This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2020.3012910, IEEE Transactions on Wireless Communications

The Max-Product Algorithm Viewed as Linear Data-Fusion: A Distributed Detection Scenario

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

***Abstract*—In this paper, we disclose the statistical behavior of the max-product algorithm configured to solve a *maximum a posteriori* (MAP) estimation problem in a network of distributed agents. Specifically, we first build a distributed hypothesis test conducted by a max-product iteration over a binary-valued pairwise Markov random field and show that the decision variables obtained are linear combinations of the local log-likelihood ratios observed in the network. Then, we use these linear combinations to formulate the system performance in terms of the false-alarm and detection probabilities. Our findings indicate that, in the hypothesis test concerned, the optimal performance of the max-product algorithm is obtained by an optimal linear data-fusion scheme and the behavior of the max-product algorithm is very similar to the behavior of the sum-product algorithm. Consequently, we demonstrate that the optimal performance of the max-product iteration is closely achieved via a linear version of the sum-product algorithm, which is optimized based on statistics received at each node from its one-hop neighbors. Finally, we verify our observations via computer simulations.**

***Index Terms*—Statistical inference, distributed systems, max-product algorithm, sum-product algorithm, linear data-fusion, Markov random fields, factor graphs, spectrum sensing.**

I. Introduction

STANDARD optimization methods are computationally demanding when dealing with a large collection of correlated random variables. This is a well-known challenge in designing statistical inference techniques used in a wide range of signal-processing applications such as channel decoding, image processing, spread-spectrum communications, distributed detection, etc. Alternatively, message-passing algorithms over factor graphs provide a powerful low-complexity approach to characterizing and optimizing the collective impact of those variables on the desired system performance, see e.g., [1]–[3]. Consequently, a better understating of the statistical behavior of the message-passing algorithms leads to statistical inference systems with better performance. Two widely-used message-passing algorithms are the so-called sum-product and max-product algorithms. We have analyzed the behavior of the sum-product algorithm, a.k.a., the belief propagation algorithm, in [4].

Our main focus in this paper is on the max-product algorithm, which is an iterative method for approximately solving the problem of *maximum a posteriori probability* (MAP) estimation [5]. We analyze the behavior of this algorithm in a distributed detection scenario where every network node estimates a binary-valued random variable based on noisy observations collected throughout the entire network. The correlations between the random variables are modeled by a pairwise Markov random field (MRF) [1] whose structure fits well into pairwise interactions between the nodes in an ad-hoc network configuration. An MRF is an undirected graph where vertices correspond to the random variables of interest and edges represent the correlations between them. By using the max-product algorithm, the estimation problem concerned is decomposed into a number of small optimizations performed locally at each node based on information provided by other nodes in the network via one-hop communications per iteration.

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).

A. Max-Product v.s. Sum-Product

Let $\boldsymbol{x} = [x_1, ..., x_N]^T$ denote a vector of N discrete-valued random variables to be estimated given the observations $\boldsymbol{Y} \triangleq [\boldsymbol{y}_1, ..., \boldsymbol{y}_N]$ where $\boldsymbol{y}_i \triangleq [y_i(1), ..., y_i(K)]^T$ denotes K samples collected at node i and $i = 1, ..., N$. The MAP estimation of $\boldsymbol{x}$ is formally stated as

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{Y}). \tag{1}$$

This is an integer program, which is NP-hard. When $p(\boldsymbol{x}|\boldsymbol{Y})$ is stated in the form of an MRF, (1) can be solved with low complexity by using two commonly-used message-passing algorithms, i.e., the sum-product and max-product algorithms. We provide a brief overview of the origins and differences of the two algorithms here. An interested reader may refer to [1]–[3], [5] for further details and more comprehensive discussions.

The optimization in (1) can be approximated by the so-called *Bethe variational problem* [1, Sec. 4] and also by a *linear program* [1, Sec. 8]. For a tree-structured MRF, both of these approximations turn out to be convex, can be solved by the Lagrangian dual method, and provide exact solutions for (1). The sum-product iteration solves the dual of the Bethe problem while the max-product algorithm solves the dual of the linear program. When the MRF contains cycles, the fixed points of both message-passing algorithms provide approximate solutions for the MAP estimation problem concerned.

The sum-product algorithm solves (1) by finding the marginal distributions associated with $p(\boldsymbol{x}|\boldsymbol{Y})$ whereas the max-product is an attempt to find the so-called max-marginals. More specifically, the sum-product algorithm gives the marginal distribution, at node i, defined as

$$p(x_i|\boldsymbol{Y}) \triangleq \sum_{\{\boldsymbol{x}'|x'_i=x_i\}} p(\boldsymbol{x}'|\boldsymbol{Y}) \tag{2}$$

that is used to solve (1) by

$$\hat{x}_i = \arg\max_{x_i} p(x_j|\boldsymbol{Y}). \tag{3}$$

The outcome of the max-product algorithm at node i is the max-marginal distribution of x_i defined as

$$q(x_i|\boldsymbol{Y}) \triangleq \kappa \max_{\{\boldsymbol{x}'|x'_i=x_i\}} p(\boldsymbol{x}'|\boldsymbol{Y}), \tag{4}$$

where κ denotes a positive arbitrary normalization constant. If for each node, the maximum of $q(x_i|\boldsymbol{Y})$ is attained at a unique value, then the MAP configuration is unique and can be obtained by maximizing the corresponding max-marginal at each node [5], i.e.,

$$\hat{x}_i = \arg\max_{x_i} q(x_i|\boldsymbol{Y}). \tag{5}$$

In case there is a node at which the maximum of $q(x_i|\boldsymbol{Y})$ is not attained at a unique value, Eq. (5) provides a sub-optimal solution. We discuss this case in Section IV-C. The analysis proposed in this paper along with the work in [4] shows that, in a distributed detection scenario, both methods are equivalent to a linear data-fusion.

B. Related Work and Motivation

Many existing works on the max-product algorithm attempt to pave the way towards theoretical guarantees on the convergence of the algorithm and on the quality of the resulting fixed points on graphs with arbitrary topology and with arbitrary probability distributions, see e.g., [5]–[8]. This is still an open and growing research field. There also exist numerous works that tailor the max-product iteration into a particular statistical inference scenario taking into account the graph structure and available resources in that particular setting. Examples of such works can be found in LDPC decoding [9], multi-sensor target tracking [10], [11], clock synchronization in wireless sensor networks (WSN) [12], [13], sparse code multiple access [14], etc. Moreover, several works in the literature use some sort of approximation in modeling various message-passing structures to offer a deeper insight into the behavior of message-passing algorithms or to propose better distributed inference methods, see e.g., [15]–[19].

To the best of our knowledge, the existing works do not offer a comprehensive analysis and optimization framework for the max-product algorithm in the context of distributed detection. Formulating the performance of a WSN that employs distributed detection requires understanding the statistical behavior of the underlying data-exchange process between the sensing nodes. When the max-product algorithm is used, this data-exchange process is built based on the structure of the factor graph that models the network behavior. Therefore, an optimal system design calls for finding the relation between the parameters of the factor graph and the network performance metrics. More specifically, an optimal design requires answering the following questions:

- How to best represent the network behavior by a pairwise MRF and how to impose certain constraints on the system performance, in terms of the desired false-alarm or detection probabilities, when the data-exchange process between the nodes is realized by the max-product algorithm over that MRF?

We answer these questions in this paper. The importance of the research gap discussed here is highlighted by noting that distributed detection is a major functionality in many advanced communication scenarios such as industrial internet of things, internet of vehicles, mobile crowdsensing, and cognitive radio (CR) networks, see e.g., [20]–[24].

C. Contribution

We show that the max-product algorithm works as a linear data-fusion process. Linear fusion schemes are commonly used in distributed detection systems to achieve near-optimal performance with low implementation complexity, see e.g., [25]–[28]. Therefore, we indicate that the knowledge already developed in linear distributed detection methods can be used to better understand the behavior of the max-product algorithm. The proposed analysis is supported by a strong connection between the sum-product and max-product operations. In particular, we show that, in the distributed detection scenario concerned, the behavior of the max-product algorithm is very similar to the behavior of the sum-product algorithm and that the decision variables built by the max-product operation are linear combinations of the local likelihoods in the network—a behavior we have already observed in the sum-product algorithm [4]. By using this linearity, we make the following contributions:

- We show that the message-update rule in the max-product algorithm is almost the same as its counterpart in the sum-product algorithm.
- We show that when performing a distributed MAP estimation via the max-product algorithm over a network modeled by a pairwise MRF, under certain practical conditions, the decision variables obtained are linear combinations of the local log-likelihood ratios (LLR) in the network.
- We find the probability distribution function of the decision variables in a practical detection scenario and formulate the detection performance in closed form.
- We show how to set the detection threshold to achieve a predefined detection performance.
- We show that the optimal linear message-passing algorithm in [4] attains the optimal detection performance of the max-product algorithm in the distributed detection scenario concerned.

As in [4], [24], and [29], we clarify our findings by considering a spectrum sensing scheme in a CR network. In these networks, the wireless nodes perform spectrum sensing in bands allocated to the so-called primary users (PU) to discover vacant parts of the radio spectrum and to establish communication on those temporarily- or spatially-available spectral opportunities [30]. 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.

The rest of the paper is organized as follows. In Section II, we discuss how to solve the MAP estimation problem in a network of distributed agents via the sum-product and max-product algorithms. In addition, we illustrate in Section

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2020.3012910, IEEE Transactions on Wireless Communications

II the connection between the sum-product and max-product operations. Then, we analyze the behavior of the max-product algorithm in Section III to show that it works as a linear fusion scheme. In Section IV, we briefly discuss the use of linear data-fusion in distributed detection along with the proposed optimization framework. We then verify our analysis by computer simulations in Section V and present our concluding remarks in Section VI.

