the factor graph on the system performance. In this paper, we fill this gap by using linear data-fusion techniques [7]–[10]. Specifically, through linear approximations we simplify the message update rule in the BP algorithm and reveal important aspects of its behavior and demonstrate how its outcome is affected by the parameters of the underlying factor graph. The link between the linear fusion and the BP algorithm enables us to optimize the performance of the BP-based statistical inference systems in different design scenarios.

A. Related Work

Various approximations of BP have been considered in the literature. In [11] small perturbations are inserted into a BP algorithm, whose messages are formulated by first- and second-order linear approximations, to estimate the joint probability densities of the nodes located far away from each other. In [12] and [13], Gaussian approximations for message probability density functions (pdf) are used to simplify the analysis of the BP-based decoding algorithm for low-density parity-check (LDPC) codes. For turbo codes, a one-dimensional Gaussian approximation based on extrinsic information transfer chart (EXIT chart) is used in [14]–[17]. In [18]–[20] the use of a linear approximate message-passing (AMP) algorithm is discussed assuming that a linear mixing problem is to be solved. In addition, [21] and [22] offer a generalization of the AMP, termed GAMP, which can be used when the mixing structure involves arbitrary, possibly nonlinear, known operations.

Those works on BP-based channel coding schemes are mainly based on empirical results while dealing with factor graphs that are practically cycle-free. In the design scenarios involving AMP and GAMP, the inference problem to be solved is typically well-specified, and approximations are aimed at building a tailored BP to solve the underlying inference problem by using a known mixing structure. In this paper, the aim of the approximation is to achieve a near-optimal detection performance in a wireless sensor network with an unknown topology. The mixing structure is learned through fitting a Markov random field (MRF) [23] to the statistical behavior of the observations in the network while the proposed method effectively optimizes the parameters of the MRF in a distributed detection scenario. We do not assume that the factor graph describing the network configuration is cycle-free.

To clarify the proposed approach, we consider the problem of distributed spectrum sensing in cognitive radio (CR) networks. In these networks, the wireless nodes perform spectrum sensing in bands allocated to the so-called primary users (PU) in order to discover vacant parts of the radio spectrum and establish communication on those temporarily- or spatially-available spectral opportunities [24]. In this context, CRs are considered secondary users (SU) in the sense that they have to vacate the spectrum to avoid making any harmful interference once the PUs are active. Hence, the spectrum sensing capability plays a crucial role in CR networks. The use of BP for distributed spectrum sensing is discussed in [25] and [3] where the dependency among the heterogeneous spectral occupancy states observed by the network nodes is modeled by a binary-valued pairwise MRF. We use a similar graphical model and design a distributed spectrum sensing scheme based on a linear BP algorithm.

B. Belief Propagation vs. Linear Fusion

We consider a statistical inference scenario where a binary hypothesis test is conducted over binary-valued random variables. The test can be formulated as $\hat{\boldsymbol{x}} = \max_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{Y}) = \max_{\boldsymbol{x}} p(\boldsymbol{Y}|\boldsymbol{x})p(\boldsymbol{x})$, where $\boldsymbol{x} = [x_1, ..., x_N]^T$ denotes the vector of N random variables and $\boldsymbol{Y} = [\boldsymbol{y}_1, ..., \boldsymbol{y}_N]$ contains the corresponding observations in the network while $\boldsymbol{y}_i = [y_i(1), ..., y_i(K)]^T$ denotes K observation samples at node i.

Such an optimization, even if all the joint distributions are available, incurs prohibitive computational complexity. In practice, the test is conducted by estimating the marginal distributions, which are then used in a likelihood-ratio test (LRT). That is, $\hat{x}_j = \max_{x_j \in \{0,1\}} p(x_j|\boldsymbol{Y})$ is solved. This test can be stated in vector form as

$$\hat{\boldsymbol{x}} = \mathbf{1}\{\boldsymbol{\lambda} > 0\} \tag{1}$$

where $\mathbf{1}\{X\}$ denotes the indicator function, which equals to one if X is true and to zero otherwise, and $\boldsymbol{\lambda} = [\lambda_1, ..., \lambda_N]^T$ while $\lambda_j \triangleq \ln \frac{p(x_j=1|\boldsymbol{Y})}{p(x_j=0|\boldsymbol{Y})}$ is referred to as the log-likelihood ratio (LLR) of x_j.

In a wireless sensor network, modeling the correlations between the random variables of interest is commonly realized by using a pairwise MRF. Such an MRF is defined on a graph $G = (\mathcal{V}, \mathcal{E})$ composed of a set of vertices or nodes $\mathcal{V} = \{1, ..., N\}$ and a set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. Node $i \in \mathcal{V}$ corresponds to x_i and edge $(i,j) \in \mathcal{E}$ connects nodes i and j to represent a possible correlation between x_i and x_j. Based on this structure, the *a priori* probability measure of $\boldsymbol{x}$ is considered as an exponential function defined as [23]

$$p(\boldsymbol{x}) \propto \exp\left(\sum_{n=1}^{N} \theta_n x_n + \sum_{(i,j) \in \mathcal{E}} J_{ij} x_i x_j\right) \tag{2}$$

where $\propto$ denotes proportionality up to a multiplicative constant. The product over couples (i,j) is only over $i < j$, for which $i \in \mathcal{N}_j$ where $\mathcal{N}_j$ refers to the set of neighbors of node j in the graph, i.e., $\mathcal{N}_j \triangleq \{k : (k,j) \in \mathcal{E}\}$.

Given $\boldsymbol{x}$, the local observations are assumed independent from each other. Consequently, their conditional probability

distribution can be factorized as $p(\boldsymbol{Y}|\boldsymbol{x}) = \prod_{n=1}^{N} p(\boldsymbol{y}_n|x_n)$. This model is commonly used based on the fact that $\boldsymbol{y}_n$ is evaluated at node n, when calculating the local LLR, solely based on the status of x_n. Consequently, in this detection structure x_n works as a sufficient statistic for $\boldsymbol{y}_n$, i.e., $p(\boldsymbol{y}_n|\boldsymbol{x}) = p(\boldsymbol{y}_n|x_n)$. Since the *a posteriori* probability distribution of $\boldsymbol{x}$ can be stated as $p(\boldsymbol{x}|\boldsymbol{Y}) = p(\boldsymbol{Y}|\boldsymbol{x})p(\boldsymbol{x})/p(\boldsymbol{Y})$, from (2) we have

$$p(\boldsymbol{x}|\boldsymbol{Y}) \propto \prod_{n=1}^{N} p(\boldsymbol{y}_n|x_n)e^{\theta_n x_n} \prod_{(i,j) \in \mathcal{E}} e^{J_{ij} x_i x_j} \tag{3}$$

where the proportionality sign includes but is not limited to $1/p(\boldsymbol{Y})$. This model captures the correlations between the random variables of interest in a pairwise structure, which fits well into an ad-hoc network configuration and is used to build a BP algorithm to estimate the required marginals $p(x_j|\boldsymbol{Y})$ in (1). The BP algorithm is described in Section II.

Such an approach to a distributed binary hypothesis test is studied in [3], [25]–[27] in the context of distributed detection. These works mainly focus on how to build the messages and beliefs in a BP iteration to conduct, with low implementation complexity, the required information exchange between nodes. Moreover, they define the elements of the BP algorithm based on simple heuristic approaches to learning the MRF parameters.

Linear data-fusion is an alternative low-complexity suboptimal approach to realizing the desired hypothesis test. In this method, the decision variables are built by a linear combination of the *local* LLRs and then compared against some thresholds. Specifically,

$$\hat{\boldsymbol{x}} = \mathbf{1}\{\boldsymbol{A}^T \boldsymbol{\gamma} - \boldsymbol{\tau} > 0\} \tag{4}$$

where $\boldsymbol{\gamma} = [\gamma_1, ..., \gamma_N]^T$ while $\gamma_j \triangleq \ln \frac{p(\boldsymbol{y}_j|x_j=1)}{p(\boldsymbol{y}_j|x_j=0)}$ denotes the local LLR measured at node j and $\boldsymbol{\tau} = [\tau_1, ..., \tau_N]^T$ denotes the detection thresholds. By optimizing $\boldsymbol{A}$ and $\boldsymbol{\tau}$ a near-optimal detection performance is obtained [7]–[10].

C. Contribution

In this paper, we see the MRF parameters, i.e, θ_n's and J_{ij}'s in (2), as valuable degrees of freedom that can be optimized to achieve a better detection performance. To this end, we use a linear approximation of the BP message update rule to show that learning the MRF parameters to best represent the network behavior can be viewed as optimizing a distributed linear data-fusion scheme in that network. More specifically, we show that when BP is used to realize (1), the detection process achieved is almost equivalent to the one in (4) where $\boldsymbol{A}$ and $\boldsymbol{\tau}$ are determined in terms of θ_n's and J_{ij}'s. Consequently, we show that optimizing θ_n's and J_{ij}'s to have the best possible BP-based detection in (1) is almost equivalent to optimizing $\boldsymbol{A}$ and $\boldsymbol{\tau}$ in (4).

The proposed approach enables us to study the statistical behavior of the messages in the BP algorithm in a practical distributed detection scenario and to formulate the system false-alarm and detection probabilities in terms of the MRF parameters. Consequently, we offer the following contributions:

- We show that a distributed inference based on the BP algorithm and the factor graph defined by (3) can be approximated by a distributed linear data-fusion scheme.
- We propose a linear BP algorithm, derive its convergence conditions, and provide an effective framework for distributed optimization of its performance.
- We design a novel cooperative spectrum sensing scheme in CR networks based on the proposed framework and show that it closely obtains the optimal detection performance.
- We develop a blind learning-optimization scheme for performance optimization of the BP-based distributed spectrum sensing when no information is available a priori regarding the wireless environment.
- We build a blind threshold adaptation mechanism, independent of the proposed linear BP, by which a certain detection performance level is guaranteed when no information is available a priori regarding the wireless environment.

D. Organization

The rest of the paper is organized as follows. In Section II, we give a brief overview of the BP algorithm and demonstrate the link between the BP and distributed linear data-fusion. In Section III, we discuss how to design a BP-based linear fusion scheme in both centralized and decentralized network configurations. In Section IV, we show how to implement the proposed optimization framework when no statistical information is available a priori. In Section V, we derive the convergence conditions for the proposed linear BP. We discuss the impact of cycles in Section VI, illustrate the performance of the proposed method via computer simulations in Section VII, and provide the concluding remarks in Section VIII.

II. BELIEF PROPAGATION ON A FACTOR GRAPH

A general pairwise MRF defined on factor graph G to factorize the a posteriori distribution function $p(\boldsymbol{x}|\boldsymbol{Y})$ into single-variable and pairwise terms, can be expressed as

$$p(\boldsymbol{x}|\boldsymbol{Y}) \propto \prod_{n=1}^{N} \phi_n(x_n) \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j) \tag{5}$$

where each single-variable term $\phi_n(x_n)$ models the impact of x_n in the joint distribution whereas each pairwise term $\psi_{ij}(x_i, x_j)$ represents the interdependency of x_i and x_j connected by an edge in the graph. Note that, $\boldsymbol{Y}$ is implicitly included in the right side of (5) within the structures of ϕ_n's and ψ_{ij}'s and also in the multiplicative constant.

