A BRANCH AND BOUND APPROACH TO SPEED UP THE SPHERE DECODER

M. Stojnic, H. Vikalo, and B. Hassibi

California Institute of Technology, Pasadena, CA

ABSTRACT

It is well known that maximum-likelihood decoding in many communications applications reduces to solving an integer least-squares problem which is NP hard in the worst-case. On the other hand, it has recently been shown that, over a wide range of dimensions and SNRs, the sphere decoder can be used to find the exact solution with an expected complexity that is roughly cubic in the dimension of the problem. However, the computational complexity becomes prohibitive if the SNR is too low and/or if the dimension of the problem is too large. In this paper, we target these two regimes and attempt to find faster algorithms by pruning the search tree beyond what is done in the standard sphere decoder. The search tree is pruned by computing lower bounds on the possible optimal solution as we proceed to go down the tree. We observe a trade-off between the computational complexity required to compute the lower bound and the size of the pruned tree: the more effort we spend in computing a tight lower bound, the more branches that can be eliminated in the tree. Thus, even though it is possible to prune the search tree (and hence the number of points visited) by several orders of magnitude, this may be offset by the computations required to perform the pruning. All of which suggests the need for computationally-efficient tight lower bounds. We present three different lower bounds based on spherical-relaxation, on polytope-relaxation and on duality, simulate their performances and discuss their relative merits.

1. INTRODUCTION

In this paper we are interested in solving *exactly* the following problem

$$\min_{\mathbf{s}\in\mathcal{D}\subset\mathcal{Z}^m} \|\mathbf{x}-H\mathbf{s}\|_2, \tag{1}$$

where $\mathbf{x}\in\mathcal{R}^m$, $H\in\mathcal{R}^{m\times m}$ and $\mathcal{D}$ refers to some subset of the integer lattice $\mathcal{Z}^m$. The main idea of the sphere decoder algorithm [1] for solving the previous problem is based on finding all points $\mathbf{s}$ such that $\|\mathbf{x}-H\mathbf{s}\|_2$ lies within some adequately chosen radius d, i.e., on finding all $\mathbf{s}$ such that

$$d^2 \geq \|\mathbf{x}-H\mathbf{s}\|_2^2, \tag{2}$$

and then choosing the one that minimizes the objective function. Using the QR-decomposition $H=QR$, with Q unitary and R upper triangular, we can reformulate (2) as

$$d^2 \geq \|\mathbf{y}-R\mathbf{s}\|_2^2, \tag{3}$$

where we have defined $\mathbf{y}=Q^*\mathbf{x}$.

Now using the upper-triangular property of R, (3) can be further rewritten as

$$\begin{aligned} d^2 \quad \geq \quad & \|\mathbf{y}_{k:m}-R_{k:m,k:m}\mathbf{s}_{k:m}\|^2 + \\ & \|\mathbf{y}_{1:k-1}-R_{1:k-1,1:k-1}\mathbf{s}_{1:k-1}-R_{1:k-1,k:m}\mathbf{s}_{k:m}\|^2, \end{aligned} \tag{4}$$

for any $2\leq k\leq m$, where the subscripts determine the entries the various vectors and matrices run over. A necessary condition for (3) can therefore be obtained by omitting the second term on the RHS of the above expression to yield

$$d^2 \geq \|\mathbf{y}_{k:m}-R_{k:m,k:m}\mathbf{s}_{k:m}\|^2. \tag{5}$$

The sphere decoder finds all points $\mathbf{s}$ in (2) by proceeding inductively on (5), starting from $k=m$ and proceeding to $k=1$. In other words for $k=m$, it determines all one-dimensional lattice points $\mathbf{s}_m$ such that

$$d^2 \geq (\mathbf{y}_m - R_{m,m}\mathbf{s}_m)^2,$$

and then for each such one-dimensional lattice point $\mathbf{s}_m$ determines all possible values for $\mathbf{s}_{m-1}$ such that

$$\begin{aligned} d^2 \quad &\geq \quad \|\mathbf{y}_{m-1:m}-R_{m-1:m,m-1:m}\mathbf{s}_{m-1:m}\|^2 \\ &= \quad (\mathbf{y}_m - R_{m,m}\mathbf{s}_m)^2 + \\ & \quad\quad (\mathbf{y}_{m-1}-R_{m-1,m-1}\mathbf{s}_{m-1}-R_{m-1,m}\mathbf{s}_m)^2. \end{aligned}$$

This gives all possible two-dimensional lattice points and one then proceeds in a similar fashion until $k=1$. The sphere decoder thus generates a tree, where the branches at the $m-k+1$th level of the tree correspond to all $m-k+1$-dimensional lattice points satisfying (5). In this manner at the bottom of the tree (the m-th level) all points satisfying

This work was supported in part by the National Science Foundation under grant no. CCR-0133818, by the Office of Naval Research under grant no. N00014-02-1-0578, and by Caltech's Lee Center for Advanced Networking.

