EFFICIENT COMPUTATION OF THE BINARY VECTOR THAT MAXIMIZES A RANK-DEFICIENT QUADRATIC FORM

George N. Karystinos and Athanasios P. Liavas

Department of Electronic and Computer Engineering
Technical University of Crete
Kounoupidiana, Chania, 73100, Greece
E-mail: {karystinos, liavas}@telecom.tuc.gr

ABSTRACT

The maximization of a full-rank quadratic form over a finite alphabet is NP-hard in both a worst-case sense and an average sense. Interestingly, if the rank of the form is not a function of the problem size, then it can be maximized in polynomial time. An algorithm for the efficient computation of the binary vector that maximizes a rank-deficient quadratic form is developed based on an analytic procedure. Auxiliary spherical coordinates are introduced and the multi-dimensional space is partitioned into a polynomial-size set of regions; each region corresponds to a distinct binary vector. The binary vector that maximizes the rank-deficient quadratic form is shown to belong to the polynomial-size set of candidate vectors. Thus, the size of the feasible set is efficiently reduced from exponential to polynomial.

Index Terms — Optimization.

1. INTRODUCTION

The maximization of a full-rank quadratic form over a finite alphabet is NP-hard in both a worst-case sense [1] and an average sense [2]. Interestingly, it has been recently proven that the maximization of a quadratic form with a binary vector argument is no longer NP-hard if the rank of the form is not a function of the problem size. Indeed, [3] presents an algorithm that computes the binary vector that maximizes a rank-2 quadratic form with log-linear complexity. In [4], the same idea is extended to the maximization of a rank-3 quadratic form, resulting in an algorithm that computes the optimal binary vector with log-quadratic complexity. It does so by utilizing auxiliary spherical coordinates and partitioning the three-dimensional space into a quadratic-size set of regions, where each region corresponds to a distinct binary vector. The binary vector that maximizes the rank-3 quadratic form is shown to belong to the quadratic-size set of candidate vectors. Thus, the method in [4] efficiently reduces the size of the feasible set from exponential to quadratic. From a different perspective, based on results from computational geometry (CG), it has been identified that the maximization of any reduced-rank quadratic form over the binary field can be attained in polynomial time through a variety of CG algorithms, such as the incremental algorithm for cell enumeration in arrangements [5] and the reverse search [6].[1]

In the present work, we follow an analytic procedure to generalize the approach in [3], [4] and build an efficient algorithm for the computation of the binary vector that maximizes a reduced-rank quadratic form.[2] We prove that the proposed algorithm is at least one order of magnitude faster than reverse search. In addition, the proposed method is completely parallelizable and rank-scalable. Finally, due to its nature, it can be appropriately modified to perform ML block noncoherent MPSK detection [11] (the algorithm in [8] treats only BPSK and QPSK).

2. PROBLEM STATEMENT

We consider the quadratic form

$$\mathbf{x}^T\mathbf{A}\mathbf{x} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$ is a symmetric matrix and $\mathbf{x} \in \{\pm 1\}^N$ is a binary vector argument. Since $\mathbf{A}$ is symmetric, it can be decomposed as $\mathbf{A} = \sum_{n=1}^{N} \lambda_n \mathbf{q}_n \mathbf{q}_n^T$, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$, $\|\mathbf{q}_n\| = 1$, $\mathbf{q}_n^T\mathbf{q}_k = 0$, $n \neq k$, $n, k = 1, 2, \ldots, N$, where λ_n and $\mathbf{q}_n$ are its nth eigenvalue and eigenvector, respectively.