II. Distributed Detection via Message-Passing

We consider a pairwise MRF defined on an undirected graph $G = (\mathcal{V}, \mathcal{E})$ composed of a set of N vertices or nodes $\mathcal{V} \triangleq \{1, ..., N\}$ and a set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. Each node $i \in \mathcal{V}$ corresponds to a random variable x_i and each edge $(i, j) \in \mathcal{E}$, which connects nodes i and j, represents a possible correlation between random variables x_i and x_j. The MRF is used to factorize the a posteriori distribution function $p(\boldsymbol{x}|\boldsymbol{Y})$ into single-variable and pairwise terms, i.e.,

$$p(\boldsymbol{x}|\boldsymbol{Y}) \propto \prod_{n\in\mathcal{V}} \phi_n(x_n) \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j), \tag{6}$$

where $\propto$ denotes proportionality up to a multiplicative constant. In our detection scenario, the main goal of each node, say node i, is to find its max-marginal a posteriori distribution $q(x_i|\mathbf{y})$. This goal is achieved by the max-product algorithm where the messages sent from node k to node j in the network are built as

$$\mu^{(l)}_{k\to j}(x_j) \propto \max_{x_k} \left[\phi_k(x_k)\psi_{kj}(x_k, x_j) \prod_{n\in\mathcal{N}^j_k} \mu^{(l-1)}_{n\to k}(x_k) \right], \tag{7}$$

where $\mathcal{N}^j_k$ denotes the set of neighbors of node k except for node j. Fig. 1 illustrates this process. The *belief* of node j at iteration l, denoted $b^{(l)}_j(x_j)$, is formed by multiplying the local inference result $\phi_j(x_j)$ by all the messages received from its neighbors, i.e., $b^{(l)}_j(x_j) \propto \phi_j(x_j) \prod_{k\in\mathcal{N}_j} \mu^{(l)}_{k\to j}(x_j)$. The resulting belief is then used to estimate the desired max-marginal distribution, i.e., $q(x_j|\boldsymbol{Y}) \approx b^{(l)}_j(x_j)$. We can express $b^{(l)}_j(x_j)$ in the logarithm form as

$$\ln b^{(l)}_j(x_j) = \ln \phi_j(x_j) + \sum_{k\in\mathcal{N}_j} m^{(l)}_{k\to j}(x_j), \tag{8}$$

where

$$\begin{aligned} m^{(l)}_{k\to j}(x_j) &\triangleq \ln \mu^{(l)}_{k\to j}(x_j) \\ &= \max_{x_k} \left[\ln \phi_k(x_k) + \ln \psi_{kj}(x_k, x_j) + \sum_{n\in\mathcal{N}^j_k} m^{(l-1)}_{n\to k}(x_k) \right]. \end{aligned} \tag{9}$$

We have replaced $\propto$ by equality in our formulations since the proportionality constant turns into an offset value in the log domain with no impact on the proposed analysis. We adopt the commonly-used exponential model to represent the probability measure defined on $\boldsymbol{x}$, i.e.,

$$p(\boldsymbol{x}) \propto \exp\left(\sum_{n\in\mathcal{V}} \theta_n x_n + \sum_{(i,j)\in\mathcal{E}} J_{ij} x_i x_j \right). \tag{10}$$

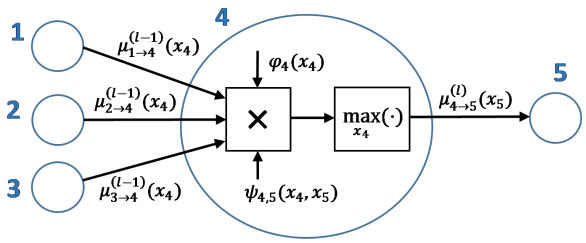


Fig. 1: An schematic diagram of the max-product algorithm illustrating how the messages are generated.

Consequently, from $p(\boldsymbol{x}|\boldsymbol{Y}) = p(\boldsymbol{Y}|\boldsymbol{x})p(\boldsymbol{x})/p(\boldsymbol{Y})$, we obtain [4]

$$p(\boldsymbol{x}|\boldsymbol{Y}) \propto \prod_{k\in\mathcal{V}} p(\boldsymbol{y}_k|x_k) \prod_{(i,j)\in\mathcal{E}} e^{J_{ij}x_i x_j}, \tag{11}$$

where the proportionality sign covers $\frac{1}{p(\boldsymbol{Y})}$. Since θ_k does not affect the proposed analysis, we have set $\theta_k = 0$ for all k. By comparing (11) to (6), we obtain

$$\phi_k(x_k) \triangleq p(\boldsymbol{y}_k|x_k), \tag{12}$$

$$\psi_{kj}(x_k, x_j) \triangleq e^{J_{kj}x_k x_j}. \tag{13}$$

Assuming Gaussian observations at the nodes, we have

$$p(\boldsymbol{y}_k|x_k) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-1}{2\sigma^2} \left\| \boldsymbol{y}_k - \boldsymbol{\mu}_{\boldsymbol{y}_k|x_k} \right\|^2 \right), \tag{14}$$

where $\boldsymbol{\mu}_{\boldsymbol{y}_k|x_k} \triangleq E[\boldsymbol{y}_k|x_k]$. For $i = 1, ..., K$, we have $y_k(i) = \xi_k s_k(i) + \nu_k(i)$ where $\xi_k \triangleq \frac{1}{2}(x_k + 1)$ and $\nu_k(i) \sim \mathcal{N}(0, \sigma^2)$. In the vector format, we have $\boldsymbol{y}_k = \xi_k \boldsymbol{s}_k + \boldsymbol{\nu}_k$ where $\boldsymbol{s}_k \triangleq [s_k(1), ..., s_k(K)]^T$ denotes a deterministic but unknown sequence of PU signal samples received at node k and $\boldsymbol{\nu}_k \triangleq [\nu_k(1), ..., \nu_k(K)]^T$. Hence, $\boldsymbol{\mu}_{\boldsymbol{y}_k|x_k} = \xi_k \boldsymbol{s}_k$. We have $x_k \in \{-1, +1\}$ for all $k \in \mathcal{V}$ while $\xi_k \in \{0, 1\}$ maps the state of x_k to the occupancy state of the radio spectrum sensed by node k. See Table I for a list of symbols used in this paper.

The max-marginals obtained by the max-product algorithm are used to conduct a distributed MAP estimation as in (5). Specifically, after l iterations, at each node the approximate max-LLR is built and compared, as a decision variable, to a predefined threshold, i.e.,

$$\lambda^{(l)}_j \triangleq \ln \frac{b^{(l)}_j(x_j = +1)}{b^{(l)}_j(x_j = -1)} \gtrless \tau_j. \tag{15}$$

which means that $\hat{x}_j = +1$ if $\lambda^{(l)}_j > \tau_j$ and $\hat{x}_j = -1$ otherwise. Note that $\lambda^{(l)}_j \approx \ln \frac{q(x_j=+1|\boldsymbol{Y})}{q(x_j=-1|\boldsymbol{Y})}$. To see the impact of messages on the decision variable, we express $\lambda^{(l)}_j$ as

$$\lambda^{(l)}_j = \gamma_j + \sum_{k\in\mathcal{N}_j} \delta^{(l)}_{k\to j}, \tag{16}$$

where γ_j denotes the local LLR obtained at node j, i.e., $\gamma_j \triangleq \ln \frac{\phi_j(x_j=+1)}{\phi_j(x_j=-1)} = \ln \frac{p(\boldsymbol{y}_j|x_j=+1)}{p(\boldsymbol{y}_j|x_j=-1)}$ while $\delta^{(l)}_{k\to j}$ denotes the LLR

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2020.3012910, IEEE Transactions on Wireless Communications

TABLE I: Main Parameters Specifying the Detector Structure

Symbol	Meaning
N	Number of sensing nodes in the network
K	Number of samples collected at each sensing node
$\mathcal{V}$	Set of vertices in the factor graph
$\mathcal{E}$	Set of edges in the factor graph
$\mathcal{N}_j$	Set of neighbors of node j
$\mathcal{N}_j^k$	Set of neighbors of node j except for node k
$\mu_{k\to j}^{(l)}(x_j)$	Max-product message sent to node j from node k
$\bar{\mu}_{k\to j}^{(l)}(x_j)$	Sum-product message sent to node j from node k
$b_j^{(l)}(x_j)$	Max-product beliefs
$\bar{b}_j^{(l)}(x_j)$	Sum-product beliefs
$m_{k\to j}^{(l)}(x_j)$	Max-product message in the log domain
$\bar{m}_{k\to j}^{(l)}(x_j)$	Sum-product message in the log domain
$\lambda_j^{(l)}$	Max-product decision variable at node j
$\bar{\lambda}_j^{(l)}$	Sum-product decision variable at node j
$\delta_{k\to j}^{(l)}$	LLR of max-product messages
$\bar{\delta}_{k\to j}^{(l)}$	LLR of sum-product messages
τ_j	Max-product detection threshold
$\bar{\tau}_j$	Sum-product detection threshold
γ_j	Local sensing outcome at node j
E_j	Energy of the PU signal received at node j
$\boldsymbol{s}_j$	Signal to be detected at node j
$\boldsymbol{y}_j$	Noisy received signal at node j
$\boldsymbol{\nu}_j$	Noise received at node j
$\boldsymbol{\mu}_{\boldsymbol{y}_j\|x_j}$	Conditional mean of $\boldsymbol{y}_j$ given x_j
σ^2	Noise variance
ξ_j	Auxiliary variable that maps x_j to $\{0,1\}$
$f(\boldsymbol{x})$	Transfer function built by the sum-product operation
$P_{\mathrm{f}}^{(j)}$	False-alarm probability at node j
$P_{\mathrm{d}}^{(j)}$	Detection probability at node j
$\phi_n(x_n)$	Single-variable factor in $p(\boldsymbol{x}\|\boldsymbol{Y})$
$\psi_{ij}(x_i,x_j)$	Pairwise factor in $p(\boldsymbol{x}\|\boldsymbol{Y})$
θ_n	Single-variable exponent factor in $p(\boldsymbol{x})$
J_{ij}	Pairwise exponent factor in $p(\boldsymbol{x})$
T	Sample-window size in updating J_{ij}'s
ζ	Determines the impact of T samples on J_{ij}'s
$\hat{x}_k^{(l)}(x_j)$	Outcome of the MLE at node k, equals to $u_{kj}^{(l)}+v_{kj}^{(l)}x_j$
$\hat{x}_k$	Outcome of the desired MAP estimation at node k

of the messages at iteration l, i.e.,

$$\begin{aligned}\delta_{k\to j}^{(l)} &\triangleq \ln\frac{\mu_{k\to j}^{(l)}(x_j=+1)}{\mu_{k\to j}^{(l)}(x_j=-1)}\\ &= m_{k\to j}^{(l)}(x_j=+1)-m_{k\to j}^{(l)}(x_j=-1).\end{aligned}\tag{17}$$