The main goal of each node, say node i, is to find its marginal a posteriori distribution $p(x_i|\boldsymbol{Y})$. This goal can be achieved by using the BP algorithm in which the message from node k to node j, at the lth iteration, is formulated as

$$\mu_{k\to j}^{(l)}(x_j) \propto \sum_{x_k} \phi_k(x_k)\psi_{kj}(x_k, x_j) \prod_{n\in\mathcal{N}_k^j} \mu_{n\to k}^{(l-1)}(x_k) \tag{6}$$

where $\mathcal{N}_k^j \triangleq \mathcal{N}_k \backslash \{j\}$ denotes all neighbors of node k except for node j. The so-called *belief* of node j regarding x_j,

denoted $b_j^{(l)}(x_j)$, is formed by multiplying the local inference result at node j by all the messages received from its neighbors, i.e.,

$$b_j^{(l)}(x_j) \propto \phi_j(x_j) \prod_{k\in\mathcal{N}_j} \mu_{k\to j}^{(l)}(x_j) \tag{7}$$

which is used to approximate the desired marginal distribution, i.e., $b_j^{(l)}(x_j) \approx p(x_j|\boldsymbol{Y})$. The proportionality signs in (6) and (7) indicate that the beliefs and messages are expressed up to a constant, which can be found by normalizing $b(\cdot)$ and $\mu(\cdot)$ so as to sum to 1.

From (3), we see that for the binary hypothesis test concerned the BP message update rule can be stated as

$$\mu_{k\to j}^{(l)}(x_j) \propto \sum_{x_k} p(\boldsymbol{y}_k|x_k)e^{\theta_k x_k}e^{J_{kj}x_k x_j} \prod_{n\in\mathcal{N}_k^j} \mu_{n\to k}^{(l-1)}(x_k) \tag{8}$$

while the beliefs are updated according to

$$b_j^{(l)}(x_j) \propto p(\boldsymbol{y}_j|x_j)e^{\theta_j x_j} \prod_{k\in\mathcal{N}_j} \mu_{k\to j}^{(l)}(x_j) \tag{9}$$

The message update rule can be expressed in terms of the belief and message LLRs defined, respectively, as

$$\lambda_j^{(l)} \triangleq \ln \frac{b_j^{(l)}(x_j = 1)}{b_j^{(l)}(x_j = 0)} \tag{10}$$

$$m_{k\to j}^{(l)} \triangleq \ln \frac{\mu_{k\to j}^{(l)}(x_j = 1)}{\mu_{k\to j}^{(l)}(x_j = 0)} \tag{11}$$

where $\lambda_j^{(l)}$ can be used as an estimation of the LLR of x_j and $m_{k\to j}^{(l)}$ can be used as the message sent from node k to node j. By this change of variables, we have

$$m_{k\to j}^{(l)} = S\left(J_{kj},\ \gamma_k + \sum_{n\in\mathcal{N}_k^j} m_{n\to k}^{(l-1)}\right) \tag{12}$$

$$\lambda_j^{(l)} = \gamma_j + \sum_{k\in\mathcal{N}_j} m_{k\to j}^{(l)} \tag{13}$$

where $S(a,b) \triangleq \ln \frac{1+e^{a+b}}{e^a+e^b}$. After l^* iterations, $\lambda_j^{(l^*)}$ is used at node j as a decision variable to perform an LRT, i.e., $\hat{x}_j = 1$ if $\lambda_j^{(l^*)} \geq \tau_j$ and $\hat{x}_j = 0$ otherwise. In (12), we have replaced $2\theta_k + \gamma_k$ by γ_k and $2J_{kj}$ by J_{kj} since $2\theta_k$ can be merged into γ_k and 2 can be merged into J_{ij} with no impact on the rest of the analysis. Optimization of θ_k's is covered in the proposed framework through considering the detection thresholds as optimization variables.

In order to formulate the system performance in terms of the commonly-used metrics, the probability distribution of λ_j is needed. This is a major challenge since, first, the probability distribution of γ_i's are not known a priori. And second, even if they are known somehow, the nonlinearity of the message-update rule in (12) does not allow us to find the distribution of λ_j and the resulting performance in closed form. In [25], a BP-based distributed detection method is introduced. However, due to these challenges, the detection threshold is determined through a heuristic approach and the
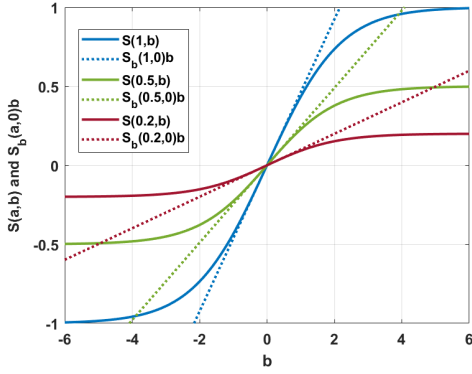
Fig. 1: Nonlinear transformation $S(a, b)$ along with its linear approximation $S_b(a, 0)b$. We replace S, which acts as a sigmoid function, with a linear transfer function.

system performance optimization is not discussed. In this paper we build an analytical framework to address these issues.

Eq. (13) shows that the decision variable at node j, i.e., λ_j, is built by the local LLR obtained at node j plus a linear combination of the messages received from the neighboring nodes. The messages, however, pass through a nonlinear transformation S, see Fig. 1. In (12) and (13) we clearly see how the nonlinearity of the BP algorithm transforms the likelihood values in the factor graph.

A closer look at (12) reveals that $S(a, b)$ plays a role similar to the role of the sigmoid activation functions typically used in artificial neural networks [28]. In certain neural-network structures, each neuron generates an outcome by applying a bounded nonlinear transformation on the sum of its inputs received from other neurons in the network. This transfer function keeps the network stable by smoothening the neuron outcomes as well as by restricting them to reside between an upper and a lower bound. Consequently, the signals passed from one neuron to another become less sensitive to sharp variations in the inputs and the mapping learned by the network does not result in unbounded values for the variables of interest.

We propose to replace the nonlinear functions in the BP algorithm by linear ones. However, we design the proposed linear transformations in the network while taking into account the convergence condition of the distributed message-passing algorithm obtained. Having linear message-update rules in the BP algorithm enables us to analyze the network dynamics and develop effective optimization techniques to enhance system performance.

We realize the proposed linearization by approximating the message transfer function S. Specifically, we use the first-order Taylor series expansion of $S(a, b)$ with respect to its second argument, i.e., $S(a, b) \approx S_b(a, 0)b$, where $S_b(a, b) \triangleq \frac{\partial S(a,b)}{\partial b}$. This leads to a linear message-update rule as

$$m_{k \to j}^{(l)} \approx c_{jk} \left(\gamma_k + \sum_{n \in \mathcal{N}_k^j} m_{n \to k}^{(l-1)} \right) \tag{14}$$

where $c_{jk} = \frac{(e^{2J_{kj}} - 1)}{(1 + e^{J_{kj}})^2}$.

By applying this approximation on (12) and (13), while iterating the resulting linear message-update rule, we obtain

$$\lambda_j \approx \gamma_j + \sum_{k \in \mathcal{N}_j} c_{jk} \gamma_k + \sum_{k \in \mathcal{N}_j} \sum_{n \in \mathcal{N}_k^j} c_{jk} c_{kn} \gamma_n$$
$$+ \sum_{k \in \mathcal{N}_j} \sum_{n \in \mathcal{N}_k^j} \sum_{m \in \mathcal{N}_n^k} c_{jk} c_{kn} c_{nm} \gamma_m + ... \tag{15}$$

which shows that, given enough time, the overall result of the BP algorithm at a node is obtained, approximately, by a linear combination of all the local LLRs in the network. Note that we do not need the iteration index (l) in this expression since (15) denotes the final result of the iterations.

Let $(V, \| \cdot \|)$ denote a normed space where $f : V \to V$ is defined and let $\boldsymbol{m}^{(l)}$ denote a vector containing all the messages at iteration l. A general message-passing algorithm is defined by $\boldsymbol{m}^{(l)} = f^l(\boldsymbol{m}^{(0)})$ where f indicates the message-update rule. Convergence of such an iteration to a so-called *fixed point* means that the sequence $\boldsymbol{m}, f(\boldsymbol{m}), f^2(\boldsymbol{m}), ...$ obtained by iterating f converges to a unique fixed point for any $\boldsymbol{m} \in V$. Consequently, λ_j in (15) denotes (the decision variable made by) the fixed point of the linear BP algorithm in (14) and an approximation of (the decision variable made by) the fixed point attained via the BP in (12).

The linearization in (14) can be interpreted in two ways. One can see it as a linear approximation to the message-update rule in (12). Such an interpretation is valid since when γ_i's are Gaussian (which is a common practical case), one can normalize them without affecting the detection result. Specifically, when the decision variable follows a Gaussian distribution, for a constant false-alarm probability the probability of detection does not change by normalizing the decision variable. Consequently, by proper normalization of γ_i's, the messages can be made small enough to experience a linear behavior in the transfer functions.

Alternatively, one can opt not to normalize the local LLRs and to consider (14) as a new message-update rule, which is obtained directly by replacing the nonlinear transfer functions in the network by linear ones. Note that when the message values are too large to fall within the linear part of the transfer functions, the saturation behavior in S acts as a thresholding process leading to a so-called hard fusion of the local LLRs. By using the linear transfer function in (14), the hard fusion rule is replaced by a soft one in which the sensing outcomes are directly combined without passing through that threshold. Since a soft fusion method outperforms a hard fusion one in general, one can expect a better overall detection result obtained by this linearization. A hard fusion scheme refers to a combination of binary-valued variables obtained by comparing the local LLRs against (local) decision thresholds. See [29] for a comparison between the hard and soft decision methods.

Remark 1: Regardless of how (14) is interpreted, since the sigmoid functions are now removed, the linear transfer functions have to be designed properly to guarantee the convergence of the new message-update rule. We will discuss the convergence condition in Section V.

Remark 2: We have configured the linear combination in (15) such that it reveals the effect of the network topology

on how the fusion coefficients are arranged. Specifically, for the one-hop neighbors of node j we have one coefficient c_{jk} affecting the local likelihoods received, for the two-hop neighbors we have two coefficients $c_{jk}c_{kn}$ and so on. We will use this observation in Section III-B to develop a decentralized distributed inference method.

From (15) it is clear that, determining c_{jk}'s for all $(k,j) \in \mathcal{E}$ in the network based on a set of performance criteria can be viewed as determining the values of J_{kj}'s in the MRF. In [3], [25], the BP is designed and analyzed without directly considering the problem of fitting an MRF to the network behavior. Specifically, in those works, J_{kj}'s are set by an empirical estimation of the correlations between the neighboring nodes in a window of T time slots, i.e.,

$$J_{kj} \triangleq \frac{\zeta}{T} \sum_{t=1}^{T} \mathbf{1}\{\hat{x}_j(t) = \hat{x}_k(t)\} - \mathbf{1}\{\hat{x}_j(t) \neq \hat{x}_k(t)\} \quad (16)$$

where ζ, referred to as the learning factor in this paper, is considered as a small constant and T denotes the number of samples used in this training process. We will see in Section VII why ζ should be small. Eq. (12) shows that J_{kj}'s determine the behavior of the transfer functions applied on the BP messages. Consequently, optimizing J_{kj}'s can be viewed as finding an optimal set of linear transfer functions for the BP algorithm.

III. LINEAR FUSION BASED ON BELIEF PROPAGATION

In this section, we use the link between BP and linear data-fusion to design distributed inference methods in both *centralized* and *decentralized* network configurations. In a centralized setting [24], there is a special node in the network, referred to as the fusion center (FC), which collects data from the sensing nodes, coordinates the distributed inference, and builds the overall inference outcome based on the collected data. In a decentralized (i.e., ad-hoc) network, however, there is no FC and the statistical inference is realized through the so-called peer-to-peer communication between the nodes.

The optimization framework developed in this section is inline with practical design scenarios wisely-used in the literature. Our foundation here is the linear fusion structure in (15), which is mainly based on (2) and (5), which together build a relatively generic model for a distributed binary hypothesis test. The use of the exponential form in (2) to model the behavior of correlated binary-valued random variables is supported by the *principle of maximum entropy* [23]. The use of the false-alarm and detection probabilities facilitates the design of the CR network concerned since the former is directly linked to the CR network throughput while the latter determines the level of interference experienced by the PUs. However, as explained in *Remark 5*, we are not concerned with the type of the objective and constraint functions in our optimization framework. Therefore, we do not lose generality by using these performance metrics.