We are interested in computing the binary vector that maximizes the quadratic form

$$\mathbf{x}_{\rm opt} \stackrel{\triangle}{=} \arg\max_{\mathbf{x}\in\{\pm 1\}^N} \mathbf{x}^T\mathbf{A}\mathbf{x}. \tag{2}$$

Without loss of generality (w.l.o.g.) we assume that $\lambda_N = 0$. Thus, $\mathbf{A}$ is semidefinite positive with rank $D \leq N-1$, i.e. $\mathbf{A} = \sum_{n=1}^{D} \lambda_n \mathbf{q}_n \mathbf{q}_n^T$, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D > 0$. Furthermore, since $\lambda_n > 0, n = 1, 2, \ldots, D$, we define the weighted principal component $\mathbf{v}_n \stackrel{\triangle}{=} \sqrt{\lambda_n}\mathbf{q}_n$, $n = 1, 2, \ldots, D$, and the corresponding $N \times D$ matrix $\mathbf{V} \stackrel{\triangle}{=} [\mathbf{v}_1\mathbf{v}_2\ldots\mathbf{v}_D]$ such that

[1] The reverse-search-based maximization over the $0/1$ field has been used for near-ML MUD [7] and ML block noncoherent detection of BPSK and QPSK signals [8] while the incremental algorithm [5] has been identified as a tool for ML block noncoherent detection of MPSK signals [9].

[2] Due to space limitation, we refer the interested reader to the journal version [10] for the proofs of our arguments.

This work was supported in part by the Sixth European Union Framework Programme under Project MC-IRG-046563-PREMIUM.

$\mathbf{A} = \mathbf{V}\mathbf{V}^T$ and

$$\mathbf{x}_{\text{opt}} = \arg \max_{\mathbf{x}\in\{\pm 1\}^N} \{\mathbf{x}^T\mathbf{V}\mathbf{V}^T\mathbf{x}\}. \quad (3)$$

Notice that $\mathbf{V}$ is full-rank and the matrices $\mathbf{A}$ and $\mathbf{V}$ have the same rank $D \leq N-1$.

3. EFFICIENT MAXIMIZATION OF A RANK-DEFICIENT QUADRATIC FORM WITH A BINARY VECTOR ARGUMENT

Since $\mathbf{x}^T\mathbf{V}\mathbf{V}^T\mathbf{x} = \|\mathbf{V}^T\mathbf{x}\|^2$, from (3) our optimization problem becomes

$$\mathbf{x}_{\text{opt}} = \arg \max_{\mathbf{x}\in\{\pm 1\}^N} \|\mathbf{V}^T\mathbf{x}\|. \quad (4)$$

We recall that $\mathbf{V}$ is a full-rank $N \times D$ matrix, $D \leq N-1$. W.l.o.g. we assume that each row of $\mathbf{V}$ has at least one nonzero element, i.e. $\mathbf{V}_{n,1:D} \neq \mathbf{0}_{1\times D}$, and $V_{n,1} \neq 0$, $n = 1, 2, \ldots, N$. To develop an efficient method for the maximization in (4), we introduce the spherical coordinates $\phi_1 \in (-\pi, \pi]$, $\phi_2, \ldots, \phi_{D-1} \in (-\frac{\pi}{2}, \frac{\pi}{2}]$, define $\boldsymbol{\phi}_{i:j} \triangleq [\phi_i, \phi_{i+1}, \ldots, \phi_j]^T$ and the hyperpolar vector

$$\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) \triangleq \begin{bmatrix} \sin\phi_1 \\ \cos\phi_1 \sin\phi_2 \\ \cos\phi_1\cos\phi_2\sin\phi_3 \\ \vdots \\ \cos\phi_1\cos\phi_2\ldots\sin\phi_{D-1} \\ \cos\phi_1\cos\phi_2\ldots\cos\phi_{D-1} \end{bmatrix}, \quad (5)$$

and turn our interest into the equivalent problem

$$\max_{\mathbf{x}\in\{\pm 1\}^N} \max_{\boldsymbol{\phi}_{1:D-1}\in(-\pi,\pi]\times(-\frac{\pi}{2},\frac{\pi}{2}]^{D-2}} \left\{\mathbf{x}^T\mathbf{V}\mathbf{c}(\boldsymbol{\phi}_{1:D-1})\right\} \quad (6)$$

