Distributed Decoding in a Cellular Multiple-Access Channel

Emre Aktas, Jamie Evans, and Stephen Hanly

Abstract— **This paper considers the problem of joint detection in the uplink of cellular multiaccess networks with base-station cooperation. Distributed multiuser detection algorithms with local message passing among neighbor base stations are proposed and compared in terms of computational complexity required in the base stations, the amount of serial communications among them, error rate performance, and convergence speed. The algorithms based on the belief propagation algorithm result in complexity and delay per base station which do not grow as the network size increases. In addition, it is observed that these algorithms have near single-user error rate performance for the fading channels considered. Thus it is illustrated that using the belief propagation algorithm, it is possible to have full frequency re-use and achieve near-optimal performance with moderate computational complexity and a limited amount of message passing between base stations of adjacent cells.**

Index Terms— **Cellular communication, distributed detection.**

I. Introduction

In [1], Wyner considered the uplink of a cellular network, and proposed the concept of a joint processing global receiver which has access to all the received signals, and optimally decodes all the transmitted codewords in the entire network. In this model, each base station acts as an antenna of a global receiver which performs optimum multiuser decoding. It was shown that a cellular network with full frequency re-use and such a joint-processing receiver significantly outperforms a network with individual processing where different frequency bands are assigned to neighboring cells to avoid co-channel interference [1]–[3].

Even though the joint processing global receiver is desirable, its implementation poses several challenges from the practical point of view. Firstly, all the received signals are required for joint processing. Since the cells in the cellular system are geographically separated, possibly over a large area, collecting all the received signals at one location for joint processing might not be feasible. Secondly, even if all the received data is collected at a central processor, the fact that the complexity of the general multiuser detection problem grows exponentially in the number of users [4] suggests that multiuser decoding of all users may be intractable as the size of the network grows.

Despite the aforementioned challenges, the fact that a mobile only causes significant interference to *nearby* base stations suggests the possibility of a practical solution. In the present paper, we wish to exploit the local interactions between cells, both in terms of the interference between cells, but also in terms of the communication available between adjacent base stations, in order to obtain global demodulation algorithms that do not grow in complexity as the size of the array grows large.

The main idea is that the global problem of demodulating all the symbols in the network can be distributed across a network of interacting base stations, each of which performs local computations, and then passes the results on to their neighbors for further processing. The theoretical foundation for this approach is provided by Pearl's work on statistical inference using Bayesian networks [5].

Pearl's belief propagation (BP) algorithm is a local message passing algorithm based on a Bayesian network graph representation of a probabilistic model. The goal of the BP algorithm is to calculate the *a posteriori* probability (APP) of each random variable given the observation of some of the random variables - a common statistical inference problem. The algorithm exploits the local dependency properties of the variables in order to reduce the complexity of the APP calculations, and for graphs with no loops, it converges to the desired result in a finite number of steps.

We link the network-wide problem of demodulating the users' data symbols, conditional on the received signals at the base stations, to the computation of the APP for a corresponding graph. It turns out that our models are not loop-free, and in this sense the algorithms we propose do not compute exact APPs. However, it appears that the exact calculation of the APPs is exponential in the size of the array [6], and, as observed in many other contexts [7]–[9], the existence of loops in the graph need not prevent the algorithms we propose from generating very good approximate solutions to the problem. We provide numerical results, and lower bounds, to demonstrate the accuracy of our algorithms.

We consider the 2-D planar array cellular model with discrete alphabet signaling in this paper. Before we study the general problem of global demodulation for this case, we consider first the much simpler one-dimensional version of the problem, for which the BP algorithm provides an exact, and well known solution to the problem, namely the

Manuscript received July 13, 2006; revised April 5, 2007 and April 25, 2007; accepted May 3, 2007. The associate editor coordinating the review of this paper and approving it for publication was S. Zhou. This paper was presented in part at the IEEE International Symposium on Information Theory, Chicago, IL, USA, June 2004, and IEEE Internatinal Conference on Communications, Istanbul, Turkey, June 2006. This work was supported by Australian Research Council Discovery Grant DP0209658.

E. Aktas was with the ARC Special Research Centre for Ultra-Broadband Information Networks (CUBIN), Department of Electrical and Electronic Engineering, University of Melbourne, Parkville VIC 3010, Australia. He is now with the Department of Electrical and Electronics Engineering, Hacettepe University, Beytepe Ankara, 06800 Turkey (e-mail: aktas@ee.hacettepe.edu.tr).

J. Evans and S. Hanly are with the ARC Special Research Centre for Ultra-Broadband Information Networks (CUBIN), Department of Electrical and Electronic Engineering, University of Melbourne, Parkville VIC 3010, Australia (e-mail: {jse, hanly}@ee.unimelb.edu.au).

Digital Object Identifier 10.1109/TWC.2008.060469.

BCJR algorithm [10]. As an extension to the application of BCJR algorithm in the one-dimensional case, we propose to isolate columns and rows of the rectangular grid and perform the BCJR algorithm on them alternatively, as a soft iterative algorithm with two modules. This method, although implementable in a distributed fashion, is serial in nature and still not completely decentralized. Thus, we also propose two graphical models upon which we can apply the BP algorithm [5] directly, in such a way that computations at the cell sites can proceed in parallel, and are continuously updated as new information arrives from the adjacent cells. We compare the implementation of the three resulting algorithms in terms of error rate performance, number of message passing steps, and complexity required at the base stations.

All of the methods we propose have the following desired properties. They can be implemented with local message passing between base stations of neighboring cells. The numerical complexity required for a base station is constant regardless of the network size, and the performance is near the single-user lower bound for fading channels at high signal to noise ratios.

The rest of the paper is organized as follows. In Section II the rectangular cellular model is presented. The BCJR decoding in the linear array model [11], and the iterative extension of it for the rectangular model are presented in Section III. The graphical models and the BP algorithm based on the models are proposed Section IV. The numerical simulations are given in Section V, followed by concluding remarks in Section VI.

A. Related work

Distributed decoding in cellular networks has been considered previously for a simplified linear model [11], [12]. The linear cellular array model assumes a hypothetical scenario in which the cells are placed on a line, each having two neighbor cells. For this model, it was shown that the maximum *a posteriori* (MAP) decoding of each user is equivalent to the inference problem of estimating the states which form a Markov chain, similar to the detection of symbols in a single user inter-symbol interference channel with a fixed channel length of three. The BCJR algorithm [10] for this Markov chain results in a distributed algorithm with local message passing [11]. Distributed implementation of the linear minimum mean squared error receiver, which is optimal for Gaussian sources but sub-optimal for discrete alphabet sources, was developed in [12]. The methods of [11], [12] are not directly applicable to the 2-D model where the cells are located on a plane.

Recent research has considered distributed global demodulation in 2-D cellular channels [13]–[15]. In [14], a reduced complexity maximum-likelihood (ML) decoder is developed, which results in the exact ML solution. The algorithm is motivated as an extension of the Viterbi algorithm which exploits the limited interference structure. Although the general large 2-D cellular structure is not treated, it seems that the algorithm, if applied to that structure, would result in increasing complexity per symbol with growing network size. In fact, since the number of loops grows with the network size, we would expect that the exact ML solution has complexity per symbol that grows unbounded in the size of the network.

In [15], Shental *et al.* have considered the application of a generalized BP algorithm, which is defined on Markov networks, to the 2-D inter-cell interference (ICI) problem. It is reported in [15] that the BP has very poor convergence when there are a large number of short cycles, and a generalized version is provided in which clusters of nodes are used to overcome this problem. This seems to contradict with the results in the present paper, since we observe near-optimal performance without clustering. This is perhaps because of the fixed model used for channel coefficients in [15] as opposed to the random coefficients considered in the present paper. In addition, in [15] BP over pairwise Markov random field is considered, while here we utilize directed Bayesian network graphs, which may result in two different message-passing schemes.