A. Centralized Distributed Inference

Eq. (15) can compactly be stated as $\lambda_j = \sum_{i=1}^{N} a_{ji}\gamma_i$ or, in matrix form, as

$$\boldsymbol{\lambda} = \boldsymbol{A}^T \boldsymbol{\gamma} \quad (17)$$

where $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_N]$ while $\boldsymbol{a}_j = [a_{j1}, ..., a_{jN}]^T$. Assuming $U(\boldsymbol{\lambda})$ as the objective function corresponding to a certain application and $C_i(\boldsymbol{\lambda})$ as one of N_C constraint functions indicating the requirements to be met by the system design, the optimal linear fusion in the network can be obtained by

$$\max_{\boldsymbol{A}, \boldsymbol{\tau}} U(\boldsymbol{\lambda}), \quad \text{s.t.,} \quad C_i(\boldsymbol{\lambda}) \leq \alpha_i, \quad i = 1, ..., N_C \quad \text{(P1)}$$

We further clarify our points by considering a distributed spectrum sensing scenario where each node, say node j, performs a binary hypothesis test on the decision variable λ_j to decide the availability of the radio spectrum.

In this scenario, x_j is a binary-valued random variable representing the occupancy state of the radio spectrum where node j is located. Let's say, $x_j = 1$ indicates that the spectrum band sensed by node j is occupied whereas $x_j = 0$ indicates the availability of that band for the secondary use. The signal samples used to form the test statistics at node j can be expressed as

$$y_j(k) = \sum_n u_n h_{jn} s_n(k) + \nu_j(k) \quad (18)$$

where $\nu_j(k) \sim \mathcal{CN}(0, \sigma_\nu^2)$ denotes a zero-mean complex-valued Gaussian random variable, which models the additive white noise at node j while $s_n(k)$, also a zero-mean complex-valued Gaussian random variable, represents the signal transmitted by the nth PU transmitter with $E[|s_n(k)|^2] = \sigma_s^2$. The channel gain from the nth PU transmitter to node j, denoted h_{jn}, is assumed constant during the sensing time interval. u_n denotes a binary random variable, which represents the state of the nth PU transmitter, i.e., the nth PU transmitter is on when $u_n = 1$ and is off otherwise. u_n is also assumed constant during the sensing time interval. According to this model, $x_j = 0$ means that no PU signal is received at node j, i.e., $u_n = 0$ for all n and the spectrum is available for the secondary use whereas $x_j = 1$ means that at least one PU is active and the spectrum band sensed by node j is occupied and cannot be used by the SUs.

Given the state of x_j, if we assume as in [25] that $\boldsymbol{y}_j$ is a vector of i.i.d. complex Gaussian signal samples, then we have

$$\gamma_j = \frac{-N}{2} \ln(1 + \rho_j) + \frac{\rho_j}{1 + \rho_j} \frac{\|\boldsymbol{y}_j\|^2}{2\sigma_\nu^2} \quad (19)$$

where $\rho_j \triangleq |h_j|^2 \sigma_s^2 / \sigma_\nu^2$ denotes the received signal-to-noise ratio (SNR) at node j. Therefore, one needs to know the channel (power) gain, the transmitted signal power, and the noise level to obtain the probability distribution of γ_j.

In order to keep the resulting false-alarm rate below a predefined constraint α, in [25] γ_j in (19) is replaced by

$$\gamma_j \triangleq \frac{1}{K} \|\boldsymbol{y}_j\|^2 - \tau_0 \quad (20)$$

where τ_0 is set such that $\Pr\{\gamma_j > 0 | x_j = 0\} = \alpha$, which leads to

$$\tau_0 = \sigma_\nu^2 \left(1 + \sqrt{\frac{2}{K}} Q^{-1}(\alpha)\right) \quad (21)$$

Hence, τ_0 can be simply calculated in terms of the noise level and without the need for the channel gain and the transmitted power level.

The proposed structure remains intact if we use $\gamma_j \triangleq \|\boldsymbol{y}_j\|^2/K$ instead of (19). All other variables in (19) along with τ_0 in (20) can be absorbed either into the coefficients in (15) or into the detection threshold τ_j. Consequently, we can justify here why the BP algorithm works well in [25] where energy detection is adopted as the local sensing method. As we show in Section VII, τ_0 in (20) works well only for small values of α. We propose a new threshold adaptation scheme in Section IV-B, which can always guarantee the false-alarm rate of the BP algorithm fall below a predefined upper bound without the need for any statistics other than the local noise level.

The performance of the binary hypothesis test discussed is formulated by the so-called false-alarm and detection probabilities defined, respectively, at node j as $P_{\mathrm{f}}^{(j)} \triangleq \Pr\{\lambda_j \geq \tau_j|x_j = 0\}$ and $P_{\mathrm{d}}^{(j)} \triangleq \Pr\{\lambda_j \geq \tau_j|x_j = 1\}$. The proposed transfer functions are linear and, given the status of x_j's, our γ_i's are Gaussian random variables. Consequently, we can express the proposed system false-alarm and detection probabilities in closed form. Specifically, at node j and for $v = 0, 1$ we have

$$
\begin{aligned}
g_j(\tau_j, v) &\triangleq \Pr\{\lambda_j > \tau_j|x_j = v\} \\
&= \sum_{\boldsymbol{b} \in \{0,1\}^{N-1}} \Pr\{\lambda_j > \tau_j|\boldsymbol{x}_{(j)} = \boldsymbol{b}, x_j = v\} p_{\boldsymbol{x}_{(j)}|x_j}(\boldsymbol{b}|v) \\
&= \sum_{\boldsymbol{b} \in \{0,1\}^{N-1}} Q\left(\frac{\tau_j - \eta_{j,v}(\boldsymbol{b})}{\sigma_{j,v}(\boldsymbol{b})}\right) p_{\boldsymbol{x}_{(j)}|x_j}(\boldsymbol{b}|v) \quad (22)
\end{aligned}
$$

where $\boldsymbol{x}_{(j)} \triangleq [x_1, x_2, ..., x_{j-1}, x_{j+1}, x_{j+2}, ..., x_N]$, $p_{\boldsymbol{x}_{(j)}|x_j}(\boldsymbol{b}|v) \triangleq \Pr\{\boldsymbol{x}_{(j)} = \boldsymbol{b}|x_j = v\}$, and for $v = 0, 1$, $\eta_{j,v}(\boldsymbol{b}) \triangleq E[\lambda_j|\boldsymbol{x}_{(j)} = \boldsymbol{b}, x_j = v]$ and $\sigma_{j,v}^2(\boldsymbol{b}) \triangleq \mathrm{Var}[\lambda_j|\boldsymbol{x}_{(j)} = \boldsymbol{b}, x_j = v]$. $Q(x) \triangleq \int_x^\infty \frac{1}{\sqrt{2\pi}}e^{-z^2/2}dz$ is the so-called Q-function. Note that $\boldsymbol{x}_{(j)}$ contains all x_i's except for x_j. It is clear that $P_{\mathrm{f}}^{(j)} = g_j(\tau_j, 0)$ and $P_{\mathrm{d}}^{(j)} = g_j(\tau_j, 1)$. By solving $g_j(\tau_j, 0) = \alpha$ or $g_j(\tau_j, 1) = \beta$ we obtain a value for τ_j that guarantees the false-alarm or detection probability at node j to be, respectively, equal to α or β. The fact that, given x_i's, λ_j is Gaussian comes from (15).

In a CR network, the higher the false-alarm probability the more vacant spectrum bands are mistakenly treated as occupied leading to a loss in the network throughput. Consequently, the opportunistic spectral utilization of the jth band is measured by $1 - P_{\mathrm{f}}^{(j)}$. In addition, the higher the misdetection probability the higher the interference level imposed on the primary users. Hence, the interference level on the jth band is measured by $1 - P_{\mathrm{d}}^{(j)}$. In a rational design strategy, interference on PUs is associated with a cost.

We collect these performance metrics regarding the entire network in vectors $\boldsymbol{P}_{\mathrm{f}} \triangleq [P_{\mathrm{f}}^{(1)}, ..., P_{\mathrm{f}}^{(N)}]^T$ and $\boldsymbol{P}_{\mathrm{d}} \triangleq [P_{\mathrm{d}}^{(1)}, ..., P_{\mathrm{d}}^{(N)}]^T$. Consequently, we can express the network

performance by the following quantities

$$
R(\boldsymbol{\lambda}, \boldsymbol{\tau}) \triangleq \boldsymbol{r}^T [\mathbf{1} - \boldsymbol{P}_{\mathrm{f}}(\boldsymbol{\lambda}, \boldsymbol{\tau})] \quad (23)
$$
$$
I(\boldsymbol{\lambda}, \boldsymbol{\tau}) \triangleq \boldsymbol{q}^T [\mathbf{1} - \boldsymbol{P}_{\mathrm{d}}(\boldsymbol{\lambda}, \boldsymbol{\tau})] \quad (24)
$$

where $\mathbf{1}$ denotes an all-one vector, $R(\boldsymbol{\lambda}, \boldsymbol{\tau})$ denotes the *aggregate network throughput* calculated over all spectrum bands sensed across the whole network and $I(\boldsymbol{\lambda}, \boldsymbol{\tau})$ denotes the *aggregate interference* caused on the PUs operating on those bands, where $\boldsymbol{\tau} \triangleq [\tau_1, ..., \tau_N]^T$ and $\boldsymbol{r} \triangleq [r_1, ..., r_N]^T$ while r_i denotes the throughput achieved by using the ith band. The cost of interference caused on the ith band is denoted by q_i which is included in $\boldsymbol{q} \triangleq [q_1, ..., q_N]^T$.

Now, the optimal sensing performance is obtained by maximizing the aggregate system throughput subject to a constraint on the aggregate interference as well as the per channel false-alarm and misdetection constraints, i.e.,

$$
\max_{\boldsymbol{A}, \boldsymbol{\tau}} R(\boldsymbol{\lambda}, \boldsymbol{\tau}) \quad (\mathrm{P2})
$$
$$
\mathrm{s.t.} \quad I(\boldsymbol{\lambda}, \boldsymbol{\tau}) \leq I_0, \quad \boldsymbol{P}_{\mathrm{f}}(\boldsymbol{\lambda}, \boldsymbol{\tau}) \leq \boldsymbol{\alpha}, \quad \boldsymbol{P}_{\mathrm{d}}(\boldsymbol{\lambda}, \boldsymbol{\tau}) \geq \boldsymbol{\beta}
$$

where $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_N]^T$ and $\boldsymbol{\beta} = [\beta_1, ..., \beta_N]^T$ impose per-channel constraints on the false-alarm and misdetection probabilities. The *sequential optimization* method in [8], which maximizes the so-called *deflection coefficient* [30] of the detector, can be used to solve (P2) for a good suboptimal performance.

The deflection coefficient associated with λ_j is defined as

$$
\Delta_j \triangleq \frac{E[\lambda_j|x_j = 1] - E[\lambda_j|x_j = 0]}{\sqrt{\mathrm{Var}[\lambda_j|x_j = 0]}} \quad (25)
$$

where, from (17), $E[\lambda_j|x_j] = \sum_{i=1}^N a_{ji}E[\gamma_i|x_j]$ and $\mathrm{Var}[\lambda_j|x_j] = \sum_{i=1}^N \sum_{k=1}^N a_{ji}a_{jk}\mathrm{cov}(\gamma_i, \gamma_k|x_j)$. Hence, to maximize the deflection coefficients in the network, we only need $E[\gamma_i|x_j]$ and $\mathrm{cov}(\gamma_i, \gamma_k|x_j)$ for all $j \in \mathcal{V}$ and $i, k \in \mathcal{E}$. These statistics are clearly easier to obtain than $\eta_{j,v}(\boldsymbol{b})$, $\sigma_{j,v}^2(\boldsymbol{b})$, and $p_{\boldsymbol{x}_{(j)}|x_j}(\boldsymbol{b}|v)$ in (22).

