Iterative algorithms for SDP relaxation associated with MIMO ML detection problem

Slim CHABBOUH †, Philippe CIBLAT † and Jean-Claude BELFIORE †

† Département COMELEC, Ecole Nationale Supérieure des Télécommunications (ENST), Paris, France
46 rue Barrault 75013 Paris
E-mail: {chabbouh,ciblat,belfiore}@enst.fr

Abstract

In the context of MIMO maximum-likelihood detection, efficient iterative algorithms are proposed to solve the semidefinite programming relaxation based on a reduced rank non linear formulation. The convergence of the algorithms is studied theoretically and experimentally. The influence of various parameters on the performance is also analysed. The overall complexity is evaluated and some implementation issues are pointed out.

1. Introduction

The use of multiple antennas (MIMO systems) to combat fading effects in mobile communications has attracted much attention in the past few years. Information theory, via capacity considerations, predicts exceptionnal behaviour for MIMO systems compared to single antenna systems. Therefore, developing relevant detection algorithms suited to MIMO systems is a crucial task. The sphere decoding algorithm, known to be an optimal algorithm in the maximum-likelihood (ML) sense, was first introduced by Viterbo *et al.* [1] and was modifed to reduce its complexity in [2]. Sub-optimal algorithms developed for the multiuser detection can be adapted to the problem at hand. The most important class of such algorithms relies on the semidefinite relaxation of the ML problem [3, 4, 5]. Semidefinite relaxation principle is an active research field in the mathematical community and various published algorithms can be adapted to our context [6]. In this paper, two efficient algorithms which solve the semidefinite relaxation of the MIMO ML detection problem are proposed. This paper is organized as follows : the system model, the maximum-likelihood detection problem and the semidefinite relaxation are introduced in Section II. The novel algorithms and associated theoretical convergence results are derived in Section III. In Section IV, the performance is analysed by means of extensive simulations and complexity characteristics are presented. Section V is devoted to conclusions and perspectives.

Throughout the paper, bold symbols denote vectors, capital bold symbols denote matrices. Let $\boldsymbol{M} \in \mathbb{R}^{k,n}$ be a matrix. The $(i,j)^{th}$ entry of $\boldsymbol{M}$ is denoted by M_{ij} and its i^{th} column by $\boldsymbol{m}_i$. Let $\boldsymbol{x}$ be a vector in $\mathbb{R}^{n\times r}$, x_i is the i^{th} component of $\boldsymbol{x}$ and $\boldsymbol{x}_i$ or $(\boldsymbol{x})_i$ is the i^{th} sub-vector of $\boldsymbol{x}$ of size r. Thus, we can write $\boldsymbol{x}$ as $\boldsymbol{x} = [\boldsymbol{x}_1^t, ..., \boldsymbol{x}_n^t]^t$ where the superscript $(.)^t$ stands for transposition. If $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_n]$ is a matrix, then $\text{vec}(\boldsymbol{X}) = [\boldsymbol{x}_1^t, ..., \boldsymbol{x}_n^t]^t$ is a vector. The standard Kronecker product of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is denoted by $\boldsymbol{C} = \boldsymbol{A} \otimes \boldsymbol{B}$. $\|\boldsymbol{x}\|$ is the classical euclidean norm of $\boldsymbol{x}$. The largest eigenvalue of a matrix $\boldsymbol{A}$ is $\lambda_1(\boldsymbol{A})$ and its spectral norm is $\|\boldsymbol{A}\| = \sqrt{|\lambda_1(\boldsymbol{A})|}$. Finally, $\boldsymbol{I}_n$ is the identity matrix of size n.

2. System model and ML detection problem

The considered system model is described by

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{b} \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^m$ is the baseband received vector, $\boldsymbol{H} \in \mathbb{R}^{m,n}$ is the channel matrix, $\boldsymbol{x} \in \{-1, 1\}^n$ is the information vector also called information frame, and $\boldsymbol{b}$ is a real-valued Gaussian noise vector with zero mean and covariance matrix $N_0 I_m$ (m and n are positive integers $n \leq m$). This general model may represent several system among which MIMO communications and CDMA multiuser communications. Note that only binary communications are considered. An extension to the non-binary case is still under investigation. The case of complex modulations (QPSK) can be reduced to the model of (1) by separating the real and imaginary part of the signals as done in [7].

