Application of Semi Definite relaxation and Variable Neighborhood Search for multiuser detection in synchronous CDMA

Lisser Abdel, *LRI, Université Paris-Sud 91405 Orsay Cedex France*
Lopez Rafael, *LRI, Université Paris-Sud 91405 Orsay Cedex France*

Keywords: Code-division multiple access, multiuser detection, semidefinite programming, variable neighborhood search.

Abstract

In this paper, a detection strategy based on variable neighborhood search (VNS) and semidefinite relaxation of the CDMA maximum likelihood (ML) is investigated. The VNS method provides a good method for solving the ML problem while keeping the integer constraints. A SDP relaxation is used as an efficient way to generate an initial solution in a limited amount of time, in particular using early termination. The SDP resolution tool used is the spectral bundle method developed by Helmberg. We show that using VNS can result in a better error rate, but at a cost of calculation time.

1. Introduction

In direct-sequence code-division multiple access systems, users are assigned a unique signature waveform, used to modulate their transmitted signals. However, due to several factors, it is not possible to insure the orthogonality between them. The most visible effect is that the signal quality is lowered (due to multiple access interferences), and that it poses difficulties when the signal is demodulated.
A conventional Rake detector is limited by these effects, and it has been shown that a better strategy is to detect the multiple users jointly. Verdú [11] proposed and analyzed the optimal multi-user detector, which is a maximum likelihood (ML) sequence detector. The detector minimizes, with respect to all possible transmitted bit sequences, the Euclidean distance between the actual received bit sequence and the hypothesized sent bit sequence. It allows one to take into account all K users when the signal arrives at the ML detector. This is possible because a sufficient statistics for the detection of one use contains the information about the others.
When using Binary Phase-Shift Keying (BPSK) to modulate and spread the signal, the optimal detection of a signal of length L with K users is equivalent to minimizing a quadratic program with the bit vector $\mathbf{d}$ constrained to the set $\{-1, 1\}^{LK}$, known as a binary quadratic program (BQP). It has been shown [12] that this problem is a non-deterministic polynomial (NP-hard), and is much too complex for practical use even with a moderate number of users.
It is possible to find upper bounds to the multi-user detection problem using meta heuristics such as Tabu Search, Simulated Annealing or, as we show in this paper, Variable Neighborhood Search. Such algorithms can give good results by avoiding the systematic test of every possible solution. However, in some situations, their performance may not be enough.
One solution is to lift the domain constraint on $\mathbf{d}$ to obtain a tractable nonlinear problem that can be solved approximately. In particular, the use of semidefinite programming has been shown to give tight bounds to the BQP program [5, 1] and therefore good detection results in CDMA systems [9], [13]. This paper is organized as follows: in the section below, the CDMA model is described. In section 3, we will develop the semidefinite relaxation and its use in the multi-user detection problem. In section 4, we will present the VNS algorithm and how it can be used with a SDP relaxation, then in section 5 numerical results will be presented. Concluding remarks will be found in section 6.

2. System Model

We consider a CDMA channel with K users, using pseudo-random signatures of length N, each of unit energy. The transmitted signal is assumed to be corrupted by Additive White Gaussian Noise (AWGN). We also consider the channel to be synchronous. We consider a sequence of arbitrary length L. At any given instant, let $\mathbf{d} \in \mathcal{D}^K$ be the vector of symbols sent on the channel, and $\mathcal{D}$ the set of possible symbols. For more details, see P.H. Tan and L.K. Rasmussen [9].

Using vector notation, the LK Matched Filter (MF) outputs can be written as [12]:

$$\mathbf{y} = \mathbf{RCd} + \mathbf{n} \tag{1}$$

Where $\mathbf{C}$ is the matrix of the channel coefficients of the users, $\mathbf{R}$ is the $KL \times KL$ matrix of cross correlations, $\mathbf{n}$ is the gaussian noise sequence vector, and has zero mean and autocorrelation matrix (with $\sigma^2 = N_0/2$):

$$E\left[\mathbf{n}\mathbf{n}^H\right] = \sigma^2 \mathbf{R}$$

The optimum ML detector chooses the hypothesis $\hat{\mathbf{d}}$ given the MF output, and assumes perfect knowledge of $\mathbf{C}$ and of $\mathbf{R}$:

$$\hat{\mathbf{d}} = \arg\max_{\mathbf{d} \in \mathcal{D}} p(\mathbf{y} \mid \mathbf{d}).$$