Marrow and Wolf proposed the idea of using 1-D MAP algorithms on rows and columns iteratively for the 2-D equalization problem [16]. This is very similar to the method we use in our iterative BCJR implementation in Section III-B. However, since the actual signal structure does not allow a Markov chain representation, application of BCJR algorithm along the rows or columns is an approximation, resulting in unaccounted interference, and the way this unaccounted interference is treated is different in [16] and the method we use here.

Demodulation of 2-D cellular networks in the presence of ICI is closely related to the problem of 2-D equalization in page-oriented optical memories [17]–[20]. The use of BP for 2-D equalization has been considered in [19], [20]. The graphical representations in Sections IV-A and IV-B of the current paper had been suggested in [19] and Chapter 5 of [20], respectively. Thus the resulting algorithms we consider in this paper are structurally the same as the corresponding algorithms in [19], [20]. We emphasize that the application of BP to the problem of distributed decoding in cellular networks is novel.

II. Rectangular Cellular Array Model

We consider a cellular network model where the cells are located on a rectangular grid, and each cell has four neighbors (extending results to the more realistic hexagonal case where each cell has six neighbors is conceptually straightforward, and we consider the rectangular case for its relative simplicity). We assume flat fading, where the baseband channel is represented by a complex coefficient. Within a cell, multiple users' signals are assumed orthogonal, hence there is no intra-cell interference. The received signal at the base station of any cell, in any channel, is the superposition of the signal from its own user, and the signals the four adjacent cell co-channel users. Each base station is primarily interested in obtaining the APP of the symbol of its user, given the signal observed by the network of all base stations. We exclude coding in order to focus on dealing with the ICI at a symbol level.

In the following, uppercase letters denote random variables, and lowercase of the same letter denotes a realization of the random variable. For cell (i, j) in the rectangular grid, let $B_{i,j}$ denote the transmitted symbol drawn from a constellation of size M, $Y_{i,j}$ the received signal, $h_{i,j}(m, n)$ the channel to base state in cell (i, j) from the user in cell $(i + m, j + n)$, and $N_{i,j}$ additive Gaussian noise. We assume that the channel

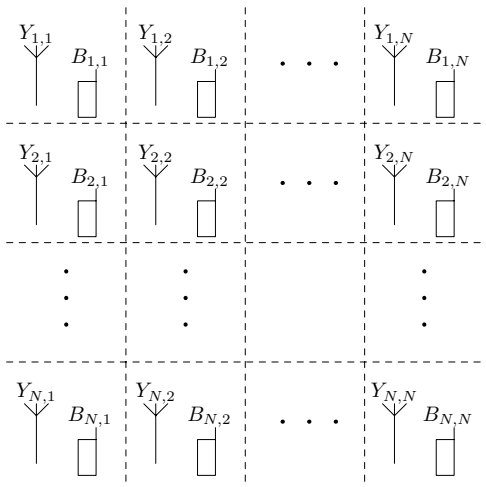

Fig. 1. Rectangular cellular array.

coefficients $h_{i,j}(m,n)$ and the noise variance σ^2 are known at the base station of cell (i,j). We consider an $N \times N$ rectangular network as in Fig. 1. The relationship between the observation $Y_{i,j}$ and the transmitted bit $B_{i,j}$ is expressed as:

$$Y_{i,j} = h_{i,j}(-1,0)B_{i-1,j} + h_{i,j}(0,0)B_{i,j} + h_{i,j}(1,0)B_{i+1,j} + h_{i,j}(0,-1)B_{i,j-1} + h_{i,j}(0,1)B_{i,j+1} + N_{i,j}. \quad (1)$$

For the cells at the edges of the rectangular network, we add dummy symbols $B_{0,j}$, $B_{N+1,j}$, $B_{i,0}$, and $B_{i,N+1}$ for $i = 1,\ldots,N$, and set the corresponding $h_{i,j}(m,n)$'s to zero.

Let Y denote the set of all observations $Y = \{Y_{1,1},\ldots,Y_{N,1},\ldots,Y_{1,N},\ldots,Y_{N,N}\}$. The goal is to calculate the APP of any transmitted symbol given $Y = y$, that is, $P(b_{i,j}|y)$. Brute force calculation of the APP has complexity that grows exponentially with N^2, which is clearly intractable. Although, by clustering along the width of the network and applying the BCJR algorithm, it is possible to obtain exact inference with complexity that grows exponentially with $3N$, this is still intractable as the network size grows.

III. The BCJR Algorithm

A. The BCJR algorithm on a one-dimensional array

In order to emphasize that it may be possible to obtain an optimum global receiver using local processing and message passing, we now consider the special case of a one-dimensional array of cells, and present the method of [11] for this case, which is the application of the BCJR algorithm to this problem. Consider only one column of cells in the rectangular array. Omitting the column index in (1), the signal model reduces to

$$Y_i = h_i(-1)B_{i-1} + h_i(0)B_i + h_i(1)B_{i+1} + N_i.$$

This model can be represented as a hidden Markov chain as follows:

$$Y_i = m_i(S_i) + N_i, \quad (2)$$

where S_i is the "state" for cell i, $S_i = (B_{i-1}, B_i, B_{i+1})$, and where m_i is a deterministic function of the state since the channel coefficients are known: $m_i(s_i) = h_i(-1)b_{i-1} + h_i(0)b_i + h_i(1)b_{i+1}$. The states form a Markov chain because $p(s_i|s_{i-1},\ldots,s_1) = p(s_i|s_{i-1})$.

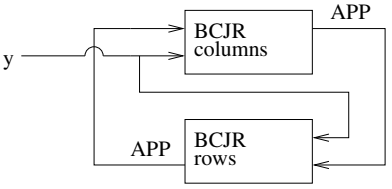

Fig. 2. Iterative implementation of BCJR algorithm along the columns and rows of the rectangular array.

The BCJR algorithm is an iterative method for models of this type. It is used to obtain the APP $p(s_i|y)$ and can be directly applied here. The desired APP $p(b_i|y)$ can be obtained from the marginalization of $p(s_i|y)$. It can be shown that [10], [11] the following recursion

$$\alpha(s_1) = p(y_1|s_1)p(s_1) \quad (3)$$

$$\alpha(s_i) = p(y_i|s_i)\sum_{s_{i-1}} p(s_i|s_{i-1})\alpha(s_{i-1}) \quad \text{for } i = 2,3,\ldots,N \quad (4)$$

and

$$\beta(s_N) = 1 \quad (5)$$

$$\beta(s_{i-1}) = \sum_{s_i} p(y_i|s_i)p(s_i|s_{i-1})\beta(s_i) \quad \text{for } i = N-1, N-2,\ldots,1 \quad (6)$$

results in

$$p(s_i|y) = \alpha \cdot \alpha(s_i)\beta(s_i) \quad (7)$$

where α is a normalizing constant to obtain a probability density function. Note that it should not be confused with $\alpha(s_i)$.

This recursion can be implemented locally where the base station of cell i performs the calculations in (4), (6), and (7), receives local $\alpha(s_{i-1})$ message from cell $i-1$, and transmits $\beta(s_{i-1})$ message to cell $i-1$. Notice that the implementation is serial, *i.e.* cell i has to wait to receive the $\alpha(s_{i-1})$ message from cell $i-1$ before it can transmit the $\alpha(s_i)$ message to cell $i+1$, and similarly for the $\beta(s_{i+1})$ and $\beta(s_i)$ messages going in the opposite direction.