Availability of the statistics discussed may require a centralized configuration in which an FC collects data from the nodes. As explained in [31, Sec. II-C], the standard preambles or synchronization symbols, which typically reside within the PU signals, can facilitate the estimation of the desired statistics in a centralized setting. In Section III-B, we show how to optimize the system performance in a decentralized network and when those statistics are not available a priori.

Remark 3: Due to the nonlinearity of the message update rule in (12), the system false-alarm and detection probabilities are not available in closed form when the BP algorithm is used and, therefore, the performance optimization in (P2) or even guaranteeing a predefined level of performance in a BP-based system analytically is a very challenging task, if possible.

B. Decentralized Distributed Inference

Message-passing algorithms are of special interest in the decentralized network configurations where there is no FC, the communication between nodes is limited to one-hop neighbors, and, typically, there are no statistics available a priori regarding the wireless environment.

In order to build a decentralized distributed inference, we make three particular observations here. Firstly, note that $|c_{jk}| < 1$ for all $(j, k) \in \mathcal{E}$. Moreover, as shown in Section VI, we can use ρc_{jk}'s as the fusion coefficients, where $0 < \rho < 1$, without affecting the detection probability obtained. Scaling down the coefficients is an effective method to mitigate the impact of graph cycles on the system performance.

Secondly, our MRF in (5) captures the correlations between the random variables by a set of pairwise links. In particular, if node n is connected to node j through node k, the correlation between x_n and x_j is accounted for in (5) by two factors $\psi_{k,n}(x_k, x_n)$ and $\psi_{jk}(x_j, x_k)$ multiplied together within $p(\boldsymbol{x}|\boldsymbol{Y})$. A similar process is observed in (15) where the impact of γ_n on λ_j, which depends on the correlation between x_n and x_j, is determined by multiplying two factors c_{kn} and c_{jk} corresponding, respectively, to the link from node n to node k and the link from node k to node j.

Thirdly, the fusion process in (15) is inherently symmetric. That is, each node performs the same local fusion on the data received from its neighbors and passes the result on along the path towards node j. We use this symmetric structure, which is clarified in (26), to distribute the load of the desired optimization across all nodes in the network. Specifically, we assign each node the task of optimizing its own local fusion process. In this manner, each node has to deal with a few variables and with the statistics received from its one-hope neighbors only. The fact that $|c_{jk}| < 1$ makes this collection of localized optimizations a reliable distributed approach to the system performance optimization. The reason is that γ_i's received from neighbors with longer distances are multiplied by higher-order terms in the data-fusion process. Consequently, the system inherently favors, at each node, the data received from the close neighbors over the data from the more-distant ones.

Therefore, firstly, we further approximate (15) by keeping the first-order terms, i.e.,

$$\lambda_j \approx \gamma_j + \sum_{k \in \mathcal{N}_j} c_{jk}\gamma_k \qquad (27)$$

We use $\lambda_j \approx \sum_{k \in \mathcal{M}_j} c_{jk}\gamma_k$, where $\mathcal{M}_j \triangleq \{j\} \cup \mathcal{N}_j$, when optimizing the coefficients. In the vector form we have $\lambda_j \approx \mathbf{c}_j^T \boldsymbol{\gamma}_j$ where $\mathbf{c}_j$ contains all c_{jk}'s such that $k \in \mathcal{M}_j$ and $\boldsymbol{\gamma}_j$ contains the local likelihood ratios at node j and at all its neighbors.

Remark 4: The approximation in (27) is only used to realize the optimization of c_{jk}'s in a decentralized configuration. The resulting coefficients are then used in a linear message-update rule as in (14). Hence, the overall impact of the distributed linear BP obtained is still approximated by (15).

Secondly, we optimize $\mathbf{c}_j$ by considering the one-hop neighbors of node j. In this manner, the correlation between x_j and x_k is captured in c_{jk} while the correlation between x_k and x_n is accounted for by c_{kn}. Hence, both of the correlations concerned are taken into account in the system design by the multiplication $c_{jk}c_{kn}$ in (15) while each node sees its immediate neighbors only. This optimization can be performed conveniently in a distributed fashion since each node has to deal with only a few variables.

Contrary to the linear fusion method discussed in Section III-A, here we have an iterative message-passing algorithm whose convergence needs to be guaranteed. In Section V, we show that the convergence is guaranteed when we have $|c_{j,k}| < \frac{1}{\max_n |\mathcal{N}_n| - 1}, \forall (j, k) \in \mathcal{E}$, which is a convex constraint on the coefficients. Therefore, the proposed decentralized linear fusion scheme is obtained by solving the following optimization at node j, $j = 1, ..., N$,

$$\max_{\mathbf{c}_j, \tau_j} U_j(\lambda_j) \qquad (P3)$$
$$\text{s.t.,} \quad C_{ji}(\lambda_j) \leq \alpha_{ji}, \quad \text{for } i = 1, ..., N_{C_j}$$
$$|c_{j,k}| < \frac{1}{\max_n |\mathcal{N}_n| - 1}, \forall k \in \mathcal{N}_j$$

where $U_j(\lambda_j)$, $C_{ji}(\lambda_j)$, and N_{C_j} denote the application-specific local objective function, the local constraint functions, and the number of constraints at node j, respectively. The major advantage of this optimization is that the local objective and constraint functions are calculated in terms of the statistics of λ_j and these statistics are now conveniently obtained due to the linearity of the proposed message update rule.

It is worth noting that, considering the overall system performance, the use of (27) does not mean that the impact of the higher-order terms on λ_j is ignored. Instead, the task of optimizing those terms are assigned to the other nodes in the network. It is clear that by this approximation we achieve a suboptimal solution. However, this solution distributes the optimization load well over the entire network by exploiting the symmetry in the overall data-fusion process achieved.

In the rest of the paper, we use c_{jk}^{BP} to denote the coefficients in the linear approximation of the BP algorithm, i.e., $c_{jk}^{\text{BP}} \triangleq (e^{2J_{kj}} - 1)/(1 + e^{J_{kj}})^2$, to distinguish those coefficients from the ones obtained by the proposed optimization. In addition, we use $\mathbf{c}_j^{\text{BP}}$ to contain all c_{jk}^{BP}'s such that $k \in \mathcal{M}_j$.

In a spectrum sensing scenario, the proposed optimization is built by the well-known Neyman-Pearson hypothesis test [30] at each node. Specifically, at each node we aim at maximizing the detection probability subject to an upper bound on the false-alarm probability, i.e.,

$$\max_{\mathbf{c}_j, \tau_j} P_{\text{d}}^{(j)}(\lambda_j) \qquad (P4)$$
$$\text{s.t.,} \quad P_{\text{f}}^{(j)}(\lambda_j) \leq \alpha_j$$
$$|c_{j,k}| < \frac{1}{\max_n |\mathcal{N}_n| - 1}, \forall k \in \mathcal{N}_j$$

where $P_{\text{f}}^{(j)}$ and $P_{\text{d}}^{(j)}$ are derived, approximately, in a decentralized fashion as described in the following.

The conditional probabilities in (22), which govern the statistical behavior of the test summary λ_j at node j, take into account any possible correlations between x_j and all other x_i's. Some nodes may, in fact, be located far away from node j and their observations may have, if any, a little or negligible impact on the detection result at node j. Hence, we simplify the calculations by limiting the summation in (22) to include only the correlations between x_j and x_i's observed by the one-hop neighbors of node j. These statistics can be obtained by each node in a network with a decentralized configuration.

$$\lambda_j \approx \gamma_j + \underbrace{\sum_{k \in \mathcal{N}_j} c_{jk} \left(\gamma_k + \underbrace{\sum_{n \in \mathcal{N}_k^j} c_{kn} \left(\gamma_n + \underbrace{\sum_{m \in \mathcal{N}_n^k} c_{nm} \left(\gamma_m + ... \right.}_{\text{optimized by node } n} \right.}_{\text{optimized by node } k} \right.}_{\text{optimized by node } j} \qquad (26)$$

The proposed approximation is expressed as

$$g_j(\tau_j, v) \approx \sum_{\boldsymbol{b} \in \{0,1\}^{|\mathcal{N}_j|}} Q\left(\frac{\tau_j - \tilde{\eta}_{j,v}(\boldsymbol{b})}{\tilde{\sigma}_{j,v}(\boldsymbol{b})} \right) p_{\tilde{\boldsymbol{x}}_{(j)}|x_j}(\boldsymbol{b}|v) \quad (28)$$

where $\tilde{\boldsymbol{x}}_{(j)}$ is a vector which contains x_i's with $i \in \mathcal{N}_j$ while for $v = 0, 1$, we have $\tilde{\eta}_{j,v}(\boldsymbol{b}) \triangleq E[\lambda_j | \tilde{\boldsymbol{x}}_{(j)} = \boldsymbol{b}, x_j = v]$ and $\tilde{\sigma}_{j,v}^2(\boldsymbol{b}) \triangleq \text{Var}[\lambda_j | \tilde{\boldsymbol{x}}_{(j)} = \boldsymbol{b}, x_j = v]$. Note that $|\mathcal{N}_j|$ denotes the number of one-hop neighbors of node j while $\tilde{\eta}_{j,v}$ and $\tilde{\sigma}_{j,v}$ denote an estimation of the first- and second-order conditional statistics of the test summary at node j given the value of x_j and its immediate neighbors. λ_j is approximated as in (27).

Based on (27) we have $E[\lambda_j | x_j] \approx \sum_{k \in \mathcal{M}_j} c_{jk} E[\gamma_k | x_j]$ and $\text{Var}[\lambda_j | x_j] \approx \sum_{n \in \mathcal{M}_j} \sum_{k \in \mathcal{M}_j} c_{jn} c_{jk} \text{cov}(\gamma_n, \gamma_k | x_j)$. Hence, the use of the deflection coefficient here can simplify the design even further compared to solving (P4). Maximizing the deflection coefficient is a classic low-complexity approach to optimal linear fusion which is considered in e.g., [7]–[9].

Remark 5: Since each node is typically connected to only a few neighbors, regardless of the scale of the network, each node has to deal only with a few variables when running the proposed optimization (and, as discussed later, when estimating the required statistics). Consequently, the scalability of the proposed detection scheme does not depend on the low-complexity optimization techniques in [7]–[9].

IV. BLIND DISTRIBUTED DETECTION

We propose the notions of *adaptive linear belief propagation* and *detector calibration* here to realize a near-optimal distributed detection performance when there is no information available a priori regarding the statistical behavior of the radio environment.

A. Offline Learning and Optimization

As we saw in Section III, in order to optimize the fusion weights in the proposed linear message-passing algorithm, we need the first- and second-order statistics of the local LLRs. Since we do not assume any prior knowledge about the channel statistics or the behavior of the PUs, we have to estimate those parameters based on the detection outcome. This is a challenging task since the detection outcome itself depends on those estimates. Erroneous detection outcomes lead to erroneous estimates of the desired statistics, which, in turn, deteriorate the detection results. We overcome this challenge by implanting an adaptation process within the linear BP algorithm, which is run offline on a set of data samples collected and processed for channel estimation and detector optimization purposes. This adaptation is realized by an offline linear BP algorithm whose outcome is a set of near-optimal

coefficients used in the main linear BP algorithm, which is responsible for spectrum sensing in real time.