Since we are working with an AWGN channel, the log-likelihood function based on $p(\mathbf{y} \mid \mathbf{d})$ may be written:

$$F(\mathbf{d}) = 2\text{Re}\{\mathbf{y}^H \mathbf{Cd}\} - \mathbf{d}^H \mathbf{C}^H \mathbf{RCd}.$$

The constrained ML problem associated with the negative log-likelihood function is described as:

$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d} \in \mathcal{D}^{KL}} \mathbf{d}^H \mathbf{C}^H \mathbf{RCd} - 2\text{Re}\{\mathbf{y}^H \mathbf{Cd}\} \tag{2}$$

Solving problem (2) requires a search over the $\mathcal{D}^{KL}$ possible combinations of the components of $\mathbf{d}$. Even when considering a single symbol interval (synchronous channel), we still have to search over $\mathcal{D}^K$ possibilities. It is clear that computational complexity grows exponentially with the number of users. Verdú showed in [10] that the complexity, however, does not necessarily grow exponentially with the block length L.

In single path AWGN channels, it is sufficient to consider the signal received during one signal interval. When the channel coefficients are real valued and the symbols $\mathbf{d}$ are binary ($\mathcal{D} \in \{-1, 1\}$), the optimization problem (2) can be rewritten as:

$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d} \in \{\pm 1\}^K} \mathbf{d}^T \mathbf{C}^T \mathbf{RCd} - 2\mathbf{y}^T \mathbf{Cd} \tag{3}$$

3. Semidefinite programming for multi-user detection

We will use the real channel - binary symbol formulation, with $\mathbf{Q} = \mathbf{C}^T\mathbf{RC}$, $\mathbf{c} = \mathbf{Cy}$ and $\mathbf{u} = \hat{\mathbf{d}}$. We get:

$$\begin{aligned} \mathbf{u} &= \arg\min_{\mathbf{u}} \mathbf{u}^T \mathbf{Qu} - 2\mathbf{c}^T \mathbf{u} \\ &\text{s.t. } \mathbf{u} \in \{-1, 1\}^{n-1} \end{aligned} \tag{4}$$

with $K = n - 1$. The following results are true for any matrix $\mathbf{Q} = \mathbf{Q}^T$ [5]. By adding a dummy variable u_n, we may reformulate the preceding program as:

$$[\mathbf{u}^*, u_n^*] = \arg\min_{[\mathbf{u}, u_n]} \begin{bmatrix} \mathbf{u}^T & u_n \end{bmatrix} \begin{bmatrix} \mathbf{Q} & -\mathbf{c} \\ -\mathbf{c}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ u_n \end{bmatrix}$$
$$\text{s.t.} \begin{bmatrix} \mathbf{u} \\ u_n \end{bmatrix} \in \{-1, 1\}^N, \quad u_n = 1.$$

Given that the cost function is symmetrical, we do not have to maintain explicitly that $u_n = 1$, we then have:

$$\mathbf{L} = \begin{bmatrix} \mathbf{Q} & -\mathbf{c} \\ -\mathbf{c}^T & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} \mathbf{u}^T & u_n \end{bmatrix}^T$$

The problem (4) becomes:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \mathbf{x}^T \mathbf{L} \mathbf{x} \quad \text{s.t. } \mathbf{x} \in \{-1, 1\}^N. \tag{5}$$

A semidefinite relaxation of this program uses the rewriting: $\mathbf{x}^T\mathbf{L}\mathbf{x} = \text{tr}\{\mathbf{L}\mathbf{x}\mathbf{x}^T\}$. For all $\mathbf{x} \in \{-1, 1\}^n$, the matrix $\mathbf{x}\mathbf{x}^T$ is positive semidefinite, its diagonal elements are equal to 1, and is a rank-1 matrix. Let $\mathbf{X} = \mathbf{x}\mathbf{x}^T$, such as $\mathbf{X}$ satisfies the properties we just listed. This may be rewritten as:

$$\mathbf{X} = \begin{bmatrix} \mathbf{u}\mathbf{u}^T & \mathbf{u} \\ \mathbf{u}^T & 1 \end{bmatrix}$$