By using the signal model in (14), we obtain

$$\gamma_j=\boldsymbol{s}_j^T\boldsymbol{y}_j-\frac{1}{2}E_j,\tag{18}$$

where $E_j\triangleq\|\boldsymbol{s}_j\|^2$. Consequently, it is clear that, given x_j, the local LLR γ_j follows a Gaussian distribution. For simplicity we assume that $\sigma^2=1$. Eq. (18) indicates a matched filtering process, a.k.a., coherent detection [31] performed locally at each sensing node. In practice, since $\boldsymbol{s}_k$ is unknown, energy detection is used as the local sensing scheme [4], [24]. That is, the local sensing outcome is formed as

$$\gamma_j\triangleq\frac{1}{K}\left\|\boldsymbol{y}_j\right\|^2-\tau_0,\tag{19}$$

where τ_0 is set such that $\Pr\{\gamma_k>0|x_k=0\}=\alpha$, i.e., $\tau_0=\sigma_\nu^2\left(1+\sqrt{\frac{2}{K}}Q^{-1}(\alpha)\right)$ where $Q^{-1}(\cdot)$ denotes the inverse of the Q-function. Assuming the number of signal samples K is large enough [25]–[27], the central limit theorem states that, given x_j, the sensor outcome γ_j in (19) follows a Gaussian distribution.

The sum-product algorithm has a similar structure except that the max operator in (9) is replaced by a summation. This message-update rule is given by

$$\bar{\mu}_{k\to j}^{(l)}(x_j)\propto\sum_{x_k}\left[\phi_k(x_k)\psi_{kj}(x_k,x_j)\prod_{n\in\mathcal{N}_k^j}\bar{\mu}_{n\to k}^{(l-1)}(x_k)\right].\tag{20}$$

The beliefs made by the sum-product algorithm are denoted $\bar{b}_j(x_j)$ in this paper and calculated by (8) in which $\mu_{k\to j}^{(l)}$ is replaced by $\bar{\mu}_{k\to j}^{(l)}$. The sum-product algorithm approximates the marginal distributions of the random variables of interest, i.e., $\bar{b}_j(x_j)\approx p(x_j|\boldsymbol{Y})$. The detection process is conducted by comparing the resulting decision variable to a predefined threshold as in (15) where λ_j, $b_j(x_j)$, and τ_j are replaced by $\bar{\lambda}_j$, $\bar{b}_j(x_j)$, and $\bar{\tau}_j$, respectively.

Similar to (16), the detection variable build by the sum-product iteration can be expressed as

$$\bar{\lambda}_j^{(l)}=\gamma_j+\sum_{k\in\mathcal{N}_j}\bar{\delta}_{k\to j}^{(l)},\tag{21}$$

where $\bar{\delta}_{k\to j}^{(l)}\triangleq\bar{m}_{k\to j}^{(l)}(+1)-\bar{m}_{k\to j}^{(l)}(-1)$ while $\bar{m}_{k\to j}^{(l)}(x_j)\triangleq\ln\bar{\mu}_{k\to j}^{(l)}(x_j)$. Through some algebra, we obtain

$$\bar{\delta}_{k\to j}^{(l)}=S\left(J_{kj},\ \gamma_k+\sum_{n\in\mathcal{N}_k^j}\bar{\delta}_{n\to k}^{(l-1)}\right),\tag{22}$$

where $S(a,b)\triangleq\ln\frac{1+e^{a+b}}{e^a+e^b}$. J_{kj} is determined by a moving average of length T time slots, i.e.,

$$J_{kj}\triangleq\frac{\zeta}{T}\sum_{t=1}^{T}\left[\mathbf{1}\{\hat{x}_j(t)=\hat{x}_k(t)\}-\mathbf{1}\{\hat{x}_j(t)\neq\hat{x}_k(t)\}\right],\tag{23}$$

where ζ is a constant and $\mathbf{1}\{\cdot\}$ denotes the indicator function.

We use the first-order Taylor series expansion of S to linearize the message-update rule as

$$\bar{\delta}_{k\to j}^{(l)}\approx c_{jk}\left(\gamma_k+\sum_{n\in\mathcal{N}_k^j}\bar{\delta}_{n\to k}^{(l-1)}\right),\tag{24}$$

where $c_{jk}=\frac{(e^{2J_{kj}}-1)}{(1+e^{J_{kj}})^2}$ [4]. Consequently, at node j we have

$$\begin{aligned}\bar{\lambda}_j\approx\gamma_j&+\sum_{k\in\mathcal{N}_j}c_{jk}\gamma_k+\sum_{k\in\mathcal{N}_j}\sum_{n\in\mathcal{N}_k^j}c_{jk}c_{kn}\gamma_n\\&+\sum_{k\in\mathcal{N}_j}\sum_{n\in\mathcal{N}_k^j}\sum_{m\in\mathcal{N}_n^k}c_{jk}c_{kn}c_{nm}\gamma_m+...,\end{aligned}\tag{25}$$

where $\bar{\lambda}_j\triangleq\lim_{l\to\infty}\bar{\lambda}_j^{(l)}$. Eq. (25) shows that the sum-product algorithm is approximately a linear fusion scheme. Since the local LLRs are normal random variables, given the state of x_i's, the decision variable $\bar{\lambda}_j$ is, approximately, a normal random variable as well. We can express this linear fusion

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2020.3012910, IEEE Transactions on Wireless Communications

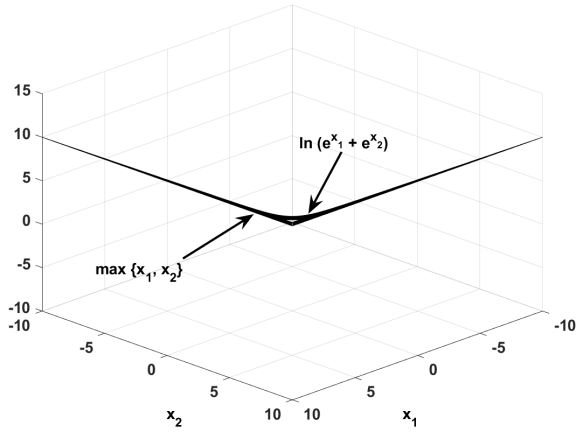


Fig. 2: $\max\{x_1, x_2\}$ provides a piece-wise linear approximation for $\ln(e^{x_1} + e^{x_2})$.

in compact form as $\bar{\lambda}_j \approx \sum_{k\in\mathcal{V}} \bar{w}_{jk}\gamma_k$ where $\bar{w}_{jk}$ denotes the weight of γ_k in this combination.

To see the connection between the sum-product and max-product operations, let us take a closer look at the messages in the sum-product algorithm. In the log domain, we have

$$\bar{m}^{(l)}_{k\to j}(x_j) \triangleq \ln \bar{\mu}^{(l)}_{k\to j}(x_j) = \ln \sum_{x_k} \exp\left[\ln \phi_k(x_k) + \ln \psi_{kj}(x_k, x_j) + \sum_{n\in\mathcal{N}_k^j} \bar{m}^{(l-1)}_{n\to k}(x_k)\right], \tag{26}$$

which shows that the messages, received at node k and combined with the likelihoods $\ln \phi_k(x_k)$ and $\ln \psi_{kj}(x_k, x_j)$, pass through the following transformation to form the message sent to node j

$$f(\boldsymbol{x}) \triangleq \ln \sum_k e^{x_k}. \tag{27}$$

As shown in Fig. 2, due to the highly selective nature of the exponential function, $f(\cdot)$ behaves like a max operator, i.e.,

$$f(\boldsymbol{x}) \approx \max_k x_k. \tag{28}$$

Consequently, we can approximate the message-update rule in the sum-product algorithm as

$$\bar{m}^{(l)}_{k\to j}(x_j) \approx \max_{x_k}\left[\ln \phi_k(x_k) + \ln \psi_{kj}(x_k, x_j) + \sum_{n\in\mathcal{N}_k^j} \bar{m}^{(l-1)}_{n\to k}(x_k)\right], \tag{29}$$

which clearly shows that *the message-update rule in the sum-product algorithm is almost the same as its counterpart in the max-product algorithm.*

Therefore, we expect the max-product algorithm to work as a linear fusion as well. More specifically, we expect to have $\lambda_j = \sum_{k\in\mathcal{V}} w_{jk}\gamma_k$ where w_{jk} denotes the weight of γ_k in this linear fusion. In the following section, we formally establish that the max-product algorithm is a distributed linear fusion scheme.

III. Analysis of the Max-Product Operation

Performance of a binary hypothesis test is commonly measured by two parameters: the probability of detection and the probability of false alarm. These performance metrics are calculated based on the statistical behavior of the decision variable λ_j. Specifically, at node j we have

$$P_{\mathrm{f}}^{(j)} = \Pr\{\lambda_j > \tau_j | x_j = -1\}, \tag{30}$$

$$P_{\mathrm{d}}^{(j)} = \Pr\{\lambda_j > \tau_j | x_j = +1\}, \tag{31}$$

where $P_{\mathrm{f}}^{(j)}$ denotes the false-alarm probability of node j and $P_{\mathrm{d}}^{(j)}$ denotes the corresponding detection probability.

