Heuristic Tree Search for Detection and Decoding of Uncoded and Linear Block Coded Communication Systems

Tao Cui, Tracey Ho
Department of Electrical Engineering
California Institute of Technology
Pasadena, CA, USA 91125
Email: {taocui, tho}@caltech.edu

Chintha Tellambura
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, AB, Canada T6G 2V4
Email: chintha@ece.ualberta.ca

***Abstract*— A heuristic tree search algorithm is developed for the maximum likelihood detection and decoding problem in a general communication system. We propose several "cheap" heuristic functions using constrained linear detectors and the minimum mean square errors (MMSE) detector. Even though the MMSE heuristic function does not guarantee the optimal solution, it has a negligible performance loss and provides a good complexity-performance tradeoff. For linear block coded systems, heuristic tree search is modified for soft decision decoding. High rate codes are decoded via the minimum state trellis, and low rate codes via the minimum complexity tree. Preprocessing is also discussed to further speed up the algorithms.**

I. Introduction

Maximum-likelihood (ML) detection achieves significant performance gains over suboptimal algorithms in many communications applications. For example, in a $M_T \times M_R$ uncoded multiple-input and multiple output (MIMO) system, the ML detector extracts M_R order (full) diversity while the zero forcing detector extracts only $M_R - M_T + 1$ order diversity. The performance advantage and high complexity of ML detector have motivated the research on reduced complexity detectors.

The sphere decoder (SD), a special instance of Branch and Bound (BnB), has been applied to solve the ML detection problem in communications [1]. Recent results show that the average complexity of the SD is exponential for fixed SNR [2]. Although the SD is efficient for high SNRs and problems of moderate size, in practice, low SNR operating regimes are common and the dimension of the applications is large. The computational SD complexity in those cases can be high. New detection algorithms should be developed for low SNR applications and/or for large dimensional problems.

Linear block codes that may sometimes be decoded using hard-decision decoding (HDD). However, compared to HDD, soft-decision decoding (SDD) provides about a 2-dB gain in additive white Gaussian noise (AWGN) channels. Similar to the ML detection in uncoded systems, the SDD complexity increases exponentially in the code length. Hence, much effort has been expended on developing efficient ML SDD algorithms. For instance, one such algorithm [3] represents a block code by a trellis and searches it using the Viterbi algorithm. This is successful only for short codes as the trellis complexity increases rapidly with the code length. The heuristic tree search algorithm A* is used in [4] to reduce the Viterbi-search complexity. However, all these papers treat SDD over AWGN channels only. Only a few papers consider SDD of linear block codes in fading channels. In [5], a SDD algorithm using SD is developed over Rayleigh block fading channels. An efficient ML decoder for Golay codes is proposed in [6] for OFDM systems. SDD algorithms for frequency selective and multiple-antenna channels are not available.

In this paper, we use the heuristic tree search algorithms [7] widely-known in the artificial intelligence community to reduce the complexity of ML detection. Although a heuristic tree search algorithm has recently been proposed in [8], since those heuristic functions incur high computational overhead, the overall complexity reduction achievable by their use is offset by the high computations required. Consequently, we propose several "cheap" heuristic functions. Heuristic functions based on constrained least squares (CLS) [9] and generalized minimum mean square errors (GMMSE) [10] are used for unitary and non-unitary constellations, respectively. To further alleviate the computational burden, a minimum mean square errors (MMSE) heuristic function is proposed, which does not guarantee optimal detection. The resulting complexity saving is significant, yet the performance loss is negligible. For coded systems over fading channels, we propose a class of efficient SDD algorithms by modifying the heuristic tree search algorithms for uncoded systems. Both high rate and low rate codes are efficiently decoded using the trellis and tree structures of the codes, respectively. We give a "minimum complexity tree" (MCT) description of a block code. Preprocessing is also discussed to reduce the decoding complexity.

Notation: $(\cdot)^T$ and $(\cdot)^H$ denote transpose and conjugate transpose, respectively. The imaginary unit is $\jmath = \sqrt{-1}$. A circularly complex Gaussian variable with mean m and variance σ^2 is denoted by $z \sim \mathcal{CN}(m, \sigma^2)$. The set of complex numbers is $\mathcal{C}$, and the $N \times N$ identity matrix is $\mathbf{I}_N$.

II. System model

We consider a general detection problem. At the transmitter, k' independent information symbols $\mathbf{b}' \in \mathrm{GF}(2^q)^{k'}$ are encoded via a systematic generator matrix $\mathbf{G}'$ to produce a