The problem (5) becomes:

$$\begin{aligned} &\mathbf{X}_1^* = \arg\min_{\mathbf{X}} \text{tr}\{\mathbf{L}\mathbf{X}\} \\ &\text{s.t.} \quad \text{diag}(\mathbf{X}) = \mathbf{e}_n, \quad \text{rank}(\mathbf{X}) = 1, \quad \mathbf{X} \succeq 0. \end{aligned} \tag{6}$$

where $\text{diag}(\mathbf{X})$ is the vector of diagonal elements of $\mathbf{X}$ and $\mathbf{e}_n$ the all ones vector of length n. $\mathbf{X} \succeq 0$ means that $\mathbf{X}$ is positive semidefinite. The subscript of $\mathbf{X}_1^*$ refers to the rank one constraint. The equivalence between (5) and (6) is shown in general in Lemma 3.1 of [7]. If we drop the rank one constraint, we get the following semidefinite program:

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \text{tr}\{\mathbf{L}\mathbf{X}\} \quad \text{s.t. diag}(\mathbf{X}) = \mathbf{e}_n, \quad \mathbf{X} \succeq 0, \tag{7}$$

with $\text{diag}(\mathbf{X})$ the vector of diagonal elements of $\mathbf{X}$.

We use the program SBmethod, developed by Helmberg [4] to solve the problem, with an early termination criterion, given by Jaldèn's method in [6], followed by the approximation method of Ma and al. [13]. We will refer to the algorithm as early SDP, which we will note eSDP.

4. Variable Neighborhood Search

Variable Neighborhood Search (VNS) is a recently developed meta heuristic [2, 3, 8] that uses the idea of neighborhood change during the descent toward local optima and to get out of the valleys that contain them. We define our neighbor structure $N_k(\mathbf{x})$ as the vectors $\mathbf{x}'$ at a distance k from $\mathbf{x}$. The distance is the number of bits differing between vectors $\mathbf{x}$ and $\mathbf{x}'$.

VNS is a meta heuristic that allows us to find an upper bound to the function we minimize. By comparing that bound with the one found by the SDP method, we can see whether or not we have a good approximation of the optimal solution. VNS uses the following observations:
Obs 1: A local minimum for a given neighborhood structure is not necessarily one for another.
Obs 2: A global minimum is a local minimum for every possible neighborhood structure.
Obs 3: In numerous problems, local minima in one or more neighborhood structures are close to each other.

4..1 Basic VNS algorithm

Initialization. We pick a set of neighborhood structures N_k, with $k = 1, \ldots, k_{max}$, that we will use during the shaking phase, a set of neighborhood structures N_l with $l = 1, \ldots l_{max}$ used during the local search. In our case, we let $k_{max} = 1$ and $l_{max} = K$, and our initial solution is a randomly generated vector.

Repeat the following sequence:

(1) $k \leftarrow 1$

(2) *Repeat* the following steps:

(a) *Shaking*: we randomly generate a vector $\mathbf{x}'$ in the k-th neighborhood of $\mathbf{x}$ ($\mathbf{x}' \in N_k(\mathbf{x})$).

(b) *Local search*:

(b1) $l \leftarrow 1$

(b2) *Repeat* the following steps:
- Exploration of the neighborhood: Find the best neighbor $\mathbf{x}''$ of $\mathbf{x}'$ in $N_l(\mathbf{x}')$;
- Move or not: If $f(\mathbf{x}'') < f(\mathbf{x}')$ then $\mathbf{x}' \leftarrow \mathbf{x}''$ and $l \leftarrow 1$; else $l \leftarrow l+1$ (here $f(\mathbf{u}) = \mathbf{u}^T\mathbf{Q}\mathbf{u} - 2\mathbf{c}^T\mathbf{u}$; c.f. (4));

Until $l = l_{max}$:

(c) *Move or not*: If the local optimum $\mathbf{x}''$ is better than the original vector $\mathbf{x}$, then we move there ($\mathbf{x} \leftarrow \mathbf{x}''$), and the search restarts in N_1 ($k \leftarrow 1$); else we increase k ($k \leftarrow k+1$).