Hence, we need to find the probability distribution of λ_j to measure the system performance analytically. To realize this goal, we calculate the outcome of each iteration and show that even though the iteration process involves nonlinear transformations, its outcome is a linear combination of the local LLRs. The analysis of the max-product process provided in this section does not require the node variables to be binary-valued. We only use binary-valued x_i's when evaluating the result of the proposed analysis.

Recall that the local observation at node k is represented by $\phi_k(x_k)$ while the correlation between the observations at nodes k and j is captured by $\psi_{kj}(x_k, x_j)$. Since in the beginning there are no messages received, i.e., $m^{(0)}_{k\to j}(x_j) = 0$, each node builds its message only based on its own local observation and the correlation of its random variable with the ones of the neighboring nodes. That is, at $l = 1$ the messages are created based on

$$\begin{aligned} m^{(1)}_{k\to j}(x_j) &= \max_{x_k}\Big[\ln \phi_k(x_k) + \ln \psi_{kj}(x_k, x_j)\Big] \\ &= \ln \phi_k(\hat{x}^{(1)}_k(x_j)) + \ln \psi_{kj}(\hat{x}^{(1)}_k(x_j), x_j), \end{aligned} \tag{32}$$

where $\hat{x}^{(1)}_k(x_j)$ is found by solving $\frac{\partial}{\partial x_k}[\ln \phi_k(x_k) + \ln \psi_{kj}(x_k, x_j)] = 0$ that leads to

$$\hat{x}^{(1)}_k(x_j) = u^{(1)}_{kj} + v^{(1)}_{kj} x_j, \tag{33}$$

where, by using $\frac{\partial}{\partial x_k} \ln \phi_k(x_k) = \boldsymbol{s}_k^T(\boldsymbol{y}_k - \xi_k \boldsymbol{s}_k)$, we have

$$u^{(1)}_{kj} = \frac{2\gamma_k}{E_k} - 1, \tag{34}$$

$$v^{(1)}_{kj} = \frac{2J_{kj}}{E_k}. \tag{35}$$

Consequently, at the beginning of the iteration, the message sent form node k to node j is a linear function of two components: *i*) the local LLR at node k, denoted γ_k, and *ii*) the realization of the random variable concerned at node j, i.e., x_j. We see $\hat{x}^{(1)}_k(x_j)$ as the outcome of a maximum-likelihood estimation (MLE) process at node k. This estimation provides a point at which the likelihood functions at node k are evaluated to build a message sent to node j. Please make sure to distinguish between the MLE performed locally at each node and the MAP estimation discussed earlier.

As we show in the following, the linear behavior observed in (33) propagates throughout the entire iteration. Specifically, at the l'th iteration, the MLE results at node k, which build the messages sent to node j, are in the form of linear combinations

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2020.3012910, IEEE Transactions on Wireless Communications

of x_j and local LLRs obtained at nodes located within less than l hops from node j. Consequently, given x_j, the decision variable at node j (i.e., λ_j) is built by a linear fusion of the local LLRs obtained at node j and at all the nodes located within less than l hops from node j. In other words, the hypothesis test result obtained by the max-product algorithm is equivalent to the one obtained by a distributed linear data-fusion scheme whose scope is increased by every iteration.

We now clarify this observation by solving the iterative optimizations in (9). For $l = 2$, we have

$$m^{(2)}_{k\to j}(x_j) = \max_{x_k}\Bigg\{ \ln\phi_k(x_k) + \ln\psi_{kj}(x_k,x_j) + \sum_{n\in\mathcal{N}_k^j} \max_{x_n}\Big[\ln\phi_n(x_n) + \ln\psi_{nk}(x_n,x_k)\Big]\Bigg\}, \tag{36}$$

which leads to

$$m^{(2)}_{k\to j}(x_j) = \ln\phi_k(\hat{x}^{(2)}_k(x_j)) + \ln\psi_{kj}(\hat{x}^{(2)}_k(x_j),x_j) + \sum_{n\in\mathcal{N}_k^j}\Big[\ln\phi_n(\hat{x}^{(1)}_n(\hat{x}^{(2)}_k(x_j))) + \ln\psi_{nk}(\hat{x}^{(1)}_n(\hat{x}^{(2)}_k(x_j)),\hat{x}^{(2)}_k(x_j))\Big], \tag{37}$$

where $\hat{x}^{(2)}_k(x_j)$ is found by solving

$$\frac{\partial}{\partial x_k}\Bigg\{ \ln\phi_k(x_k) + \ln\psi_{kj}(x_k,x_j) + \sum_{n\in\mathcal{N}_k^j}\Big[\ln\phi_n(\hat{x}^{(1)}_n(x_k)) + \ln\psi_{nk}(\hat{x}^{(1)}_n(x_k),x_k)\Big]\Bigg\} = 0, \tag{38}$$

which leads to

$$\hat{x}^{(2)}_k(x_j) = u^{(2)}_{kj} + v^{(2)}_{kj}x_j, \tag{39}$$

where

$$u^{(2)}_{kj} = \frac{u^{(1)}_{kj} + \frac{1}{E_k}\sum_{n\in\mathcal{N}_k^j} E_n u^{(1)}_{nk} v^{(1)}_{nk}}{1 - \frac{1}{E_k}\sum_{n\in\mathcal{N}_k^j} E_n\Big[v^{(1)}_{nk}\Big]^2}, \tag{40}$$

$$v^{(2)}_{kj} = \frac{v^{(1)}_{kj}}{1 - \frac{1}{E_k}\sum_{n\in\mathcal{N}_k^j} E_n\Big[v^{(1)}_{nk}\Big]^2}. \tag{41}$$

Consequently, $\hat{x}^{(2)}_k(x_j)$ is built as a linear function of x_j plus a linear combination of the local LLRs obtained at node k and at its one-hop neighbors. More specifically, from (39), (40), and (41) we see that $\hat{x}^{(2)}_k(x_j)$ is formed as a linear combination of γ_k with γ_n's for $n \in \mathcal{N}_k^j$. In addition, note that $v^{(2)}_{kj}$ is a constant whereas $u^{(2)}_{kj}$ is a random variable that captures the statistical behavior of the local observations.

Through iterative calculations for $l = 3, 4, ...$, we see that the MLE result $\hat{x}^{(l)}_k(x_j)$ has similar components, i.e., a linear function of x_j plus a linear combination of the local LLRs obtained within less than l hops from node j, i.e.,

$$\hat{x}^{(l)}_k(x_j) = u^{(l)}_{kj} + v^{(l)}_{kj}x_j, \tag{42}$$

where $v^{(l)}_{kj}$ is a constant and $u^{(l)}_{kj}$ can be expressed as a linear combination of the local likelihood values, i.e.,

$$u^{(l)}_{kj} = \sum_{i\in\mathcal{V}} \omega^{(l)}_{kj}(i)\gamma_i, \tag{43}$$

where $\omega^{(l)}_{kj}(i)$ denotes the weight of γ_i in this linear combination. Moreover, $\omega^{(l)}_{kj}(i)$ is zero if node i is located more than $l-1$ hops away from node j. Therefore, by increasing l, we expand the maximum radius around node j within which the local likelihoods are combined to build $u^{(l)}_{kj}$. Consequently, *to include all the local LLRs in the fusion process, the maximum number of iterations does not need to be greater than the length of the longest path in the network graph.* This justifies the observation in [24] where the desired detection performance is achieved by only a few iterations.

It is worth noting that, one does not need to perform many iterative calculations to see the linearity of the final result. Starting from $m^{(1)}_{k\to j}(x_j)$ in (32), we see that the term $\ln\phi_k(x_k) + \ln\psi_{kj}(x_k,x_j)$ is concave quadratic in x_k. Hence, its partial derivative leads to a linear equation, which, in turn, leads to a linear expression for $\hat{x}^{(1)}_k(x_j)$ in terms of γ_k and x_j. Moreover, in order to build $m^{(l)}_{k\to j}$ from $m^{(l-1)}_{k\to j}$ one needs to add some terms, inside the max operator in (9), with similar concave quadratic attributes. The only difference is that, these new terms have as their arguments some linear expressions with positive coefficients. Note that x_k in (9) is calculated by (42). Since these linear transformations preserve the concave quadratic nature of the whole expression inside the max operator, the maximum in (9) is found by solving a linear equation that leads to a linear expression in terms of $u^{(l)}_{kj}$ and x_j.

Based on this observation, we propose a set of formulas to recursively calculate $\hat{x}^{(l)}_k(x_j)$. This calculation is realized by using a quadratic form to represent the messages, which can be expressed as

$$m^{(l)}_{k\to j}(x_j) = a^{(l)}_{kj}x_j^2 + b^{(l)}_{kj}x_j. \tag{44}$$

The partial derivative of the messages is then a linear expression as

$$\frac{\partial}{\partial x_j} m^{(l)}_{k\to j}(x_j) = 2a^{(l)}_{kj}x_j + b^{(l)}_{kj}. \tag{45}$$

Now, by solving the following equation

$$\frac{\partial}{\partial x_k}\Big[\ln\phi_k(x_k) + \ln\psi_{kj}(x_k,x_j) + \sum_{n\in\mathcal{N}_k^j} m^{(l-1)}_{n\to k}(x_k)\Big] = 0, \tag{46}$$

we link $a^{(l)}_{kj}$ and $b^{(l)}_{kj}$ to $u^{(l)}_{kj}$ and $v^{(l)}_{kj}$ as

$$u^{(l)}_{kj} = \frac{u^{(1)}_{kj} + \frac{2}{E_k}\sum_{n\in\mathcal{N}_k^j} b^{(l-1)}_{nk}}{1 - \frac{1}{E_k}\sum_{n\in\mathcal{N}_k^j} E_n a^{(l-1)}_{nk}}, \tag{47}$$

$$v^{(l)}_{kj} = \frac{v^{(1)}_{kj}}{1 - \frac{1}{E_k}\sum_{n\in\mathcal{N}_k^j} a^{(l-1)}_{nk}}. \tag{48}$$

Hence, by using $u^{(l-1)}_{kj}$ and $v^{(l-1)}_{kj}$ we calculate $a^{(l-1)}_{kj}$ and $b^{(l-1)}_{kj}$, which are then used to obtain $u^{(l)}_{kj}$ and $u^{(l)}_{kj}$. This recursive calculation starts from $u^{(1)}_{kj}$ and $v^{(1)}_{kj}$ in (34) and (35).