which results from Cauchy-Schwartz Inequality, since, for any $\mathbf{a} \in \mathbb{R}^D$, $\mathbf{a}^T\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) \leq \|\mathbf{a}\|\|\mathbf{c}(\boldsymbol{\phi}_{1:D-1})\|$ with equality if and only if $\phi_1, \ldots, \phi_{D-1}$ are the hyperspherical coordinates of $\mathbf{a}$. We interchange the maximizations in (6) to obtain the equivalent problem

$$\max_{\boldsymbol{\phi}_{1:D-1}\in(-\pi,\pi]\times(-\frac{\pi}{2},\frac{\pi}{2}]^{D-2}} \sum_{n=1}^{N} \max_{x_n=\pm 1} \left\{x_n\mathbf{V}_{n,1:D}\mathbf{c}(\boldsymbol{\phi}_{1:D-1})\right\}. \quad (7)$$

For a given point $\boldsymbol{\phi}_{1:D-1} \in (-\pi, \pi] \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]^{D-2}$, the maximizing argument of each term of the sum in (7) depends *only* on the corresponding row of $\mathbf{V}$ and is determined by

$$\mathbf{V}_{n,1:D}\,\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) \underset{x_n=-1}{\overset{x_n=+1}{\gtrless}} 0, \qquad n = 1, \ldots, N. \quad (8)$$

Motivated by the decision rule in (8), for each $D \times 1$ vector $\mathbf{v}$ we define the *decision function* x that maps $\boldsymbol{\phi}_{1:D-1}$ to $+1$ or -1 according to

$$\begin{aligned} x(\mathbf{v}^T; \boldsymbol{\phi}_{1:D-1}) &\triangleq \arg \max_{x=\pm 1} \left\{x\mathbf{v}^T\mathbf{c}(\boldsymbol{\phi}_{1:D-1})\right\} \\ &= \text{sgn}(\mathbf{v}^T\mathbf{c}(\boldsymbol{\phi}_{1:D-1})). \end{aligned} \quad (9)$$

Then, for the given $N \times D$ matrix $\mathbf{V}$, each point in $(-\pi, \pi] \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]^{D-2}$ is mapped to a candidate binary vector[3]

$$\mathbf{x}(\mathbf{V}_{N\times D}; \boldsymbol{\phi}_{1:D-1}) \triangleq \text{sgn}(\mathbf{V}_{N\times D}\mathbf{c}(\boldsymbol{\phi}_{1:D-1})) \quad (10)$$

and the optimal vector $\mathbf{x}_{\text{opt}}$ in (4) belogs to

$$\bigcup_{\boldsymbol{\phi}_{1:D-1}\in(-\pi,\pi]\times(-\frac{\pi}{2},\frac{\pi}{2}]^{D-2}} \mathbf{x}(\mathbf{V}_{N\times D}; \boldsymbol{\phi}_{1:D-1}). \quad (11)$$

Before we proceed, we note that

$$x(\mathbf{v}^T; \phi_1 - \pi, \boldsymbol{\phi}_{2:D-1}) = -x(\mathbf{v}^T; \phi_1, \boldsymbol{\phi}_{2:D-1}) \quad (12)$$

for any $\mathbf{v} \in \mathbb{R}^D$ and $\boldsymbol{\phi}_{1:D-1} \in (-\pi, \pi] \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]^{D-2}$, implying that $\mathbf{x}(\mathbf{V}_{N\times D}; \phi_1 - \pi, \boldsymbol{\phi}_{2:D-1}) = -\mathbf{x}(\mathbf{V}_{N\times D}; \phi_1, \boldsymbol{\phi}_{2:D-1})$ for any real matrix $\mathbf{V}_{N\times D}$ and $\boldsymbol{\phi}_{1:D-1} \in (-\pi, \pi] \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]^{D-2}$. Since opposite binary vectors $\mathbf{x}$ and $-\mathbf{x}$ result in the same metric in (4), we can ignore the values of ϕ_1 in $\left(-\pi, -\frac{\pi}{2}\right] \cup \left(\frac{\pi}{2}, \pi\right]$ and rewrite the optimization problem in (7) as