It is well known that the maximum likelihood detector associated with model (1) can be written via a binary quadratic programming problem as follows :

$$\text{BQP:} \qquad \arg \min_{\boldsymbol{x}\in\{-1,1\}^n} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2 .$$

As shown in [3], the previous problem BQP can be reformulated :

$$\arg \min_{\boldsymbol{x}\in\{-1,1\}^{n+1},\boldsymbol{x}(1)=1} \frac{1}{2}\boldsymbol{x}^t\boldsymbol{Q}\boldsymbol{x}$$

where

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{y}^t\boldsymbol{y} & -\boldsymbol{y}^t\boldsymbol{H} \\ -\boldsymbol{H}^t\boldsymbol{y} & \boldsymbol{H}^t\boldsymbol{H} \end{pmatrix}.$$

For a generic semidefinite positive matrix $\boldsymbol{Q}$, BQP cannot be solved in polynomial time. To overcome this difficulty, eq. (2) is rewritten by using $\boldsymbol{x}^t\boldsymbol{Q}\boldsymbol{x} = \text{Tr}(\boldsymbol{x}\boldsymbol{x}^t\boldsymbol{Q}) = \text{Tr}(\boldsymbol{X}\boldsymbol{Q})$ with $\boldsymbol{X} \geq 0$, $\boldsymbol{X}_{i,i} = 1$, and $\text{rank}(\boldsymbol{X}) = 1$. Dropping the constraint $\text{rank}(\boldsymbol{X}) = 1$ leads to the so-called semidefinite relaxation (SDP) [6].

$$\text{SDP} : \arg \min_{\substack{\boldsymbol{X}\in\mathcal{S}_+^{n+1} \\ \forall i\in\{1,n+1\}\ X_{ii}=1}} \frac{1}{2}\text{Tr}\left(\boldsymbol{X}\boldsymbol{Q}\right)$$

Obviously, SDP gives rise to a sub-optimal detector but is a convex linear program. Many algorithms have been proposed in the litterature to solve this formulation [6]. Some of these have been adapted to solve the CDMA multiuser detection problem [3, 4] but the complexity of the resulting algorithms is prohibitive. Since the constraint of semidefinite positivity is difficult to handle, Burer [8] proposed the change of variable $\boldsymbol{X} \rightarrow \boldsymbol{V}$ where $\boldsymbol{V}$ is a square root of $\boldsymbol{X}$ i.e. $\boldsymbol{X} = \boldsymbol{V}^t\boldsymbol{V}$. The drawback of this approach is that the number of variables (n^2) in the optimisation is prohibitive. Thanks to [9], Burer proposed to add the following constraint : $\text{rank}(\boldsymbol{X}) \leq r$ where r is a "small" integer. To force $\boldsymbol{V}$ to satisfy $\text{rank}(\boldsymbol{X}) \leq r$, only its r main rows are kept. Thus, $\boldsymbol{V}$ belongs to $\mathbb{R}^{r,n+1}$ instead of $\mathbb{R}^{n+1,n+1}$. Finally, the following non linear formulation of SDP is obtained :

$$\text{NLSDPr} : \min_{\substack{\boldsymbol{V}=[\boldsymbol{v}_1,...,\boldsymbol{v}_{n+1}] \\ \boldsymbol{v}_i\in\mathbb{R}^r,\|\boldsymbol{v}_i\|^2=1}} \frac{1}{2}\text{Tr}\left(\boldsymbol{Q}\boldsymbol{V}^t\boldsymbol{V}\right)$$