B. BCJR algorithm in the rectangular array

Encouraged by the elegance of the implementation of the BCJR algorithm for the one-dimensional array, we consider using this algorithm along the columns and rows of a rectangular array in an iterative manner. The APP outputs of the BCJR along one direction will be used as *a priori* probabilities for the BCJR along the other direction. Thus the global decoder is built as an iterative decoder where the two modules of the iterative decoder are the BCJR in each direction (Fig. 2). In order to describe this decoder, suppose that the algorithm is at the stage where it has just completed the BCJR along the rows, and it will next start the BCJR along the columns. The resulting APPs from the row-BCJR will be passed to the column-BCJR, to be used as *a priori* probabilities. For base station of cell (i,j), the required priors are $p(b_{i,j})$, $p(b_{i+1,j})$,

$p(b_{i,j-1})$, and $p(b_{i,j+1})$. Consider the operation on column j of cells only, since in each column identical operations will be performed, and let us rewrite the BCJR equations of (3)-(6) with the column index for clarity:

$$\alpha(s_{1,j}) = p(y_{1,j}|s_{1,j})p(s_{1,j}) \tag{8}$$

$$\alpha(s_{i,j}) = p(y_{i,j}|s_{i,j}) \sum_{s_{i-1,j}} p(s_{i,j}|s_{i-1,j})\alpha(s_{i-1,j}) \quad \text{for } i = 2, 3, \ldots, N \tag{9}$$

$$\beta(s_{N,j}) = 1 \tag{10}$$

$$\beta(s_{i-1,j}) = \sum_{s_{i,j}} p(y_{i,j}|s_{i,j})p(s_{i,j}|s_{i-1,j})\beta(s_{i,j}) \quad \text{for } i = N-1, N-2, \ldots, 1 \tag{11}$$

The state for cell (i, j) is $s_{i,j} = (b_{i-1,j}, b_{i,j}, b_{i+1,j})$ for the BCJR along the columns. The *a priori* probabilities $p(b_{1,j})$, $p(b_{2,j})$ are used to calculate the initial state probability $p(s_{1,j})$ in (8) and the *a priori* probability $p(b_{i+1,j})$ is used to calculate the transition probability $p(s_{i,j}|s_{i-1,j})$ in (9) and (11).

Notice that the received signal is not only a function of three consecutive transmitted symbols on the column but the symbols from the two neighboring cells on the same row as well. Thus the likelihood of state $S_{i,j}$ is calculated by averaging out the row interference:

$$p(y_{i,j}|s_{i,j}) = \sum_{b_{i,j-1}, b_{i,j+1}} p(y_{i,j}|s_{i,j}, b_{i,j-1}, b_{i,j+1}) \times p(b_{i,j-1})p(b_{i,j+1}). \tag{12}$$

The priors $p(b_{i,j-1})$ and $p(b_{i,j+1})$ are used to calculate $p(y_{i,j}|s_{i,j})$ in (12). The likelihood $p(y_{i,j}|s_{i,j})$ is used in (9) and (11).

The averaging in (12) can be interpreted as a way of handling the row interference unaccounted for by the BCJR along the column. The treatment of the row ICI in this manner is distinct from the way it is treated in [16]. In [16], uniform distributions of $B_{i,j-1}$, $B_{i,j+1}$ are used as opposed to our approach where the APP from the previous iteration is used.

When the BCJR along the columns are completed, the base station (i, j) will have the APP of $S_{i,j}$, and thus of $B_{i-1,j}$, $B_{i,j}$, and $B_{i+1,j}$ via marginalization. Next, the BCJR along the rows will be performed. For the row-BCJR, the state for cell (i, j) is $S_{i,j} = (B_{i,j-1}, B_{i,j}, B_{i,j+1})$. Following similar derivation, it can be seen that the base station of cell (i, j) will require the following *a priori* distributions for row-BCJR: $p(b_{i,j})$, $p(b_{i,j+1})$, $p(b_{i-1,j})$, and $p(b_{i+1,j})$. It already has $p(b_{i,j})$, $p(b_{i-1,j})$, and $p(b_{i+1,j})$ upon completion of the column-BCJR, and the distribution $p(b_{i,j+1})$ should be passed from base station of cell $(i, j + 1)$. When the row-BCJR is completed, the base station (i, j) will have the APPs of $B_{i,j-1}$, $B_{i,j}$, and $B_{i,j+1}$. Thus, only the distribution $p(b_{i+1,j})$ should be passed from base station $(i+1, j)$ to base station (i, j) for the base station (i, j) to have all the required prior probabilities required by the column-BCJR that will follow.

IV. Belief Propagation Algorithm

We next discuss the distributed decoding in the rectangular model using the BP algorithm of [5]. Our notation is based on [5], where the algorithm is defined via messages among the nodes of a Bayesian network graph representation of the probabilistic model. Bayesian networks are directed acyclic graphs where each node represents a random variable in the system, and the arcs signify a direct probabilistic dependency between the linked variables. If there is an arc from node P to node C in the graph, node P is named the *parent node*, and C is the *child node*. An arc from a parent node P to child node C represents the existence of a direct causal influence between P and C, expressed by the conditional probability mass function $p(c|p)$ of C given P. The message from node C to node P is denoted by $\lambda_C(p)$ and the message from node P to node C is denoted by $\pi_C(p)$. The message $\pi_C(p)$ is the probability distribution of P conditioned on all messages received by node P, except the message from node C. The message $\lambda_C(p)$ is the likelihood distribution of P based on all messages received by node C, except the message from node P. The messages are sent for every p in the sample space of P, so they are vectors when P is discrete.

Note that there can be more than one Bayesian network representation of a probabilistic model. We will consider two different graphical representations, which will lead to two different algorithms. As mentioned in Section I, the BP algorithm eventually results in the APP only in the absence of loops in the graph, but can be implemented as an iterative algorithm to obtain approximate results for loopy graphs, as will be the case for our two graphical representations. The first representation is one where the transmitted symbols from neighboring cells are clustered to form a single node in the associated graph. The second graphical representation is a decomposed model in which transmitted symbols and received signals are represented by separate nodes. See Fig. 3 and Fig. 5, respectively, for the two graphical representations.

For a complete characterization of the use of the BP algorithm for an inference problem, one needs to specify not only the graphical model, but also the initialization and the order of activation of the nodes. This we now do for the two different graphical models.

A. Clustered graphical representation

For each cell (i, j), we define a state $S_{i,j}$ such that it d-separates (as defined in [5]) the observation $Y_{i,j}$ from the rest of the graph. The state consists of the transmitted symbols:

$$S_{i,j} = \{B_{i,j}, B_{i-1,j}, B_{i+1,j}, B_{i,j-1}, B_{i,j+1}\}. \tag{13}$$

The clustered nodes then form the loopy graph in Fig. 3. Since each $Y_{i,j}$ node has only a single parent node and no child nodes, it will send a message to its parent node only once. That message is

$$\lambda_{Y_{i,j}}(s_{i,j}) = \alpha p(y_{i,j}|s_{i,j}) = \alpha \mathcal{N}(y_{i,j}, m_{i,j}(s_{i,j}), \sigma^2), \tag{14}$$

where $\mathcal{N}(y, m_Y, \sigma_Y^2)$ denotes the probability density function of complex Gaussian random variable Y with mean m_Y and variance σ_Y^2. Equation (14) is due to the fact that $Y_{i,j} = m_{i,j}(S_{i,j}) + N_{i,j}$, where

$$m_{i,j}(s_{i,j}) = h_{i,j}(-1, 0)b_{i-1,j} + h_{i,j}(0, 0)b_{i,j} + h_{i,j}(1, 0)b_{i+1,j} + h_{i,j}(0, -1)b_{i,j-1} + h_{i,j}(0, 1)b_{i,j+1}. \tag{15}$$

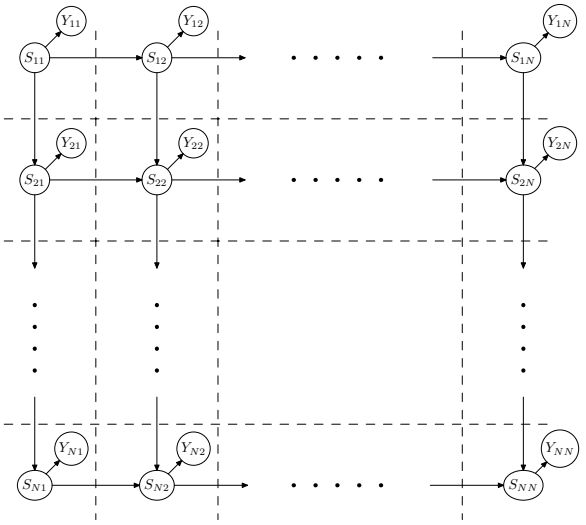

Fig. 3. Clustered graph representation of the rectangular array model.