$$\max_{\boldsymbol{\phi}_{1:D-1}\in\Phi^{D-1}} \sum_{n=1}^{N} \max_{x_n=\pm 1} \left\{x_n\mathbf{V}_{n,1:D}\mathbf{c}(\boldsymbol{\phi}_{1:D-1})\right\} \quad (13)$$

where $\Phi \triangleq (-\frac{\pi}{2}, \frac{\pi}{2}]$. Finally, we collect all candidate binary vectors into set

$$\mathcal{X}(\mathbf{V}_{N\times D}) \triangleq \bigcup_{\boldsymbol{\phi}_{1:D-1}\in\Phi^{D-1}} \left\{\mathbf{x}(\mathbf{V}_{N\times D}; \boldsymbol{\phi}_{1:D-1})\right\} \quad (14)$$

and observe that $\arg\max_{\mathbf{x}\in\{\pm 1\}^N} \left\{\mathbf{x}^T\mathbf{V}\mathbf{V}^T\mathbf{x}\right\} \in \mathcal{X}(\mathbf{V})$, i.e.

$$\mathbf{x}_{\text{opt}} = \arg \max_{\mathbf{x}\in\mathcal{X}(\mathbf{V})} \left\{\mathbf{x}^T\mathbf{V}\mathbf{V}^T\mathbf{x}\right\}. \quad (15)$$

In the following, we *(i)* show that $|\mathcal{X}(\mathbf{V}_{N\times D})| = \sum_{d=0}^{D-1}\binom{N-1}{d}$ and *(ii)* develop an algorithm for the construction of $\mathcal{X}(\mathbf{V}_{N\times D})$ with complexity $\mathcal{O}(N^D)$.

We begin by observing that the decision function x in (9) determines a hypersurface that partitions the $(D-1)$-dimensional hypercube Φ^{D-1} into two regions; one corresponds to $x(\mathbf{v}^T; \boldsymbol{\phi}_{1:D-1}) = +1$ and the other corresponds to $x(\mathbf{v}^T; \boldsymbol{\phi}_{1:D-1}) = -1$. Indeed, it can be shown that for any $\mathbf{v} \in \mathbb{R}^D$ with $v_1 \neq 0$ the function $\phi_1 = \tan^{-1}\left(-\frac{\mathbf{v}_{2:D}^T\mathbf{c}(\boldsymbol{\phi}_{2:D-1})}{v_1}\right)$ is equivalent to $\mathbf{v}^T\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) = 0$ and determines a hypersurface $S(\mathbf{v}^T)$ which partitions Φ^{D-1} into two regions that correspond to two opposite values $x(\mathbf{v}^T; \boldsymbol{\phi}_{1:D-1}) = \pm 1$. As a result, the $N \times D$ matrix $\mathbf{V}_{N\times D}$ is associated with N hypersurfaces $S(\mathbf{V}_{1,1:D})$, $S(\mathbf{V}_{2,1:D})$, $\ldots$, $S(\mathbf{V}_{N,1:D})$ that constitute a simple arrangement and partition the hypercube Φ^{D-1} into K cells $C_1, C_2, \ldots, C_K$

[3] When the dimensions of the $N \times D$ matrix $\mathbf{V}$ matter we denote it by $\mathbf{V}_{N\times D}$, otherwise we denote it by $\mathbf{V}$.

such that $\bigcup_{k=1}^{K} C_k = \Phi^{D-1}$, $C_k \cap C_j \neq 0$ if $k \neq j$, and each cell C_k corresponds to a *unique* $\mathbf{x}_k \in \{\pm 1\}^N$.