In a blind detection scheme the only source of information is the detector outcome. Hence, we have to use $E[\lambda_j | \hat{x}_j]$ and $\text{cov}(\gamma_i, \gamma_k | \hat{x}_j)$ to estimate $E[\gamma_i | x_j]$ and $\text{cov}(\gamma_i, \gamma_k | x_j)$ for all $i, k \in \mathcal{M}_j$ for all $j \in \mathcal{V}$. These estimates are erroneous in nature due to the non-zero false-alarm and misdetection probabilities in the system. As an example note that, to estimate $E[\gamma_i | x_j = 0]$ node j needs to calculate an average over the samples of γ_i received when $x_j = 0$. However, node j does not exactly know when $x_j = 0$ and it can only calculate $E[\gamma_i | \hat{x}_j = 0]$. When the sensing outcome indicates $\hat{x}_j = 0$, it may be, in fact, a misdetection and including the γ_i sample in the averaging process may lead to a significant error in that case. Note that γ_i samples can vary widely in size depending on the status of x_j. Similar errors can also occur due to the false-alarm incidents.

Therefore, we need to enhance the quality of sensing in order to have high-quality estimates of the desired statistics which, in turn, are used to achieve a better sensing quality. Clearly, we are facing a loop here. Hence, we design an iterative learning-optimization loop to optimize the proposed message-passing algorithm. This loop works as an offline linear BP algorithm that operates on a window of T samples of $\boldsymbol{\gamma}$.

In the following, we use κ (instead of l) to denote the iteration index of the offline linear BP algorithm in order to distinguish this iteration from the main linear BP algorithm. In addition, we use $\boldsymbol{\gamma}_T$ to denote a window of T samples of $\boldsymbol{\gamma}$ stored for the proposed offline data processing and $\hat{\boldsymbol{x}}^{(\kappa)} \triangleq [\hat{x}_1^{(\kappa)}, ..., \hat{x}_N^{(\kappa)}]$ to denote the offline sensing outcomes, while we use $\boldsymbol{c}_j^{(\kappa)}$ and $\tau_j^{(\kappa)}$ to denote respectively the fusion coefficients and the detection threshold at node j at (offline) iteration κ. For notational convenience, we do not explicitly show the timing of the samples here. However, it should be noted that by $\boldsymbol{\gamma}_T$ and $\hat{\boldsymbol{x}}^{(\kappa)}$ we are referring, respectively, to T samples of $\boldsymbol{\gamma}(t)$ and $\hat{\boldsymbol{x}}^{(\kappa)}(t)$ corresponding to $t = 1, 2, ..., T$. It is also worth noting the difference between $\hat{\boldsymbol{x}}^{(\kappa)}$ and $\hat{\boldsymbol{x}}$. Throughout the paper, we use $\hat{\boldsymbol{x}}$ to denote the sensing outcomes resulted from the main linear BP. However, $\hat{\boldsymbol{x}}^{(\kappa)}$ is used here to denote the offline sensing outcomes based on which the desired statistics are estimated.

Algorithm I provides a pseudo-code of the proposed adaptation algorithm in which per-node computations are clarified. The adaptation is designed in a way that each node processes samples of γ_i only from its immediate neighbors. Hence, in the proposed distributed optimization no further information is used at each node other than those assumed available in [25].

The loop starts with initializing $\hat{\boldsymbol{x}}^{(0)}$ by the local sensing process on $\boldsymbol{\gamma}_T$. Specifically, for $t = 1, ..., T$, initial offline

Algorithm I: Blind Distributed Optimization of Linear BP by an Offline Adaptive Linear BP

```
Blind Distributed Optimization of Linear Belief Propagation
Input: γ_T, τ^(0), κ_max, η
Output: Near-optimal coefficients for linear BP
1.   Let κ ← 0 and initialize x̂^(0) by comparing γ_T against τ^(0) ;
2.   while κ ≤ κ_max
3.      for node j ∈ {1, 2, ..., N}
4.         Calculate E[γ_i|x̂_j^(κ)] and cov(γ_i, γ_k|x̂_j^(κ)) for all i, k ∈ M_j;
5.         Solve (P4) to find coefficients c_j^(κ) and threshold τ_j^(κ);
6.      end
7.      Run the resulting linear BP on γ_T to find x̂^(κ+1);
8.      κ ← κ + 1;
9.   end
10.  for node j ∈ {1, 2, ..., N}
11.     for k ∈ M_j
12.        if c_j^BP(k)/c_j^(κ_max)(k) > η
13.           c_j(k) ← c_j^BP(k);
14.        else
15.           c_j(k) ← c_j^(κ_max)(k);
16.        end
17.     end
18.  end
19.  Output c_j for j ∈ 1, 2, ...N;
```

sensing outcome $\hat{x}_j^{(0)}(t)$ is created at node j by comparing $\gamma_j(t)$ against the corresponding local threshold $\tau_j^{(0)}$ calculated by (21), which only requires the noise variance. The noise levels can be measured in practice [25]. At iteration κ, all T samples of $\hat{x}^{(\kappa)}$ are used to calculate $E[\gamma_i|\hat{x}_j^{(\kappa)}]$ and $\text{cov}(\gamma_i, \gamma_k|\hat{x}_j^{(\kappa)})$ for all $i, k \in \mathcal{M}_j$, $j = 1, 2, ..., N$. These estimates are then fed into (P4) to find the coefficients $\boldsymbol{c}_j^{(\kappa)}$ and thresholds $\tau_j^{(\kappa)}$. The resulting linear BP is then run over γ_T to find $\hat{x}^{(\kappa+1)}$. The iteration stops after either a predetermined number of cycles or when there is no significant change in the updated parameters and statistics concerned. Lines 1 to 9 in Algorithm I describe this iteration.

Due to the multipath fading or shadowing effects typically associated with the wireless environment, we may experience very low SNR levels at some sensing nodes. In such cases, the signals to be detected may be buried under a heavy noise and the channel estimation process may not be able to reliably capture the correlations between the desired random variables. In order to ensure that the system performance is not degraded in such conditions – to below the performance level of the BP algorithm – we compare the coefficients obtained by the adaptive linear BP against their counterparts in the linearized version of the main BP algorithm. If a coefficient obtained by the offline iteration is too small compared to its counterpart in the main BP, then that coefficient is not used in the adaptation and the one from the main BP is used instead. Lines 10 to 18 in Algorithm I illustrate this process. Specifically, if $c_j^{\text{BP}}(k)/c_j^{(\kappa_{\max})}(k)$ is greater than a predefined value denoted by η, then $c_j^{\text{BP}}(k)$ is used as the final fusion weight. Otherwise, $c_j^{(\kappa_{\max})}(k)$ is used. The final coefficients $\boldsymbol{c}_j$, $j = 1, 2, ...N$, are then used in the main linear BP algorithm for spectrum sensing.

Time is considered slotted in this paper and we refer to the proposed adaptation process as offline to distinguish it from the main linear BP algorithm, which is executed at the beginning of every time slot. Unlike the main linear BP iteration (or the legacy BP), Algorithm I is not run at every time slot. Instead, it is executed with a longer periodicity and in step with the changing characteristics of the radio environment. Since we assume that the statistical behavior of the environment changes slowly, the period over which Algorithm I is repeated is much longer than the time-slot duration. Specifically, if the spectrum sensing is performed at $t = nt_0$, $n = 0, 1, 2, ...$, with t_0 being the slot duration, then Algorithm I is executed at $t = kT_0$ for $k = 1, 2, ...$, while $T_0/t_0 \gg 1$. Moreover, this adaptation is based on a window of data samples generated in the past compared to the data samples used for spectrum sensing. Hence, with respect to the spectrum sensing process, the fusion coefficients are adapted offline.

Remark 6: Since the statistics of the radio environment change slowly compared to the instantaneous status of x_j's, the proposed adaptation is run far less frequently than is the main linear BP algorithm. Consequently, the proposed learning-optimization scheme does not lead to any significant increase in the computational complexity or communication overhead of the existing BP-based distributed detection methods.

It is worth noting that, if the channel statistics are known then there is no need for Algorithm I and the linear BP is run with the messages updated as in (14) where the coefficients are derived by (P4). In case the channel statistics are not known a priori, the spectrum sensing process is still realized by the same linear BP process while c_{jk}'s are updated by Algorithm I. Therefore, the spectrum sensing is realized by the same iterative process and with the same time scale in both cases. In the blind case, however, there is an extra BP iteration withing Algorithm I, which is run far less frequently than the main BP. The outcome of this offline linear BP is not used to realize the sensing process and is only used for channel estimation. Therefore, having a slower BP iteration here does not mean that the spectrum sensing is performed in a different time scale. The offline BP is slower since the channel statistics are constant during many time slots. During this time, $\hat{x}_j^{(\kappa)}$ and stored samples of γ_i, $i \in \mathcal{M}_j$, are iteratively refined and combined together at node j to build good estimates of the required statistics $E[\gamma_i|x_j]$ and $\text{cov}(\gamma_i, \gamma_k|x_j)$, see line 4 in Algorithm I.

We do not use $\boldsymbol{\tau}^{(\kappa_{\max})}$ to set the detection thresholds in the main linear BP algorithm. The reason is that, to guarantee a certain level of the false-alarm or misdetection probability in the system we need a more accurate threshold adaptation mechanism. In the following, we propose a blind threshold adaptation scheme that guarantees the required performance.

B. Detector Calibration

As discussed in Section IV-A, when we rely only on the detector outcome to derive the required conditional probabilities and statistics, we have to deal with inevitable errors. Moreover, by evaluating (22) and (28) we realize that the threshold values aimed at guaranteeing a certain false-alarm or detection probability appear to be too sensitive to these errors.
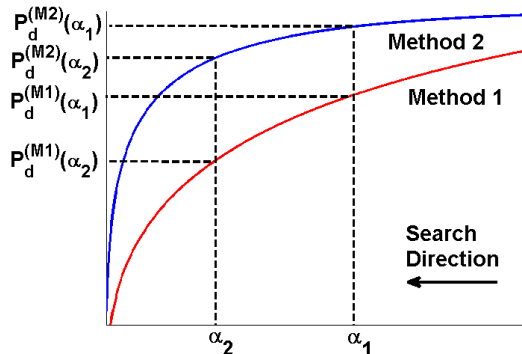
Fig. 2: ROC curves of two hypothetical sensing methods. Method 2 is a better detector but the false-alarm probability of Method 1 can be set arbitrarily.

Specifically, we have found through computer simulations that when $\eta_{j,v}(\boldsymbol{b})$, $\sigma_{j,v}^2$, and $\Pr\{\boldsymbol{x}_j = \boldsymbol{b} | x_j = v\}$ in (22) or $\hat{\eta}_{j,v}(\boldsymbol{b})$, $\hat{\sigma}_{j,v}^2$, and $\Pr\{\boldsymbol{z}_j = \boldsymbol{b} | x_j = v\}$ in (28) are estimated solely based on the detection outcomes, i.e., without prior knowledge about the joint statistical behavior of x_i's, the thresholds obtained by $g_j(\tau_j, 0) \leq \alpha$ or $g_j(\tau_j, 1) \geq \beta$ cannot always guarantee the desired performance levels. However, they work well when the required statistics are available.