For each node $S_{i,j}$, the set of child nodes are $\{Y_{i,j}, S_{i+1,j}, S_{i,j+1}\}$. Excluding the observed node $Y_{i,j}$, we define the set

$$C_{i,j} = \{C^1_{i,j}, C^2_{i,j}\}, \tag{16}$$

where $C^1_{i,j}$ and $C^2_{i,j}$ denote the two child nodes $S_{i+1,j}$ and $S_{i,j+1}$, respectively. The set of parent nodes is denoted in a similar fashion: $P_{i,j} = \{P^1_{i,j}, P^2_{i,j}\} = \{S_{i-1,j}, S_{i,j-1}\}$.

In order to characterize the BP algorithm for this graph, for each state node $S_{i,j}$, we next express the messages sent to child nodes, messages sent to parent nodes, and how these messages are combined. The message that node $S_{i,j}$ sends to child node $C^k_{i,j}$ for $k = 1, 2$ is

$$\pi_{C^k_{i,j}}(s_{i,j}) = \alpha \pi(s_{i,j}) \lambda_{Y_{i,j}}(s_{i,j}) \lambda_{C^l_{i,j}}(s_{i,j}), \tag{17}$$

where $l = \{1,2\}\backslash k$, $\lambda_{C^l_{i,j}}(s_{i,j})$ is the message from child node $C^l_{i,j}$ to node $S_{i,j}$, and $\pi(s_{i,j})$ is the combination of messages from parent nodes of $S_{i,j}$ to node $S_{i,j}$:

$$\pi(s_{i,j}) = \sum_{p_{i,j}} p(s_{i,j}|p_{i,j}) \prod_{k=1}^{2} \pi_{S_{i,j}}(p^k_{i,j}). \tag{18}$$

Note that $p_{i,j}$ in the summation satisfies $p_{i,j} = (p^1_{i,j}, p^2_{i,j})$ and $\pi_{S_{i,j}}(p^k_{i,j})$ is the message from parent node $P^k_{i,j}$ to $S_{i,j}$. It is unnecessary in (18) to sum over $p_{i,j}$ for which $p(s_{i,j}|p_{i,j}) = 0$. Note that for $p(s_{i,j}|p_{i,j}) > 0$, the $b_{l,m}$'s that are both in $s_{i,j}$ and in $p_{i,j}$ must be the same. For example, consider Fig. 4, where $S_{i,j} = S_{33}$, and $P_{i,j} = \{S_{23}, S_{32}\}$. In order to conform with s_{33}, $\{b_{23}, b_{33}\}$ in s_{23} and $\{b_{32}, b_{33}\}$ in s_{32} should be the same as the $\{b_{23}, b_{32}, b_{33}\}$ in s_{33}. We denote $p_{i,j} : s_{i,j}$ to be set of all values of $p_{i,j}$ which conform with $s_{i,j}$. Utilizing the fact that the transmitted symbols have uniform *a priori* probability, we can therefore write (18) as

$$\pi(s_{i,j}) = \alpha \sum_{p_{i,j}:s_{i,j}} \prod_{k=1}^{2} \pi_{S_{i,j}}(p^k_{i,j}), \tag{19}$$

$$\simeq \alpha \sum_{\substack{p^1_{i,j}:s_{i,j} \\ p^2_{i,j}:s_{i,j}}} \prod_{k=1}^{2} \pi_{S_{i,j}}(p^k_{i,j}), \tag{20}$$

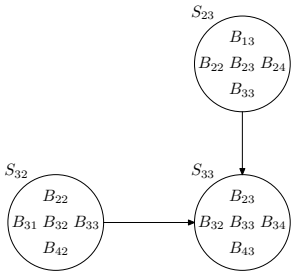

Fig. 4. Example state node and its parent nodes in the clustered graph.

$$= \alpha \prod_{k=1}^{2} \bar{\pi}_{P^k_{i,j}}(s_{i,j}), \tag{21}$$

where $\bar{\pi}_{P^k_{i,j}}(s_{i,j})$ can be considered as a pre-processed message from parent node $P^k_{i,j}$ to node $S_{i,j}$:

$$\bar{\pi}_{P^k_{i,j}}(s_{i,j}) = \sum_{p^k_{i,j}:s_{i,j}} \pi_{S_{i,j}}(p^k_{i,j}). \tag{22}$$

Note that (19) and (20) are in general not equal because in (19) the summation is over $p^1_{i,j}$ and $p^2_{i,j}$ that conform with each other as well as with $s_{i,j}$, whereas in (20) we also include $p^1_{i,j}$ and $p^2_{i,j}$ that does not conform with each other. Specifically, $b_{i-1,j-1}$s in $p^1_{i,j}$ and $p^2_{i,j}$ should be the same for the summation in (19), but they may be different in (20). The additional terms in (20) lead to the simplification in (21), which results in a considerable saving in complexity. In numerical simulations, we observe near optimal performance so (20) is a good approximation.

The messages to be sent from node $S_{i,j}$ to the parent nodes $P^k_{i,j}$ for $k = 1, 2$ are:

$$\lambda_{S_{i,j}}(p^k_{i,j}) = \alpha \sum_{s_{i,j}} \lambda(s_{i,j}) \sum_{p^l_{i,j}} p(s_{i,j}|p^k_{i,j}, p^l_{i,j}) \pi_{S_{i,j}}(p^l_{i,j}) \tag{23}$$

$$= \alpha \sum_{s_{i,j}:p^k_{i,j}} \lambda(s_{i,j}) \sum_{\substack{p^l_{i,j}:s_{i,j} \\ p^l_{i,j}:p^k_{i,j}}} \pi_{S_{i,j}}(p^l_{i,j}) \tag{24}$$

$$\simeq \alpha \sum_{s_{i,j}:p^k_{i,j}} \lambda(s_{i,j}) \bar{\pi}_{P^l_{i,j}}(s_{i,j}), \tag{25}$$

where again $l = \{1,2\}\backslash k$, $\lambda(s_{i,j})$ is the combination of messages from child nodes at node $S_{i,j}$, and (24) is due to the fact that $p(s_{i,j}|p^k_{i,j}, p^l_{i,j})$ is identical for all $s_{i,j} : p^k_{i,j}$ and $p^l_{i,j} : s_{i,j}$. The approximation is (25) is similar to (20) in that we have extra terms in the summation with non-conforming $p^k_{i,j}$ and $p^l_{i,j}$. The messages from child nodes are combined at node $S_{i,j}$ as:

$$\lambda(s_{i,j}) = \lambda_{Y_{i,j}}(s_{i,j}) \prod_{k=1}^{2} \lambda_{C^k_{i,j}}(s_{i,j}). \tag{26}$$

Finally, combining everything, the belief of the state $S_{i,j}$ is

$$\text{BEL}(s_{i,j}) = \alpha \pi(s_{i,j}) \lambda(s_{i,j}), \tag{27}$$

from the marginalization of which the belief for $B_{i,j}$ can be obtained.