Until $k = k_{max}$:

Until the stopping condition is met (in our case, a number of iterations N_{iter}):

4..2 VNS + SDP algorithm

One of the steps of the VNS algorithm (the choice of the initial solution) can be generated randomly, which may in some cases allow fast convergence toward the solution, but which may also generate a waste of time if the initial solution is in a valley far from the optimal solution. The consequence is that the calculation time is subject to important variations. A solution to that problem is to pick an initial solution x_0 using a different method. Here, we used the SDP method to find an approximate solution, which we then round off (simple rounding) from where the VNS algorithm will start. It should be noted that we are not doing a full SDP calculation, but that we limit the execution time (using the method given by J. Jaldèn in [6] to find the precision termination criterion). The resulting algorithm is almost the same as the one above, with the difference that the initial solution is not randomly generated, but is instead obtained from an SDP relaxation. An interesting effect is that it lowers the numbers of iterations N_{iter} necessary to converge toward an optimal solution. We will note this algorithm VNS+.

5. Simulations

The measurement of interest here is the Bit Error Rate (BER), which, from the user's point of view, is the main aspect. It allows us to calculate the performance of our detection method. The BER is defined as the number of errors over the number of transmitted bits ratio. Both the eSDP and VNS+ algorithm rely on the early termination method of J. Jaldèn [6], which allows to make comparisons between both regarding their capacity to find the best solution not only on the exact same sequences, but on the same SDP solution (the one returned by SBmethod) as well. We choose to present here the BER and signal to noise ratio (measured in dB, noted SNR/dB) relation curve, for two reasons: it allows us to see how the quality of the signal progresses with the signal strength. Moreover, it is a widely used performance measurement which allows comparisons with other methods, such as the one proposed by P.H. Tan and L.K. Rasmussen [9] or by Ma et al. [13].
We studied three situations: the first two are light load cases (Fig. 1), with a number of users K equal to 36 or 48. The third is a heavy load case (Fig. 2), with the number of users equal to 60. All three cases use a signature of length $N = 64$. In the first case ($K = 36$), we can see that both algorithms perform similarly, with the VNS+ algorithm giving slightly better results. In the second case ($K = 48$), we can see that the gap between the two methods starts to widen, while the eSDP algorithm gives good results, the VNS+ outperforms it. In the heavy load case ($K = 60$), we can see that the eSDP algorithm is clearly outperformed by the VNS+.

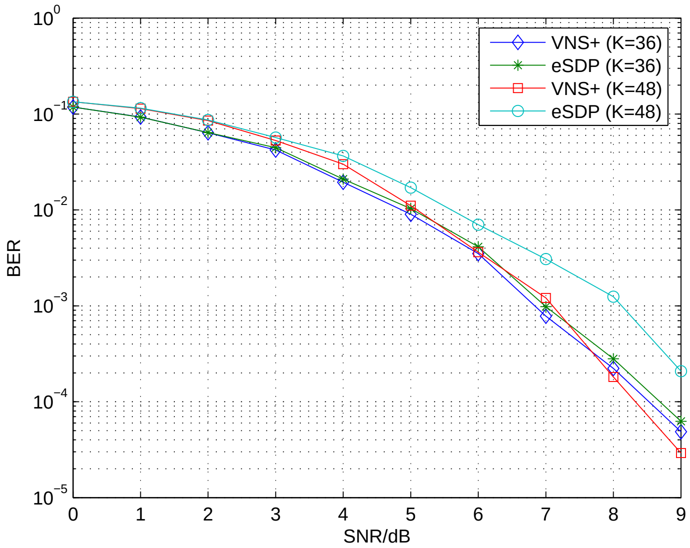

Figure 1: Average bit error rate in the light load cases ($N = 64$)