Note that the change of variables transforms the convex program SDP into a non-convex one. An algorithm has been proposed for solving NLSDPr in the context of asynchronous CDMA based on the coordinate ascent method [5]. We propose here a different approach, leading to a less complex algorithm. We adapt the projected gradient algorithm, a well known algorithm solving convex constrained programs [10], to our nonconvex optimisation problem. First of all, by vectorizing the matrix $\boldsymbol{V}$, we get a new but equivalent formulation as follows:

$$\text{NLSDPr} : \min_{v\in S} F\left(\boldsymbol{v}\right) = \frac{1}{2}\boldsymbol{v}^t\tilde{\boldsymbol{Q}}\boldsymbol{v}$$

with $\boldsymbol{v} = \text{vec}\left(\boldsymbol{V}\right)$, $\tilde{\boldsymbol{Q}} = \boldsymbol{Q} \otimes \boldsymbol{I}_r$ and $S = \{\boldsymbol{v} \in \mathbb{R}^{(n+1)r} |\ \boldsymbol{v} = \left[\boldsymbol{v}_1^t, ..., \boldsymbol{v}_{n+1}^t\right]^t, (\forall i \in \{1, n+1\})\ \boldsymbol{v}_i \in \mathbb{R}^r, \text{ and } \|\boldsymbol{v}_i\| = 1\}$.

The original formulation above highlights the quadratic nature of the optimization problem with non-convex constraints. In order to solve the constrained optimisation problems, the standard unconstrained gradient algorithm can be modified to take into account the constraints. The modified gradient algorithm is called the *projected* gradient algorithm and its iteration is defined as follows [10]:

$$\boldsymbol{v}^{k+1} = \left[\boldsymbol{v}^k - \gamma\boldsymbol{M}^{-1}\boldsymbol{\nabla}F\left(\boldsymbol{v}^k\right)\right]_P^+ \tag{2}$$

In our context, $\boldsymbol{\nabla}F\left(\boldsymbol{v}\right) = \tilde{\boldsymbol{Q}}\boldsymbol{v}$ and $\boldsymbol{\nabla}^2F\left(\boldsymbol{v}\right) = \tilde{\boldsymbol{Q}}$. The matrix $\boldsymbol{M}$ is usually chosen as the diagonal of the Hessian matrix, i.e., here $\boldsymbol{M} = \text{diag}\left(\tilde{\boldsymbol{Q}}\right) = \text{diag}\left(\boldsymbol{Q}\right) \otimes \boldsymbol{I}_r$. The parameter γ ensures the convergence of the algorithm. Finally, $[.]_P^+$ is a projection onto S which is defined below.

The iteration is divided into two steps : the first one is the standard gradient iteration step and the second one consists of a projection onto the constraint set S. For quadratic programming, these algorithms are well studied if S is convex. Unfortunately, in our case, S is nonconvex : no convergence result can be found in the literature. However, we show that these algorithms keep some of their theoretical convergence properties despite the non-convexity.

The projection $[.]_P^+$ *must be carefully chosen* for the algorithm to converge to a local minimum of the function F [10]. A classical choice is the projection induced by the norm $\|\boldsymbol{x}\|_M = \sqrt{\boldsymbol{x}^t\boldsymbol{M}\boldsymbol{x}}$, defined by $[\boldsymbol{v}]_M^+ = \arg\min_{\|\boldsymbol{y}\|\in S} \|\boldsymbol{y} - \boldsymbol{v}\|_M^2$. Some properties of this projection are listed below.

Proposition 2.1 *For $\boldsymbol{M} = diag\left(\boldsymbol{Q}\right) \otimes \boldsymbol{I}_r$, the projection $[\boldsymbol{v}]_M^+$ onto S of any $\boldsymbol{v} \in \mathbb{R}^{(n+1)r}$ is well defined if $(\forall i \in \{1, n+1\})\ \boldsymbol{v}_i \neq 0$ and the projected vector is :*

$$[\boldsymbol{v}]_M^+ = \left[\left(\left[\boldsymbol{v}_1\right]^+\right)^t, ..., \left(\left[\boldsymbol{v}_{n+1}\right]^+\right)^t\right]^t$$

where $\left[\boldsymbol{v}_i\right]^+ = \arg\min_{\|\boldsymbol{y}\|=1} \|\boldsymbol{y} - \boldsymbol{v}_i\|^2 = \frac{\boldsymbol{v}_i}{\|\boldsymbol{v}_i\|}$.

Projecting a vector $\boldsymbol{v}$ is done by projecting each block $\boldsymbol{v}_i$ onto the unit sphere. Because the projection is not defined for the vectors $\boldsymbol{v} \in \mathbb{R}^{(n+1)r}$ such that $\boldsymbol{v}_i = 0$ for some i, we must modify eq. (2).