Hence, we need a blind threshold adaptation method. We develop such an adaptation here by showing that the desired thresholds can be found through evaluating the outcomes of two sensing schemes that work in the same environment. It is clear that the sensing outcomes are always available and can be measured with no error. In this design, we use the fact that *if the false-alarm (misdetection) probability of a sensing method can be set arbitrarily, that sensing method can be used to find a proper detection threshold for a better sensing scheme whose false-alarm (misdetection) probability needs to be below a certain level.*

To clarify this observation, see Fig. 2 in which the receiver operational characteristic (ROC) curves of two hypothetical sensing methods, Method 1 and Method 2, are depicted. Both methods are used to decide whether a binary-valued variable of interest, denoted here by x, equals 1. We assume that Method 2 exhibits a better performance and that we can set the detection threshold in Method 1 such that its false-alarm probability is fixed at α_1. Let $\tau_{\alpha_1}^{(\text{M1})}$ denote that threshold. When $\tau_{\alpha_1}^{(\text{M1})}$ is adopted for Method 1, the probability of observing 1 as the outcome of Method 1 can be stated as

$$
\begin{aligned}
\Pr\{&\hat{x}_{\text{M1}} = 1\} \\
&= P_0 \Pr\{\hat{x}_{\text{M1}} = 1 | x = 0\} + P_1 \Pr\{\hat{x}_{\text{M1}} = 1 | x = 1\} \\
&= P_0 P_{\text{f}}^{(\text{M1})} + P_1 P_{\text{d}}^{(\text{M1})} = P_0 \alpha_1 + P_1 P_{\text{d}}^{(\text{M1})} (\alpha_1) \quad (29)
\end{aligned}
$$

where $\hat{x}_{\text{M1}}$ denotes an estimation of x given by Method 1 (i.e., the outcome of Method 1), $P_0 \triangleq \Pr\{x = 0\}$ and $P_1 \triangleq \Pr\{x = 1\}$, while $P_{\text{f}}^{(\text{M1})}$ and $P_{\text{d}}^{(\text{M1})}$ denote, respectively, the false-alarm and detection probabilities of Method 1. $P_{\text{d}}^{(\text{M1})} (\alpha_1)$ denotes the detection probability of Method 1 when its false-alarm probability equals α_1. In the following, we use similar notations regarding Method 2.

$\Pr\{\hat{x}_{\text{M1}} = 1\}$ can be measured in a simple fashion. We just need to count the number of times $\hat{x}_{\text{M1}}$ equals 1 over a long enough period of time. Now, let $\tau^{(\text{M2})}$ denote the detection threshold in Method 2. We are looking for $\tau_{\alpha_2}^{(\text{M2})}$ at which $P_{\text{f}}^{(\text{M2})} = \alpha_2 \leq \alpha_1$. We start with $\tau^{(\text{M2})} = 0$ and gradually increase the threshold level. At the starting point, Method 2 gives only 1 as its outcome. The reason is that when $\tau^{(\text{M2})} = 0$, we have $P_{\text{f}}^{(\text{M2})} = P_{\text{d}}^{(\text{M2})} = 1$. Hence, at the beginning $\Pr\{\hat{x}_{\text{M2}} = 1\} = P_0 + P_1 = 1$. It is also clear that, at the beginning we have $\Pr\{\hat{x}_{\text{M2}} = 1\} \geq \Pr\{\hat{x}_{\text{M1}} = 1\}$.

By increasing $\tau^{(\text{M2})}$ both values of $P_{\text{f}}^{(\text{M2})}$ and $P_{\text{d}}^{(\text{M2})}$ decrease reducing the gap between $\Pr\{\hat{x}_{\text{M2}} = 1\}$ and $\Pr\{\hat{x}_{\text{M1}} = 1\}$. We stop the search when the two values coincide, which happens when we have $P_{\text{f}}^{(\text{M2})} = \alpha_2$ and $\alpha_2 \leq \alpha_1$. When the gap is zero we have $P_0 \alpha_2 + P_1 P_{\text{d}}^{(\text{M2})} = P_0 \alpha_1 + P_1 P_{\text{d}}^{(\text{M1})} (\alpha_1)$. This leads to $P_{\text{d}}^{(\text{M2})} = \frac{P_0}{P_1}(\alpha_1 - \alpha_2) + P_{\text{d}}^{(\text{M1})} (\alpha_1)$. Therefore, now we have found a detection threshold that guarantees $P_{\text{f}}^{(\text{M2})} \leq \alpha_1$ while $P_{\text{d}}^{(\text{M2})} \geq P_{\text{d}}^{(\text{M1})} (\alpha_1)$.

The proposed threshold adaptation is based on measuring the difference in the behaviors of two sensing schemes operating in the same environment. Therefore, we refer to this process as the *calibration of Method 2 by using Method 1.*

Remark 7: Assuming no prior knowledge about the environment means that none of the following probabilities are (perfectly) available: $P_0, P_1, P_{\text{d}}^{(\text{M1})}, P_{\text{f}}^{(\text{M2})}$, and $P_{\text{d}}^{(\text{M2})}$. The proposed calibration finds the desired threshold level without the need for any of those parameters.

In our spectrum sensing system, we have two options to use for calibrating the linear BP algorithm. The first option is the local sensing performed at each node. And the second one is the legacy BP-based distributed detection when the required false-alarm probability is small. In both of those detection schemes, we can guarantee a predefined level of false alarm only by knowing the noise level at each node. As mentioned before, the noise levels can be measured in practice. Since a linear BP with near-optimal coefficients exhibits a better performance level than the level shown by either of those methods, either of them can be used to calibrate the linear BP. We use the first option not to be limited to a small value of α.

V. CONVERGENCE CONDITIONS

In order to guarantee the convergence of the linear BP to a fixed point, we need to guarantee the so-called *contracting mapping condition* [2], [32] of the overall transformation obtained. To this end, we collect all the messages $m_{i \to j}^{(l)}$, $(i, j) \in \mathcal{E}$ in the vector $\boldsymbol{m}^{(l)}$. Then, we can express the message-update rule (12) in matrix form as $\boldsymbol{m}^{(l)} = f(\boldsymbol{m}^{(l-1)})$ and the linear approximation as $\mathbf{m}^{(l)} \approx \boldsymbol{T} \boldsymbol{m}^{(l-1)} + \boldsymbol{\xi}$ where $\boldsymbol{T} = [T_{ij}]$ denotes an $|\mathcal{E}| \times |\mathcal{E}|$ matrix that contains c_{ji}'s while $\boldsymbol{\xi}$ denotes the offset in this transformation. Assuming that the ith and jth elements of $\boldsymbol{m}^{(l)}$ are, respectively, $m_{k \to n}^{(l)}$ and $m_{p \to q}^{(l)}$, we have

$$
T_{ij} = \frac{\partial m_{k \to n}^{(l)}}{\partial m_{p \to q}^{(l-1)}} = \begin{cases} c_{nk}, & p \in \mathcal{N}_k^n, q = k \\ 0, & \text{otherwise} \end{cases} \quad (30)
$$

Recall from Section II that we denote a general message-update rule as $\boldsymbol{m}^{(l)} = f^l(\boldsymbol{m}^{(0)})$ where $f : V \to V$. From [2, Theorem 1] we know that if there exist $0 \le K < 1$ such that $\|f(\boldsymbol{x}) - f(\boldsymbol{y})\| \le K\|\boldsymbol{x} - \boldsymbol{y}\|$ for any $\boldsymbol{x}, \boldsymbol{y} \in V$, then the sequence convergence is guaranteed. This is known as the contraction condition. Moreover, for a differentiable mapping f, for any point $\boldsymbol{z}$ on a segment between $\boldsymbol{x}$ and $\boldsymbol{y}$ denoted by $[\boldsymbol{x}, \boldsymbol{y}] \triangleq \{\theta\boldsymbol{x} + (1-\theta\boldsymbol{y}) : \theta \in [0,1]\}$ we have $\|f(\boldsymbol{x}) - f(\boldsymbol{y})\| \le \|\boldsymbol{x}-\boldsymbol{y}\| \cdot \sup_{\boldsymbol{z} \in [\boldsymbol{x}, \boldsymbol{y}]} \|f'(\boldsymbol{z})\|$. Therefore, the following condition guarantees the convergence of a message passing algorithm defined by f

$$\sup_{\boldsymbol{m} \in V} \|f'(\boldsymbol{m})\| < 1 \qquad (31)$$

For simplicity, we work with the ℓ_∞ matrix norm $\|\cdot\|_\infty$. By using the the proposed linear approximation in (14), we have $f(\boldsymbol{m}) = \boldsymbol{T}\boldsymbol{m} + \boldsymbol{\xi}$ which leads to $\|f'(\boldsymbol{m})\| = \|\boldsymbol{T}\|$. Consequently, the contraction condition holds if $\max_i \sum_j |T_{ij}| < 1$ which, according to (30), is equivalent to $\max_{k,n} \sum_p \sum_q \left|\frac{\partial m_{k \to n}}{\partial m_{p \to q}}\right| < 1$ which can be restated as $\max_{k,n} \sum_{p \in \mathcal{N}_k^n} |c_{k,n}| < 1$ which is equivalent to $\max_{k,n}(|\mathcal{N}_k| - 1)|c_{k,n}| < 1$ where $|\mathcal{N}_k|$ indicates the number of neighbors of node k. Therefore, by imposing the following constraint on the proposed optimizations we guarantee the convergence of the proposed linear message-passing algorithm

$$|c_{jk}| < \frac{1}{\max_n |\mathcal{N}_n| - 1}, \forall(j,k) \in \mathcal{E} \qquad (32)$$

This is a convex constraint and does not add any significant challenge to the optimization problems at hand.

It is worth noting that, (32) is the last piece we need to develop an optimization framework for the BP-based distributed inference systems, which takes the advantages of both the BP and linear fusion. In particular, the BP algorithm is an effective approach to take into account the correlations between a group of spatially-distributed nodes while linear fusion is known to enable building low-complexity near-optimal decision variables based on the data exchanged between such nodes. By combining these capabilities, we develop a message-passing algorithm that takes into account the impact of pairwise correlations in accordance with the MRF model and is easy to optimize while its convergence is guaranteed.

VI. IMPACT OF CYCLES

It is known that the presence of cycles in the graph can lead to overconfidence in beliefs, which may degrade the system performance [25]. The impact of cycles can be mitigated by using the so-called uniformly tree-reweighted-BP (UTRW-BP) algorithm [3], which is a special case of the so-called tree-reweighted BP [33]. This method reduces the amount of information exchanged between the nodes to maintain an acceptable performance level when the factor graph contains cycles. In this method, the message from node j to node k is given by

$$\mu_{k \to j}^{(l)}(x_j) \propto \sum_{x_k} \left\{ \phi_k(x_k)[\psi_{kj}(x_k, x_j)]^{\frac{1}{\rho}} \left[\mu_{j \to k}^{(l-1)}(x_k)\right]^{\rho-1} \right.$$
$$\left. \prod_{n \in \mathcal{N}_k^j} \left[\mu_{n \to k}^{(l-1)}(x_k)\right]^\rho \right\} \qquad (33)$$

while the beliefs are built as

$$b_j^{(l)}(x_j) \propto \phi_j(x_j) \prod_{k \in \mathcal{N}_j} \left[\mu_{k \to j}^{(l)}(x_j)\right]^\rho \qquad (34)$$

where ρ is referred to as the *edge appearance probability* (EAP). In this method, we have $0 < \rho \le 1$ and the BP algorithm is a special case of the UTRW-BP with $\rho = 1$. Using smaller values for the EAP reduces the impact of cycles at the expense of cooperation between the sensing nodes. The performance of the UTRW-BP can be optimized over the EAP. It is shown in [3] that the optimal EAP is given by

$$\rho = \min\{1, \frac{1}{2n_D}\} \qquad (35)$$

where n_D is the average degree of the graph, i.e., the average number of connections of the nodes in the network. We know that, n_D can be computed in a decentralized architecture by an average consensus algorithm.

Following the same approach as in Section II, we see that the decision variable build by UTRW-BP at node j approaches approximately to

$$\lambda_j \approx \gamma_j + \rho \sum_{k \in \mathcal{N}_j} c_{jk}\gamma_k + \rho^2 \sum_{k \in \mathcal{N}_j} \sum_{n \in \mathcal{N}_k^j} c_{jk}c_{kn}\gamma_n$$
$$+ \rho^3 \sum_{k \in \mathcal{N}_j} \sum_{n \in \mathcal{N}_k^j} \sum_{m \in \mathcal{N}_n^k} c_{jk}c_{kn}c_{nm}\gamma_m + ... \qquad (36)$$

where $c_{jk} = \frac{(e^{2J_{kj}/\rho} - 1)}{(1 + e^{J_{kj}/\rho})^2}$.