1-4244-0355-3/06/$20.00 (c) 2006 IEEE

codeword $\mathbf{c}' \in \mathrm{GF}(2^q)^{n'}$. For some important block codes, e.g., RS codes, we need to convert $\mathbf{c}'$ into a vector $\mathbf{c} \in \mathrm{GF}(2^p)^n$, where $qn' = pn$. Let $\mathbf{G}$ be the equivalent $n \times k$ generator matrix for the (n, k) linear block code over $\mathrm{GF}(2^p)$, and $\mathbf{b} \in \mathrm{GF}(2^p)^k$ be the information data vector. $\mathbf{b}$ is encoded to the $n \times 1$ codeword $\mathbf{c}$ over $\mathrm{GF}(2^p)$ as $\mathbf{c} = \mathbf{G} \odot \mathbf{b}$, where $\odot$ denotes the multiplication over $\mathrm{GF}(2^p)$. Each symbol in $\mathbf{c}$ is subsequently mapped to an element in a constellation $\mathcal{Q}$ with 2^p elements. The resulting modulated vector $\mathbf{x} = \mathcal{M}(\mathbf{c}) \in \mathcal{Q}^n$, where $\mathcal{M}(\cdot)$ denotes the mapping operation. The resulting modulated vector is then transmitted across a vector channel $\mathbf{H} \in \mathcal{C}^{m \times n}$. We assume a discrete-time baseband model. The received signal $\mathbf{r}$ can be expressed as

$$\mathbf{r} = \mathbf{H}\mathbf{x} + \mathbf{n} = \mathbf{H}\mathcal{M}(\mathbf{c}) + \mathbf{n} = \mathbf{H}\mathcal{M}(\mathbf{G} \odot \mathbf{b}) + \mathbf{n} \tag{1}$$

where $\mathbf{n} \in \mathcal{C}^{m \times 1}$ is the additive Gaussian noise vector. Note the $\mathbf{H}$ in (1) denotes a general channel model. An identity matrix $\mathbf{H}$ represents an additive white Gaussian noise (AWGN) channel. A diagonal matrix $\mathbf{H}$ denotes flat fading channels. A band matrix $\mathbf{H}$ represents a frequency selective fading channel. When $\mathbf{H}$ is a full matrix, it represents MIMO channels. Throughout this paper, we assume that the perfect channel knowledge is available at the receiver side and AWGN. If $\mathbf{G}$ is a block identity matrix $\mathbf{G} = \mathbf{I}_n$ and $k = n$, (1) reduces to an uncoded system. Therefore, (1) is a general model that represents several existing coded and uncoded systems.

The ML soft decision decoder that minimizes the average bit error probability is thus given by

$$\hat{\mathbf{b}} = \underset{\mathbf{b} \in \mathrm{GF}(2^p)^k}{\arg\min} \|\mathbf{r} - \mathbf{H}\mathcal{M}(\mathbf{G} \odot \mathbf{b})\|^2. \tag{2}$$

Due to the discrete alphabet nature of $\mathbf{b}$, linear detectors such as least-squares detectors with HDD generally do not give the optimal solution. Exhaustive search for (2) has a complexity exponential in pk.

To proceed, let the QR factorization of $\mathbf{G}$ be

$$\mathbf{H} = [\mathbf{Q}_1, \mathbf{Q}_2] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \tag{3}$$

where $\mathbf{R}$ is an $n \times n$ upper-triangular matrix with (i, j)-th entry $r_{i,j}$, $\mathbf{0}$ is an $(m - n) \times n$ zero matrix, $\mathbf{Q}_1$ is an $m \times n$ unitary matrix and $\mathbf{Q}_2$ is an $m \times (m - n)$ unitary matrix. Eq. (2) is equivalent to

$$\hat{\mathbf{b}} = \underset{\mathbf{b} \in \mathrm{GF}(2^p)^k}{\arg\min} \|\mathbf{y} - \mathbf{R}\mathcal{M}(\mathbf{G} \odot \mathbf{b})\|^2 \tag{4}$$

where $\mathbf{y} = \mathbf{Q}_1^H \mathbf{r}$. Eq. (4) and the upper triangular form of $\mathbf{R}$ are the basis for the heuristic tree search based detectors developed in this paper.

For uncoded systems, the ML detector is given by

$$\hat{\mathbf{x}} = \underset{\mathbf{x} \in \mathcal{Q}^n}{\arg\min} \|\mathbf{y} - \mathbf{R}\mathbf{x}\|^2 \tag{5}$$

where $\mathbf{y}$ and $\mathbf{R}$ are obtained as in (4). In (5), we assume that $n \leq m$. If $n > m$, we can transform the original problem (2) into an equivalent full rank problem using the approach adopted in [11].

In this paper, we focus on ML detection of coded and uncoded MIMO systems with BLAST architecture [12], where n denotes the number of transmit antennas and m denotes the number of receive antennas. For coded BLAST, we consider the vertical coding architecture in [13]. Our approach in this paper can be readily modified to the horizontal coding architecture in [13]. In the following, we only consider $\mathcal{Q}$ to be real, i.e., pulse amplitude modulation (PAM). If $\mathcal{Q}$ is complex, we can decouple (1) into a real system. The SD or the modified SD in [14] can be applied to the resulting system.