Let $\mathcal{I}_{D-1} \triangleq \{i_1, i_2, \ldots, i_{D-1}\} \subset \{1, 2, \ldots, N\}$ denote a subset of $D-1$ indices (that correspond to hypersurfaces) and $\boldsymbol{\phi}(\mathbf{V}_{N\times D}; \mathcal{I}_{D-1}) \in \Phi^{D-1}$ equal the vector of coordinates of the intersection of hypersurfaces $S(\mathbf{V}_{i_1,1:D}), S(\mathbf{V}_{i_2,1:D}), \ldots, S(\mathbf{V}_{i_{D-1},1:D})$. Then, $\boldsymbol{\phi}(\mathbf{V}_{N\times D}; \mathcal{I}_{D-1})$ "leads" a cell, say $C(\mathbf{V}_{N\times D}; \mathcal{I}_{D-1})$, associated with a unique vector $\mathbf{x}(\mathbf{V}_{N\times D}; \mathcal{I}_{D-1})$ in the sense that $\mathbf{x}(\mathbf{V}_{N\times D}; \boldsymbol{\phi}_{1:D-1}) = \mathbf{x}(\mathbf{V}_{N\times D}; \mathcal{I}_{D-1}) \ \forall\, \boldsymbol{\phi}_{1:D-1} \in C(\mathbf{V}_{N\times D}; \mathcal{I}_{D-1})$. We collect all such vectors into

$$J(\mathbf{V}_{N\times D}) \triangleq \bigcup_{\mathcal{I}_{D-1} \subset \{1,\ldots,N\}} \{\mathbf{x}(\mathbf{V}_{N\times D}; \mathcal{I}_{D-1})\} \tag{16}$$

and observe that $J(\mathbf{V}_{N\times D}) \subseteq \{\pm 1\}^N$ and $|J(\mathbf{V}_{N\times D})| = \binom{N}{D-1}$. In other words, $J(\mathbf{V}_{N\times D})$ contains $\binom{N}{D-1}$ binary vectors; each vector is associated with a cell in Φ^{D-1} that minimizes its ϕ_{D-1} component at a single point which constitutes the intersection of the corresponding $D-1$ hypersurfaces. We also note that there exist cells that are not associated with such a vertex and contain uncountably many points of the form $(\phi_1, \ldots, \phi_{D-2}, -\frac{\pi}{2})$. However, every such a cell can be ignored since there exists another cell that contains points of the form $(-\phi_1, \ldots, -\phi_{D-2}, \frac{\pi}{2})$, is associated with the opposite vector, and is "led" by a vertex-intersection (thus, it belongs to $J(\mathbf{V}_{N\times D})$) unless $\phi_{D-2} = \pm\frac{\pi}{2}$. Indeed, if $\phi_{D-2} = \pm\frac{\pi}{2}$ for a particular cell, then this cell "exists" for any $\phi_{D-1} \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$, implying that we can ignore ϕ_{D-1} (or, say, set it to an arbitrary value ϕ'_{D-1}), set ϕ_{D-2} to $\pm\frac{\pi}{2}$, and consider cells defined on $\Phi^{D-3} \times \left\{\pm\frac{\pi}{2}\right\} \times \left\{\phi'_{D-1}\right\}$. Finally, the cells that are defined when $\phi_{D-2} = -\frac{\pi}{2}$ are associated with vectors that are opposite to the vectors that are associated with cells defined when $\phi_{D-2} = \frac{\pi}{2}$. Therefore, we can ignore the case $\phi_{D-2} = -\frac{\pi}{2}$, set ϕ_{D-2} to $\frac{\pi}{2}$, ignore ϕ_{D-1}, and identify the cells that are determined by the *reduced-size* matrix $\mathbf{V}_{N\times(D-2)}$ over the hypercube Φ^{D-3}. Hence, $\mathcal{X}(\mathbf{V}_{N\times D}) = J(\mathbf{V}_{N\times D}) \cup \mathcal{X}(\mathbf{V}_{N\times(D-2)})$ and, by induction, $\forall d = 3, 4, \ldots, D$