The fusion weights in (43) are determined in terms of $v_{kj}^{(l)}$'s. As we saw in (35) and (41), $v_{kj}^{(l)}$'s are determined in terms of J_{kj}'s, which capture the inter-dependencies of the random variables in the MRF. We will further discuss this point and its implications on the system design later.

Eqs. (47) and (48) indicate that, *the higher the received SNR level at node k, the lower the impact of other nodes on the data sent from node k to node j*. Hence, node k relies more on its own local observation when it is operating under good SNR conditions. Otherwise, it relies more on the data received from its neighbors. In addition, according to (40) and (41), each LLR received from a neighbor is scaled by the SNR level perceived at that neighbor. Consequently, the message-update rule in (9) works like a maximal-ratio combining (MRC) scheme.

Since the outcomes of the MLEs are derived in closed form, we can now see their impact on the binary hypothesis test. To this end, we show that $\delta_{k\to j}^{(l)}$ is a linear combination of the local LLRs. First note that for all k, n we have

$$\ln\phi_k(u+v)-\ln\phi_k(u-v)=2v\left[\gamma_k-\frac{E_k}{2}(u+1)\right], \quad (49)$$

$$\ln\psi_{nk}(u_1+v_1,u_2+v_2)-\ln\psi_{nk}(u_1-v_1,u_2-v_2)= 2J_{nk}(u_1v_2+u_2v_1), \quad (50)$$

which are linear expressions in u, u_1, and u_2.

Then, recall that $\hat{x}_k^{(l)}(x_j=\pm1)=u_{kj}^{(l)}\pm v_{kj}^{(l)}$. Consequently, $\delta_{k\to j}^{(l)}$ in (17) contains expressions, in the form of (49) and (50), that are linear functions of $u_{kj}^{(l)}$'s. To clarify this observation, we focus on $l=2$ here. A similar argument can be made for $l>2$. We see that,

$$\begin{aligned}\delta_{k\to j}^{(2)} &= m_{k\to j}^{(2)}(x_j=+1)-m_{k\to j}^{(2)}(x_j=-1)= \\ &\ln\phi_k(\hat{x}_k^{(2)}(+1))-\ln\phi_k(\hat{x}_k^{(2)}(-1)) \\ &+\ln\psi_{kj}(\hat{x}_k^{(2)}(+1),+1)-\ln\psi_{kj}(\hat{x}_k^{(2)}(-1),-1) \\ &+\sum_{n\in\mathcal{N}_k^j}\Big[\ln\phi_n(\hat{x}_n^{(1)}(\hat{x}_k^{(2)}(+1)))-\ln\phi_n(\hat{x}_n^{(1)}(\hat{x}_k^{(2)}(-1)))\Big] \\ &+\sum_{n\in\mathcal{N}_k^j}\Big[\ln\psi_{nk}(\hat{x}_n^{(1)}(\hat{x}_k^{(2)}(+1)),\hat{x}_k^{(2)}(+1)) \\ &\qquad -\ln\psi_{nk}(\hat{x}_n^{(1)}(\hat{x}_k^{(2)}(-1)),\hat{x}_k^{(2)}(-1))\Big], \end{aligned} \quad (51)$$

where

$$\hat{x}_k^{(2)}(\pm1)=u_{kj}^{(2)}\pm v_{kj}^{(2)}, \quad (52)$$

$$\begin{aligned}\hat{x}_n^{(1)}(\hat{x}_k^{(2)}(\pm1)) &= u_{nk}^{(1)}+v_{nk}^{(1)}\hat{x}_k^{(2)}(\pm1) \\ &= u_{nk}^{(1)}+v_{nk}^{(1)}u_{kj}^{(2)}\pm v_{nk}^{(1)}v_{kj}^{(2)}. \end{aligned} \quad (53)$$

Comparing (51) to (49) and (50) makes it clear that, (51) is a linear combination of $u_{nk}^{(1)}$ and $u_{kj}^{(2)}$, for $k\in\mathcal{N}_j$ and $n\in\mathcal{N}_k^j$. Since $u_{nk}^{(1)}$ and $u_{kj}^{(2)}$ are, respectively, linear combinations of the local likelihoods γ_k's for $k\in\mathcal{N}_j$ and γ_n's for $n\in\mathcal{N}_k^j$, we conclude that $\lambda_j^{(2)}$ is a linear combination of γ_k's for $k\in\{j\}\cup\mathcal{N}_j$ and γ_n's for $n\in\mathcal{N}_k^j$.

Through similar arguments, we can show that the resulting decision variable after l iterations is constructed as a linear combination of the local likelihoods, i.e.,

$$\lambda_j^{(l)}=\sum_{i\in\mathcal{M}_j^{(l)}}w_{ji}^{(l)}\gamma_i, \quad (54)$$

where $w_{ji}^{(l)}$'s denote the weights in this linear combination while $\mathcal{M}_j^{(l)}$ denotes the set of indices referring to node j and all its neighbors within its $(l-1)$-hop distance. Consequently, given enough time or when the max-product algorithm converges to a fixed point, the decision variable is built as

$$\lambda_j=\sum_{i\in\mathcal{V}}w_{ji}\gamma_i, \quad (55)$$

where $\lambda_j\triangleq\lim_{l\to\infty}\lambda_j^{(l)}$. We can summarize these observations in the following proposition.

Proposition I: The max-LLRs obtained by running the max-product algorithm, over a network described by the factor graph in (6) where $\ln\phi_k(x_k)+\ln\psi_{kj}(x_k,x_j)$ is concave quadratic in x_k, are built as linear combinations of the local LLRs in that network. Moreover, at the l'th iteration, each node combines local LLRs from its neighbors located within less than l hops away from itself.

We know that J_{kj}'s specify the factor graph, which models the stochastic behavior of the network. The proposed analysis shows that the fusion weights in (11) are determined in terms of J_{kj}'s. Therefore, finding the optimal J_{kj}'s to best represent the network behavior is equivalent to optimizing the fusion weights in (55). This observation gives us a deeper insight into the impact of the MRF parameters on the system performance and enables us to offer our second proposition as follows.

Proposition II: Learning the parameters of the pairwise factor graph in (11) to best represent the statistical correlations in a network of distributed agents and running the max-product algorithm based on that graph can be viewed as the optimization of a distributed linear data-fusion scheme in that network.

Linear data-fusion has been extensively investigated in the literature. For the completeness of presentation, we briefly explain how to realize optimal linear data-fusion in the following section. An interested reader may refer to [25]–[27] for more comprehensive discussions. We have explained in detail the proposed optimization framework in [4] where we optimize the sum-product algorithm. In the following section, we discuss why that framework can be applied to the max-product algorithm as well.

IV. Linear Data-Fusion

The fact that the decision variable λ_j is the result of a linear fusion facilitates analyzing the system behavior and optimizing its performance. Given the status of x_i's, the local LLRs follow Gaussian distributions. Consequently, the decision variable λ_j follows a Gaussian distribution and we only need its first- and second-order statistics to derive its probability distribution. We can find the impact of fusion weights (i.e., w_{jk}'s) on the system performance by noting that they determine the contribution of each node on the mean and variance of λ_j. The system false-alarm and detection probabilities at node j depend not only on the state of x_j, but also, in general, on

the state of all other x_i's being sensed throughout the entire network. Consequently, based on the total probability theorem, we have

$$\begin{aligned} g_j(\tau_j, v) &\triangleq \Pr\{\lambda_j > \tau_j | x_j = v\} \\ &= \sum_{\boldsymbol{b}\in\{-1,1\}^{N-1}} \Pr\{\lambda_j > \tau_j | \boldsymbol{x}_{(j)} = \boldsymbol{b}, x_j = v\} p_{\boldsymbol{x}_{(j)}|x_j}(\boldsymbol{b}|v) \\ &= \sum_{\boldsymbol{b}\in\{-1,1\}^{N-1}} Q\left(\frac{\tau_j - \eta_{j,v}(\boldsymbol{b})}{\sigma_{j,v}(\boldsymbol{b})}\right) p_{\boldsymbol{x}_{(j)}|x_j}(\boldsymbol{b}|v), \end{aligned} \tag{56}$$

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 $v = -1, 1$, $\eta_{j,v}(\boldsymbol{b}) \triangleq E[\lambda_j|\boldsymbol{x}_{(j)} = \boldsymbol{b}, x_j = v]$ and $\sigma^2_{j,v}(\boldsymbol{b}) \triangleq \text{Var}[\lambda_j|\boldsymbol{x}_{(j)} = \boldsymbol{b}, x_j = v]$. $Q(x) \triangleq \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz$ is the so-called Q-function. Note that $\boldsymbol{x}_{(j)} \in \{-1, 1\}^{N-1}$ contains all x_i's except for x_j. It is clear that $P_{\text{f}}^{(j)} = g_j(\tau_j, -1)$ and $P_{\text{d}}^{(j)} = g_j(\tau_j, 1)$. By solving $g_j(\tau_j, -1) = \alpha$ 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 be, respectively, equal to α or β. We have discussed how to find the detection threshold to guarantee a predefined performance level in [4].

A. Centralized Linear Fusion

In a centralized distributed detection [30], the local sensing outcomes are constantly reported to a so-called fusion center (FC), which is usually a more powerful node like a base station or an access point. The FC uses the received information from the cooperating nodes to estimate the statistics required for a linear fusion scheme and then directly combines the received local sensing results into a global decision variable. In this fusion process, the vector of decision variables is built as

$$\boldsymbol{\lambda} = \boldsymbol{W}\boldsymbol{\gamma}, \tag{57}$$

where $\boldsymbol{W}$ denotes the weighting coefficients, see (55).