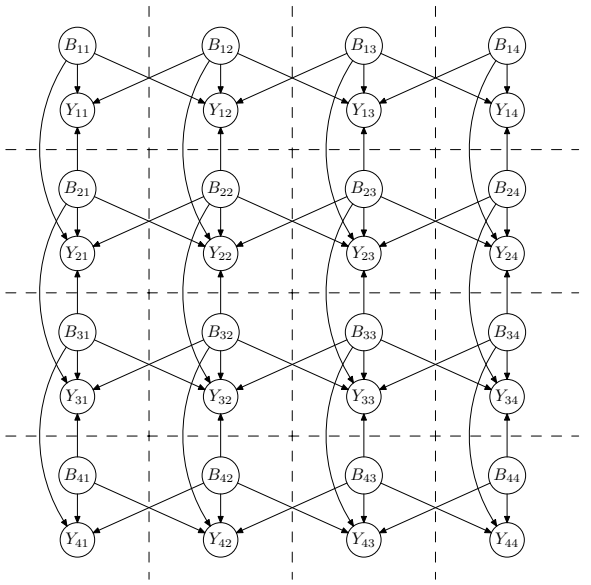

Fig. 5. Decomposed graph representation of the rectangular array model.

We consider the ordering of activation of nodes where, at each step, each state node sends and receives messages from all neighbor state nodes in parallel. For each node $S_{i,j}$ the incoming messages $\bar{\pi}_{P^k_{i,j}}(s_{i,j})$ and $\lambda_{C^k_{i,j}}(s_{i,j})$ are initialized to uniform distributions, and $\pi(s_{i,j})$ and $\lambda(s_{i,j})$ are calculated from this initialization, which constitutes the first iteration. Afterward, for each iteration

1) $\bar{\pi}_{S_{i,j}}(c^k_{i,j})$ and $\lambda_{S_{i,j}}(p^k_{i,j})$ are calculated and transmitted in parallel for $k = 1, 2$ using computations in (17), (22), and (25).
2) $\pi(s_{i,j})$ and $\lambda(s_{i,j})$ are calculated using computations in (21) and (26).

After a predetermined number of iterations, the belief of the state is computed using (27).

In the cellular system, this can be implemented in a distributed fashion where the calculations (17)-(26) done by the state $S_{i,j}$ are performed by the base station of cell (i, j), and the messages among state nodes are implemented as communication among base stations of neighboring cells. For each iteration, the base station of cell (i, j) transmits $\bar{\pi}_{S_{i,j}}(c^k_{i,j})$ and $\lambda_{S_{i,j}}(p^k_{i,j})$ to its four neighbors, while it is receiving the messages $\bar{\pi}_{P^k_{i,j}}(s_{i,j})$ and $\lambda_{C^k_{i,j}}(s_{i,j})$ from its four neighbors.

B. Decomposed graphical representation

In this graphical representation, each transmitted symbol $B_{i,j}$ and received signal $Y_{i,j}$ at the base station are represented by separate nodes, see Fig. 5. Note that, for cell (i, j), the received signal $Y_{i,j}$ is d-separated from the rest of the random variables by the set of transmitted symbols from the base stations in the following set:

$$\begin{aligned} v_{i,j} &= \{(i,j),(i-1,j),(i+1,j),(i,j-1),(i,j+1)\} \\ &= \{v_{i,j}(0), v_{i,j}(1), v_{i,j}(2), v_{i,j}(3), v_{i,j}(4)\}. \end{aligned} \tag{28}$$

These are the cells immediately adjacent to cell (i, j), together with cell (i, j) itself. Denote the corresponding set of transmitted symbols by $B_{v_{i,j}}$.

The belief graph can be constructed whereby the observed $Y_{i,j}$ nodes are the child nodes with the set of parents $B_{v_{i,j}}$, and the transmitted symbol nodes $B_{i,j}$ are the parent nodes with the set of children $Y_{v_{i,j}}$. Since $B_{i,j}$'s are independent in the absence of the observations $Y_{i,j}$, they are not directly connected to each other in the graph. A complete graph can be obtained this way, and an example is depicted for the case of a 4×4 rectangular network in Fig. 5.

We next employ the BP algorithm for this graph. With the assumption that $B_{i,j}$'s have uniform *a priori* distribution, and the fact that $B_{i,j}$'s are nodes with no parent nodes, the belief for $B_{i,j}$ becomes

$$\text{BEL}(b_{i,j}) = \alpha \lambda(b_{i,j}), \tag{29}$$

where the overall likelihood $\lambda(b_{i,j})$ is the product of the likelihood messages from all child nodes of $B_{i,j}$:

$$\lambda(b_{i,j}) = \prod_{k=0}^{4} \lambda_{Y_{v_{i,j}(k)}}(b_{i,j}). \tag{30}$$

The message from the observed node $Y_{i,j}$ to its parent node $B_{v_{i,j}(k)}$ for $k = 0, 1, \ldots, 4$ is

$$\lambda_{Y_{i,j}}(b_{v_{i,j}(k)}) = \alpha \sum_{b_{v_{i,j}} : b_{v_{i,j}(k)}} p(y_{i,j}|b_{v_{i,j}}) \prod_{l=0,\, l\neq k}^{4} \pi_{Y_{i,j}}(b_{v_{i,j}(l)}). \tag{31}$$

Finally, the message sent from node $B_{i,j}$ to its child node $Y_{v_{i,j}}(k)$ for $k = 0, 1, \ldots, 4$ is

$$\pi_{Y_{v_{i,j}(k)}}(b_{i,j}) = \alpha \prod_{l=0,\, l\neq k}^{4} \lambda_{Y_{v_{i,j}(l)}}(b_{i,j}), \tag{32}$$

which is the product of all messages sent to node $B_{i,j}$, except from node $Y_{v_{i,j}(k)}$.

In order to complete the definition of the algorithm, one needs to choose the initialization and the order of activation of the nodes in the graph. We consider initializing all probability messages $\pi_{Y_{v_{i,j}(k)}}(b_{i,j})$ to uniform distributions, and activating the nodes in the following order:

1) Activate all $Y_{i,j}$ nodes in parallel: Calculate and pass the likelihood messages $\lambda_{Y_{i,j}}(b_{v_{i,j}(k)})$ to all parent nodes, using computations in (31).
2) Activate all $B_{i,j}$ nodes in parallel: Calculate and pass the probability messages $\pi_{Y_{v_{i,j}(k)}}(b_{i,j})$ to all the child nodes, using computations in (32).

These two steps are then repeated in the next iteration. This algorithm can be implemented in the cellular system in a distributed fashion, where the base station of cell (i, j) performs the calculation in (31) and propagates $\lambda_{Y_{i,j}}(b_{v_{i,j}(k)})$ for $k = 1, \ldots, 4$ to its four neighboring base stations. At the same instant, it is receiving $\lambda_{Y_{v_{i,j}(k)}}(b_{i,j})$ for $k = 1, \ldots, 4$ from its four neighboring base stations. Next, the calculation (32) is performed and $\pi_{Y_{v_{i,j}(k)}}(b_{i,j})$ messages are propagated to neighboring base stations, while receiving $\pi_{Y_{i,j}}(b_{v_{i,j}(k)})$ messages from them. After a sufficient number of iterations, to obtain the belief of $B_{i,j}$, the base station (i, j) performs (30)

TABLE I
COMPARISON OF ALGORITHMS

	BCJR	BP-clustered	BP-decomposed
complexity per base station per iteration	$4M^5 + 6M^4 + 4M^3$	$8M^5 + M^3$	$5M^5 + 5M^4 + 6M^2 + 9M$
communication per base station per iteration	$4M^3 + M$	$4M^2$	$8M$
serial communication per iteration	$2M + 2(N-1)M^3$	M^2	$2M$

and (29). Notice that activating $B_{i,j}$ nodes is not necessary for the final iteration.

The algorithms developed are for the scenario where multiple users within a cell use orthogonal signaling. In the scenario where non-orthogonal signaling is used, signals from more than one mobile users from a cell will contribute to the received signal at the base stations of that cell and the neighboring cells. For this scenario, the algorithms developed in this section can be extended as follows. For the decomposed graph, there are more than one B nodes for each cell, and they are the parent nodes of Y nodes of their own cell and the neighboring cells. For the clustered graph, the clustered state node for each state encompasses the transmitted bits from all the active users in the neighborhood of the cell. Using these graphs, the corresponding algorithms can be developed similarly to the algorithms in this section.