3. Iterative algorithms for the NLSDPr problem

The two proposed iterative algorithms are based on the modification/adaptation of the iteration step of the algorithm introduced in (2) for the projected gradient algorithm: the first one refers to the so-called Jacobi iteration and the second one relies on the so-called Gauss-Seidel iteration.

3.1. Jacobi iterations of the projected gradient

Here, unlike eq. (2), we define a modified projection taking into account vectors having null sub-blocks. The algorithm first computes $\tilde{\boldsymbol{v}}^{k+1} = \boldsymbol{v}^k - \gamma \boldsymbol{M}^{-1} \boldsymbol{\nabla} F\left(\boldsymbol{v}^k\right)$, then a modified projection along M onto S is performed which gives $\boldsymbol{v}^{k+1}$ as follows : if the sub-block $\tilde{\boldsymbol{v}}_i^{k+1} = 0$ then $\boldsymbol{v}_i^{k+1} = \boldsymbol{v}_i^k$ else $\boldsymbol{v}_i^{k+1} = \frac{\tilde{\boldsymbol{v}}_i^{k+1}}{\left\|\tilde{\boldsymbol{v}}_i^{k+1}\right\|}$ according to proposition 2.1. The overall mapping defined over S by $\boldsymbol{v}^k \rightarrow \boldsymbol{v}^{k+1}$ is easily shown to be continuous over S. We call this algorithm the Jacobi iterations of the Gradient Projected Algorithm (J-GPA) and summarize its operation in the flowchart Algorithm 1:

$k = -1$;
$\boldsymbol{v}^0 = \left[\left(\boldsymbol{v}_1^0\right)^t, ..., \left(\boldsymbol{v}_{n+1}^0\right)^t\right]^t$, $\boldsymbol{v}_i^0 \in \mathbb{R}^r, \left\|\boldsymbol{v}_i^0\right\| = 1$;
repeat
  $k \leftarrow k + 1$;
  $\tilde{\boldsymbol{v}}^{k+1} = \left(\boldsymbol{I}_{(n+1)\times r} - \gamma \boldsymbol{M}^{-1} \tilde{\boldsymbol{Q}}\right) \boldsymbol{v}^k$;
  if $\tilde{\boldsymbol{v}}_i^{k+1} = 0$ **then**
    $\boldsymbol{v}_i^{k+1} = \boldsymbol{v}_i^k$;
  else
    $\boldsymbol{v}_i^{k+1} = \frac{\tilde{\boldsymbol{v}}_i^{k+1}}{\left\|\tilde{\boldsymbol{v}}_i^{k+1}\right\|}$;
  end
until $\left|F\left(\boldsymbol{v}^{k+1}\right) - F\left(\boldsymbol{v}^k\right)\right| < \epsilon F\left(\boldsymbol{v}^k\right)$;

Algorithm 1: Jacobi iterations of the gradient projected algorithm

The following proposition shows that every iteration of the algorithm reduces the objective function.

Proposition 3.1 *If* $\gamma \leq \frac{\min_{i\in\{1,n+1\}} Q_{ii}}{\|\boldsymbol{Q}\|^2}$*, then the sequence* $\left\{F\left(\boldsymbol{v}^k\right)\right\}_k$ *decreases and converge.*

This proposition gives the range of theoretically admissible values for the step size γ. In practice, convergence is observed even with higher values of γ.

The sequence of vectors generated by the algorithm is bounded, so it admits limit points. Here is a proposition giving an important property of these points :

Proposition 3.2 *Any limit point* $\boldsymbol{v}^\infty$ *of the sequence* $\left\{\boldsymbol{v}^k\right\}_k$ *is stationary.*

Stationnary points are points which satisfy first order Lagrange necessary conditions for local optimality. We cannot prove that the stationary points are local minimums because the problem is non-convex but we show through simulations in section 4 that the algorithm offers good convergence behaviour in practice.