III. Sphere decoding algorithm

We first discuss the ML detection of uncoded MIMO systems. The SD only searches those points that lie within a hypersphere of radius d or find all $\mathbf{x}$ such that

$$\|\mathbf{y} - \mathbf{R}\mathbf{x}\|^2 \leq d^2. \tag{6}$$

Note the set $\{\mathbf{R}\mathbf{x} \,|\, \mathbf{x} \in \mathcal{Q}^n\}$ is called a lattice generated by $\mathbf{R}$. The members of the lattice are called lattice points. Thus, an alternative viewpoint is that (6) specifies all the lattice points that are within a distance d from the received signal vector. Eq. (6) can be expanded as

$$\sum_{i=1}^{n} \left(y_i - \sum_{j=i}^{n} r_{i,j} x_j \right)^2 \leq d^2 \tag{7}$$

where $r_{i,j}$ denotes the (i, j)-th entry of $\mathbf{R}$. The first term of (7) depends only on x_n, the second term on x_n, x_{n-1} and so forth. Therefore, a necessary condition for a lattice point to lie inside the sphere is that $d^2 \geq (y_n - r_{n,n} x_n)^2$, which is equivalent to x_n belonging to the interval

$$\left\lceil \frac{-d + y_n}{r_{n,n}} \right\rceil \leq x_n \leq \left\lfloor \frac{d + y_n}{r_{n,n}} \right\rfloor \tag{8}$$

where $\lceil \cdot \rceil$ denotes the smallest integer greater than or equal to its argument and $\lfloor \cdot \rfloor$ denotes the largest integer less than or equal to its argument.

We continue in the same process for x_{n-1} and so on. The bounds for x_k are

$$LB_k \leq x_k \leq UB_k \tag{9}$$

where

$$LB_k = \left\lceil \frac{-d_k + y_k - \sum_{j=k+1}^{n} r_{k,j} x_j}{r_{k,k}} \right\rceil, \tag{10}$$

$$UB_k = \left\lfloor \frac{d_k + y_k - \sum_{j=k+1}^{n} r_{k,j} x_j}{r_{k,k}} \right\rfloor, \tag{11}$$

and $d_k^2 = d_{k+1}^2 - \left(y_{k+1} - \sum_{j=k+1}^{n} r_{k+1,j} x_j \right)^2$. If there is no valid candidate for x_k (i.e., $[LB_k, UB_k] \cap \mathcal{Q} = \emptyset$), the SD comes back to x_{k+1} and chooses another candidate value from the corresponding region for x_{k+1}. If the SD reaches x_1, the SD finds a candidate vector $\mathbf{x}'$, where the lattice point $\mathbf{R}\mathbf{x}'$ lies inside the hypersphere of radius d. The SD checks the value of $\|\mathbf{y} - \mathbf{R}\mathbf{x}'\|^2$. If this value is less than d, we replace the radius

d by this value. The search space is now limited by the new radius. This process continues until no further lattice points is found within the hypersphere. The candidate vector with the smallest value of $\|\mathbf{y}-\mathbf{Rx}\|^2$ is output as the ML solution. If no lattice point inside the sphere is found, the sphere is empty and the search fails. In this case, the initial search radius d must be increased and the search is restarted with the new squared radius until a valid point is found.

Apparently, when n or σ_n^2 is large, (9) contains all the points in $\mathcal{Q}$ in the first k dimensions, and SD cannot prune any nodes. It searches almost all the $|\mathcal{Q}|^k$ points. The ratio k/n increases as increasing n.

IV. Heuristic tree search

The ML detection problem in (5) can be viewed as a tree search problem to find the leaf node with the minimum cost. The conventional SD is a modification of depth-first branch-and-bound (DFBnB) search [7].

In the SD, the bounding range in (9) for x_k is loose primarily because we have neglected the term

$$\sum_{i=1}^{k-1}\left(y_i-\sum_{j=i}^{n}r_{i,j}x_j\right)^2. \tag{12}$$

The sum (12) is the cost from the current node to an end (leaf) nodes. A cost estimate from the current node to the leaf node is called a heuristic function, which can be used to guide the search. If the heuristic function is a **lower bound** to the sum (12), it can be used in (9) for determining the bounding range for x_k. In this case, any nodes pruned by the algorithm would not affect its optimality. The A* search algorithm is an example of a heuristic tree search algorithm. It visits the nodes in order of the heuristic estimate, and it is an example of best-first search. A search algorithm that guarantees the optimal solution is called **admissible**. A heuristic that makes the A* search admissible is called an admissible heuristic (i.e., the heuristic function must be a true lower bound on the cost from the current node to a leaf node).

A lower bound on (12) would be an admissible heuristic function. To proceed, we write (12) into a vector form as

$$\|\mathbf{y}_k-\mathbf{R}_k\mathbf{x}_k-\mathbf{r}_kx_k\|^2 \tag{13}$$

where $\mathbf{y}_k=[y_1,\dots,y_{k-1}]^T$, $\mathbf{x}_k=[x_1,\dots,x_{k-1}]^T$, $\mathbf{R}_k=\mathbf{R}(1:k-1,1:k-1)$, and $\mathbf{r}_k=\mathbf{R}(1:k-1,k)$. In [8], several heuristic functions are proposed by the use of high complexity relaxations, and (13) needs to be computed for each x_k. We use linear constrained detectors as cheap heuristic functions.