Now, the problem is to find the optimal fusion weights and detection thresholds to have the best detection performance. A linear fusion scheme can be optimized by assigning rewards to the detection and costs to the false-alarm incidents. In particular, we assume that the system obtains reward r_i for performing a correct detection at node i and incurs cost c_i when a false alarm happens. Therefore, the average reward obtained by the system regarding the detection performance at node i is $r_i P_{\text{d}}^{(i)}$ whereas the average cost of false alarms happening at that node is $c_i P_{\text{f}}^{(i)}$. Consequently, the aggregate reward obtained throughout the entire network is stated as $R(\boldsymbol{\lambda}, \boldsymbol{\tau}) \triangleq \mathbf{r}^T \mathbf{P}_{\text{d}}$ where $\mathbf{r} = [r_1, ..., r_N]^T$ and $\mathbf{P}_{\text{d}} = [P_{\text{d}}^{(1)}, ..., P_{\text{d}}^{(N)}]^T$ while the aggregate cost of false alarms is taken into account by $C(\boldsymbol{\lambda}, \boldsymbol{\tau}) = \mathbf{c}^T \mathbf{P}_f$ where $\mathbf{c} = [c_1, ..., c_N]^T$ and $\mathbf{P}_{\text{f}} = [P_{\text{f}}^{(1)}, ..., P_{\text{f}}^{(N)}]^T$. Accordingly, the system performance optimization is formulated as

$$\begin{aligned} &\max_{\mathbf{W}, \boldsymbol{\tau}} R(\boldsymbol{\lambda}, \boldsymbol{\tau}), \\ \text{s.t.,} \quad &C(\boldsymbol{\lambda}, \boldsymbol{\tau}) \le C_0, \quad \mathbf{P}_{\text{f}}(\boldsymbol{\lambda}, \boldsymbol{\tau}) \le \boldsymbol{\alpha}, \quad \mathbf{P}_{\text{d}}(\boldsymbol{\lambda}, \boldsymbol{\tau}) \ge \boldsymbol{\beta}, \end{aligned} \tag{P1}$$

where C_0 denotes the maximum cost allowed while $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_N]^T$ and $\boldsymbol{\beta} = [\beta_1, ..., \beta_N]^T$ denote the per-node constraints the system has to meet when performing the hypothesis test. That is, we optimize the aggregate system performance while maintaining the constraints $P_{\text{f}}^{(j)} \le \alpha_j$ and $P_{\text{d}}^{(j)} \ge \beta_j$ for $j = 1, ..., N$. The optimization problem in (P1) is solved in [32] in the context of distributed multiband spectrum sensing in CR networks.

B. Decentralized Linear Fusion

The message-passing algorithms are of special interest in decentralized distributed settings where there is no FC and the network has to conduct the detection process based on limited computation and communication resources offered only by the sensing nodes. Here we discuss how to optimize the message-passing process in such a design scenario.

Since the max-product algorithm works like the sum-product algorithm, we first explain how to derive the optimal detection performance by optimizing the sum-product algorithm. This optimization framework is based on the fact that the resulting linear fusion favors the LLRs received from shorter distances, especially, the ones generated at the one-hop neighbors. Then, we show that the linear fusion realized by the max-product algorithm has the same property. Therefore, the same optimization framework gives the optimal detection performance of the max-product algorithm as well.

Eq. (25) reveals the effect of the network topology on how the fusion coefficients are arranged by a sum-product iteration. Specifically, for the one-hop neighbors of node j we have one coefficient c_{jk} affecting the local LLRs received, for the two-hop neighbors we have two coefficients $c_{jk}c_{kn}$ and so on. Since $|c_{jk}| < 1$, the system favors LLRs received through the shortest paths when building the decision variables. Moreover, 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. Accordingly, we decompose the problem of optimizing c_{jk}'s into N small optimizations, each carried out locally at a sensing node, that collectively lead to a near-optimal performance in a decentralized distributed network configuration.

In the proposed optimization framework each node is focused on the fusion of the LLRs received from its one-hop neighbors by using the following approximation [4]

$$\bar{\lambda}_j \approx \gamma_j + \sum_{k\in\mathcal{N}_j} c_{jk}\gamma_k, \tag{58}$$

which is used to formulate the local optimizations as

$$\begin{aligned} &\max_{\mathbf{c}_j, \tau_j} P_{\text{d}}^{(j)}(\bar{\lambda}_j), \\ \text{s.t.,} \quad &P_{\text{f}}^{(j)}(\bar{\lambda}_j) \le \alpha_j, \\ &|c_{jk}| < \tfrac{1}{\max_n |\mathcal{N}_n| - 1}, \forall k \in \mathcal{N}_j, \end{aligned} \tag{P2}$$

where we maximize the detection probability at node j while maintaining its false-alarm probability below the predefined threshold α. This optimization is based on the Neyman-Pearson method [33]. $P_{\text{d}}^{(j)}$ and $P_{\text{f}}^{(j)}$ are derived approximately

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2020.3012910, IEEE Transactions on Wireless Communications

by

$$g_j(\tau_j, v) \approx \sum_{\boldsymbol{b}\in\{0,1\}^{|\mathcal{N}_j|}} Q\left(\frac{\tau_j - \tilde{\eta}_{j,v}(\boldsymbol{b})}{\tilde{\sigma}_{j,v}(\boldsymbol{b})}\right) p_{\tilde{\boldsymbol{x}}_{(j)}|x_j}(\boldsymbol{b}|v), \quad (59)$$

where $\tilde{\boldsymbol{x}}_{(j)}$ is a vector that contains x_i's with $i \in \mathcal{N}_j$ while for $v = 0, 1$, we have $\tilde{\eta}_{j,v}(\boldsymbol{b}) \triangleq E[\bar{\lambda}_j|\tilde{\boldsymbol{x}}_{(j)} = \boldsymbol{b}, x_j = v]$ and $\tilde{\sigma}^2_{j,v}(\boldsymbol{b}) \triangleq \text{Var}[\bar{\lambda}_j|\tilde{\boldsymbol{x}}_{(j)} = \boldsymbol{b}, x_j = v]$. 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 $\bar{\lambda}_j$ given the value of x_j and its immediate neighbors. This optimization can be solved by the blind adaptation method provided in [4] when the required statistics are not available a priori. The last constraint in (P2) is imposed by the *contracting mapping principle* [34], which guarantees the convergence of the message-passing iteration.

By solving (P2), the optimal c_{jk}'s are found in terms of the correlations between the LLRs made at nodes j and k. Consequently, if node n is connected to node j through node k, 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 (25) while each node sees its immediate neighbors only when optimizing its own fusion coefficients. This fusion process is inline with the Markovian structure of the factor graph. Recall that, the correlation between x_n and x_j is accounted for in (6) by two factors $\psi_{k,n}(x_k, x_n)$ and $\psi_{jk}(x_j, x_k)$ multiplied together within $p(\boldsymbol{x}|\boldsymbol{Y})$.

The same approach can be used to optimize the max-product algorithm. The reason is that the local LLRs received from the one-hop neighbors have dominant effects on the decision variables build by the max-product operation. This is a behavior we saw earlier in the sum-product algorithm. To clarify this observation, when formulating the decision variable $\lambda_j^{(l)}$, we use the fact that $v_{kj}^{(1)}$ is proportional to $\frac{1}{K}$ and, assuming the number of samples K to be large [25]–[27], we can see that the system favors data received from closer distances. Again, we focus on $l = 2$ for simplicity. By using (49) and (50) while approximating the terms proportional to $\frac{1}{K}$ by zero, we have

$$\begin{aligned}&\ln\phi_k(\hat{x}_k^{(2)}(+1)) - \ln\phi_k(\hat{x}_k^{(2)}(-1))\\ &\approx v_{kj}^{(1)}\left[\gamma_k - \frac{E_k}{2}\left(u_{kj}^{(1)} + 1\right)\right] = 0, \quad (60)\end{aligned}$$

$$\ln\psi_{kj}(\hat{x}_k^{(2)}(+1), +1) - \ln\psi_{kj}(\hat{x}_k^{(2)}(-1), -1) \approx 2J_{kj}u_{kj}^{(1)}, \quad (61)$$

$$\begin{aligned}&\ln\phi_n(\hat{x}_n^{(1)}(\hat{x}_k^{(2)}(+1))) - \ln\phi_n(\hat{x}_n^{(1)}(\hat{x}_k^{(2)}(-1)))\\ &= v_{nk}^{(1)}v_{kj}^{(2)}\left[\gamma_n - \frac{E_n}{2}\left(u_{nk}^{(1)} + v_{nk}^{(1)}u_{kj}^{(2)} + 1\right)\right] \approx 0, \quad (62)\end{aligned}$$

$$\begin{aligned}&\ln\psi_{nk}(\hat{x}_n^{(1)}(\hat{x}_k^{(2)}(+1)), \hat{x}_k^{(2)}(+1))\\ &- \ln\psi_{nk}(\hat{x}_n^{(1)}(\hat{x}_k^{(2)}(-1)), \hat{x}_k^{(2)}(-1))\\ &= 2J_{kj}\left[(u_{nk}^{(1)} + v_{nk}^{(1)}u_{kj}^{(2)})v_{kj}^{(2)} + u_{kj}^{(2)}v_{nk}^{(1)}\right] \approx 0. \quad (63)\end{aligned}$$

By plugging these approximations into (51), we derive the fusion result in (25) as[1]

$$\lambda_j \approx \gamma_j + \sum_{k\in\mathcal{N}_j} J_{kj}\gamma_k, \quad (64)$$

which shows that, firstly, the decision variable of each node can be derived approximately by linearly combining its local LLR with the LLRs obtained at its one-hop neighbors; secondly, the likelihoods are scaled in this combination by J_{kj}'s, which capture the correlation between the node variables; and thirdly, we only need the statistics of the one-hop neighbors here to analyze the stochastic behavior of λ_j.