(2) are found. (For more details on the sphere decoder and for an explicit description of the algorithm the reader may refer to [1, 10, 4].)

The computational complexity of the sphere decoder depends on how d is chosen. In communications we usually can assume

$$\mathbf{x} = H\mathbf{s} + \mathbf{w}, \tag{6}$$

where the entries of $\mathbf{w}$ are independent $\mathcal{N}(0, \sigma^2)$ random variables. In [4] it is shown that, if the radius is chosen appropriately based on the statistical characteristics of the noise $\mathbf{w}$, then over a wide range of SNRs and problem dimensions the expected complexity of the sphere decoder is roughly cubic.

The above assertion unfortunately fails and the computational complexity becomes increasingly prohibitive if the SNR is too low and/or if the dimension of the problem is too large. Increasing the dimension of the problem clearly is useful. Moreover, the use of the sphere decoder in low SNR situations is also important when one is interested in obtaining soft information to pass onto an iterative decoder (see, e.g., [8, 7]). To reduce the computational complexity one approach is to resort to suboptimal methods based either on heuristics (see, e.g., [6]) or some form of statistical pruning (see [9]).

In this paper, we attempt to reduce the computational complexity of the sphere decoder while still finding the *exact* solution. Let us surmise on how this may be done. As mentioned above, the sphere decoder generates a tree whose number of branches at each level corresponds to the number of lattice points satisfying (5). Clearly, the complexity of the algorithm depends on the size of this tree since each branch in the tree is visited and appropriate computations are then performed. Thus, one approach would be to reduce the size of the tree beyond that which is suggested by (5). To do so, suppose that we had some way of computing a lower bound on the optimal value of the second term of the RHS of (4):

$$\begin{aligned} LB = LB(\mathbf{y}_{1:k-1}, R_{1:k-1,1:m}, \mathbf{s}_{k:m}) \leq \\ \min_{\mathbf{s}_{1:k-1} \in \mathcal{D} \subset \mathcal{Z}^{k-1}} \|\mathbf{y}_{1:k-1} - R_{1:k-1,1:k-1}\mathbf{s}_{1:k-1} - R_{1:k-1,k:m}\mathbf{s}_{k:m}\|^2, \end{aligned}$$

where we have emphasized the fact that the lower bound is a function of $\mathbf{y}_{1:k-1}$, $R_{1:k-1,1:m}$, and $\mathbf{s}_{k:m}$. Provided our lower bound is nontrivial, i.e., $LB > 0$, then we can replace (5) by [1]

$$d^2 - LB \geq \|\mathbf{y}_{k:m} - R_{k:m,k:m}\mathbf{s}_{k:m}\|^2. \tag{7}$$

This is certainly a more restricted condition than (5) and so will lead to the elimination of more points from the tree. Note that (7) will not result in missing any lattice points from (2) since we have used a lower bound for the remainder of the cost in (4).

Now clearly, the tighter the lower bound LB, the more points that will be pruned from the tree. Of course, we cannot hope to find the optimal lower bound since this requires solving an integer least-squares problem (which was our original problem to begin with). Therefore in what follows we shall consider obtaining lower bounds on the integer least-squares problem

$$\min_{\mathbf{s}_{1:k-1} \in \mathcal{D} \subset \mathcal{Z}^{k-1}} \|\mathbf{z}_{1:k-1} - R_{1:k-1,1:k-1}\mathbf{s}_{1:k-1}\|^2, \tag{8}$$

where we have defined $\mathbf{z}_{1:k-1} = \mathbf{y}_{1:k-1} - R_{1:k-1,k:m}\mathbf{s}_{k:m}$. We consider two main methods of constructing lower bounds. One is to enlarge the search space (so-called relaxation). In sections 2 and 3, respectively, we relax the search space to a sphere and polytope, that include $\mathcal{D}$. In both cases the relaxation results in a convex problem that can be readily solved. The other is to use duality. In section 4, we show that the dual problem to (8) is an SDP and can also be readily solved. However, since (8) is non-convex there is a non-zero duality gap and so we obtain a strict lower bound. As expected, we will observe that the more computational effort that is devoted to obtaining a lower bound, the more pruning of the tree that will be obtained.