A. Heuristics for unitary constellations

We first consider a unitary constellation with unit modulus $|x_i|^2=1$, i.e., M-PSK. Using the approach in [11], a system using quadrature amplitude modulation (QAM) can be transformed into an equivalent system with quadrature phase-shift keying (QPSK) and the constrained detector developed for unitary constellations can also be applied to QAM. The CLS detector relaxes the candidate vectors to be on the hypersphere $\mathbf{x}_k^H\mathbf{x}_k=k-1$. It is the optimal value of the following problem It is thus given by

$$\hat{\mathbf{x}}_k=\underset{\mathbf{x}_k^H\mathbf{x}_k=k-1}{\arg\min}\ \|\mathbf{y}_k-\mathbf{R}_k\mathbf{x}_k-\mathbf{r}_kx_k\|^2. \tag{14}$$

The lagrangian $\mathcal{L}(\mathbf{x}_k,\lambda)$ for this minimization problem is

$$\mathcal{L}(\mathbf{x},\lambda_1,\dots,\lambda_g)=\|\mathbf{y}_k-\mathbf{R}_k\mathbf{x}_k-\mathbf{r}_kx_k\|^2+\lambda\left(\mathbf{x}_k^H\mathbf{x}_k-k+1\right). \tag{15}$$

Taking partial derivatives with respect to $\mathbf{x}_k$ the solution for $\mathbf{x}_k$ can be derived as

$$\hat{\mathbf{x}}_k(\lambda,x_k)=\left(\mathbf{R}_k^H\mathbf{R}_k+\lambda\mathbf{I}_{k-1}\right)^{-1}\mathbf{R}_k^H(\mathbf{y}_k-\mathbf{r}_kx_k) \tag{16}$$

where $\hat{\mathbf{x}}_k(\lambda,x_k)$ means that it is a function of λ and x_k. The optimal value for λ has to be computed so that the constant modulus constraints are fulfilled. Substituting $\hat{\mathbf{x}}_k(\lambda,x_k)$ into (14), we need to find the zeros of the equation

$$\|\hat{\mathbf{x}}_k(\lambda,x_k)\|^2-k+1=0. \tag{17}$$

Let the singular value decomposition (SVD) decomposition of $\mathbf{R}_k$ be $\mathbf{U}_k^T\mathbf{V}_k\mathbf{U}_k$, where $\mathbf{U}_k$ is an unitary matrix and $\mathbf{V}_k$ is a diagonal matrix with diagonal entries σ_i^k, $i=1,\dots,k-1$. Define $\tilde{\mathbf{y}}_k=\mathbf{U}_k\mathbf{y}_k$ and $\tilde{\mathbf{r}}_k=\mathbf{U}_k\mathbf{r}_k$. Eq. (17) becomes

$$\sum_{i=1}^{k-1}\frac{|\tilde{y}_i^k-\tilde{r}_i^kx_k|^2(\sigma_i^k)^2}{((\sigma_i^k)^2+\lambda)^2}=k-1. \tag{18}$$

Eq. (18) is a polynomial with order $2(k-1)$. From [15], it was shown that the global minimum of (14) is achieved by the maximal real root, which can be found using one-dimensional Newton method within the region $[\min((\sigma_i^k)^2),+\infty)$.

B. Heuristics for non-unitary constellations

For a non-unitary constellation, we denote $\rho_{\max}$ and $\rho_{\min}$ as the largest and smallest modulus of the constellation, respectively. We first consider using the constraint on $\rho_{\max}$. We thus relax $\mathbf{x}_k$ within a $k-1$-dimensional hypercube $\mathbf{x}_k^H\mathbf{x}_k\le\rho_{\max}^2(k-1)$. The constrained detector is given by as

$$\hat{\mathbf{x}}_k=\underset{\mathbf{x}_k^H\mathbf{x}_k\le\rho_{\max}^2(k-1)}{\arg\min}\ \|\mathbf{y}_k-\mathbf{R}_k\mathbf{x}_k-\mathbf{r}_kx_k\|^2 \tag{19}$$

The Lagrangian dual function for (19) can be expressed as

$$\mathcal{L}(\mathbf{x}_k,\lambda)=\|\mathbf{y}_k-\mathbf{R}_k\mathbf{x}_k-\mathbf{r}_kx_k\|^2+\lambda\left(\mathbf{x}_k^H\mathbf{x}_k-\rho_{\max}^2(k-1)\right) \tag{20}$$

where λ is the Lagrangian multiplier. The Lagrange dual function is given by

$$g(\lambda)=\inf_{\mathbf{x}_k\in\mathbb{C}^{k-1}}\mathcal{L}(\mathbf{x}_k,\lambda) \tag{21}$$

Minimizing (20) for $\mathbf{x}$, the solution is the same as (16). Substituting it back to (20), λ is solved by maximizing $g(\lambda)$. $g(\lambda)$ can be simplified by using the SVD of $\mathbf{R}_k$. Since $g(\lambda)$ is an unconstrained one-dimensional optimization problem, the simple Newton method can be used. The constrained detector (19) is similar to the GMMSE in [10].

To have tighter relaxation and better performance, the constraint on $\rho_{\min}$ can be taken into account. We can pose another g constraints $\rho^2_{\min}(k-1) \leq \mathbf{x}^H_{k-1}\mathbf{x}_{k-1}$ on (19). Similar primal-dual approach can be used.

