

# MIMO Detection for High-Order QAM Based on a Gaussian Tree Approximation

Jacob Goldberger and Amir Leshem

**Abstract**—This paper proposes a new detection algorithm for MIMO communication systems employing high-order QAM constellations. The factor graph that corresponds to this problem is very loopy; in fact, it is a complete graph. Hence, a straightforward application of the Belief Propagation (BP) algorithm yields very poor results. Our algorithm is based on an optimal tree approximation of the Gaussian density of the unconstrained linear system. The finite-set constraint is then applied to obtain a cycle-free discrete distribution. Simulation results show that even though the approximation is not directly applied to the exact discrete distribution, applying the BP algorithm to the cycle-free factor graph outperforms current methods in terms of both performance and complexity. The improved performance of the proposed algorithm is demonstrated on the problem of MIMO detection.

**Index Terms**—Integer Least Squares, High-order QAM, MIMO communication systems, MIMO-OFDM systems.

## I. INTRODUCTION

Finding a linear least squares fit to data is a well-known problem, with applications in almost every field of science. When there are no restrictions on the variables, the problem has a closed-form solution. In many cases, a-priori knowledge on the values of the variables is available. One example is the existence of priors, which leads to Bayesian estimators. Another example of great interest in a variety of areas is when the variables are constrained to a discrete finite set. This problem has many diverse applications such as the decoding of multi-input-multi-output (MIMO) digital communication systems. In contrast to the continuous linear least squares problem, this problem is known to be NP hard [1].

We consider a MIMO communication system with  $n$  transmit antennas and  $m$  receive antennas. The tap gain from transmit antenna  $j$  to receive antenna  $i$  is denoted by  $\mathbf{H}_{ij}$ . In each use of the MIMO channel a vector  $x = (x_1, \dots, x_n)^\top$  is independently selected from a finite set of complex numbers  $\mathcal{A}$  according to the data to be transmitted, so that  $x \in \mathcal{A}^n$ . We further assume that in each use of the MIMO channel,  $x$  is uniformly sampled from  $\mathcal{A}^n$ . The received vector  $y$  is given by

$$y = \mathbf{H}x + \epsilon. \quad (1)$$

Here, noise is modeled by the random vector  $\epsilon$  which is independent of  $x$  and whose components are assumed to be i.i.d.

The material in this work was presented in part at the Neural Information Processing Systems Conference, Vancouver, BC, Canada, Dec 2009.

J. Goldberger is with the School of Engineering, Bar-Ilan University, 52900, Ramat-Gan. Email: goldbej@eng.biu.ac.il

A. Leshem is with the School of Engineering, Bar-Ilan University, 52900, Ramat-Gan. Email: leshema@eng.biu.ac.il

according to a complex Gaussian distribution with mean zero and with known variance  $\sigma^2$ . The  $m \times n$  matrix  $\mathbf{H}$  comprises i.i.d. elements drawn from a complex normal distribution of unit variance. The MIMO detection problem consists of finding the unknown transmitted vector  $x$  given  $\mathbf{H}$  and  $y$ . The task, therefore, boils down to solving a linear system in which the unknowns are constrained to a discrete finite set. It is convenient to reformulate the complex-valued model into a real-valued one. It can be translated into an equivalent double-size real-valued representation that is obtained by considering the real and imaginary parts separately

$$\begin{bmatrix} \text{Re}(y) \\ \text{Im}(y) \end{bmatrix} = \begin{bmatrix} \text{Re}(\mathbf{H}) & -\text{Im}(\mathbf{H}) \\ \text{Im}(\mathbf{H}) & \text{Re}(\mathbf{H}) \end{bmatrix} \begin{bmatrix} \text{Re}(x) \\ \text{Im}(x) \end{bmatrix} + \begin{bmatrix} \text{Re}(\epsilon) \\ \text{Im}(\epsilon) \end{bmatrix}.$$

Hence we assume hereafter that  $\mathbf{H}$  has real values without any loss of generality. The maximum likelihood (ML) solution is

$$\hat{x} = \arg \min_{x \in \mathcal{A}^n} \|\mathbf{H}x - y\|^2. \quad (2)$$

However, going over all the  $|\mathcal{A}|^n$  vectors is practically unfeasible when either  $n$  or  $|\mathcal{A}|$  are large.

A simple sub-optimal solution, known as the zero-forcing algorithm, is based on a linear decision that ignores the finite-set constraint

$$z = (\mathbf{H}^\top \mathbf{H})^{-1} \mathbf{H}^\top y \quad (3)$$

and then, neglecting the correlation between the symbols, finding the closest point in  $\mathcal{A}$  for each symbol independently

$$\hat{x}_i = \arg \min_{a \in \mathcal{A}} |z_i - a|. \quad (4)$$

This scheme performs poorly due to its inability to handle ill-conditioned realizations of the matrix  $\mathbf{H}$  (if  $m < n$  then  $\mathbf{H}^\top \mathbf{H}$  is even singular). Somewhat better performance can be obtained by using a minimum mean square error (MMSE) Bayesian estimation for the continuous linear system. Let  $e$  be the mean symbol energy. We can partially incorporate the information that  $x \in \mathcal{A}^n$  by using the prior Gaussian distribution  $x \sim \mathcal{N}(0, eI)$ . The MMSE estimation becomes

$$E(x|y) = (\mathbf{H}^\top \mathbf{H} + \frac{\sigma^2}{e} I)^{-1} \mathbf{H}^\top y \quad (5)$$

and then the finite-set solution is obtained by finding the closest lattice point in each component independently. A vast improvement over the linear approaches described above can be achieved by the MMSE with Successive Interference Cancellation (MMSE-SIC) algorithm that is based on sequential decoding with optimal ordering [2]. These linear type algorithms can also easily provide probabilistic (soft-decision) estimates for each symbol. However, there is still a significant

gap between the detection performance of the MMSE-SIC algorithm and the performance of the ML detector.

Many alternative methods have been proposed to approach the ML detection performance. The sphere decoding (SD) algorithm finds the exact ML solution by searching the nearest lattice point [1], [3], [4], [5]. Although SD reduces computational complexity compared to the exhaustive search of the ML solution, sphere decoding is not feasible for high-order QAM constellations. While SD has been empirically found to be computationally very fast for small to moderate problem sizes (say, for  $n < 20$  for 16-QAM), the sphere decoding complexity would be prohibitive for large  $n$ , higher order QAM and/or low SNRs [6]. Another family of MIMO decoding algorithms is based on semidefinite relaxation (e.g., [7], [8], [9]). Although the theoretical computational complexity of semidefinite relaxation is a low degree polynomial, in practice the running time is very high. Thus, there is still a need for low-complexity detection algorithms that perform well.

This study attempts to solve the MIMO decoding problem using the Belief Propagation (BP) paradigm. It is well-known (see e.g., [10]) that a straightforward implementation of the BP algorithm to the MIMO detection problem yields very poor results since there are a large number of short cycles in the underlying factor graph. In this study we introduce a novel approach to utilize the BP paradigm for MIMO detection. The proposed variant of the BP algorithm is both computationally efficient and achieves improved results. The paper proceeds as follows. In Section II we discuss previous attempts to apply variants of the BP algorithm to the MIMO decoding problem. The proposed algorithm which we dub ‘The Gaussian-Tree-Approximation (GTA) Algorithm’ is described in Section III. Experimental results are presented in Section IV.