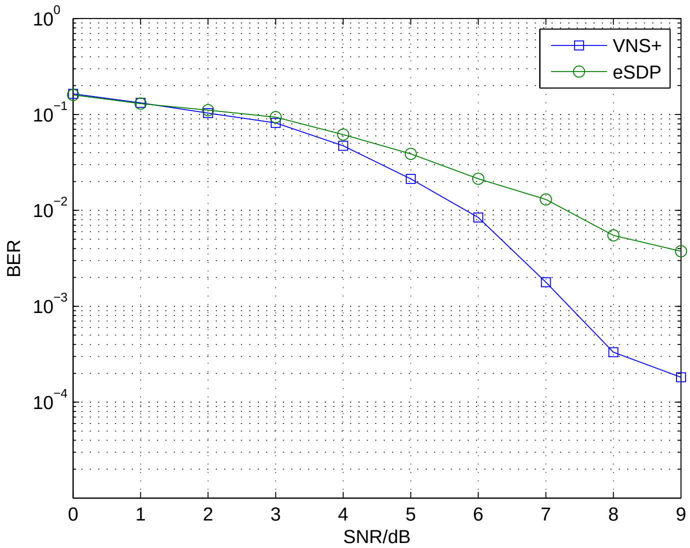

Figure 2: Average bit error rate in the heavy load case ($K = 60$, $N = 64$)

The complexity performance, on the other hand, is completely opposite. Indeed, if the VNS+ gives better results, it comes with a complexity cost. For $K = 10$, VNS+ takes 3.4 times more time to end than eSDP. The more we raise the number of users, the wider the gap becomes. For $K = 48$, the ratio of the execution times is 23.8, and for $K = 60$, VNS+ is 32.5 times slower than eSDP. We believe this ratio can be lowered with finer tuning of the VNS algorithm, in particular to avoid recalculations of the log-likelihood in the local search phase.

6. Conclusion

In this paper, we have introduced a semidefinite relaxation of the optimal ML problem coupled to a variable neighborhood search method as a way to obtain better BER performances. The partial result of the semidefinite relaxation problem with early termination was fed to the VNS algorithm as a way to improve its performance in heavily loaded cases. The numerical examples presented show that our method can yield better BER results, at the cost of increased computation cost.

References

[1] M. X. Goemans and D. P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. Assoc. Comput. Mach.*, 42:1115–1145, 1995.

[2] P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. *European Journal of Operations Research*, 130:449–467, 2001. Invited Review.

[3] P. Hansen and N. Mladenović. Variable neighbourhood search. In Fred Glover and Gary A. Kochenberger, editors, *Handbook of Metaheuristics*, chapter 6. Kluwer, 2003.

[4] C. Helmberg. SBmethod – A C++ implementation of the spectral bundle method. Technical report, Takustraße 7, 14195 Berlin, Germany, 2000.

[5] C. Helmberg and F. Rendl. Solving quadratic (0,1)- problems by semidefinite programs and cutting planes. *Mathematical Programming*, 82:291–315, 1998.

[6] Joakim Jaldén and Björn Ottersten. Channel dependent termination of the semidefinite relaxation detector. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, January 2006.

[7] M. Laurent and S. Poljak. On a positive semidefinite relaxation of the cut polytope. *Linear Algebra and its Applications*, 223/224(1–3):439–461, 1995.

[8] N. Mladenović and P. Hansen. Variable neighborhood search. *Comput. Oper. Res.*, 24(11):1097–1100, 1997.

[9] Peng Hui Tan and Lars K. Rasmussen. The application of semidefinite programming for detection in CDMA. *IEEE Journal on Selected Areas in Communications*, 19(8):1442–1449, 2001.

[10] Sergio Verdú. Minimum probability of error for asynchronous gaussian multiple-access channels. *IEEE Transactions on Information Theory*, 32(1):85–96, 1986.

[11] Sergio Verdú. Multiple-access channels with point-process observations: Optimum demodulation. *IEEE Transactions on Information Theory*, 32(5):642–651, 1986.

[12] Sergio Verdú. *Multiuser Detection*. Cambridge University Press, New York, NY, USA, 1998.

[13] Ma W-K, Davidson TN, Wong KM, Luo Z-Q, and Ching P-C. Quasi-maximum-likelihood multiuser detection using semidefinite relaxation with application to synchronous cdma. *IEEE Transactions on Signal Processing*, 50(4):912–922, 2002.