Both (14) and (19) result in linear detectors given $\lambda^*(x_k)$. Clearly, both of them provide a lower bound on (15). Therefore, the heuristic function can be defined as

$$h(x_k) = \left\| \left(\mathbf{I}_{k-1} - \mathbf{R}_k \left(\mathbf{R}^H_k \mathbf{R}_k + \lambda^*(x_k)\mathbf{I}_{k-1} \right)^{-1} \mathbf{R}^H_k \right) (\mathbf{y}_k - \mathbf{r}_k x_k) \right\|^2 . \tag{22}$$

In the procedure of SD, the bound on x_k is modified as

$$g(x_k) + h(x_k) < d^2_k \tag{23}$$

where $g(x_k) = (y_k - r_{k,k}x_k)^2$. Instead of deciding the bounds on x_k directly using (14), we must test each $x_k \in \mathcal{Q}$ to obtain the candidate set in (23). The optimality using the heuristic search (23) is guaranteed by the following theorem.

Theorem 1: If $h(x_k)$ is admissible, the heuristic search always returns the optimal solution.

Proof: The proof is trivial, and it is omitted for brevity.

C. MMSE Heuristic

If $\lambda^*(x_k)$ is set to σ^2_n, (16) returns the MMSE solution and $\hat{\mathbf{x}}_k$ is very simple to obtain. We can also obtain

$$(y_k - r_{k,k}x_k)^2 + h(x_k) = (a_k x_k - b_k)^2 + c^2_k \tag{24}$$

where

$$a_k = \sqrt{\left\| \left(\mathbf{I}_{k-1} - \mathbf{R}_k \left(\mathbf{R}^H_k \mathbf{R}_k + \sigma^2_n \mathbf{I}_{k-1} \right)^{-1} \mathbf{R}^H_k \right) \mathbf{r}_k \right\|^2 + r^2_{k,k}}$$

$$b_k = \frac{y^H_k \left(\mathbf{I}_{k-1} - \mathbf{R}_k \left(\mathbf{R}^H_k \mathbf{R}_k + \sigma^2_n \mathbf{I}_{k-1} \right)^{-1} \mathbf{R}^H_k \right)^2 \mathbf{r}_k + y^*_k r_{k,k}}{a_k}$$

$$c_k = \sqrt{\left\| \left(\mathbf{I}_{k-1} - \mathbf{R}_k \left(\mathbf{R}^H_k \mathbf{R}_k + \sigma^2_n \mathbf{I}_{k-1} \right)^{-1} \mathbf{R}^H_k \right) \mathbf{y}_k \right\|^2 + y^2_k - b^2_k}. \tag{25}$$

Like (14), the bounds for x_k are

$$LB_k \leq x_k \leq UB_k \tag{26}$$

where

$$LB_k = \left\lceil \frac{-\sqrt{d^2_k - c^2_k} + b_k}{a_k} \right\rceil, \; UB_k = \left\lfloor \frac{\sqrt{d^2_k - c^2_k} + b_k}{a_k} \right\rfloor . \tag{27}$$

However, the heuristic function using MMSE solution is not admissible, and hence the heuristic tree search does not guarantee optimal solution. Using the Chernoff bound and the union bound approach in [16], we find the symbol error rate (SER) of the heuristic tree search is at most $e^{\sigma^4_n}$ of that of the SD. Therefore, the performance loss compared to the SD is negligible and full diversity order can be achieved.

Note that the use of the heuristic function allows more nodes to be pruned than the standard SD, albeit at the cost of more computations of the heuristic function. Therefore, the use of the heuristic tree search does not always guarantee complexity saving. In particular, as the search proceeds to the bottom of the tree, the original bound by the SD in (9) becomes tighter. Thus, it is not worthwhile to compute the heuristic function values in this case. To get a tradeoff between complexity saving and cost, we can set a level, say, K, if $k < K$ we do not evaluate the heuristic function $h(x_k)$ to save complexity. K also depends on the SNR. In high SNR, the bound by SD is tight and K can be close to n, while K should be close to 1 in low SNR. In a specific SNR, there exists a K that achieves the minimum complexity.

Remarks:

- For unitary constellations, we can use the quadratic eigenvalue approach in [17] to avoid the SVD decomposition.
- If the channel remains constant over several blocks, the SVD of $\mathbf{R}_k$ only needs to be computed once, which reduces the average complexity of the heuristic search. The mean computation complexity comes from solving (19) for the CLS heuristic function. Similar results also hold for GMMSE heuristic function.
- The main difficulty with the CLS and GMMSE heuristic functions is $\lambda^*(x_k)$ depends on x_k. For each $x_k \in \mathcal{Q}$, we must compute $h(x_k)$ and test whether (23) holds. When the constellation size is large, this approach is inefficient. However, if the MMSE heuristic function is used, we can obtained the bounds from (26) directly without resorting to test all the elements in $\mathcal{Q}$.
- The Schnorr and Euchner strategy [18] can also be applied to all the heuristic tree search algorithms by ordering the candidate set according to $g(x_k) + h(x_k)$.
- We can also apply A* to MIMO detection using our heuristic functions. It visits fewer nodes than any other heuristic tree search algorithm with the same heuristic function. However, this approach has a exponential memory requirement.
- The function $h(x_k) = 0$ is also a heuristic function. The heuristic tree search reduces to the DFBnB.