## II. THE LOOPY BELIEF PROPAGATION APPROACH

Given the constrained linear system  $y = \mathbf{H}x + \epsilon$ , and a uniform prior distribution on  $x$  over a finite set of points  $\mathcal{A}$ , the posterior probability function of the discrete random vector  $x$  given  $y$  is

$$p(x|y) \propto \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{H}x - y\|^2\right), \quad x \in \mathcal{A}^n. \quad (6)$$

The notation  $\propto$  stands for equality up to a normalization constant. Observing that  $\|\mathbf{H}x - y\|^2$  is a quadratic expression, it can be easily verified that  $p(x|y)$  can be factorized into a product of two- and single-variable potentials

$$p(x_1, \dots, x_n|y) \propto \prod_i \psi_i(x_i) \prod_{i < j} \psi_{ij}(x_i, x_j) \quad (7)$$

such that

$$\begin{aligned} \psi_i(x_i) &= \exp\left(-\frac{1}{2\sigma^2}y^\top h_i x_i\right) \\ \psi_{ij}(x_i, x_j) &= \exp\left(-\frac{1}{\sigma^2}h_i^\top h_j x_i x_j\right) \end{aligned} \quad (8)$$

where  $h_i$  is the  $i$ -th column of the matrix  $\mathbf{H}$ . Since the obtained factors are simply functions of pairs of random variables, we obtain a Markov Random Field (MRF) representation [11]. In the MIMO application the (known) matrix  $\mathbf{H}$  is randomly

selected and therefore the MRF graph is usually a completely connected graph (see an MRF graph illustration in Fig. 1).

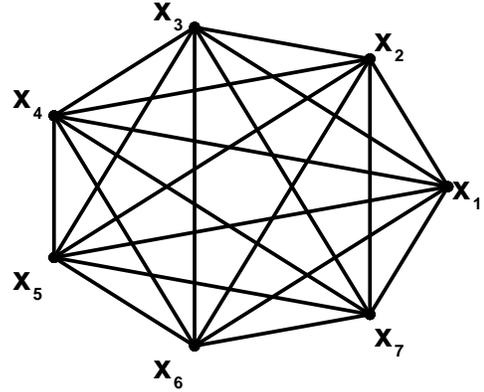


Fig. 1. The MRF undirected graphical model corresponds to the MIMO detection problem with  $n = 7$ .

The Belief Propagation (BP) algorithm aims to solve inference problems by propagating information throughout this MRF via a series of messages sent between neighboring nodes (see [12] for an excellent tutorial on BP). In the sum-product variant of the BP algorithm applied to the MRF (7), the message from  $x_j$  to  $x_i$  is

$$m_{j \rightarrow i}(x_i) = \sum_{x_j \in \mathcal{A}} \psi_j(x_j) \psi_{ij}(x_i, x_j) \prod_{k \neq i, j} m_{k \rightarrow j}(x_j), \quad x_i \in \mathcal{A}. \quad (9)$$

In each iteration messages are passed along all the graph edges in both edge directions. In every iteration, an estimate of the posterior marginal distribution (‘belief’) for each variable can be computed by multiplying together all the incoming messages from all the other nodes

$$b_i(x_i) = \psi_i(x_i) \prod_{k \neq i} m_{k \rightarrow i}(x_i), \quad x_i \in \mathcal{A}. \quad (10)$$

A variant of the sum-product algorithm is the max-product algorithm in which the summation in (9) is replaced by a maximization over all the symbols in  $\mathcal{A}$ . In a cycle-free MRF graph the sum-product algorithm always converges to the exact marginal probabilities (which in the case of MIMO detection corresponds to a soft decision of each symbol, i.e.,  $p(x_i|y)$ ). In a cycle-free MRF graph the max-product variant of the BP algorithm always converges to the most likely configuration [13] (which corresponds to ML decoding in our case). For cycle-free graphs, BP is essentially a distributed variant of dynamic programming. The BP message update equations only involve passing messages between neighboring nodes. Computationally, it is thus straightforward to apply the same local message updates in graphs with cycles. In most such models, however, this loopy BP algorithm will not compute exact marginal distributions; hence, there is almost no theoretical justification for applying the BP algorithm (one exception is that, for Gaussian graphs, if BP converges, then the means are correct [14]). However, the BP algorithm applied to loopy graphs has been found to have outstanding empirical success in many applications, e.g., in decoding of LDPC

codes [15]. The performance of BP in this application may be attributed to the sparsity of the graphs. The cycles in the graph are long, hence the graph has tree-like properties, so that messages are approximately independent and inference may be performed as though the graph was cycle-free. The BP algorithm has also been used successfully in image processing and computer vision (e.g., [16]) where the image is represented by a grid-structured MRF that is based on local connections between neighboring nodes.

However, when the graph is not sparse, and is not based on local grid connections, loopy BP almost always fails to converge. Unlike the sparse graphs of LDPC codes, or grid graphs in computer vision applications, the MRF graphs of MIMO channels are *completely connected graphs* and therefore the associated detection performance is poor. This has prevented the BP from being an asset for the MIMO problem. Fig. 2 shows performance curves for a BPSK MIMO system based on an  $8 \times 8$  matrix and  $\mathcal{A} = \{-1, 1\}$  (see Section IV for a detailed description of the simulation set-up). As can be seen in Fig. 2, the BP decoder based on the MRF representation (7) yields very poor results. Standard techniques to stabilize the BP iterations such as damping the message updates [17] do not help here. Even applying more advanced versions of BP (e.g., Generalized BP and Expectation Propagation [18]) to inference problems on complete MRF graphs yields poor results. The problem here is not in the optimization method but in the cost function that needs to be modified to yield a good approximate solution (see e.g., [19]).

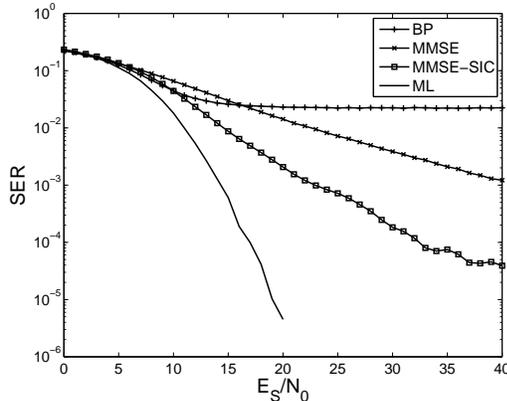


Fig. 2. Decoding results for  $8 \times 8$  BPSK real valued system,  $\mathcal{A} = \{-1, 1\}$ .

Shental et al. [10] analyzed the case where the matrix  $\mathbf{H}$  is relatively sparse (and has a grid structure) (see Fig. 3). They showed that even under this restricted assumption the BP still does not perform well. As an alternative method they proposed the generalized belief propagation (GBP) algorithm that does work well on the sparse matrix if the algorithm regions are carefully chosen. There are situations where the sparsity assumption makes sense (e.g., 2D intersymbol interference (ISI) channels). However, in the MIMO channel model we assume that the channel matrix elements are i.i.d. and Gaussian; hence we cannot assume that the channel matrix  $\mathbf{H}$  is sparse.