V. NUMERICAL RESULTS

Simulations are conducted for a fading channel where each $H_{i,j}(m,n)$ is complex Gaussian distributed with zero mean and variance 1 if $(m,n) = (0,0)$ and variance α^2 otherwise. Thus α^2 represents the average power of the ICI from each neighbor. The additive noise $N_{i,j}$ is complex Gaussian with zero mean and variance σ^2. The signal to noise ratio (SNR) is defined as $1/\sigma^2$. For each simulation trial, $H_{i,j}(m,n)$, and $N_{i,j}$ are drawn independently from the Gaussian distribution, and $B_{i,j}$ from the binary distribution $(M = 2)$. The performance of the algorithms in the form of probability of bit error is obtained. In addition, a single user lower bound, applicable to any multiuser detection algorithm, is obtained for given α^2, σ^2 by Monte-Carlo averaging of the Q-function $Q\left(\sqrt{\sum_{m,n} 2\|H_{i-m,j-n}(n,m)\|^2/\sigma^2}\right)$ over channel realizations. All simulation results were obtained by repeating the realizations until obtaining $N_e = 500$ bit errors for the cell with the smallest number of errors. We have observed that by increasing N_e the estimate of bit error probability P_b varies very slightly. Therefore the deviation in the simulation results is small.

Before we present the simulation results, we first compare the BCJR algorithm, BP algorithm on the clustered graph, and BP algorithm on the decomposed graph in terms of complexity and the amount of message passing required among neighboring base stations per iteration. Numerical complexity comparison is listed in the first row of Table I, where we take the number of real multiplications per base station as a measure. Since all the messages required among neighboring base stations are normalized probabilities in the range $[0,1]$, they can be represented as fixed point real numbers, and we measure the amount of communication among neighboring base stations in terms of number of fixed point real number transmissions to its neighbors for a base station. We list the total amount of communication per base station per iteration in the second row of Table I. We recognize that the base station communications may be required to be in series, or they may proceed in parallel, depending on whether the algorithm is serial or parallel in nature. We therefore also list the amount of communication required in serial in order to complete an iteration for each algorithm in the third row of Table I. The messages in the BCJR algorithm need to proceed on the rows and columns in serial, therefore the amount of serial communication among base stations grow with the size of the array, N. On the other hand, the BP algorithms have a parallel nature, which is demonstrated by the fact that the amounts in row 3 are row 2 divided by four for these algorithms in Table I. Thus for the BP algorithms, the total amount of communication required for a base station to complete an iteration is evenly divided into communication with its four neighbors, which are done in parallel. This is significant, since the amount of serial communication per iteration, along with the number of iterations, determines the required rate of land-line communication among the base-stations in terms of the symbol rate of the uplink communication from the mobile units to the base stations. Comparing the BP algorithms with clustered and decomposed graphical representations, we observe that the decomposed graph can be more advantageous for larger values of M. We explain this by the fact that clustering results in some of the structure being hidden in the nodes, while the decomposed model lets the BP algorithm work on the true nature of the problem.

We examine the amount of serial local communication among base stations required for the iterative algorithms to converge in Fig. 6, where the error rate performance is shown for cell $(2,2)$ in a 4×4 network. From Table I, it is seen that the amount of serial communication required in order to complete an iteration are M^2 and $2M$ real numbers for the algorithms based on clustered and decomposed models, respectively. Therefore, for binary signaling, four real numbers are transmitted in serial per iteration for both BP algorithms. For the clustered graph, the first iteration is completed without any message passing. For the decomposed graph, the first beliefs of the symbols are obtained after the transmission of the first $\lambda_{Y_{i,j}}(b_{v_{i,j}(k)})$ messages, which requires M serial transmissions of real numbers. After the first iteration, each algorithm has a constant number of transmissions per iteration. It is clear from Fig. 6 that both algorithms are quickly achieving performance that is close to the single-user performance. The BP algorithm on the decomposed graph converges slightly faster than the clustered graph case, approximately ten serial real number transmissions are required among base stations per symbol transmitted in the uplink. It is observed that for the

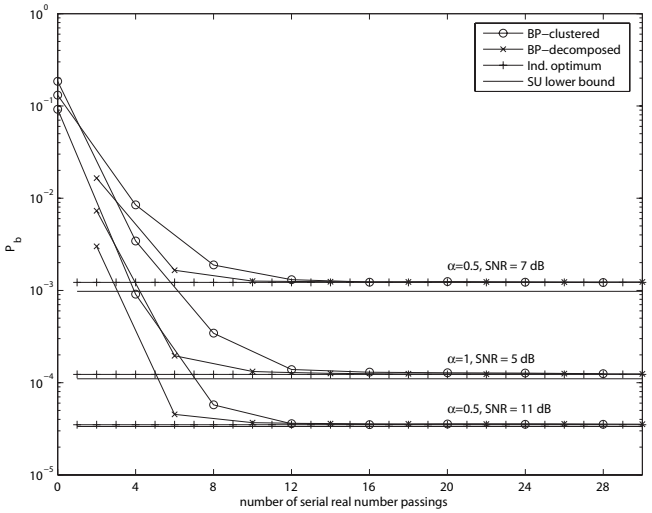

Fig. 6. Probability of error of the algorithms and the single user lower bound as a function of number of serial real number transmissions between base stations for node $(2, 2)$ of a 4×4 network.

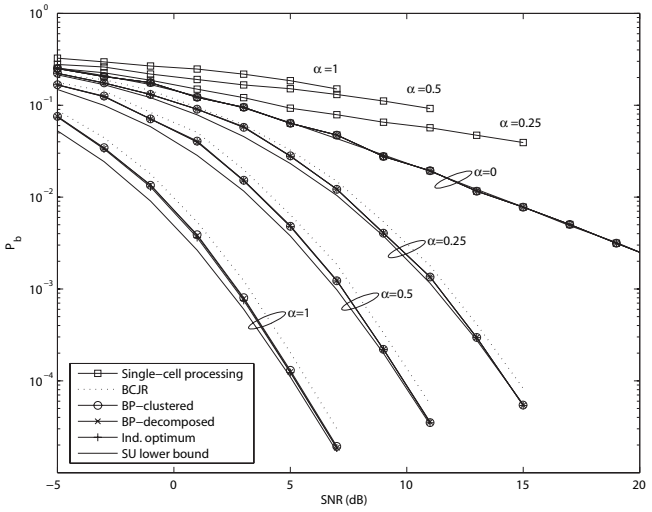

Fig. 7. Probability of error of the algorithms and the single user lower bound as a function of signal to noise ratio for cell $(2, 2)$ of a 4×4 network.

decomposed graph case, after four iterations, performance no longer improves with more iterations. Therefore the algorithm can be stopped after four iterations. For the the clustered graph case, it is observed that five iterations are needed for the performance to converge, after which the algorithm can be stopped. It is also observed that the convergence speeds of the algorithms are not altered by the strength of the ICI. For the iterative BCJR algorithms the amount of serial transmissions required is much larger, and it gets even larger with increasing array size N, so they are not shown in the plot. For this relatively small network size, the individually optimum detector which calculates the exact APP is also simulated. In Fig. 7, the bit error rate performances of the algorithms are illustrated for cell $(2, 2)$ of a 4×4 planar array. The values plotted are the performances after enough iterations have occurred to satisfy the convergence criteria that the performance no longer improves with more iterations. As a benchmark for comparison, we also plot the performance of the optimal single-cell processing receiver which calculates the exact APP $p(b_{i,j}|y_{i,j})$ by averaging out the ICI. Note that this is the result of the first iteration of the BP algorithm on the clustered graph, which is obtained before the first message passing among the base stations (At the end of the first iteration the belief the state is $\mathrm{BEL}(s_{i,j}) = \alpha p(y_{i,j}, s_{i,j})$, from the marginalization of which the distribution of $B_{i,j}$ is obtained.)