From (36) we obtain some interesting observations. In particular, we observe a stronger decaying effect by distance from node j than we had before. As we saw in (15), since $|c_{jk}| < 1$, the impact of other nodes on λ_j decreases by the distance of those nodes from node j. Now, this decaying is stronger and in favor of the one-hop neighbors even more than before. Specifically, if γ_i arrives at node j through a path of length L, then it is multiplied by ρ^L in the data-fusion process. Consequently, in the presence of loops, i.e., when there is more than one path from node i to node j, the impact of γ_i on λ_j received through the shortest path is amplified compared to the ones received through the other paths. In other words, in the UTRW-BP the decision variable at a node is built mainly by the data received through the tree sub-graph(s) connecting that node to all the other nodes via the shortest paths. Hence, ρ reduces the overconfidence in nodes by alleviating the impact of cycles.

We can accommodate this decaying effect in the proposed linear BP scheme. Note that, for a fixed false-alarm probability $P_f^{(j)}$ the detection probability $P_d^{(j)}$ is positively homogeneous of degree zero with respect to $\boldsymbol{c}_j$. To show this we set the detection threshold in (28) as

$$\tau_j = \mathsf{q}_\alpha \tilde{\sigma}_{j,0}(\boldsymbol{b}) + \tilde{\eta}_{j,0}(\boldsymbol{b}) \qquad (37)$$
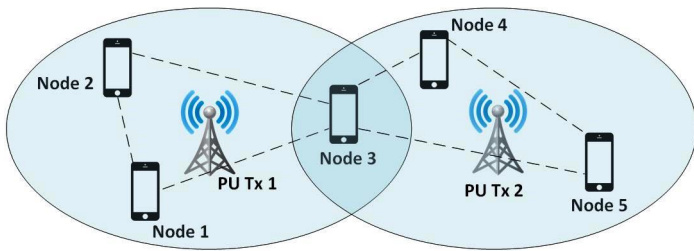
Fig. 3: The network configuration considered in the simulations. The dashed lines depict the links between the sensing nodes through which the distributed detection is conducted.

where $\mathfrak{q}_\alpha \triangleq Q^{-1}(\alpha)$. Consequently, $P_f^{(j)} \approx \alpha$ and

$$P_d^{(j)} \approx \sum_{\boldsymbol{b} \in \{0,1\}^{|\mathcal{N}_j|}} Q\left(\frac{\mathfrak{q}_\alpha \tilde{\sigma}_{j,0}(\boldsymbol{b}) - \Delta\tilde{\eta}_j(\boldsymbol{b})}{\tilde{\sigma}_{j,1}(\boldsymbol{b})}\right) p_{\tilde{\boldsymbol{x}}_{(j)}|x_j}(\boldsymbol{b}|1) \tag{38}$$

where $\Delta\tilde{\eta}_j \triangleq \tilde{\eta}_{j,1}(\boldsymbol{b}) - \tilde{\eta}_{j,0}(\boldsymbol{b})$.

Now we see that by using $\rho\boldsymbol{c}_j$ instead of $\boldsymbol{c}_j$ as the fusion vector at node j, the detection probability $P_d^{(j)}$ remains unchanged. The reason is that, the variables $\tilde{\sigma}_{j,0}, \tilde{\sigma}_{j,1}$ and $\Delta\tilde{\eta}_j$ are all multiplied by ρ, thus, canceling the effect of this scaling on $P_d^{(j)}$. Since the detection probability is a monotonically increasing function of the false-alarm probability, the global optimum in (P4) occurs when $P_f^{(j)} = \alpha$. Therefore, given $\boldsymbol{c}_j$ as a solution for (P4) its scaled version $\rho\boldsymbol{c}_j$ is also a global optimum. Hence, by using ρc_{jk} in (14) we obtain a similar decaying effect to the one observed in UTRW-BP without altering the proposed optimization framework. Note also that, using the deflection coefficient as the objective function in the proposed distributed optimization does not affect this argument since the deflection coefficient is a homogeneous function of $\boldsymbol{c}_j$ as well.

As the final point, note that when the number of iterations is small there is no need to be concerned of the impact of cycles [25].

VII. NUMERICAL RESULTS

Fig. 3 depicts the network configuration used in our simulations. We consider two PU transmitters and five sensing nodes as SUs. Nodes 1 and 2 are located within the range of PU transmitter 1, nodes 4 and 5 are located within the range of PU transmitter 2, while node 3 is located within the range of the both PU transmitters. These nodes exchange messages with each other over a set of wireless links, depicted as the dashed lines in the figure, to conduct a distributed detection scheme through which the transmission opportunities are discovered within the spectrum bands allocated to the PUs.

Energy detection is used as the local detection method at each node with $K = 100$ samples to estimate the received signal energy. The SNR levels of the signals received at nodes 1 and 5 are -5 dB, at nodes 2 and 4 are -8 dB and at node 3 are -10 and -10 dB. We realize a spatially-correlated occupancy pattern by making the primary transmitters exhibit correlated random on and off periods. This is an extension of the simulation scenario in [25] where one of the PU
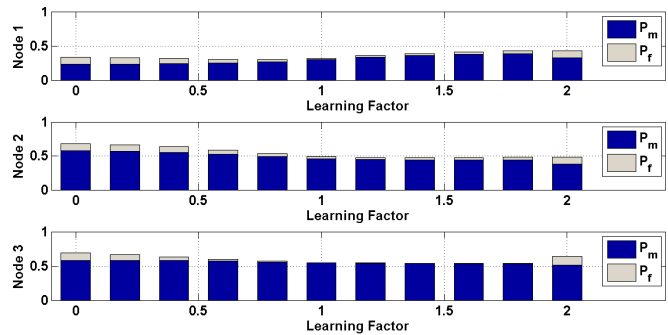

Fig. 4: Different levels of system performance, in terms of the false-alarm and misdetection probabilities, denoted P_f and P_m respectively, are observed for different values of the learning factor ζ.

transmitters is constantly on while the other one is off all the time. The channel coefficients are constant during the sensing period as in [25]. The number of iterations is 3 for all the message-passing algorithms considered in this section. In addition, the offline BP algorithm is repeated only 4 times, i.e., $\kappa_{max} = 4$. Each data point in the simulation results is obtained by averaging the detection outcomes over 20,000 time slots. A window of $T = 2000$ time slots is used to train all the message-passing algorithms.

Fig. 4 depicts the system performance in terms of the false-alarm and misdetection probabilities vs. the learning factor in (16). We observe different performance levels for different values of the learning factor, which indicates the need for optimization. To cover a wide range of possible performance levels of the BP algorithm, we adopt three values as $\zeta = 0.2, 0.4, 1$. As reported in [25], the BP algorithm works better when the learning factor is small. However, note that how to determine the best value for ζ is unknown. For the sake of brevity, we only focus on nodes 1, 2, and 3. Nodes 4 and 5 experience similar conditions as those in which nodes 1 and 2 operate. The optimal linear fusion scheme is realized in this section by solving (P2) via the sequential optimization method described in [8].

We use ROC curves to demonstrate and compare the detection performance achieved by the different methods. In Fig. 5a we see the performance of node 1 which operates under a relatively better SNR regime than those experienced by nodes 2 and 3. The performance of the proposed linear BP along with those of the optimal linear fusion and the local detection are shown for comparison. We first observe that the performance of the BP algorithm improves by increasing the value of ζ from 0.2 to 0.6 and then degrades by a further increase in ζ. Moreover, we observe that both the BP with $\zeta = 0.6$ and the proposed linear BP achieve the performance of the optimal linear fusion scheme without any prior knowledge available concerning the radio environment. In comparing these performance results it should be noted that, firstly, $\zeta = 0.6$ is found by a numerical search and secondly, as it is shown later in this section, it fails to always guarantee the performance constraint on the system false-alarm probability. The proposed method achieves the optimal performance blindly while guaranteeing
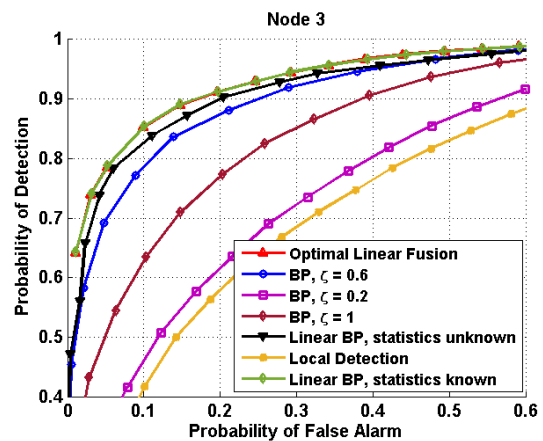
(a) Detection performance of node 1.
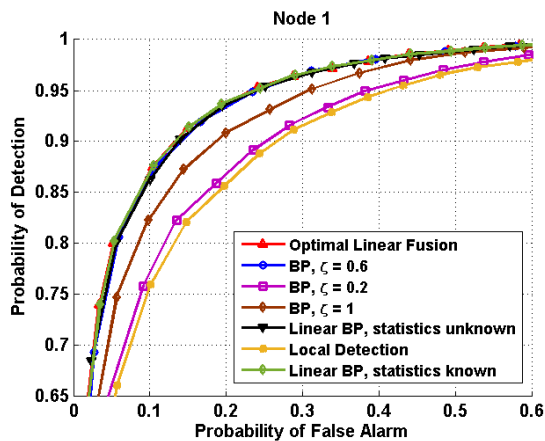
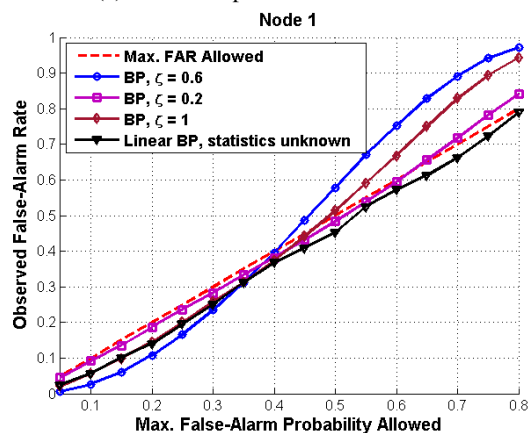

(a) Detection performance of node 3.
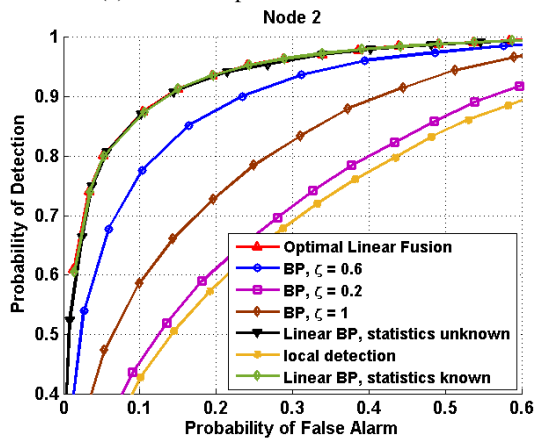
(b) Detection performance of node 2.

(b) FAR observed at node 1.

Fig. 5: Comparison of the different distributed detection methods discussed in this paper.

Fig. 6: Detection performance at node 3 and the result of different threshold adaptation schemes at node 1.

that the required constraint on the false-alarm probability is always met.