There is yet another way to write the MIMO detection problem (6) as a factor graph [20]. It can be easily verified

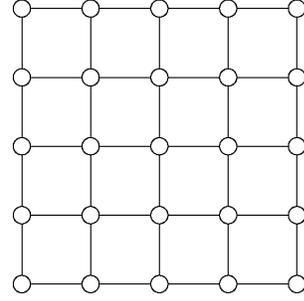


Fig. 3. The MRF grid model corresponds to  $5 \times 5$  2D intersymbol interference (ISI) channels.

that

$$p(x|y) \propto \prod_i p(y_i|x) \propto \prod_i \exp\left(-\frac{1}{2\sigma^2}(h_i x - y_i)^2\right), \quad x \in \mathcal{A}^n, \quad (11)$$

with  $h_i$  is the  $i$ -th row of  $\mathbf{H}$ . Each factor corresponds to a single linear equation and therefore each factor is a function of all the unknown variables. An illustration of the resulting factor graph is shown in Fig. 4. The variables of the factor graph correspond to the transmit antennas and the factors correspond to the receive antennas. Note that the factor graph representation of (11) is a completely connected bipartite graph. Applying the BP algorithm on the factor graph (11) yields the following messages. The BP message from the factor associated with  $y_j$  to the variable  $x_i$  is

$$m_{j \rightarrow i}(a) = \sum_{x|x_i=a} \exp\left(-\frac{1}{2\sigma^2}(h_j x - y_j)^2\right) \prod_{k \neq i} m_{k \rightarrow j}(x_k), \quad (12)$$

and the BP message from variable  $x_i$  to factor  $y_j$  is

$$m_{i \rightarrow j}(a) = \prod_{k \neq j} m_{k \rightarrow i}(a), \quad a \in \mathcal{A}. \quad (13)$$

A major drawback of applying BP to the factor graph (11) is the complexity of computing the factor-to-variable BP messages (12), which is exponential in the number of variables  $n$  [21]. Several recent studies suggested an efficiently computed approximation of the factor-to-variable BP messages based on a Gaussian approximation. The Gaussian approximation can be justified for large values of  $n$  by the central limit theorem (see e.g., [22], [23], [24]). While these algorithms were originally described for BPSK constellations, they can be generalized to arbitrary QAM constellations.

In recent years there were also several attempts to apply BP for densely connected Gaussian graphs. For example consider the MIMO detection problem (6) without the finite-set constraint on the unknown variable  $x$  (see e.g., [25]). The Gaussian MRF model, however, is much easier. In that case the computational complexity of finding the exact solution is a low-degree polynomial in the number of variables. The BP algorithm can be utilized to further reduce the complexity.

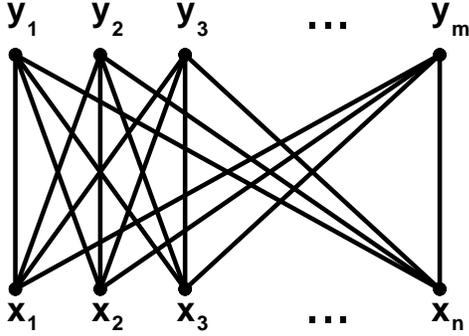


Fig. 4. Factor graph representation of the MIMO detection system. The variables of the factor graph  $x_1, \dots, x_n$  correspond to the transmit antennas and the factors  $y_1, \dots, y_m$  correspond to the receive antennas.

### III. THE GAUSSIAN TREE APPROXIMATION ALGORITHM

Our approach is based on an approximation of the exact probability function

$$p(x_1, \dots, x_n | y) \propto \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{H}\mathbf{x} - y\|^2\right), \quad x \in \mathcal{A}^n \quad (14)$$

that enables a successful implementation of the Belief Propagation paradigm. Since the BP algorithm is optimal on connected cycle-free factor graphs, i.e., on trees, a reasonable approach is finding an optimal tree approximation of the exact distribution (14). Chow and Liu [26] proposed a method to find a tree approximation of a given distribution that has the minimal Kullback-Leibler (KL) divergence to the true distribution. They showed that the optimal tree can be found efficiently by searching for the maximum-weight spanning tree in the weighted complete graph on  $n$  vertices where the weight of an edge is given by the mutual information of the two random variables associated with the edge's endpoints. The problem is that the Chow-Liu algorithm is based on the two-dimensional marginal distributions. However, finding the marginal distribution of the probability function (14) is, unfortunately, NP hard and it is (equivalent to) our final target.

To overcome this obstacle, our approach is based on applying the Chow-Liu algorithm on the distribution corresponding to the unconstrained linear system. This distribution is Gaussian and therefore it is straightforward in this case to compute the two-dimensional marginal distributions. Given the Gaussian tree approximation, the next step of our approach is to apply the finite-set constraint and utilize the Gaussian tree distribution to form a discrete loop free approximation of  $p(x|y)$  which can be efficiently globally maximized using the BP algorithm. To motivate this approach we first show that the zero-forcing decoding algorithm (4) can be derived by utilizing a Gaussian approximation which is a simplified version of our Gaussian tree approximation.

Let  $z(y) = (\mathbf{H}^\top \mathbf{H})^{-1} \mathbf{H}^\top y$  be the least-squares estimator (3) and  $\mathbf{C} = \sigma^2 (\mathbf{H}^\top \mathbf{H})^{-1}$  be its covariance matrix. It can be easily verified that  $p(x|y)$  (14) can be written as

$$p(x|y) \propto f(x; z, \mathbf{C}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{C}|}} \times \quad (15)$$

$$\exp\left(-\frac{1}{2}(z-x)^\top \mathbf{C}^{-1}(z-x)\right)$$

where  $f(x; z, \mathbf{C})$  is a Gaussian density with mean  $z$  and covariance matrix  $\mathbf{C}$ .  $f(x; z, \mathbf{C})$  can be viewed as a posterior distribution of  $x$  assuming a non-informative prior. Note that (15) is only defined if  $m \geq n$ , otherwise the matrix  $\mathbf{H}^\top \mathbf{H}$  is singular and therefore  $\mathbf{C}$  is undefined. Now, instead of marginalizing the true distribution  $p(x|y)$ , which is an NP hard problem, we approximate it by the product of the marginals of the Gaussian density  $f(x; z, \mathbf{C})$

$$\begin{aligned} f(x; z, \mathbf{C}) &\approx \prod_i f(x_i; z_i, \mathbf{C}_{ii}) \quad (16) \\ &= \prod_i \frac{1}{\sqrt{2\pi \mathbf{C}_{ii}}} \exp\left(-\frac{(z_i - x_i)^2}{2\mathbf{C}_{ii}}\right). \end{aligned}$$

At this stage we apply the finite-set constraint. From the Gaussian approximation (16) we can extract a discrete approximation

$$\hat{p}(x|y) \propto \prod_i \exp\left(-\frac{(z_i - x_i)^2}{2\mathbf{C}_{ii}}\right), \quad x \in \mathcal{A}^n. \quad (17)$$

Since this joint probability function is obtained as a product of marginal probabilities, we can decode each variable separately

$$\hat{p}(x_i = a|y) \propto \exp\left(-\frac{(z_i - a)^2}{2\mathbf{C}_{ii}}\right), \quad a \in \mathcal{A}. \quad (18)$$

Taking the most likely symbol we obtain the sub-optimal zero-forcing solution (4).

Motivated by the simple product-of-marginals approximation described above, we suggest approximating the discrete distribution  $p(x|y)$  via a tree-based approximation of the Gaussian distribution  $f(x; z, \mathbf{C})$ . Although the Chow-Liu algorithm was originally stated for discrete distributions, one can verify that it also applies for the Gaussian case. For the sake of completeness we give a detailed derivation below.