$$\mathcal{X}(\mathbf{V}_{N\times d}) = J(\mathbf{V}_{N\times d}) \cup \mathcal{X}(\mathbf{V}_{N\times(d-2)}) \tag{17}$$

which implies that

$$\begin{aligned}\mathcal{X}(\mathbf{V}_{N\times D}) &= J(\mathbf{V}_{N\times D}) \cup \ldots \cup J(\mathbf{V}_{N\times(D-2\lfloor\frac{D-1}{2}\rfloor)}) \\ &= \bigcup_{d=0}^{\lfloor\frac{D-1}{2}\rfloor} J(\mathbf{V}_{N\times(D-2d)}),\end{aligned} \tag{18}$$

since $\mathcal{X}(\mathbf{V}_{N\times 1}) = J(\mathbf{V}_{N\times 1})$, $|\mathcal{X}(\mathbf{V}_{N\times 1})| = |J(\mathbf{V}_{N\times 1})| = 1$ and $\mathcal{X}(\mathbf{V}_{N\times 2}) = J(\mathbf{V}_{N\times 2})$, $|\mathcal{X}(\mathbf{V}_{N\times 2})| = |J(\mathbf{V}_{N\times 2})| = N$ [3]. As a result, the cardinality of $\mathcal{X}(\mathbf{V}_{N\times D})$ is

$$\begin{aligned}|\mathcal{X}(\mathbf{V}_{N\times D})| &= |J(\mathbf{V}_{N\times D})| + \ldots + |J(\mathbf{V}_{N\times(D-2\lfloor\frac{D-1}{2}\rfloor)})| \\ &= \binom{N}{D-1} + \ldots + \binom{N}{D-1-2\lfloor\frac{D-1}{2}\rfloor} \\ &= \sum_{d=0}^{\lfloor\frac{D-1}{2}\rfloor} \binom{N}{D-1-2d} = \sum_{d=0}^{D-1} \binom{N-1}{d}.\end{aligned} \tag{19}$$

To summarize the developments so far, we have utilized $D-1$ auxiliary spherical coordinates, partitioned the hypercube Φ^{D-1} into $\sum_{d=0}^{D-1}\binom{N-1}{d}$ cells that are associated with distinct binary vectors which constitute $\mathcal{X}(\mathbf{V}_{N\times D}) \subseteq \{\pm 1\}^N$, and proved that $\mathbf{x}_{\text{opt}} \in \mathcal{X}(\mathbf{V}_{N\times D})$. Therefore, the initial problem in (4) has been converted into numerical maximization of $||\mathbf{V}^T\mathbf{x}||$ among all vectors $\mathbf{x} \in \mathcal{X}(\mathbf{V}_{N\times D})$. Such an optimization costs $\mathcal{O}\left(\sum_{d=0}^{D-1}\binom{N-1}{d}\right) = \mathcal{O}(N^{D-1})$ comparisons upon construction of $\mathcal{X}(\mathbf{V}_{N\times D})$. An efficient algorithm for the construction of $\mathcal{X}(\mathbf{V}_{N\times D})$ follows.