Since J_{kj}'s are our degrees of freedom in this design, (58) and (64) are the same from an optimization point of view. Note that, c_{jk} is a monotonically-increasing function of J_{kj}.

To sum up, the max-product and sum-product operations provide almost the same message-passing algorithms in which the local LLRs are combined linearly to build the decision variables while the closer neighbors having a more significant contribution on the decision variables obtained. Consequently, the optimal detection performance of a max-product-based system can be achieved by the optimal linear message-passing algorithm defined by (24) in which the coefficients are obtained by (P2).

C. Discussion

In case there is a node at which the maximum of $q(x_j|\boldsymbol{y})$ is not attained at a unique value, then the hypothesis test conducted by using the max-marginals is not necessarily equal to the MAP estimation in (1). This means that compared to the MAP estimation we may have some extra error in the system performance. However, by using the proposed analysis, now we can control this error and even minimize it. In addition, this performance optimization can be realized with low computational complexity and in a distributed setting. Hence, in practice, the proposed message-passing process provides performance guarantees and ease of implementation not typically available when dealing with a generic MAP estimation process. In the following section, we show that the proposed detector closely achieves the optimal performance level.

Moreover, the detection performance is affected by the parameters adopted for the MRF and finding the optimal or even near-optimal values for those parameters is certainly a challenge, see e.g., [35]. Based on the proposed analysis, now we can optimize the resulting data-fusion process and that is equivalent to optimizing the parameters of the MRF. This was not possible before.

V. Numerical Results

In this section, we verify our analysis by computer simulations. Fig. 3 demonstrates the network configuration considered in our simulations. We use the same network structure as in [4]. Specifically, five SUs are cooperating in an ad-hoc setting via a parallel message-passing iteration, in sensing the radio spectrum to find vacant bands temporarily not in

[1]We have merged 2 into J_{kj}.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2020.3012910, IEEE Transactions on Wireless Communications

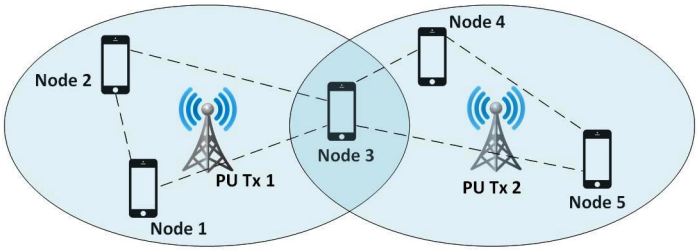
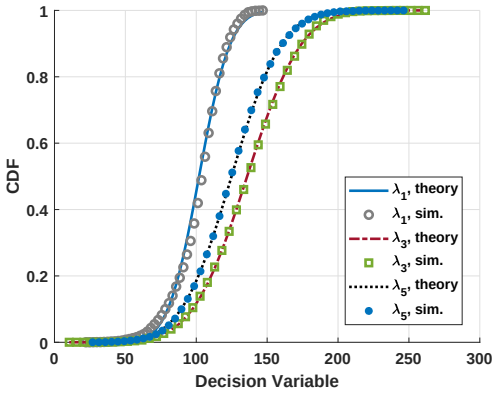


Fig. 3: The network configuration considered in the simulations. Five sensing nodes cooperate to find transmission opportunities within the spectrum bands allocated to two primary transmitters. The dashed lines depict the links between the sensing nodes through which the distributed detection is conducted.

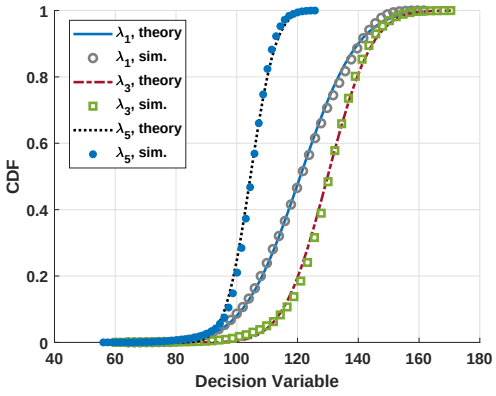
use by two PU transmitters. Nodes 1 and 2 are located within the range of PU transmitter 1 and nodes 4 and 5 are located within the range of PU transmitter 2 while node 3 can receive signals from both of the PU transmitters. The distributed detection is conducted through the one-hop links depicted in Fig. 3 by dashed lines. We realize a spatially-correlated occupancy pattern by making the PU transmitters exhibit correlated random on and off periods. This is an extension of the simulation scenario in [24] where one of the PU transmitters is constantly on while the other one is off all the time.

The spatial diversity in the network structure is accounted for by assigning different SNR levels to different nodes. Specifically, the SNR levels, in dB, of the signals received from PU transmitter 1 at nodes 1, 2, and 3 are $\rho+\Delta\rho$, ρ, and $\rho-\Delta\rho$, while the SNR levels of the signals received from PU transmitter 2 at nodes 3, 4, and 5 are ρ, $\rho-\Delta\rho$, and $\rho+\Delta\rho$, respectively. Consequently, ρ denotes the average SNR level in the network while $\Delta\rho$ measures the level of SNR dispersion among the network nodes.

We first study the statistical behavior of the decision variables built by the max-product algorithm. Specifically, we show that, given the status of the PU transmitters, λ_i's follow Gaussian distributions when coherent detection or energy detection is used in the sensing nodes. Figs. 4a and 4b depict the cumulative distribution functions (CDF) of the decision variables obtained at nodes 1, 3, and 5 when coherent detection is used at each node by processing 100 samples of the received signal, i.e., when $K = 100$. In this simulation, $\rho = -5$ dB and $\Delta\rho = 1$ dB while J_{kj} values are randomly drawn, with a uniform distribution, from $(0, 100)$. For a given realization of J_{kj}'s, each data point is obtained by averaging over 10,000 detection outcomes. Fig. 4a depicts the case in which only PU transmitter 1 is active while in the case depicted in Fig. 4b both PU transmitters are active. For each decision variable, we have provided two curves. The curves labeled "sim." depict the CDFs of the decision variables observed in our simulations whereas the ones labeled "theory" depict Gaussian CDFs fitted to the behavior of those decision variables. We can now clearly see the Gaussian behavior in the decision variables and this behavior validates our argument regarding the linearity of the max-product operation.



(a) PU Tx 1 is active and PU Tx 2 is inactive.



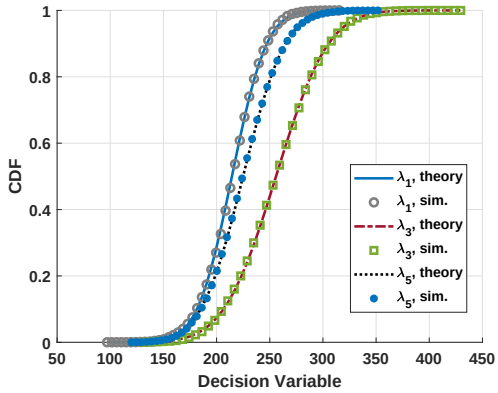
(b) Both PU transmitters are active.

Fig. 4: Cumulative distribution functions of the decision variables built by the max-product algorithm while coherent detection is used as the local sensing method. The resulting decision variables follow Gaussian distributions given the status of the PU transmitters.

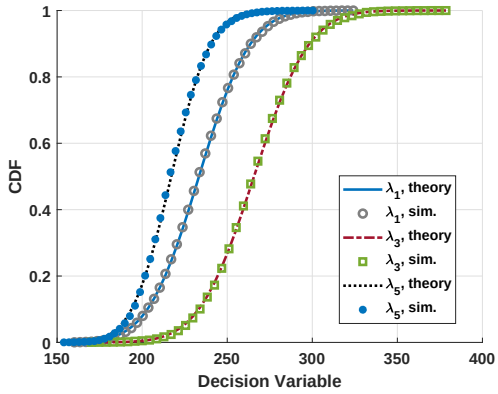
Such a linear behavior is also demonstrated in Figs. 5a and 5b where energy detection is used as the local sensing method. Again, we see that the decision variables obtained by the max-product algorithm follow Gaussian distributions. The reason is that, when we replace matched filtering by energy detection we do not alter the structure of the max-product algorithm, which leads to a linear combination of the local sensing outcomes. We only change the local sensing outcomes exchanged by the sensing nodes. Since the local sensing outcomes produced by energy detection are Gaussian random variables, a linear combination of those outcomes follows a Gaussian distribution as well.

Since we have now established that the max-product algorithm is a linear fusion method, we know that its performance level is bounded from above by the performance of the optimal linear fusion scheme. We also know from [25] that the performance of the optimal linear fusion is very close to that of the optimal detector, i.e., likelihood-ratio test. In addition, we expect the linear message-passing framework in (P2), which closely achieves the optimal detection performance, to outperform both the max-product and sum-product algorithms. We also expect the proposed method to outperform the equal-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2020.3012910, IEEE Transactions on Wireless Communications



(a) PU Tx 1 is active and PU Tx 2 is inactive.



(b) Both PU transmitters are active.

Fig. 5: Cumulative distribution functions of the decision variables built by the max-product algorithm while energy detection is used as the local sensing method. The resulting decision variables follow Gaussian distributions given the status of the PU transmitters.

gain combining (EGC) method, which is a linear fusion scheme that treats all local sensing outcomes equally [31]. All these expectations are confirmed by the results depicted in Fig. 6.

In Fig. 6, we compare the performance levels of all these methods under different SNR levels. Specifically, for the SNR dispersion level of $\Delta\rho = 0.1\rho$, the average detection rate of the different methods discussed are depicted in Fig. 6 vs. different values of the average SNR level ρ while their false-alarm rates are fixed at 0.1. The curves in Fig. 6 measure the average of the detection and false-alarm rates over all of the five sensing nodes in the network while each data point is obtained by averaging over 20,000 sensing outcomes. Each sensing outcome is calculated by processing 100 received signal samples at each node for energy detection. A window of $T = 2500$ time slots is used for training the sum-product, max-product, and the proposed linear message-passing algorithms. We realize EGC by $c_{jk} = c_0$, for all $(j,k) \in \mathcal{E}$, where c_0 is a constant.