#### A. Finding the optimal Gaussian tree approximation

We represent an  $n$ -node tree graph by the cycle-free parent relations  $\{p(i)\}_{i=1}^n$  such that  $p(i)$  is the parent node of  $i$ . To simplify notation we do not separately describe the root node. The parent of the root is implicitly assumed to be the empty set. A distribution  $g(x_1, \dots, x_n)$  is described by a tree  $\{p(i)\}$  if it can be written as  $g(x) = \prod_{i=1}^n g(x_i | x_{p(i)})$ . We start with a formula for the KL divergence between a Gaussian distribution  $f(x)$  and a distribution  $g(x)$  defined on the same space that is represented by a tree graphical model.

**Theorem 1:** Let  $f(x) = f(x_1, \dots, x_n)$  be a multivariate Gaussian distribution and let  $g(x) = \prod_{i=1}^n g(x_i | x_{p(i)})$  be another distribution that is represented by a cycle-free graphical model (tree). The KL divergence between  $f$  and  $g$  is

$$\begin{aligned} D(f \| g) &= \sum_{i=1}^n D(f(x_i | x_{p(i)}) \| g(x_i | x_{p(i)})) \quad (19) \\ &\quad -h(x) + \sum_{i=1}^n (h(x_i) - I(x_i; x_{p(i)})) \end{aligned}$$

such that  $I$  is the mutual information and  $h$  is the differential entropy, based on the distribution  $f(x)$ .

**Proof:** The definition of the KL divergence implies

$$\begin{aligned}
D(f \parallel g) &= \int f \log f - \int f(x) \sum_{i=1}^n \log g(x_i | x_{p(i)}) \\
&= -h(x) - \sum_{i=1}^n \int f(x_i, x_{p(i)}) \log g(x_i | x_{p(i)}) \\
&= -h(x) + \sum_{i=1}^n \int f(x_i, x_{p(i)}) \log \frac{f(x_i | x_{p(i)})}{g(x_i | x_{p(i)})} \\
&\quad - \sum_{i=1}^n \int f(x_i, x_{p(i)}) \log f(x_i | x_{p(i)}) \\
&= -h(x) + \sum_{i=1}^n D(f(x_i | x_{p(i)}) \parallel g(x_i | x_{p(i)})) \\
&\quad - \sum_{i=1}^n (I(x_i; x_{p(i)}) - h(x_i)). \quad \square
\end{aligned}$$

From (19) it can be easily seen that if we fix a tree graph  $\{p(i)\}_{i=1}^n$ , the tree distribution whose KL divergence to  $f(x)$  is minimal is  $g(x) = \prod_{i=1}^n f(x_i | x_{p(i)})$ . In other words, the best tree approximation of  $f(x)$  is constructed from the conditional distributions of  $f(x)$ . For that tree approximation we have

$$D\left(f \parallel \prod_{i=1}^n f(x_i | x_{p(i)})\right) = \sum_{i=1}^n h(x_i) - h(x) - \sum_{i=1}^n I(x_i; x_{p(i)}). \quad (20)$$

Moreover, since in (20)  $h(x)$  and  $h(x_i)$  do not depend on the tree structure, the tree topology  $\{p(i)\}$  that best approximates  $f(x)$  is the one that maximizes the sum

$$\sum_{i=1}^n I(x_i; x_{p(i)}). \quad (21)$$

A spanning tree of a connected graph is a subgraph that contains all the vertices and is a tree. Now suppose the edges of the graph have weights. The weight of a spanning tree is simply the sum of weights of its edges. Eq. (21) reveals that the problem of finding the best tree approximation of the Gaussian distribution  $f(x)$  can be reduced to the well-known problem of finding the maximum spanning tree of the weighted  $n$ -node graph where the weight of the  $i$ - $j$  edge is the mutual information between  $x_i$  and  $x_j$  [26]. It can be easily verified that the mutual information between two r.v.  $x_i$  and  $x_j$  that are jointly Gaussian is

$$I(x_i; x_j) = -\log(1 - \rho_{ij}^2) \quad (22)$$

where  $\rho_{ij}$  is the correlation coefficient between  $x_i$  and  $x_j$ .

There are several algorithms to find a minimum spanning tree. They all utilize a greedy approach. In this work we use the Prim algorithm [27] which is efficient and very simple to implement. The Prim algorithm begins with some vertex  $v$  in a given graph, defining the initial set of vertices  $T$ . Then, in each iteration, we choose the edge with minimal weight among all the edges  $(u, v)$ , where  $u$  is outside of  $T$  and  $v$

is in  $T$ . Then vertex  $u$  is brought in to  $T$ . This process is repeated until a spanning tree is formed. We can use a heap to remember, for each vertex, the lowest-weight edge connecting the current sub-tree  $T$  with that vertex. The complexity of Prim's algorithm, for finding the minimum-weight spanning tree of a  $n$ -vertex graph is  $O(n^2)$ .

We note in passing that the Prim algorithm only relies on the order of the weights and not on their exact values. Hence, applying a monotonically increasing function on the graph weights does not change the topology of the optimal tree. To find the optimal Gaussian tree approximation we can, therefore, use the weights  $\rho_{ij}^2$  instead of  $I(x_i; x_j) = -\log(1 - \rho_{ij}^2)$ . The optimal Gaussian tree is, therefore, the one that maximizes the sum of the square correlation coefficients between adjacent nodes. To summarize, the algorithm that finds the best Gaussian tree approximation is as follows. Define  $x_1$  as the root. Then find the edge connecting a vertex in the tree to a vertex outside the tree, such that the corresponding square correlation coefficient is maximal and add the edge to the tree. Continue this procedure until a spanning tree is obtained.

### B. Applying BP on the tree approximation

Let  $\hat{f}(x)$  be the optimal Chow-Liu tree approximation of  $f(x; z, \mathbf{C})$  (15). We can assume, without loss of generality, that  $\hat{f}(x)$  is rooted at  $x_1$ .  $\hat{f}(x)$  is a cycle-free Gaussian distribution on  $x_1, \dots, x_n$ , i.e.,

$$\hat{f}(x) = f(x_1; z, \mathbf{C}) \prod_{i=2}^n f(x_i | x_{p(i)}; z, \mathbf{C}), \quad x \in R^n \quad (23)$$

where  $p(i)$  is the 'parent' of the  $i$ -th node in the optimal tree. The Chow-Liu algorithm guarantees that  $\hat{f}(x)$  is the optimal Gaussian tree approximation of  $f(x; z, \mathbf{C})$  in the sense that the KL divergence  $D(f \parallel \hat{f})$  is minimal.

Given the Gaussian tree approximation, the next step of our approach is to apply the finite-set constraint to form a discrete loop free approximation of  $p(x|y)$  which can be efficiently globally maximized using the BP algorithm. Our approximation approach is, therefore, based on replacing the true distribution  $p(x|y)$  with the following approximation

$$\hat{p}(x_1, \dots, x_n | y) \propto \hat{f}(x) = f(x_1; z, \mathbf{C}) \prod_{i=2}^n f(x_i | x_{p(i)}; z, \mathbf{C}), \quad (24)$$

The probability function  $\hat{p}(x|y)$  is a cycle-free factor graph. Hence the BP algorithm can be applied to find its most likely configuration. We next derive the messages of the BP algorithm that is applied on  $\hat{p}(x|y)$ . An optimal BP schedule, when applied to a tree, requires passing a message once in each direction of each edge [20]. The BP messages are first sent from leaf variables 'downward' to the root. The computation begins at the leaves of the graph. Each leaf variable node sends a message to its parent. Each vertex waits for messages from all of its children before computing the message to be sent to its parent. The 'downward' BP message from a variable  $x_i$  to its parent variable  $x_{p(i)}$  is computed based on all the messages

Input: A constrained linear LS problem:  $\mathbf{H}x + \epsilon = y$ , a noise level  $\sigma^2$  and a finite symbol set  $\mathcal{A}$  whose mean symbol energy is denoted by  $e$ .