V. Decoding for linear block coded systems

The brute-force search of (4) has complexity proportional to 2^{kp}. On the other hand, search through the trellis of the code with the Viterbi algorithm has complexity $n \times \min(2^{kp}, 2^{(n-k)p})$ [3]. In the worst-case complexity sense, if $2^{kp} < n2^{(n-k)p}$ (e.g. low rate codes), we decode the code using the tree structure; otherwise, we decode the code using the trellis structure (high rate codes).

High rate codes: The minimum state trellis (MST) is first constructed by generating the minimum-span generator matrix (MSGM) via a greedy-type algorithm. The heuristic tree search algorithm uses the MST and starts from c_n. When the search goes to c_l, we first use the heuristic function to determine the candidate set on x_l using (23) or (26). For each candidate x_l from the set, we do a demapping $c_l = \mathcal{M}^{-1}(x_l)$ to obtain the candidate set for c_l. The Viterbi algorithm for the MST does not search over those branches in MST corresponding to the symbols not in the candidate set.

Low rate codes: In [19], A* algorithm is applied to the code tree generated by the systematic generator matrix over AWGN channels. However, this code tree is not optimal in terms of complexity. We introduce a minimum complexity tree (MCT). We transform the generator matrix into a block upper-triangular matrix

$$\mathbf{G} = \begin{bmatrix} \mathbf{a}_{1,1} & \mathbf{a}_{1,2} & \cdots & \mathbf{a}_{1,k} \\ & \mathbf{a}_{2,2} & \cdots & \mathbf{a}_{2,k} \\ & & \ddots & \mathbf{a}_{k-1,k} \\ & & & \mathbf{a}_{k,k} \end{bmatrix} \tag{28}$$

where $\mathbf{a}_{i,j}$ is a $d_i \times 1$ vector and $\sum_{i=1}^{k} d_i = n$. It is clear that the systematic form is a special case of (3) with $d_2 =, \ldots, = d_k = 1$ and $d_1 = n - k + 1$. The MCT is generated by maximizing $d_k, \ldots, d_2$ consecutively. Since the generator matrix for a linear block code is unique regardless of the row reordering, we can order the rows of $\mathbf{G}$ $d_k, \ldots, d_2$. For example, for $(7, 4)$ Hamming code, we obtain $d_4 = d_3 = 1$, $d_2 = 2$ and $d_1 = 3$ after reordering, and it results in

$$\begin{bmatrix} 0\,1\,1\,1\,0\,0\,1 \\ 1\,0\,1\,1\,1\,0\,0 \\ 0\,1\,0\,0\,0\,1\,1 \\ 0\,0\,1\,0\,1\,1\,1 \end{bmatrix} \rightarrow \begin{bmatrix} 1\,1\,1\,0\,0\,0\,0 \\ 0\,1\,0\,1\,1\,0\,0 \\ 1\,1\,0\,1\,0\,1\,0 \\ 1\,0\,0\,1\,0\,0\,1 \end{bmatrix} = \mathbf{G}^T \tag{29}$$

To explain the idea, we first construct a k-level tree. The cost for b_l is given by

$$g(b_l) = \sum_{i=n-e_l-d_l+1}^{n-e_l} \sum_{j=i}^{n} |y_i - r_{i,j} x_j|^2 \tag{30}$$

where $e_l = \sum_{i=k-l+1}^{k} d_i$. Due to the upper triangular form of $\mathbf{R}$ and the block upper triangular form of $\mathbf{G}$, x_j for $j > \sum_{i=1}^{l-1} d_i$ are completely determined by $b_k, \ldots, b_l$. In the heuristic tree search, we decide whether b_l is feasible by checking

$$g(b_l) + h(b_l) < d_l^2 \tag{31}$$

where $d_l^2 = d_{l+1}^2 - g(b_{l+1})$, and $g(b_l)$ is the lower bound for $x_1, \ldots, x_{n-e_l-d_l}$ with the form (22). Unlike the SD in [5], the heuristic tree search algorithm does not need an initial radius by applying the Schnorr and Euchner strategy [18], where at each node, searches the node with minimum $g(b_l) + h(b_l)$ first. When it first reaches the bottom of the tree, the cost from the root to the leaf node is recorded as the minimum cost $C_{\min}$ found so far. $C_{\min}$ is updated whenever a better leaf node is reached. With larger d_l, the average $g(b_l)$ is larger and more nodes prunes. The average complexity is thus reduced. For long codes, the complexity saving by using MCT is significant.