3.2. Gauss-Seidel iterations of the projected gradient

We introduce now a variant of the precedent algorithm, based on the observation that the Jacobi iteration updates the $\boldsymbol{v}_i^{k+1}$ by using only the old samples $\left\{\boldsymbol{v}_j^k\right\}_{j\in\{1,n+1\}}$. The Gauss-Seidel iteration updates $\boldsymbol{v}_i^{k+1}$ with the most recently computed $\left\{\boldsymbol{v}_j^{k+1}\right\}_{j\in\{1,i-1\}}$ instead of the $\left\{\boldsymbol{v}_j^k\right\}_{j\in\{1,i-1\}}$. Defining $\boldsymbol{z}^{k,i} = \left[\left(\boldsymbol{v}_1^{k+1}\right)^t, ..., \left(\boldsymbol{v}_{i-1}^{k+1}\right)^t, \left(\boldsymbol{v}_i^k\right)^t, ..., \left(\boldsymbol{v}_{n+1}^k\right)^t\right]^t$ and using the identity $\boldsymbol{M} = \mathrm{diag}\left(\boldsymbol{Q}\right) \otimes \boldsymbol{I}_r$, a Gauss-Seidel iteration of the gradient algorithm (GS-GPA) is :

$$(\forall i \in \{1, n+1\})\ \boldsymbol{v}_i^{k+1} = \left[\boldsymbol{v}_i^k - \gamma Q_{ii}^{-1} \left(\tilde{\boldsymbol{Q}} \boldsymbol{z}^{k,i}\right)_i\right]^+$$

This is summarized in the flowchart Algorithm 2.

$k = -1$;
$\boldsymbol{v}^0 = \left[\left(\boldsymbol{v}_1^0\right)^t, ..., \left(\boldsymbol{v}_{n+1}^0\right)^t\right]^t$, $\boldsymbol{v}_i^0 \in \mathbb{R}^r, \left\|\boldsymbol{v}_i^0\right\| = 1$;
repeat
  $k \leftarrow k + 1$;
  for $i \in [1, n+1]$ **do**
    $\tilde{\boldsymbol{v}}_i^{k+1} = \boldsymbol{v}_i^k - \gamma Q_{ii}^{-1} \left(\tilde{\boldsymbol{Q}} \boldsymbol{z}^{k,i}\right)_i$
    if $\tilde{\boldsymbol{v}}_i^{k+1} = 0$ **then**
      $\boldsymbol{v}_i^{k+1} = \boldsymbol{v}_i^k$;
    else
      $\boldsymbol{v}_i^{k+1} = \frac{\tilde{\boldsymbol{v}}_i^{k+1}}{\left\|\tilde{\boldsymbol{v}}_i^{k+1}\right\|}$;
    end
  end
until $\left|F\left(\boldsymbol{v}^{k+1}\right) - F\left(\boldsymbol{v}^k\right)\right| < \epsilon F\left(\boldsymbol{v}^k\right)$;

Algorithm 2: Gauss-Seidel iteration of the gradient projected algorithm

Convergence properties of this algorithm are mentioned in the propositions 3.3 and 3.4

Proposition 3.3 *If* $\gamma \leq 1$*, then the sequence* $\left\{F\left(\boldsymbol{v}^k\right)\right\}_k$ *decreases and converges.*

Proposition 3.4 *Any limit point* $\boldsymbol{v}^\infty$ *of the sequence* $\left\{\boldsymbol{v}^k\right\}_k$ *is stationnary.*

Notice that the proofs of all the propositions are omitted due to the lack of space. One can show that the particular case $\gamma = 1$ leads to the algorithm introduced in [5] for which no convergence proof was given.

4. Numerical analysis of proposed algorithms

4.1. Gauss-Seidel vs. Jacobi iterations

The J-GPA algorithm suffers from the classical slow convergence of gradient based algorithms whereas GS-JPA offers much faster convergence. Notice however that these algorithms exhibit the same performance in terms of error rates since they both solve exactly the same optimisation problem. Therefore we focus on the latter algorithm for the performance and complexity analysis.

4.2. Performance and influence of the algorithm parameters

The channel matrix $\boldsymbol{H}$ entries are gaussian i.i.d with unit variance and $m = n = 50$. This is a very hard instance of the detection problem and is therefore a good benchmark for comparing the performance of various detection algorithms. The parameters of GS-GPA were optimized by simulation. The minimum relaxation size incurring no performance loss is found to be $r = 3$, which reduces drastically the complexity compared to the full size relaxation case $r = n + 1$. The minimum number of iterations needed to achieve a precision of $\epsilon = 10^{-3}$ is 15 iterations with a step size of $\gamma = 1.6$. Using these parameters, no performance loss was observed when compared to the full size relaxation infinite iterations case. Figure 1 compares the performance (in terms of Frame Error Rate (FER)) of our algorithm to the optimal sphere decoder and other sub-optimal decoders such as the generalised MMSE and the non linear PIC described in [11]. The GS-GPA performs 3 dB better than NL-PIC at FER $= 10^{-3}$ but 2 dB worse than sphere decoder.