Algorithm:

- Compute  $z = (\mathbf{H}^\top \mathbf{H} + \frac{\sigma^2}{e} I)^{-1} \mathbf{H}^\top y$  and  $\mathbf{C} = \sigma^2 (\mathbf{H}^\top \mathbf{H} + \frac{\sigma^2}{e} I)^{-1}$ .

- Denote

$$f(x_i; z, \mathbf{C}) = \exp\left(-\frac{1}{2} \frac{(x_i - z_i)^2}{\mathbf{C}_{ii}}\right)$$

$$f(x_i|x_j; z, \mathbf{C}) = \exp\left(-\frac{1}{2} \frac{((x_i - z_i) - \mathbf{C}_{ij}/\mathbf{C}_{jj}(x_j - z_j))^2}{\mathbf{C}_{ii} - \mathbf{C}_{ij}^2/\mathbf{C}_{jj}}\right).$$

- Compute the maximum spanning tree of the  $n$ -node graph where the weight of the  $i$ - $j$  edge is the square of the correlation coefficient:  $\rho_{ij}^2 = \mathbf{C}_{ij}^2/(\mathbf{C}_{ii}\mathbf{C}_{jj})$ . Assume that the tree is rooted at node '1' and denote the parent of node  $i$  by  $p(i)$ .
- Apply BP on the loop free distribution

$$\hat{p}(x_1, \dots, x_n|y) \propto f(x_1; z, \mathbf{C}) \prod_{i=2}^n f(x_i|x_{p(i)}; z, \mathbf{C}),$$

to find the (approx. to the) most likely configuration.

Fig. 5. The Gaussian Tree Approximation (GTA) Algorithm.

$x_i$  received from its children

$$m_{i \rightarrow p(i)}(x_{p(i)}) = \sum_{x_i \in \mathcal{A}} f(x_i|x_{p(i)}; z, \mathbf{C}) \prod_{j|p(j)=i} m_{j \rightarrow i}(x_i). \quad (25)$$

If  $x_i$  is a leaf node in the tree then the message is simply

$$m_{i \rightarrow p(i)}(x_{p(i)}) = \sum_{x_i \in \mathcal{A}} f(x_i|x_{p(i)}; z, \mathbf{C}). \quad (26)$$

The ‘downward’ computation terminates at the root node.

Next, BP messages are sent ‘upward’ back to the leaves. The computation begins at the root of the graph. Each vertex waits for a message from its parent before computing the messages to be sent to each of its children. The ‘upward’ BP message from a parent variable  $x_{p(i)}$  to its child variable  $x_i$  is computed based on the ‘upward’ message  $x_{p(i)}$  received from its parent  $x_{p(p(i))}$  and from ‘downward’ messages that  $x_{p(i)}$  received from all the siblings of  $x_i$

$$m_{p(i) \rightarrow i}(x_i) = \sum_{x_{p(i)} \in \mathcal{A}} f(x_i|x_{p(i)}; z, \mathbf{C}) m_{p(p(i)) \rightarrow p(i)}(x_{p(i)}) \times \prod_{\{j|j \neq i, p(j)=p(i)\}} m_{j \rightarrow p(i)}(x_{p(i)}), \quad x_i \in \mathcal{A}. \quad (27)$$

If  $x_{p(i)}$  is the root of the tree then the message is simply

$$m_{p(i) \rightarrow i}(x_i) = \sum_{x_{p(i)} \in \mathcal{A}} f(x_i, x_{p(i)}; z, \mathbf{C}) \times \quad (28)$$

$$\prod_{\{j|j \neq i, p(j)=p(i)\}} m_{j \rightarrow p(i)}(x_{p(i)}), \quad x_i \in \mathcal{A}.$$

After the downward-upward message passing procedure is completed, we can compute the ‘belief’ at each variable which is the product of all the messages sent to the variable from its parent and from its children (if there are any).

$$\text{belief}_i(x_i) = m_{p(i) \rightarrow i}(x_i) \prod_{j|p(j)=i} m_{j \rightarrow i}(x_i), \quad x_i \in \mathcal{A}. \quad (29)$$

In the case  $x_i$  is the root node, the ‘belief’ is computed as follows

$$\text{belief}_i(x_i) = f(x_i; z, \mathbf{C}) \prod_{j|p(j)=i} m_{j \rightarrow i}(x_i), \quad x_i \in \mathcal{A}. \quad (30)$$

Since the approximated distribution  $\hat{p}(x|y)$  (24) is cycle-free, the general Belief Propagation theory guarantees that (the normalized) belief vector is exactly the marginal distribution  $\hat{p}(x_i|y)$  of the approximated distribution  $\hat{p}(x|y)$  (24). To obtain a hard-decision decoding we choose the symbol whose posterior probability is maximal

$$\hat{x}_i = \arg \max_a \text{belief}_i(a), \quad a \in \mathcal{A}. \quad (31)$$

Above we described the sum-product version of the BP algorithm that computes the marginal probabilities  $\hat{p}(x_i|y)$ . A variant of the sum-product algorithm is the max-product algorithm in which the summation in (25)-(28) is replaced by a maximization over all the symbols in  $\mathcal{A}$ . The max-product algorithm finds the most likely pattern of the approximation  $\hat{p}(x|y)$ . We did not observe any significant performance difference using either the sum-product or the max-product variants for decoding MIMO systems presented in the experiment section. The max-product is more computationally efficient since the BP messages can be entirely computed in the log-domain.

### C. An MMSE version of a tree approximation

The MMSE Bayesian approach (5) is known to be better than the zero-forcing solution (4). In MMSE we partially incorporate the information that  $x \in \mathcal{A}^n$  by using the prior Gaussian distribution  $x \sim \mathcal{N}(0, eI)$ . In a similar way we can consider a Bayesian version of the proposed Gaussian tree approximation. We can partially incorporate the information that  $x \in \mathcal{A}^n$  by using the prior Gaussian distribution  $x \sim \mathcal{N}(0, eI)$  such that  $e = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} a^2$ . This yields the posterior Gaussian distribution

$$f_{(x|y)}(x|y) = \frac{1}{\sqrt{(2\pi)^n |V(x|y)|}} \times \exp\left(-\frac{1}{2} (x - E(x|y))^\top V(x|y)^{-1} (x - E(x|y))\right) \quad (32)$$

such that  $E(x|y) = (\mathbf{H}^\top \mathbf{H} + \frac{\sigma^2}{e} I)^{-1} \mathbf{H}^\top y$  and  $V(x|y) = \sigma^2 (\mathbf{H}^\top \mathbf{H} + \frac{\sigma^2}{e} I)^{-1}$ . The MMSE method is obtained by approximating the unconstrained posterior distribution  $f_{(x|y)}(x|y)$  by a product of marginals. In our approach we use, instead, the best cycle-free Gaussian approximation. We can apply the

Chow-Liu tree approximation on the Gaussian distribution (32) to obtain a ‘Bayesian’ Gaussian tree approximation for  $p(x|y)$ . In this way we partially use the finite-set constraint when we search for the best tree approximation of the true discrete distribution  $p(x|y)$ . This is likely to yield a better approximation of the discrete distribution  $p(x|y)$  than the tree distribution which is based on the unconstrained distribution  $f(x; z, \mathbf{C})$ . Note that if there are more transmit antennas than receive antennas, i.e.,  $m < n$ , the linear system  $\mathbf{H}\mathbf{x} + \epsilon = \mathbf{y}$  is underdetermined and therefore the ZF solution does not exist and we cannot apply the Gaussian tree approximation. However, the MMSE solution and the MMSE version of the Gaussian tree approximation are still valid.