Different values of the ICI parameter α are considered in Fig. 7. For the case of $\alpha = 0$, there is no ICI, and the algorithms boil down to optimum APP estimation for a single user channel. It is observed that all algorithms perform robustly for the range of α considered. As the ICI power increases, it is observed that the distributed decoding algorithms not only can handle the ICI, but they can also exploit the extra energy and diversity provided by it. The error rate performance of the BP algorithms on the clustered and decomposed graphs are very close to the single user lower bound at low error probabilities. This tells us that the single user bound is tight for fading channels (previously observed in multiuser detection literature [21]) especially at high SNRs and also that the iterative BP algorithms achieve close to the performance of the individually optimum multiuser detector, which also cannot beat the single-user lower bound. Thus, we observe a very large gain to be accrued from local message passing to reduce the ICI. The BCJR algorithm, implemented iteratively on the columns and rows of the rectangular array, does not perform as well as the BP algorithms; a 0.5 dB gap is observed.

We next investigate whether the convergence rate varies as the size of the planar array grows. For the BP algorithms, each iteration takes the same number of serial message passings regardless of the size of the network. We consider a rectangular array of size 20×20, and observe in Fig. 8 that the performance is the same as that in the smaller network of size 4×4. The observation that the speed of convergence remains roughly the same is explained by the fact the ICI is a local effect even though the overall network size is growing. This is analogous to the equalization problem where the equalizer window size does not need to grow with the length of the signal. Finally, we investigate whether the performance varies with the location of the cell in the planar array. We have observed that as long as the cell is not at the corner or the edge of the rectangular array, the performance of the algorithms for all cells is essentially the same. This is illustrated in Fig. 9 for the BP algorithm on the decomposed graph. The performance of the cells at the edges being worse than that of the cells in the middle is the result of the fact that they have less diversity.

A. Hexagonal Network Case

Although we have considered the relatively simple rectangular model in this paper for brevity, we note that the algorithms developed are extendable to the hexagonal case. Next, we briefly present results when the rectangular model is extended to the hexagonal case.

We extended the three algorithms developed for the rectangular model in Sections III-B, IV-A, and IV-B. For the BCJR algorithm we perform BCJR in three directions successively,

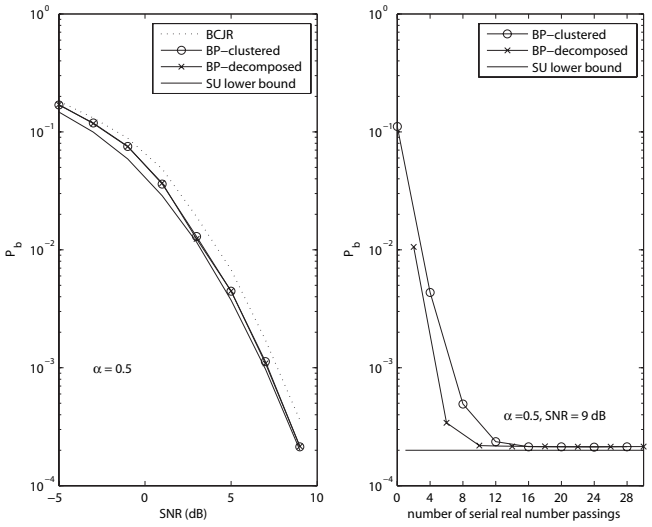

Fig. 8. Probability of error of the algorithms and the single user lower bound as a function of signal to noise ratio and as a function of number of real number transmissions in series for cell $(10, 10)$ of a 20×20 network.

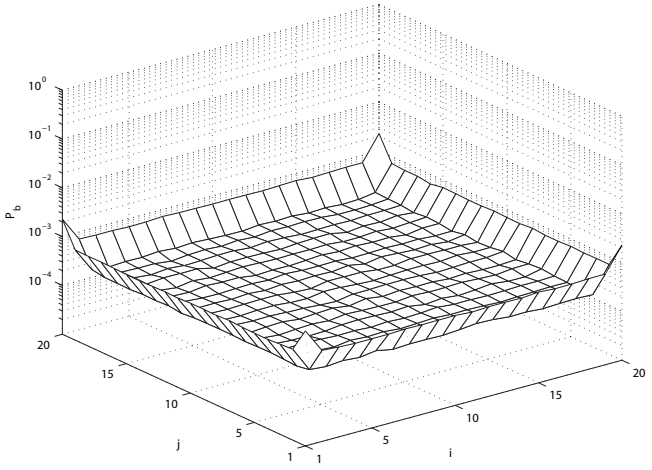

Fig. 9. Probability of error of the belief propagation algorithm on the decomposed graph for all the cells of a 20×20 network, for $\alpha = 0.5$ and SNR $= 9$ dB.

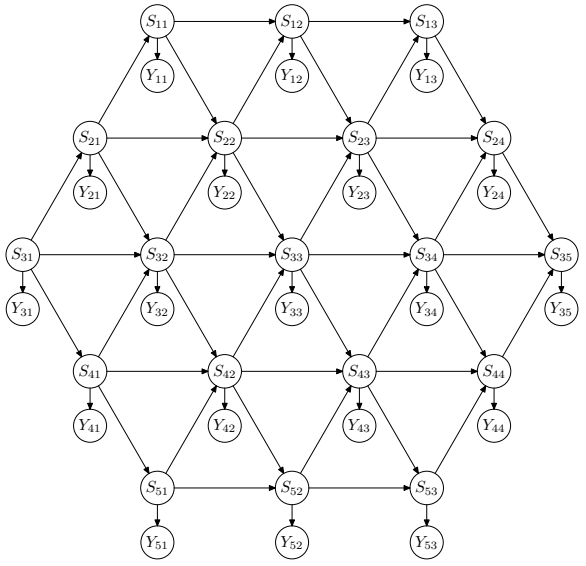

Fig. 10. Clustered model used for the hexagonal network of 19 cells. Each state S contains seven transmitted Bs.

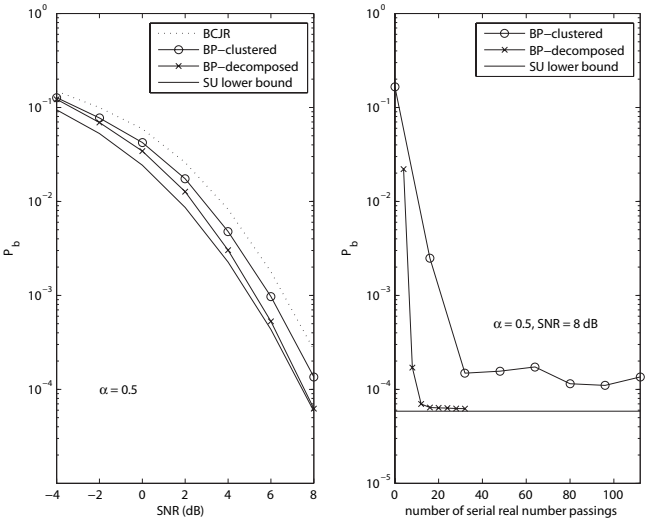

Fig. 11. Probability of error of the algorithms and the single user lower bound for cell (3,3) of the hexagonal network of 19 cells.

where the directions are east-west, southeast-northwest, and northeast-southwest. Clearly, this method is further from being parallel, so it is not feasible. For BP algorithm with clustered graph, the graph in Fig. 10 is considered, and the algorithm is derived with approximations similar to the ones in (20) and (25). For the BP algorithm with decomposed graph, straightforward extension of the graph in Fig. 5 is considered.

The results of the simulation that we conducted for the hexagonal case are given in Fig. 11. These results are for the network of Fig. 10. We observe performance behavior similar to the rectangular case, except for the fact that the performance gap between the clustered graph and the decomposed graph algorithms becomes notable for the hexagonal case. In addition, the amount of message passed among neighbor base stations increases for the hexagonal case, while it remains the same for the decomposed graph. We note that, although all simulation results are presented in this paper are averaged over random complex Gaussian channel coefficients, we have also conducted simulations for fixed channel coefficients that are identical for all cells, in the spirit of the original Wyner model. We have observed that the performance of the BP algorithm is poor for that model. Thus, we believe that the near-optimal performance we observe here is related to the randomness of the channel coefficients.