Finally, we should mention that for simplicity we will henceforth assume that $\mathcal{D} = \{-\frac{1}{2}, \frac{1}{2}\}^{k-1}$, which corresponds to BPSK modulation (or QPSK modulation, in the complex case). The lower bounds of sections 2 and 3 are straightforward to generalize to larger constellations. However, generalizing the method of section 4 will require some effort.

2. SPHERICAL RELAXATION

Let $LB_{sph} = \|\mathbf{z}_{1:k-1} - R_{1:k-1,1:k-1}\hat{\mathbf{s}}_{1:k-1}\|_2^2$ where $\hat{\mathbf{s}}_{1:k-1}$ is the solution of the following optimization problem

$$\begin{aligned} \min_{\mathbf{s}_{1:k-1}} \quad & \|\mathbf{z}_{1:k-1} - R_{1:k-1,1:k-1}\mathbf{s}_{1:k-1}\|_2^2 \\ \text{subject to} \quad & \sum_{i=1}^{k-1} \mathbf{s}_i^2 \leq \frac{k-1}{4}. \end{aligned} \tag{9}$$

The algorithm which solves (9) can be summarized as follows

1. Compute the SVD of $R_{1:k-1,1:k-1}$, $R_{1:k-1,1:k-1} = U\Sigma V^T, V = [\mathbf{v}_1, ..., \mathbf{v}_{k-1}]$,
2. Set $\mathbf{z}_{1:k-1} = U^T\mathbf{z}_{1:k-1}$ and $r = \text{rank}(R_{1:k-1,1:k-1})$
3. If $\sum_{i=1}^r (\frac{\mathbf{z}_i}{\sigma_i})^2 > \frac{k-1}{4}$
   Find λ^* such that $\sum_{i=1}^r (\frac{\sigma_i \mathbf{z}_i}{\sigma_i^2 + \lambda^*})^2 = \frac{k-1}{4}$.
   $\hat{\mathbf{s}}_{1:k-1} = \sum_{i=1}^r (\frac{\sigma_i \mathbf{z}_i}{\sigma_i^2 + \lambda^*})\mathbf{v}_i$

[1] $LB = 0$, of course, simply corresponds to the standard sphere decoder.

```
    else
    $\hat{\mathbf{s}}_{1:k-1} = \sum_{i=1}^{r} (\frac{\mathbf{z}_i}{\sigma_i})\mathbf{v}_i$
    end
```

The complete derivations of the previous algorithm and the proof that it indeed exactly solves (9) can be found in [3]. The SVD decomposition makes the complexity of the previous algorithm cubic. However, since the matrix R does not change while searching the tree, SVD's of its upper-left minors can be done before applying the sphere decoder. Then the operations that need to be done per each visited node in the tree are only of quadratic complexity. Also, it is worth mentioning (and it can be found in [3] as well) that although we constrained $\mathbf{s}_{1:k-1}$ to lie inside the ball of the radius $\frac{k-1}{4}$ in (9), the optimal value is achieved on the boundary of the ball, i.e. it holds that $\|\hat{\mathbf{s}}_{1:k-1}\|_2^2 = \frac{k-1}{4}$. We refer to the modification of the sphere decoder algorithm based on using $LB = LB_{sph}$ in (7), as the SPHSD-algorithm and give its performance in Figure 1. The simulation results presented in Figure 1 (and in figures later in the paper) are obtained with the radius initially chosen as suggested in [4] and updated each time we reach the bottom of the tree. For Figure 1 we have chosen $m = 30$ and so snr $= 10\log_{10}\frac{m}{4\sigma^2}$, where σ^2 is the variance of each component of the noise vector $\mathbf{w}$. As can be seen, the total flop count of the standard sphere decoder is improved at low SNRs. Also on Figure 1, the distributions of the number of visited nodes per level in the search tree are shown. It can be seen that a significant number of nodes have been pruned by the suggested modification. However, the large improvement in the pruning of the search tree, results in a moderate improvement in the flop count. The reason is that although the total number of visited points in the tree is decreased, the computational complexity per node is increased from the linear one in the standard sphere decoding (SD) algorithm to the quadratic one in the modified (SPHSD) version.