To summarize, our solution to the MIMO decoding problem is based on applying BP on a discrete version of the Gaussian tree approximation of the Bayesian version of the continuous least-square solution. We dub this method ‘‘The Gaussian-Tree-Approximation (GTA) Algorithm’’. The GTA algorithm is summarized in Fig. 5. We next compute the complexity of the GTA algorithm. The complexity of computing the covariance matrix  $\sigma^2(\mathbf{H}^\top \mathbf{H} + \frac{\sigma^2}{e} \mathbf{I})^{-1}$  is  $O(n^2m)$  (if  $m < n$  we can utilize the matrix inversion lemma for the matrix inversion). The complexity of the Chow-Liu algorithm (based on Prim’s algorithm for finding the minimum spanning tree) is  $O(n^2)$  and the complexity of the BP algorithm is  $O(|\mathcal{A}|^2n)$  (regardless of the number of receive antennas).

#### IV. EXPERIMENTAL RESULTS

In this section we provide simulation results for the GTA algorithm over various MIMO systems. The channel matrix comprised i.i.d. elements drawn from a zero-mean complex normal distribution of unit variance. We used 500,000 realizations of the channel matrix and each matrix was used once for sending a message. The performance of the proposed algorithm is shown as a function of the variance of the additive noise  $\sigma^2$ . The signal-to-noise ratio (SNR) is defined as  $10 \log_{10}(E_s/N_0)$  where  $E_s/N_0 = \frac{ne}{\sigma^2}$  ( $n$  is the number of variables,  $\sigma^2$  is the variance of the Gaussian additive noise, and  $e$  is the mean symbol energy). We compared the performance

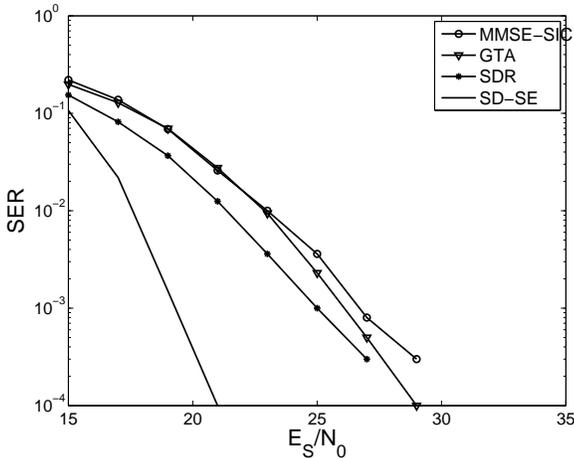


Fig. 6. Comparison of various detectors in  $12 \times 12$  system, 16-QAM symbols.

of the GTA method to the MMSE-SIC algorithm with optimal

ordering for the successive interference cancelation [2], and to the Schnorr-Euchner variant of sphere decoding (SD-SE) with infinite radius [28], [3]. We used sorting of the channel matrix using the SQRD algorithm [29] and regularization [30], which substantially reduces the computational complexity.

We also implemented the SDR detector suggested by Sidiropoulos and Luo [8]. Recently it was shown [9] that for the cases of 16-QAM and 64-QAM this SDR based method is equivalent to the SDR based detection method suggested by Wiesel, Eldar, and Shamai [7]. The SDRs were solved using the CSDP package [31]. In the SDR Gaussian randomization step, 100 independent randomizations were implemented. All the MIMO detection algorithms were implemented in C for efficiency.

Fig. 6 shows MIMO detection performance for a  $12 \times 12$  MIMO system using 16-QAM. The methods that are shown are MMSE-SIC, GTA, SDR, and sphere-decoding. In this case the SDR outperforms both the MMSE-SIC and the GTA methods. However, in this case it is still feasible to compute the optimal maximum-likelihood algorithm using the sphere decoding algorithm.

The SD-SE is the favorite detection method when the problem size is small or moderate. In this case the SD-SE can always yield the exact ML solution at acceptable computational cost. In large size problems, however, there is still a need for good approximation methods. Fig. 7 shows the SER versus SNR and worst case execution time versus SNR for the  $12 \times 12$  system using 64-QAM. Fig. 8 shows the same experimental results for the  $16 \times 16$  system using 64-QAM (time was measured on a dual quad-core Intel Xeon 2.33 GHz). To assess the computational complexity we used a measure of the worst case rather than the average case since in online applications we have to decode within a specified time. The choice between execution time or number of floating point operations is debatable. The differences in running time between the methods we implemented was in orders of magnitude and running time is easier to appreciate.

As can be seen from Figs. 7 and 8, the performance of the GTA algorithm in high SNR is significantly better than the MMSE-SIC. The computational complexity of GTA is comparable to MMSE-SIC and it is much better than SDR. Note that the performance of the SDR method [8] in these 64-QAM cases is worse than that of MMSE-SIC and the computational complexity is much higher. From the derivation of the SDR method it can be seen that the relaxation becomes more crude as at higher constellations.

We next show the performance of several variants of the GTA algorithm. The GTA algorithm differs from the ZF, MMSE, and MMSE-SIC algorithms in several ways. The first difference is that the GTA algorithm utilizes a Markovian approximation of  $f(x; z, \mathbf{C})$  instead of an approximation based on a product of independent densities. The second aspect is the use of an optimal tree. To clarify the contribution of each component we modified the GTA algorithm by replacing the Chow-Liu optimal tree by the tree  $1 \rightarrow 2 \rightarrow 3 \rightarrow \dots \rightarrow n$ . We call this method the ‘Line-Tree’. As can be seen from Fig. 9, using the optimal tree is crucial to obtain improved results. Fig. 9 also shows the results of the non-Bayesian variant of