Fig. 5b depicts the detection performance obtained at node 2. Here we have a clear performance gap which illustrates that the proposed method outperforms the conventional BP algorithm. Therefore, we can see that when operating under a lower SNR level (than the one observed in node 1), the proposed linear BP achieves the detection rate of the optimal linear fusion while the BP algorithm falls behind. This can be attributed to the better learning capability and also to the better combination of the local test summaries in the proposed linear BP algorithm. Specifically, we know that a properly designed linear fusion scheme provides a constructive discrimination [34] on the local LLRs when they are combined together to build a decision variable. This discrimination alleviates the degrading effect of the local sensing results obtained in low SNR regimes by emphasizing the effect of the ones created by the more reliable nodes. In Fig. 5b, it appears that such discrimination is better realized in the proposed linear BP algorithm compared to the conventional BP. Note that, both methods use the same data which is obtained through the exchange of information between the one-hop neighbors in the network.

Fig. 6a depicts the detection performance of node 3. Again, we see that the optimal linear BP outperforms the conventional BP algorithm. Moreover, the linear BP nearly attains the detection rate of the optimal linear fusion scheme without having any prior knowledge about the radio environment.

By comparing the performance gap between the local detection scheme and all other distributed detection methods considered in this section we see that the impact of a distributed detection scheme is more significant when the nodes operate in low SNR regimes and that is the situation in which the proposed method offers a higher performance gain than what is achieved by the conventional BP algorithm. In other words, in a network of distributed sensors, when the SNR levels are low we have a stronger need to establish some sort of cooperation between the sensing nodes to achieve more-reliable sensing results and the linear BP algorithm offers a more significant gain, over the other methods, in such a condition.

The performance of the proposed detector calibration method is depicted in Fig. 6b where the false-alarm rate (FAR) observed is depicted vs. the constraint imposed on the system false-alarm probability. We see that when the detection threshold is determined by the proposed calibration, the FAR is always guaranteed to fall below the predefined

upper bound. However, when the conventional BP is used with the detection threshold calculated by (21), it fails to achieve the required performance guarantees when the constraint is not small enough. Equivalently, Fig. 6b shows that, when (21) is used, to achieve the required performance guarantee for an arbitrary α, we have to keep the learning factor ζ relatively small. However, as we saw previously by inspecting the ROC curves, when ζ is small the detection performance is far from optimal. Hence, the proposed linear BP algorithm outperforms the existing BP algorithm when the FAR of the system is required to be controlled.

It is worth noting that, the use of (20) is inspired by (19) which is derived assuming that if $x_j = 1$, then $\boldsymbol{y}_j$ is a vector of Gaussian random variables. However, this is not always the case. From (18) we see that, when there exist more than one PU transmitter, as in our network configuration, $\boldsymbol{y}_j$ cannot be assumed necessarily to follow a Gaussian distribution. Consequently, the violation of the FAR constraint when τ_0 is used appears to be due to the error in modeling the statistical behavior of γ_j. The proposed calibration method works based on modeling the behavior of γ_j, given $x_j = 0$, by a Gaussian distribution which is, in fact, the case. Therefore, the proposed system is capable of meeting the required FAR constraint.

As the final point, the better performance of the proposed threshold adaptation scheme does not stem from the linearity of the proposed message-passing algorithm since the calibration method proposed does not depend on the behavior of the transfer functions affecting the messages. For the sake of brevity, Fig. 6b depicts only the FAR observed at node 1. All other nodes show a similar performance.

VIII. CONCLUSION

We demonstrated that a BP algorithm in a network of distributed agents can be viewed or approximated as a distributed linear data-fusion scheme. Accordingly, we addressed two particular issues in using the BP algorithm for distributed detection: First, the lack of knowledge about the statistical behavior of the BP messages exchanged between the nodes and second, the lack of a proper threshold adaptation mechanism in a BP-based detection scheme.

Regarding the first issue, our analysis and simulation results demonstrate that, when the local sensing results follow the Gaussian distribution,

- In a centralized configuration and when the first- and second-order statistics of the local detection results of all nodes are available at the FC, a linear fusion of all the local detection results, realized by (P1), can obtain the desired performance and there is no need for an iterative message-passing algorithm.
- In a decentralized configuration and when at each node the first- and second-order statistics of the local detection results of its one-hop neighbors are available, the desired performance can nearly be obtained via a linear message-passing algorithm realized by (P3).
- In a decentralized configuration and when there is no statistics available a priori, the desired performance can be obtained via a linear message-passing algorithm in

which the weighting coefficients are obtained blindly via the offline adaptive linear BP algorithm described in Section IV-A.

And, regarding the second issue, we have shown that in the proposed linear BP algorithm,

- The detection threshold to guarantee a certain false-alarm (or misdetection) probability can be found analytically at each node when the first- and second-order statistics of the one-hop neighbors of that node are available.
- The system false-alarm (or misdetection) probability can be guaranteed to fall below a certain level by the calibration method proposed in Section IV-B without the need for any statistics describing the behavior of the local test summaries or the wireless environment.

REFERENCES

[1] H. . Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 28–41, Jan 2004.

[2] J. M. Mooij and H. J. Kappen, "Sufficient conditions for convergence of the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4422–4437, Dec 2007.

[3] H. Wymeersch, F. Penna, and V. Savic, "Uniformly reweighted belief propagation for estimation and detection in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 4, pp. 1587–1595, 2012.

[4] F. R. Kschischang, B. J. Frey, and H. . Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001.

[5] N. Noorshams and M. J. Wainwright, "Stochastic belief propagation: A low-complexity alternative to the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 59, no. 4, pp. 1981–2000, April 2013.

[6] A. Y. Lokhov, M. Vuffray, S. Misra, and M. Chertkov, "Optimal structure and parameter learning of Ising models," *Science advances*, vol. 4, no. 3, p. e1700791, 2018.

[7] Z. Quan, S. Cui, and A. H. Sayed, "Optimal linear cooperation for spectrum sensing in cognitive radio networks," *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 1, pp. 28–40, Feb. 2008.

[8] Z. Quan, S. Cui, A. H. Sayed, and H. V. Poor, "Optimal multiband joint detection for spectrum sensing in cognitive radio networks," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1128–1140, March 2009.

[9] Z. Quan, W. Ma, S. Cui, and A. H. Sayed, "Optimal linear fusion for distributed detection via semidefinite programming," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2431–2436, 2010.

[10] G. Taricco, "Optimization of linear cooperative spectrum sensing for cognitive radio networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 77–86, Feb. 2011.

[11] M. Welling and Y. W. Teh, "Linear response algorithms for approximate inference in graphical models," *Neural computation*, vol. 16, no. 1, pp. 197–221, 2004.

[12] N. Wiberg, *Codes and decoding on general graphs*, 1996.

[13] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 657–670, 2001.

[14] S. Ten Brink, "Iterative decoding trajectories of parallel concatenated codes," in *3rd IEEE ITG Conf. Source and Channel Coding*, 2000, pp. 75–80.

[15] ——, "Rate one-half code for approaching the Shannon limit by 0.1 dB," *Electronics letters*, vol. 36, no. 15, pp. 1293–1294, 2000.

[16] H. El Gamal and A. R. Hammons Jr., "Analyzing the turbo decoder using the Gaussian approximation," in *Proc. Int. Symp. Information Theory*. IEEE, 2000, p. 319.

[17] ——, "Analyzing the turbo decoder using the Gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, 2001.

[18] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 764–785, 2011.

[19] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

[20] ——, "Message passing algorithms for compressed sensing: I. motivation and construction," in *IEEE Information Theory Workshop (ITW 2010)*. IEEE, 2010, pp. 1–5.

[21] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," *arXiv preprint arXiv:1010.5141*, 2010.

[22] ——, "Generalized approximate message passing for estimation with random linear mixing," in *Int. Symp. Inf. Theory Proceedings*. IEEE, 2011, pp. 2168–2172.

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

[24] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical Commun.*, vol. 4, no. 1, pp. 40–62, March 2011.

[25] F. Penna and R. Garello, "Decentralized Neyman-Pearson test with belief propagation for peer-to-peer collaborative spectrum sensing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 5, pp. 1881–1891, May 2012.

[26] F. Penna, R. Garello, and M. A. Spirito, "Likelihood-ratio propagation and consensus in wireless networks with markov random field models," in *2010 IEEE Globecom Workshops*. IEEE, 2010, pp. 1259–1263.

[27] ——, "Distributed inference of channel occupation probabilities in cognitive networks via message passing," in *2010 IEEE Symposium on New Frontiers in Dynamic Spectrum (DySPAN)*. IEEE, 2010, pp. 1–11.

[28] K. Gurney, *An introduction to neural networks*. CRC press, 1997.

[29] S. Chaudhari, J. Lunden, V. Koivunen, and H. V. Poor, "Cooperative sensing with imperfect reporting channels: Hard decisions or soft decisions?" *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 18–28, Jan. 2012.

[30] S. M. Kay, *Fundamentals of statistical signal processing*. Prentice Hall PTR, 1993.

[31] P. Paysarvi-Hoseini and N. Beaulieu, "Optimal wideband spectrum sensing framework for cognitive radio systems," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 1170–1182, March 2011.

[32] C. Huang, Y. Li, and L. Dolecek, "Belief propagation algorithms on noisy hardware," *IEEE Transactions on Communications*, vol. 63, no. 1, pp. 11–24, Jan 2015.

[33] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "A new class of upper bounds on the log partition function," *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2313–2335, July 2005.

[34] Y. Abdi and T. Ristaniemi, "Random interruptions in cooperation for spectrum sensing in cognitive radio networks," *IEEE Transactions on Communications*, vol. 65, no. 1, pp. 49–65, 2017.

Tapani Ristaniemi (SM'11) received the M.Sc. degree in mathematics, the Ph. Lic. degree in applied mathematics, and the Ph.D. degree in wireless communications from the University of Jyväskylä, Jyväskylä, Finland, in 1995, 1997, and 2000, respectively. In 2001, he was a Professor with the Department of Mathematical Information Technology, University of Jyväskylä. In 2004, he was with the Department of Communications Engineering, Tampere University of Technology, Tampere, Finland, where he was appointed as a Professor of Wireless Communications. In 2006, he moved back to the University of Jyväskylä to take up his appointment as a Professor of Computer Science. In 2013, he was a Visiting Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He is currently an Adjunct Professor with the Tampere University of Technology. He has authored or co-authored over 150 publications in journals, conference proceedings, and invited sessions. He served as a Guest Editor of the IEEE Wireless Communications in 2011. He is an Editorial Board Member of Wireless Networks and the International Journal of Communication Systems. His research interests include brain and communication signal processing and wireless communication systems.

Besides academic activities, Prof. Ristaniemi is also active in the industry. In 2005, he co-founded a start-up, Magister Solutions, Ltd., Finland, specializing in wireless systems (Research and Development) for telecom and space industries in Europe. He serves as a consultant and a member of the Board of Directors.

Younes Abdi (S'10, M'16) received the B.Sc. degree in electrical engineering from the University of Tabriz, Tabriz, Iran, in 2008, the M.Sc. degree in electrical engineering from Tarbiat Modares University, Tehran, Iran, in 2011, and the Ph.D. degree in information technology from the University of Jyväskylä, Jyväskylä, Finland, in 2016.

From 2010 to 2011, he was with the Radio Communications Group, Iran Telecommunications Research Center, Tehran, where he was involved in the standardization and regulatory issues of cognitive radio networks. Since 2012, he has been with the Faculty of Information Technology, University of Jyväskylä, where he is currently a Post-Doctoral Researcher. He is also a member of Working Group 1900.1 in the IEEE Dynamic Spectrum Access Networks Standards Committee. During spring and summer 2018, he served as a visiting researcher at Nokia Bell Labs, Espoo, Finland, where he worked, as a member of the Radio Working Group in the MulteFire Alliance, to promote MulteFire – an LTE-based technology for small cells operating solely in unlicensed spectrum. He is a recipient of research grants from Finnish National Graduate School in Electronics, Telecommunications, and Automation and Jyväskylä Doctoral Program in Computing and Mathematical Sciences. His current research interests include advanced signal processing and wireless communications.