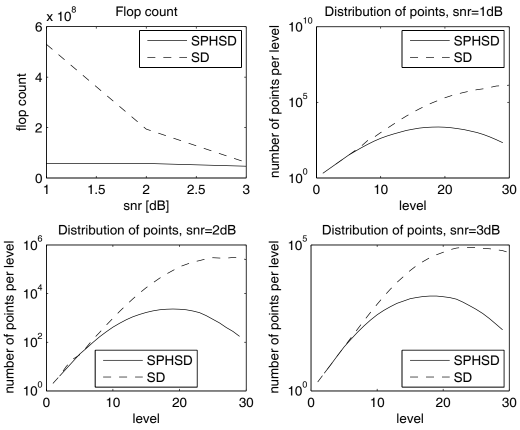

Fig. 1. Spherical relaxation for $m = 30$

3. POLYTOPE RELAXATION

Let $LB_{plt} = \|\mathbf{z}_{1:k-1} - R_{1:k-1,1:k-1}\hat{\mathbf{s}}_{1:k-1}\|_2^2$ where $\hat{\mathbf{s}}_{1:k-1}$ is the solution of the following optimization problem

$$\begin{aligned} \min_{\mathbf{s}_{1:k-1}} \quad & \|\mathbf{z}_{1:k-1} - R_{1:k-1,1:k-1}\mathbf{s}_{1:k-1}\|_2^2 \\ \text{subject to} \quad & -\frac{1}{2} \leq \mathbf{s}_i \leq \frac{1}{2}, \quad 1 \leq i \leq k-1. \quad (10) \end{aligned}$$

The optimization problem in (10) is convex and hence it is exactly solvable in polynomial time. To solve it we use the $quadprog$ function from $Matlab$ which is based on a reflective Newton method suggested for solving this type of the optimization problem [2]. We refer to the modification of the sphere decoder algorithm based on using $LB = LB_{plt}$ in (7), as the PLTSD-algorithm and give its performance in Figure 2. The simulation results presented in Figure 2 are obtained for $m = 40$ and snr $= 10\log_{10}\frac{m}{4\sigma^2}$, where σ^2 is the variance of each component of the noise vector $\mathbf{w}$. From Figure 2 it can be seen that a large number of the points in the bottom part of the tree have been pruned in comparison with the classical sphere decoder. However, the computational complexity per point is increased from linear to cubic.

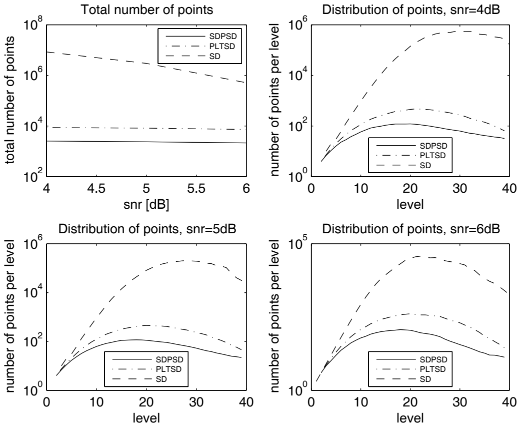

Fig. 2. Polytope relaxation and duality-based lower bound for $m = 40$

4. DUALITY-BASED LOWER BOUND

The problem given in (8) can be reformulated in the following way

$$\min_{\mathbf{s}_{1:k-1} \in \{-1,1\}^{k-1}} \begin{bmatrix} \mathbf{s}_{1:k-1} \\ 1 \end{bmatrix}^T Q \begin{bmatrix} \mathbf{s}_{1:k-1} \\ 1 \end{bmatrix}, \qquad (11)$$

where

$$Q = \begin{bmatrix} \frac{1}{4}R_{1:k-1,1:k-1}^T R_{1:k-1,1:k-1} & -\frac{1}{2}R_{1:k-1,1:k-1}^T \mathbf{z}_{1:k-1} \\ -\frac{1}{2}\mathbf{z}_{1:k-1}^T R_{1:k-1,1:k-1} & \mathbf{z}_{1:k-1}^T \mathbf{z}_{1:k-1} \end{bmatrix}.$$

Clearly, since $\begin{bmatrix} \mathbf{s}_{1:k-1} \\ 1 \end{bmatrix}^T Q \begin{bmatrix} \mathbf{s}_{1:k-1} \\ 1 \end{bmatrix} \geq \begin{bmatrix} \mathbf{s}_{1:k-1} \\ 1 \end{bmatrix}^T \Lambda \begin{bmatrix} \mathbf{s}_{1:k-1} \\ 1 \end{bmatrix}$

if $\mathbf{s}_{1:k-1} \in \{-1,1\}^{k-1}$ and $Q - \Lambda \succeq 0$, a lower bound on the optimal value of the objective function in (11) can be obtained as the solution to

$$\begin{array}{ll} \max & \mathrm{Tr}(\Lambda) \\ \text{subject to} & Q \succeq \Lambda, \quad \Lambda \quad \text{is diagonal.} \end{array} \qquad (12)$$