For SDD of systems with linear block codes, We give different heuristic functions $h(b_l)$ for different channel models. (1) AWGN: a tight heuristic function is introduced in [4] by using the constraint on Hamming weights. A simpler function can use hard-decision symbols without affecting the average complexity. (2) Flat fading channels and OFDM: For each x_k, we make hard decision $\hat{x}_i = \arg\min_{x_i \in \mathcal{Q}} |y_i - r_{i,i} x_i|^2$. The $h(b_l)$ is hence given by $h(b_l) = \sum_{i=1}^{n-e_l-d_l} |y_i - r_{i,i}\hat{x}_i|^2$. (3) Frequency selective channels: We relax $\mathbf{x} \in \mathcal{Q}^n$, which is not constrained by the code. The Viterbi algorithm with branch and bound (BnB) is used to find the minimum cost from $x_{n-e_l-d_l}$ to x_1, yielding a worst-case complexity $\mathcal{O}((n + L - 1)2^{L-1})$ (L is the channel length). The cost of each visited node is stored and reused for the heuristic tree search algorithm. (4) MIMO channels: the heuristic functions are given in Section IV. Tighter heuristic function can be found by exploiting more properties of a specific linear block code.

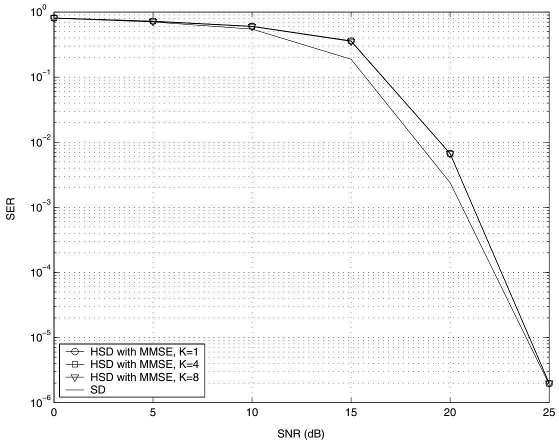

Fig. 1. Performance comparison of heuristic tree search detectors with different heuristic functions in an 8×8 uncoded MIMO system with 16QAM.

In [4], the bits with higher reliability are expanded first. The search complexity depends critically on preprocessing. The reliability is defined as $|y_i|$ in [4]. In MIMO systems, the columns of $\mathbf{H}$ can be permuted according to the Euclidean norm of each column, ZF V-BLAST and MMSE V-BLAST orderings. The ZF V-BLAST ordering is optimal for post-detection SNR. For flat fading channels, ZF V-BLAST reduces to detecting larger $|r_{i,i}|$ first.

VI. Simulation Results

We consider an $n \times n$ MIMO system over a flat Rayleigh fading channel. Our simulation is performed on a workstation with an Intel Xeon processor operating at 3.2GHz. The average CPU computation time is used as the measure of complexity. The SNR is defined to be $E\{|y_k|^2\}/\sigma_n^2$. The SD with heuristic function is denoted as HSD. The initial radius is set to infinity in all the cases.

First, we compare HSD with the SD in an 8×8 uncoded MIMO system with 16QAM. Fig. 1 shows the BER performance of HSD using MMSE heuristic function (24) with different K. We find that even though the MMSE heuristic function is not admissible, the performance loss compared to the SD is negligible. Interestingly, the HSD achieves a full diversity order. From the simulation, we find that the complexity of HSD is almost a constant in all SNRs. HSD has about 3 orders of magnitude of complexity saving over the SD in low SNR. The complexity of HSD with $K = 4$ achieves the minimum complexity even though their difference is small.

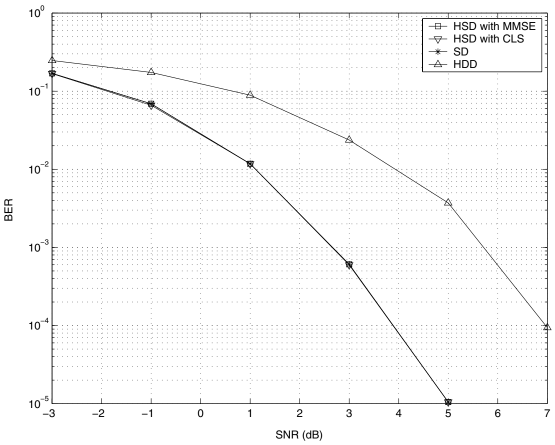

Fig. 2. Performance comparison of heuristic tree search detectors with different heuristic functions in an 23×23 Golay $(23, 12)$ coded MIMO system.

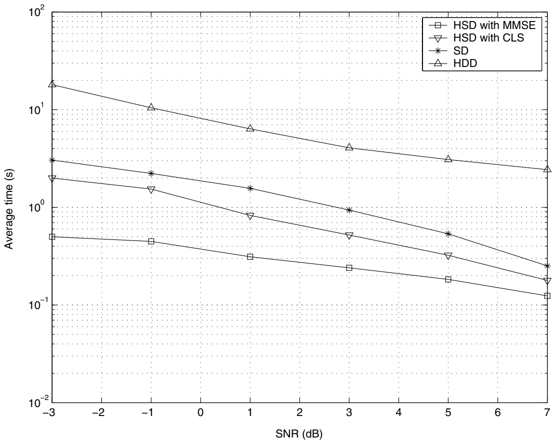

Fig. 3. Complexity comparison of heuristic tree search detectors with different heuristic functions in an 23×23 Golay $(23, 12)$ coded MIMO system.