In Fig. 6, we depict the average detection rate achieved by the max-product algorithm for $\zeta = 0.1, 0.3, 1.0$ in (23) by curves labeled, respectively, as "mp0.1", "mp0.3" and "mp1.0"

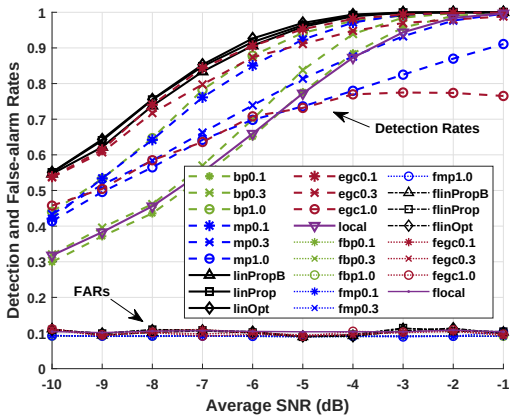


Fig. 6: Performance of different detection methods under different average SNR levels. The SNR dispersion level is $\Delta\rho = 0.1\rho$. The detection and false-alarm rates are obtained by averaging the corresponding rates over all five sensing nodes in the network.

while for the corresponding curves regarding the sum-product algorithm we use labels "bp0.1", "bp0.3" and "bp1.0". The optimal linear fusion is labeled "linOpt" and the proposed linear message-passing algorithm in [4] is labeled "linProp" while the detection rate of its blind version is labeled "linPropB". EGC is realized in our simulations for $c_0 = 0.1, 0.3, 1.0$ whose detection rates in Fig. 6 are labeled, respectively, by "egc0.1", "egc0.3" and "egc1.0". The label "local" refers to the local sensing method, which is energy detection here, performed individually at each node and without cooperating with other nodes. To refer to the false-alarm rate (FAR) of a specific method we put an "f" in the beginning of the label already used for the detection rate of that method. For instance, we use "fmp0.1" to refer to the FAR of the detection method labeled "mp0.1".

The blind optimization in [4] is conducted based on an offline iterative estimation of the required statistics where each node, say node j, processes data received from its neighbors to derive a reliable $\hat{x}_j$, which is then used in estimating the required statistics, i.e., $\mathrm{E}[\gamma_k|x_j]$ and $\mathrm{cov}(\gamma_i, \gamma_k|x_j)$ for $i,k \in \mathcal{N}_j$. This estimation can be improved by adding a majority rule [36] to the iteration. Specifically, at each iteration node j corrects its decision by applying the majority rule on $\hat{x}_j$ and the estimates of x_j inferred from γ_k's generated by its neighbors. Those estimates are obtained by thresholding γ_k's received at node j while the thresholds are simply updated by using the mean and variance of the same received data. Iterative correction of $\hat{x}_j$'s significantly increases the effectiveness of our offline adaptation and enables us to achieve near-optimal results in very low SNR regimes as shown in Fig. 6.

As we expected, Fig. 6 shows that the optimal linear fusion scheme exhibits the highest detection rates in all the SNR levels considered. Note however that, this method requires the first- and second-order statistics of all the local sensing outcomes to be available a priori [25]. The proposed linear message-passing algorithm closely achieves the optimal per-

formance when only the first- and second-order statistics of the one-hop neighbors are available at each node. When no such statistics are available, the proposed linear message-passing in (24) optimized by the blind offline adaptation scheme in [4] obtains a near-optimal detection performance.

In Fig. 6, we see that both of the sum-product and max-product algorithms exhibit different performance levels for different values of ζ and their performance is bounded from above by the detection rate of the proposed linear message-passing algorithm. Moreover, Fig. 6 indicates that the values we choose for ζ or c_0 heavily affect the performance of the message-passing algorithms concerned. For instance, the detection rates of the max-product algorithm and EGC drop below that of the local sensing method for $\zeta = 1$ and $c_0 = 1$. We have observed in our simulations that the performance of the max-product algorithm is improved when the learning factor ζ is increased from 0.01 to 0.1 and then is degraded severely by a further increase in ζ, indicating that an optimization of the message-passing iteration is needed. Optimal values for the parameters of the message-passing iteration are found by the proposed optimization method while the analysis provided in this paper justifies why the resulting detector outperforms the max-product algorithm. Note that, it is not clear how to determine the optimal value for c_0 when EGC is used. The proposed optimization framework solves this problem effectively since EGC is, in fact, a special case of linear data-fusion. We see in Fig. 6 that the optimal performance of EGC is achieved by the proposed method while no information is available a priori.

It is worth noting that, the existing works do not clarify how to best determine J_{kj}'s in the max-product operation. The analysis proposed in this paper clarifies that such an optimization is equivalent to designing an optimal linear fusion scheme. By comparing the detection rate of the max-product algorithm against that of the optimal linear fusion, the performance gain obtained by the proposed framework is visualized. Specifically, by comparing the curves labeled "mp0.1", "mp0.3", and "mp1.0" in Fig. 6 against the detection rates labeled "linPropB", "linProp", and "linOpt" we see the performance gain obtained.

VI. Conclusion

The analysis and numerical results presented in this paper demonstrate that in a distributed detection scenario and under practical assumptions the max-product algorithm works as a linear data-fusion scheme. Therefore, the knowledge already developed in the literature regarding distributed linear data-fusion can be used to better understand the behavior of the max-product algorithm.

References

[1] 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.

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

[3] M. Cetin, L. Chen, J. W. Fisher, A. T. Ihler, R. L. Moses, M. J. Wainwright, and A. S. Willsky, "Distributed fusion in sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 42–55, 2006.

[4] Y. Abdi and T. Ristaniemi, "Optimization of linearized belief propagation for distributed detection," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 959–973, 2020.

[5] M. Wainwright, T. Jaakkola, and A. Willsky, "Tree consistency and bounds on the performance of the max-product algorithm and its generalizations," *Statistics and computing*, vol. 14, no. 2, pp. 143–166, 2004.

[6] Y. Weiss and W. T. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 736–744, 2001.

[7] V. Kolmogorov and M. J. Wainwright, "On the optimality of tree-reweighted max-product message-passing," in *Uncertainty on Artificial Intelligence*. IEEE, 2005.

[8] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "Map estimation via agreement on trees: message-passing and linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3697–3717, 2005.

[9] Jianguang Zhao, F. Zarkeshvari, and A. H. Banihashemi, "On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (LDPC) codes," *IEEE Trans. Commun.*, vol. 53, no. 4, pp. 549–554, 2005.

[10] L. Chen, M. J. Wainwright, M. Cetin, and A. S. Willsky, "Multitarget-multisensor data association using the tree-reweighted max-product algorithm," in *Signal Processing, Sensor Fusion, and Target Recognition XII*, I. Kadar, Ed., vol. 5096, International Society for Optics and Photonics. SPIE, 2003, pp. 127 – 138.

[11] ——, "Data association based on optimization in graphical models with application to sensor networks," *Mathematical and computer modelling*, vol. 43, no. 9-10, pp. 1114–1135, 2006.

[12] A. Ahmad, D. Zennaro, E. Serpedin, and L. Vangelista, "A factor graph approach to clock offset estimation in wireless sensor networks," *IEEE Trans. Inf. Theory*, vol. 58, no. 7, pp. 4244–4260, 2012.

[13] D. Zennaro, A. Ahmad, L. Vangelista, E. Serpedin, H. Nounou, and M. Nounou, "Network-wide clock synchronization via message passing with exponentially distributed link delays," *IEEE Trans. Commun.*, vol. 61, no. 5, pp. 2012–2024, 2013.

[14] S. Han, Y. Huang, W. Meng, C. Li, N. Xu, and D. Chen, "Optimal power allocation for SCMA downlink systems based on maximum capacity," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1480–1489, 2019.

[15] 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.

[16] 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.

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

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

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

[20] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, 2011.

[21] B. Cao, S. Xia, J. Han, and Y. Li, "A distributed game methodology for crowdsensing in uncertain wireless scenario," *IEEE Trans. Mobile Comput.*, vol. 19, no. 1, pp. 15–28, 2020.

[22] W. Sun, J. Liu, and H. Zhang, "When smart wearables meet intelligent vehicles: Challenges and future directions," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 58–65, 2017.

[23] W. Sun, J. Liu, and Y. Yue, "AI-Enhanced offloading in edge computing: When machine learning meets industrial IoT," *IEEE Network*, vol. 33, no. 5, pp. 68–74, 2019.

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

[25] 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.

[26] 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.

[27] Y. Abdi and T. Ristaniemi, "Joint local quantization and linear cooperation in spectrum sensing for cognitive radio networks," *IEEE Trans. Signal Process.*, vol. 62, no. 17, pp. 4349–4362, Sept 2014.

[28] ——, "Random interruptions in cooperation for spectrum sensing in cognitive radio networks," *IEEE Trans. Commun.*, vol. 65, no. 1, pp. 49–65, 2017.
[29] H. Wymeersch, F. Penna, and V. Savic, "Uniformly reweighted belief propagation for estimation and detection in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 4, pp. 1587–1595, 2012.
[30] 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.
[31] J. Ma and G. Y. L. B. H. Juang, "Signal processing in cognitive radio," *Proceedings of the IEEE*, vol. 97, no. 5, pp. 805–823, 2009.
[32] 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.
[33] S. M. Kay, *Fundamentals of statistical signal processing*. Prentice Hall PTR, 1993.
[34] J. M. Mooij and H. J. Kappen, "Sufficient conditions for convergence of the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4422–4437, Dec 2007.
[35] 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.
[36] 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.

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, working on distributed detection systems. From 2013 to 2016 he served as a member of Working Group 1900.1 in the IEEE Dynamic Spectrum Access Networks Standards Committee. During spring and summer 2018, he was 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. He has recently joined Nokia Solutions and Networks, Espoo, where he is currently working on 5G technology. His current research interests include advanced signal processing and wireless communications.

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.