VI. Conclusions

We tackled the problem of designing a global receiver in a distributed fashion in the uplink of a planar cellular array with full frequency re-use. We first considered the iterative application of the BCJR algorithm (which is optimal for a one-dimensional array) along the columns and the rows of the rectangular array. The convergence time of this approach tends to be large because of the serial nature of the BCJR algorithm. Further, the resulting bit error rate after convergence leaves some room for improvement. These two observations make this iterative BCJR algorithm less attractive than those proposed later.

Next we considered the application of the BP algorithm, which is a local message passing algorithm defined on directed acyclic graphical representations of probabilistic models. Employing BP for an inference problem involves first the choice of the graphical representation, and next the choice of the order of activation of nodes and message passings on the graph. We considered a clustered graph representation where each observed node has a single parent node, and a decomposed graph representation where each single random variable is represented by a single node. The order of message passings was chosen so that the implementation was as parallel as possible. In simulations, in both cases the error rates were near the single user lower bound for fading channels. We have illustrated that, using the BP algorithm, it is possible to use non-orthogonal signaling and still achieve near single user performance with moderate complexity for base stations, and a limited amount of message passing between base stations of adjacent cells.

REFERENCES

[1] A. Wyner, "Shannon-theoretic approach to a Gaussian cellular multiple-access channel," *IEEE Trans. Inf. Theory*, vol. 40, no. 6, pp. 1713–1727, Nov. 1994.

[2] S. V. Hanly and P. Whiting, "Information-theoretic capacity of multi-receiver networks," *Telecommun. Syst. – Modeling, Analysis, Design and Management*, vol. 1, no. 1, pp. 1–42, 1993.

[3] O. Somekh and S. Shamai (Shitz), "Shannon-theoretic approach to a Gaussian cellular multiple-access channel with fading," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, p. 2000, July 2000.

[4] S. Verdú, *Multiuser Detection*. Norwell, MA: Cambridge University Press, 1998.

[5] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann Publishers, 1988.

[6] B. J. Frey, *Graphical Models for Machine Learning and Digital Communication*. Cambridge, MA: The MIT Press, 1998.

[7] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's belief propagation algorithm," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 140–152, Feb. 1998.

[8] Y. Weiss, "Correctness of local probability propagation in graphical models with loops," *Neural Computation*, vol. 12, no. 1, pp. 1–41, Jan. 2000.

[9] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

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

[11] A. Grant, S. Hanly, J. Evans, and R. Müller, "Distributed decoding for Wyner cellular," in *Proc. Australian Communications Theory Workshop*, Feb. 2004, pp. 77–81.

[12] B. Ng, J. Evans, and S. Hanly, "Distributed linear multiuser detection in cellular networks based on Kalman smoothing," in *Proc. IEEE Global Communications Conference*, Nov. 2004, pp. 134–138.

[13] E. Aktas, J. Evans, and S. Hanly, "Distributed decoding in a cellular multiple-access channel," in *Proc. IEEE International Conference on Information Theory*, June 2004, p. 484.

[14] L. Welburn, J. K. Cavers, and K. W. Sowerby, "A computational paradigm for space-time multiuser detection," *IEEE Trans. Commun.*, vol. 52, no. 9, pp. 1595–1604, Sept. 2004.

[15] O. Shental, A. J. Weiss, N. Shental, and Y. Weiss, "Generalized belief propagation receiver for near-optimal detection of two-dimensional channels with memory," in *Proc. IEEE Information Theory Workshop*, Oct. 2004, pp. 225–229.

[16] M. Marrow and J. K. Wolf, "Iterative detection of 2-dimensional ISI channels," in *Proc. IEEE Information Theory Workshop*, March 2003, pp. 131–134.

[17] X. Chen and K. Chugg, "Near-optimal data detection for two-dimensional ISI/AWGN channels using concatenated modeling and iterative algorithms," in *Proc. IEEE International Conference on Communications*, June 1998, pp. 952–956.

[18] X. C. Chen, K. Chugg, and M. Neifeld, "Near-optimal parallel distributed data detection for page-oriented optical memories," *IEEE J. Sel. Topics Quantum Electron.*, vol. 4, no. 5, pp. 866–879, Sept.-Oct. 1998.

[19] P. Thiennviboon, A. Ortega, and K. Chugg, "Simplified grid message-passing algorithm with application to digital image halftoning," in *Proc. IEEE Internatinal Conference on Image Processing*, 2001, pp. 1061–1064.

[20] K. Chugg, A. Anastasopoulos, and X. Chen, *Iterative Detection: Adaptivity, Complexity Reduction, and Applications*. Boston: Kluwer Academic Publishers, 2001.

[21] Z. Zvonar and D. Brady, "Multiuser detection in single-path fading channels," *IEEE Trans. Commun.*, vol. 42, no. 2/3/4, pp. 1729–1739, Feb./March/April 1994.

Emre Aktas (S'96-M'03) received the B.S. degree from Middle East Technical University, Ankara, Turkey, in 1996 and the M.S. and Ph.D. degrees in electrical engineering from The Ohio State University, Columbus, in 1998 and 2002, respectively. During the summer of 2000, he was an intern Engineer at Lucent Technologies, Whippany, NJ, working on an interference-mitigating receiver design for UMTS cellular communication systems. Until August 2003, he was a Postdoctoral Researcher with Stevens Institute of Technology, Hoboken, NJ. From August 2003 to August 2004, Dr. Aktas was a Research Fellow at the University of Melbourne, Australia. He joined Hacettepe University in 2004, where he is currently an assistant professor. His research interests of are in wireless communications, statistical signal processing and information theory. Dr. Aktas is the recipient of EC Sixth Framework Programme Marie Curie Fellowship and the Scientific and Research Council of Turkey Career Award.

Jamie Evans (S'93-M'98) was born in Newcastle, Australia, in 1970. He received the B.S. degree in physics and the B.E. degree in computer engineering from the University of Newcastle, Newcastle, Australia, in 1992 and 1993, respectively, and the M.S. and Ph.D. degrees from the University of Melbourne, Australia, in 1996 and 1998, respectively, both in electrical engineering.

From March 1998 to June 1999, he was a Visiting Researcher with the Department of Electrical Engineering and Computer Science, University of California, Berkeley. He returned to Australia to take up a position as Lecturer at the University of Sydney, Sydney, Australia, where he stayed until July 2001. Since that time, he has been with the Department of Electrical and Electronic Engineering, University of Melbourne, where he is now an Associate Professor. His research interests are in communications theory, information theory, and statistical signal processing with current focus on wireless communications networks. Dr. Evans was the recipient of the University Medal of the University of Newcastle and the Chancellor's Prize for Excellence for his Ph.D. dissertation.

Stephen Hanly (M'98) received the B.Sc. (Hon.) degree in mathematics and computer science and the M.Sc. degree in mathematics from the University of Western Australia, in 1988 and 1990, respectively. Under the Commonwealth Scholarship and Fellowship Plan, he undertook postgraduate research in mathematics at Cambridge University, where he received the Ph.D. degree in 1994. From 1993 to 1995, he was a Postdoctoral Member of Technical Staff at AT&T Bell Laboratories. He is presently an Associate Professor and Reader in the Department of Electrical and Electronic Engineering at the University of Melbourne, where he has been employed in teaching and research since 1996. His research interests are in information theory, and resource allocation problems in wireless networks. Dr. Hanly is an Associate Editor for *IEEE Transactions on Wireless Communications*, and in 2005 was the Technical Co-Chair for the IEEE International Symposium on Information Theory held in Adelaide, Australia. He was a co-recipient of the best paper award at the INFOCOM 1998 conference, and the 2001 Joint IEEE Communications Society and Information Theory Society best paper award, both for his joint work with D. N. C. Tse.