4.3. Complexity study and implementation considerations

The GS-GPA algorithm consists of matrix-vector products and scalar products for an overall complexity of $O(rn^2)$ floating point operations per iteration. If we denote K the number of iterations, the overall complexity is $O(Krn^2)$ per frame or $O(Krn)$ per information bit. Usually, $Kr \approx n$ so the complexity is $O(n^2)$ which is comparable to the complexity of the sphere decoder. However, from an implementation point of view, our algorithm presents the very interesting property that it is well suited for parallelization, while the sphere decoder is not. Compared to the algorithm of [5], the iterations of both algorithms are equally complex. However our algorithm requires half less iterations to converge.

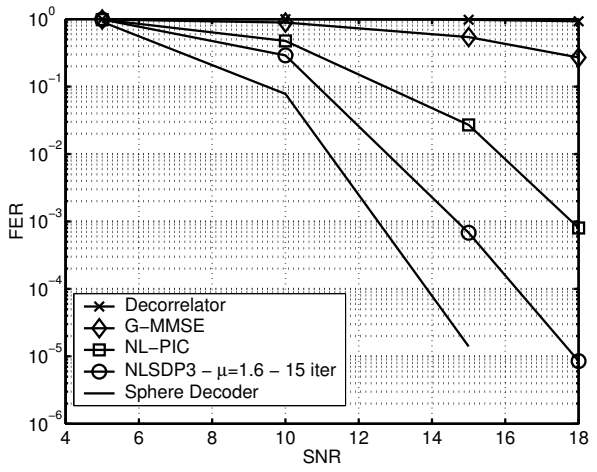

Figure 1: Performance comparison of NLSDP3 with various detectors

5. Perspectives and conclusion

We presented two sub-optimal algorithms performing close to the optimum decoder for MIMO systems, called J-GPA and GS-GPA, based on the iterative solution of a reduced rank SDP relaxation of the ML decoding. GS-GPA algorithm is less complex than other similar known algorithms in the litterature. Moreover, our algorithms presents interesting properties from an implementation point of view.

References

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

[2] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. On Information Theory*, vol. 48, no. 8, pp. 2201–2214, August 2002.

[3] P. H. Tan and L. K. Rasmussen, "The application of semidefinite programming for detection in CDMA," *IEEE J. On Selected Areas In Comm.*, vol. 19, no. 8, pp. 1442–1449, August 2001.

[4] W-K Ma, T. N. Davidson, K. M. Wong, Z-Q Luo, and P-C Ching, "Quasi-maximum-likelihood multiuser detection using semi-definite relaxation with application to synchronous CDMA," *IEEE Trans. Signal Processing*, vol. 50, no. 4, pp. 912–921, April 2002.

[5] M. Abdi, H. E. Nahas, A. Jard, and E. Moulines, "Semidefinite positive relaxation of the maximum-

likelihood criterion applied to multiuser detection in a CDMA context," *IEEE Signal Processing Letters*, vol. 9, no. 6, pp. 165–167, June 2002.

[6] R. D. C. Monteiro, "First- and second-order methods for semidefinite programming," *Math. Program.*, vol. 97, pp. 209–244, 2003.

[7] O. Damen, A. Chkeif, and J-C Belfiore, "Lattice code decoder for space-time codes," *IEEE Comm. Letters*, vol. 4, no. 5, pp. 161–163, May 2000.

[8] S. Burer and R. D. C. Monteiro, "A projected gradient algorithm for solving the maxcut SDP relaxation," *Optimization Methods and Software*, vol. 15, pp. 175–200, 2001.

[9] G. Pataki, "On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues," *Mathematics of Operations Research*, vol. 23, no. 2, pp. 339–358, 1998.

[10] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computations*, Prentice-Hall, 1989.

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