The optimization problem (12) is convex. In fact, it is the dual problem to (11). A variety of optimization methods for solving problem (12) can be found in [5]. We solve it exactly using interior point methods. Denote by $\widehat{\Lambda}$ the optimal value of the optimizing variable in (12). Let $LB_{sdp} = \mathrm{Tr}(\widehat{\Lambda})$. We refer to the modification of the sphere decoder algorithm based on using $LB = LB_{sdp}$ in (7) as the SDPSD-algorithm and give its performance in Figure 2. Again, the simulation results are obtained for $m = 40$ and snr $= 10\log_{10}\frac{m}{4\sigma^2}$, where σ^2 is the variance of each component of the noise vector $\mathbf{w}$. From Figure 2 it can be seen that the number of visited points (either in total or per level) in the search tree of SDPSD-algorithm is several order of magnitudes less than the corresponding one of the SD-algorithm. Although the number of the visited points in the tree is cut down by several thousands times the computational complexity per node is increased, since solving the optimization problem (12) in each node on the level $m-k+1$ has the complexity of $\mathcal{O}((k-1)^{3.5})$. However, comparing the numbers of the points in the search tree among the three suggested lower bounds, it is obvious that the duality-based lower bound prunes the tree the most. Hence, it would be very useful if we can find some efficient (at most quadratic, but preferably linear) way to compute the optimal value of the objective function in (12), or at least to find a good lower bound to it.

5. SUMMARY AND DISCUSSION

In this paper we attempted to improve the computational complexity of the sphere decoder in the regimes of low SNR and/or high dimensions, by further pruning points from the search tree. The main idea was based on computing a possible lower bound on the remainder of the cost function as we descend the search tree (the standard sphere decoder simply uses a lower bound of zero). If the sum of the current cost at a given node and the lower bound on the remaining cost from that node exceeds the cost of an already found solution then that node (and all its descendants) are pruned from the search tree. In this sense, we are essentially using a "branch and bound" technique.

We proposed three possible ways of computing lower bounds based on spherical relaxation, on polytope relaxation and on duality. All three methods significantly reduce the number of points in the search tree (in fact, by several orders of magnitude), with the duality-based method pruning the most and the sphere-relaxation method pruning the least. However, most of this gain is offset by the computations required to perform the pruning so that, all in all, we obtain modest computational savings (about a factor of 10) at low SNR. An interesting feature of our methods is that the number of points visited, and hence the computational complexity, is relatively independent of the SNR, something that is certainly not true of the standard sphere decoder. Our results suggest the need for computationally-efficient good lower bounds. In particular, a lower bound that requires only quadratic computations per node, but that is tighter than the sphere-relaxed lower bound, would be very useful. This seems worthy of further scrutiny and is currently being explored.

6. 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, pp. 463-471, April 1985.

[2] T. Coleman and Y. Li, "A reflective Newton method for minimizing a quadratic function subject to bounds on some of the variables," *SIAM Journal on Optimization*, vol. 6, no. 4, pp. 1040-1058, 1996.

[3] G. Golub and C. Van Loan, "*Matrix Computations*," John Hopkins University Press, 3rd edition, 1996.

[4] B. Hassibi, and H. Vikalo, "On the sphere decoding algorithm: part I, the expected complexity," to appear in *IEEE Trans. on Signal Processing*.

[5] S. Boyd and L. Vandenberghe, "*Convex Optimization*," Cambridge University Press, 2003.

[6] H. Artes, D. Seethaler and F. Hlawatsch, "Efficient detection algorithms for MIMO channels: A geometrical approach to approximate ML detection," *IEEE Trans. on Signal Processing*, November, 2003.

[7] H. Vikalo, B. Hassibi and T. Kailath, "Iterative decoding for MIMO channels via modified sphere decoder", *IEEE Trans. on Wireless Communications*, November, 2004.

[8] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel", *IEEE Trans. on Communications*, March, 2003.

[9] R. Gowaikar and B. Hassibi, "Efficient maximum-likelihood decoding via statistical pruning", submitted to *IEEE Trans. on Information Theory*.

[10] M. O. Damen, A. Chkeif and J.-C. Belfore, "Lattice code decoder for space-time codes"' *IEEE Comm. Let.* pp. 161-163, May 2000.