Fig. 2 compares the performance of HSD with different heuristic functions in an 23×23 Golay $(23, 12)$ coded MIMO system. BPSK is used. SDD performs much better than that of HDD. We do not observe performance loss in HSD with MMSE. At SNR= 1dB, HSD with MMSE and CLS is 5 and 2 times faster than the SD with code tree generated by systematic generator matrix (Fig. 3). For BCH$(63, 36)$ code, at SNR=2dB, the average nodes for SD is 65439 and time 24.33s, the average number of nodes for HSD with CLS is 12782 and time 10.44s, the average of number of nodes for HSD with MMSE is 9148 and time 4.18s. The number of nodes generated is not proportional to the computational time costed. In MIMO systems, the worst-case complexity for SDD $\mathcal{O}(2^k)$ is less than that of HDD $\mathcal{O}(2^n)$. SDD betters HDD in both performance and complexity.

VII. Conclusion

We have proposed using the heuristic tree search algorithm to solve the maximum likelihood detection and decoding problem in general communication systems. Several "cheap" heuristic functions were proposed using constrained linear detectors and minimum mean square errors (MMSE) detector. Even though the MMSE heuristic function does not guarantee the optimal solution, it has a negligible performance loss and provides a good complexity-performance tradeoff. For linear block coded systems, heuristic tree search was modified for soft decision decoding. High rate codes were decoded using the minimum state trellis, while low rate codes uses the minimum complexity tree. We show that the complexity of our new SDD algorithms are significantly less than that of the SD based algorithms. The complexity of the SD in low SNR, a major drawback, is alleviated by our new algorithms. The heuristic tree search algorithm can also be used for decoding problems with *a priori* information.

References

[1] E. Viterbo and J. Bouros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1639–1642, Jul. 1999.

[2] 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, April 2005.

[3] J. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Trans. Inform. Theory*, vol. 24, no. 1, pp. 76 – 80, Jan 1978.

[4] Y. S. Han, C. Hartmann, and C.-C. Chen, "Efficient priority-first search maximum-likelihood soft-decision decoding of linear block codes," *IEEE Trans. Inform. Theory*, vol. 39, no. 5, pp. 1514 – 1523, Sept. 1993.

[5] W. Zhao, X. Cai, and G. B. Giannakis, "An efficient maximum-likelihood decoding algorithm for linear block codes," in *Proc. of the 38th Conf. on Info. Sciences and Systems*, March 2004.

[6] A. Alavi, C. Tellambura, and I. Fair, "Near maximum-likelihood decoding of Golay sequences with reduced complexity," in *Proc. of the 16th Int. Conf. on Wireless Commun.*, July 2004.

[7] W. Zhang, *State Space Search: Algorithms, Complexity and Applications*. New York: Springer-Verlag, 1999.

[8] M. Stojnic, H. Vikalo, and B. Hassibi, "A branch and bound approach to speed up the sphere decoder," in *Proc. of ICASSP*, vol. 3, March 2005, pp. iii/429 – iii/432.

[9] S. Thoen, L. Deneire, L. V. der Perre, M. Engels, and H. D. Man, "Constrained least squares detector for OFDM/SDMA-based wireless networks," *IEEE Trans. Wireless Commun.*, vol. 2, no. 1, pp. 129 – 140, Jan. 2003.

[10] A. Yener, R. D. Yates, and S. Ulukus, "CDMA multiuser detection: a nonlinear programming approach," *IEEE Trans. Commun.*, vol. 50, no. 6, pp. 1016 – 1024, June 2002.

[11] T. Cui and C. Tellambura, "An efficient generalized sphere decoder for rank-deficient MIMO systems," *IEEE Commun. Lett.*, vol. 9, no. 5, pp. 423 – 425, May 2005.

[12] G. D. Golden, G. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky, "Detection algorithm and initial laboratory results using the V-BLAST space-time communication architecture," *Electronics Letters*, vol. 35, no. 1, pp. 14–15, Jan. 1999.

[13] X. Li, H. Huang, G. Foschini, and R. Valenzuela, "Effects of iterative detection and decoding on the performance of BLAST," in *Proc. of IEEE GLOBECOM*, vol. 2, Nov 2000, pp. 1061 – 1066.

[14] T. Cui and C. Tellambura, "Joint channel estimation and data detection for OFDM systems via sphere decoding," in *Proc. of GLOBECOM*, Dec. 2004.

[15] G. E. Forsythe and G. H. Golub, "On the stationary values of a second degree polynomial on the unit sphere," *J. Soc. Indust. Appl. Math,*, vol. 13, no. 1050-1068, 1965.

[16] T. Cui and C. Tellambura, "Generalized feedback detection for MIMO systems," in *Proc. of GLOBECOM*, 2005.

[17] F. Tisseur and K. Meerbergen, "The quadratic eigenvalue problem," *SIAM Review*, vol. 43, no. 2, pp. 235–286, 2001.

[18] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Programming*, vol. 66, pp. 181–191, 1994.

[19] L. Ekroot and S. Dolinar, "A* decoding of block codes," *IEEE Trans. Commun.*, vol. 44, no. 9, pp. 1052 – 1056, Sept. 1996.