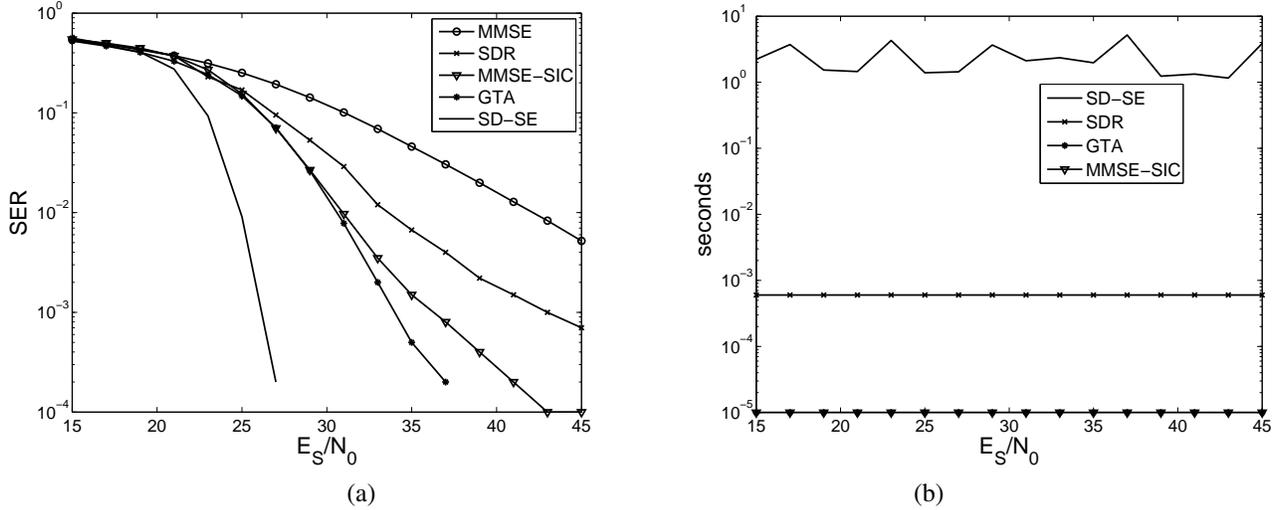


Fig. 7.  $12 \times 12$  system, 64-QAM symbols. (a) SER versus SNR, (b) max seconds for decoded symbol vector versus SNR. Note that the graphs of GTA and MMSE-SIC overlap.

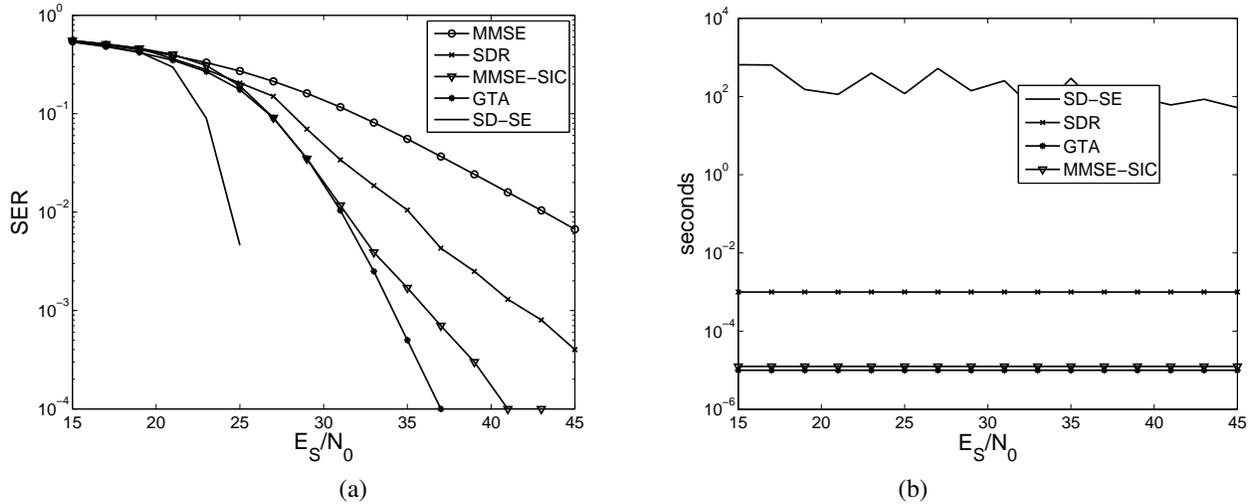


Fig. 8.  $16 \times 16$  system, 64-QAM symbols. (a) SER versus SNR, (b) max seconds for decoded symbol vector versus SNR.

the GTA algorithm. As can be seen, the Bayesian version yields better results. Fig. 9 shows the symbol error rate (SER) versus SNR for a  $20 \times 20$ ,  $|\mathcal{A}| = 4$ , real MIMO system. The performance of the GTA method and its variants was compared to the MMSE and the MMSE-SIC algorithms.

## V. CONCLUSION

We proposed a novel MIMO detection technique based on the principle of a tree approximation of the Gaussian distribution that corresponds to the continuous linear problem. The proposed method outperforms previously suggested MIMO decoding algorithms in high-order QAM constellation, as demonstrated in simulations. Finding the best tree for the integer constrained linear problem is NP hard (unless  $n \leq 2$  in which the  $n$ -node graph is a tree and therefore the Gaussian tree approximation is exact). Although the proposed method yields improved results, the tree approximation we applied may not be the best one. It is left for future research to search for a better discrete tree approximation for the constrained

linear least squares problem. This paper dealt with a tree approximation approach, more complicated approximations such as multi-parent trees could improve performance and can potentially provide a smooth performance-complexity trade-off (see e.g., [24], [32] for other methods that allow an arbitrary tradeoff of speed versus accuracy). Another aspect of the GTA algorithm that is worthwhile to be further investigated is the dependency of the performance on the channel matrix probabilistic model. In the experiments, the channel matrix was sampled from a standard normal distribution, leading to a rather dense system of equations. This is a standard simulation modeling of a MIMO communication system. The performance of the GTA algorithm, however, can be different for more sparse matrices and for binary matrices that occur in CDMA systems. We have shown that for high-order QAM the GTA method outperforms previously suggested MIMO decoding methods such as MMSE-SIC and SDR. As for potential future work, it would be interesting to compare our Gaussian tree approximation BP algorithm to other BP Gaus-

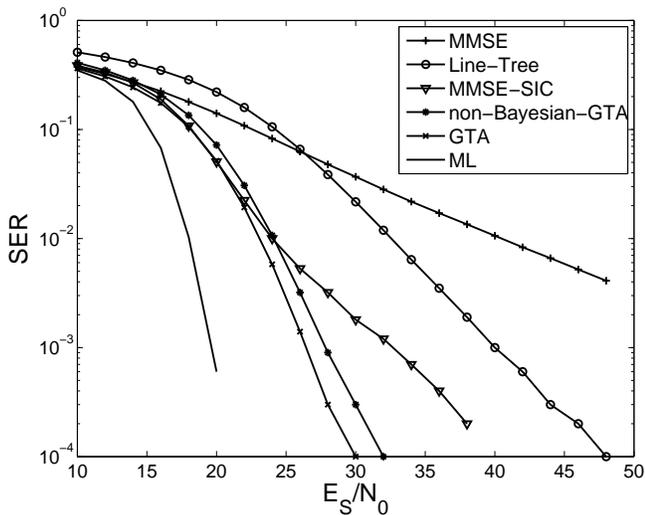


Fig. 9. Comparative results of MMSE, MMSE-SIC and variants of the GTA for a  $20 \times 20$  real system,  $\mathcal{A} = \{\pm 1, \pm 3\}$ .

sian approximation methods that are based on the factor graph representation (11), e.g., the decoding algorithm proposed by Kabashima [22].

While the method provides excellent performance, it is worthwhile to mention that the method provides estimates of the a-posteriori probabilities for each variable that can be used to improve performance. This is done by applying a Schnorr-Euchner sphere decoding algorithm [3], where we order the symbols according to their a-posteriori probabilities. This can lead to very close to optimal performance with highly reduced complexity. This is due to the fact that by utilizing the a-posteriori probabilities, we have a much higher probability of finding the true solution during the first search, therefore significantly reducing the search radius.

There are several important applications of the proposed technique. We comment here on combining it into communication systems with coding and interleaving. It is useful both for single carrier and OFDM systems. It can serve as a MIMO decoder for wireless communication systems. Using the a-posteriori probability distribution of the symbols we can easily estimate the a-posteriori probability and the likelihood ratio for the bits. The technique can be combined with MIMO-OFDM system with bit interleaved coded modulation or with trellis coded modulation by joint coding over all frequency tones of the OFDM system, and running the decoder for each tone independently.

In this paper we focused on the MIMO detection problem. The proposed method, however, can be applied to solve more general constrained linear least squares problems which is an important issue in many fields. A main concept in the GTA model is the interplay between discrete and Gaussian models. Such hybrid ideas can be considered also for discrete inference problems other than least-squares. One example is the work of Oppor and Winther [33] who applied an iterative algorithm using a model which is seen as discrete and Gaussian in turn to address Ising model inference problems. In their approach they approximated intractable probabilistic models by replacing

an average over the original intractable distribution with a tractable one. Our method can be viewed as a special case of their tree expectation-consistent approximation algorithm [33], applied to the MIMO detection problem.

## REFERENCES

- [1] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Computation*, vol. 44, no. 170, pp. 463–471, 1985.
- [2] G. D. Golden, G. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky, "Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture," *Electron. Letters*, vol. 35, no. 1, pp. 14–16, 1999.
- [3] C. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Journal of Mathematical Programming*, vol. 66, no. 1-3, pp. 181–199, 1994.
- [4] J. Boutros, N. Gresset, L. Brunel, and M. Fossorier, "Soft-input soft-output lattice sphere decoder for linear channels," *Proc. IEEE Global Communications Conference*, Dec. 2003.
- [5] C. Studer, A. Burg, and H. Bölcskei, "Soft-output sphere decoding: Algorithms and VLSI implementation," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 2, pp. 290–300, 2008.
- [6] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Processing*, vol. 53, no. 4, pp. 1474–1484, 2005.
- [7] A. Wiesel, Y. C. Eldar, and S. Shamai, "Semidefinite relaxation for detection of 16-QAM signaling in MIMO channels," *IEEE Signal Processing Letters*, vol. 12, no. 9, pp. 653–656, 2005.
- [8] N. Sidiropoulos and Z. Luo, "A semidefinite relaxation approach to MIMO detection for high-order QAM constellations," *IEEE Signal Processing Letters*, vol. 13, no. 9, pp. 525–528, 2006.
- [9] W. Ma, C. Su, J. Jalden, T. Chang, and C. Chi, "The equivalence of semidefinite relaxation MIMO detectors for higher-order QAM," *IEEE Journal of Selected Topics of Signal Processing*, vol. 3, no. 6, pp. 1038–1052, 2009.
- [10] O. Shental, N. Shental, S. S. (Shitz), I. Kanter, A. Weiss, and Y. Weiss, "Discrete-input two-dimensional Gaussian channels with memory: Estimation and information rates via graphical models and statistical mechanics," *IEEE Trans. Inf. Theory*, vol. 54, no. 4, pp. 1500–1513, 2008.
- [11] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Proc. Int. Joint Conf. on Artificial Intelligence*, Aug. 2001.
- [12] E. Sudderth and W. Freeman, "Signal and image processing with belief propagation," *IEEE Signal Processing Magazine, DSP Applications Column*, vol. 25, no. 2, pp. 114–141, 2008.
- [13] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo CA: Morgan Kaufman, 1988.
- [14] Y. Weiss and W. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Computation*, vol. 13, no. 10, pp. 2173–2200, 2001.
- [15] R. G. Gallager, *Low density parity check codes*. M.I.T. Press, 1963.
- [16] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *Int. Journal of Computer Vision*, vol. 70, no. 1, pp. 41–54, 2006.
- [17] K. Murphy, Y. Weiss, and M. Jordan, "Loopy belief propagation for approximate inference: An empirical study," *Proc. Uncertainty in Artificial Intelligence*, pp. 467–475, Jul. 1999.
- [18] T. Minka and Y. Qi, "Tree-structured approximations by expectation propagation," *Proc. Neural Information Processing Systems*, Dec. 2004.
- [19] J. Goldberger and A. Leshem, "Pseudo prior belief propagation for densely connected discrete graphs," *Proc. IEEE Information Theory Workshop*, Jan. 2010.
- [20] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, 2001.
- [21] M. Kaynak, T. Duman, and E. Kurtas, "Belief propagation over SISO/MIMO frequency selective fading channels," *IEEE Trans. Wireless Communications*, vol. 6, pp. 2001–2005, 2007.
- [22] Y. Kabashima, "A CDMA multiuser detection algorithm on the basis of belief propagation," *J. Phys. A*, vol. 36, pp. 11 111–11 121, 2003.
- [23] J. Neirotti and D. Saad, "Improved message passing for inference in densely connected systems," *Europhys. Lett.*, vol. 71, pp. 866–872, 2005.
- [24] J. Hu and T. M. Duman, "Graph-based detector for BLAST architecture," *Proc. IEEE Int. Conf. on Communications*, pp. 1018–1023, Jun. 2007.

- [25] D. Bickson, O. Shental, P. H. Siegel, J. K. Wolf, and D. Dolev, "A Gaussian belief propagation based multiuser detection," *Proc. IEEE Int. Symp. Information Theory*, Jul. 2008.
- [26] C. K. Chow and C. N. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Trans. Inf. Theory*, vol. 14, no. 3, pp. 462–467, 1968.
- [27] R. C. Prim, "Shortest connection networks and some generalizations," *J. Bell System Tech.*, pp. 1389–1401, 1957.
- [28] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, 2002.
- [29] D. Wübben, R. Böhnke, J. Rinas, V. Kühn, and K. Kammeyer, "Efficient algorithm for decoding layered space-time codes," *IEE Electronics Letters*, vol. 37, pp. 1348–1350, 2001.
- [30] D. Wübben, R. Böhnke, V. Kühn, and K. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," *Proc. IEEE Vehicular Technology Conference*, vol. 1, pp. 508–512, Oct. 2003.
- [31] B. Borchers, "A C library for semidefinite programming," *Optimization Methods and Software*, vol. 11, pp. 613–623, 1999.
- [32] P. H. Tan and L. K. Rasmussen, "Multiuser detection in CDMA - A comparison of relaxations, exact, and heuristic search methods," *IEEE Trans. on Wireless Communications*, vol. 3, no. 5, pp. 1802–1809, 2004.
- [33] M. Opper and O. Winther, "Expectation consistent approximate inference," *Journal Machine Learning Research*, vol. 6, no. 12, pp. 2177–2204, 2005.

**Jacob Goldberger** Jacob Goldberger received the B.Sc. degree (cum laude) in mathematics and computer science from the Bar-Ilan University, Ramat Gan, Israel, the M.Sc. degree (cum laude) in mathematics, and the Ph.D. degree in Engineering, from the Tel-Aviv University, Tel-Aviv, Israel. In 2004 he joined the School of Electrical and Computer Engineering, Bar-Ilan University, Ramat Gan, Israel, where he is currently a faculty member. His main research interests include machine learning, information theory, computer vision, medical imaging and speech recognition.

**Amir Leshem** Amir Leshem (M98-SM06) received the B.Sc. degree (cum laude) in mathematics and physics, the M.Sc. degree (cum laude) in mathematics, and the Ph.D. degree in mathematics, all from the Hebrew University, Jerusalem, Israel. He is one of the founders of the School of Electrical and Computer Engineering, Bar-Ilan University, Ramat Gan, Israel, where he is currently an Associate Professor and Head of the signal processing track. His main research interests include multichannel communication, applications of game theory to communication, array and statistical signal processing with applications to sensor arrays and networks, wireless communications, radio-astronomy, and brain research, set theory, logic, and foundations of mathematics.