Let $\mathbf{V}_{N\times D}$ be a real matrix that satisfies the assumptions made in the beginning of Section III. According to (18), the construction of $\mathcal{X}(\mathbf{V}_{N\times D})$ reduces to the parallel construction of $J(\mathbf{V}_{N\times D}), J(\mathbf{V}_{N\times(D-2)}), \ldots, J(\mathbf{V}_{N\times 2})$ if D is even and $J(\mathbf{V}_{N\times D}), J(\mathbf{V}_{N\times(D-2)}), \ldots, J(\mathbf{V}_{N\times 1})$ if D is odd. Recall that $J(\mathbf{V}_{N\times 1})$, $J(\mathbf{V}_{N\times 2})$, and $J(\mathbf{V}_{N\times 3})$ can be obtained with complexity $\mathcal{O}(N)$, $\mathcal{O}(N\log N)$, and $\mathcal{O}(N^2\log N)$, respectively [3], [4]. Therefore, it remains to describe a way to construct $J(\mathbf{V}_{N\times d})$ for any $d > 3$. Interestingly, from (16), we observe that the construction of $J(\mathbf{V}_{N\times d})$ can also be parallelized since the candidate vector $\mathbf{x}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ can be obtained *independently* for each $\mathcal{I}_{d-1} \subset \{1, 2, \ldots, N\}$. As a result, we only need to present a method for the computation of $\mathbf{x}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ $\forall\, \mathcal{I}_{d-1} \subset \{1, 2, \ldots, N\}$, $d \in \{3, 4, \ldots, N\}$.

Since the hypersurface arrangement is simple, only the $d-1$ hypersurfaces $S(\mathbf{V}_{i_1,1:d}), S(\mathbf{V}_{i_2,1:d}), \ldots, S(\mathbf{V}_{i_{d-1},1:d})$ pass through the "leading vertex" $\boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ of cell $C(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$. Therefore, if $n \in \{1, 2, \ldots, N\} - \mathcal{I}_{D-1}$, then the corresponding hypersurface $S(\mathbf{V}_{n,1:d})$ does not pass through $\boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$, implying that the sign of the corresponding binary element $x_n(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ is well-determined at the "leading vertex." On the other hand, if $n \in \mathcal{I}_{d-1}$, say $n = i_k$, then hypersurface $S(\mathbf{V}_{n,1:d})$ passes through $\boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ leading to an ambiguous decision $x(\mathbf{V}_{n,1:d}; \boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})) = \pm 1$. In such a case, ambiguity is avoided if we ignore $S(\mathbf{V}_{n,1:d})$ and consider the intersection of the remaining $d-2$ hypersurfaces at $\phi_{d-1} = \frac{\pi}{2}$. To describe how the vector of coordinates $\boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ is obtained efficiently, we recall that $\boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ represents the unique intersection of $S(\mathbf{V}_{i_1,1:d}), S(\mathbf{V}_{i_2,1:d}), \ldots, S(\mathbf{V}_{i_{d-1},1:d})$, i.e. the unique solution of

$$\mathbf{V}_{\mathcal{I}_{d-1},1:d}\mathbf{c}(\boldsymbol{\phi}_{1:d-1}) = \mathbf{0}_{(d-1)\times 1}. \tag{20}$$

The following proposition identifies the vector of interest.

Proposition 1 *Consider a full-rank $(d-1)\times d$ real matrix $\mathbf{V}$. Then, the equation*

$$\mathbf{V}\mathbf{c}(\boldsymbol{\phi}_{1:d-1}) = \mathbf{0}_{(d-1)\times 1} \tag{21}$$

has a unique solution $\boldsymbol{\phi}(\mathbf{V}) \in \Phi^{d-1}$ which consists of the hyperspherical coordinates of the zero left singular vector of $\mathbf{V}$. □

Therefore, to obtain $\phi(\mathbf{V}_{N\times d};\mathcal{I}_{d-1})$ we just need to compute the zero left singular vector of $\mathbf{V}_{\mathcal{I}_{d-1},1:d}$ and calculate its hyperspherical coordinates. In fact, since we are interested only in $\mathbf{c}(\phi)$, the latter conversion into hyperspherical coordinates in not necessary. Indeed, if $\mathbf{u}$ is the zero left singular vector of $\mathbf{V}_{\mathcal{I}_{d-1},1:d}$, then we only need to calculate $x_n = \text{sgn}(\mathbf{V}_{n,1:d}\mathbf{u})$ if $n \notin \mathcal{I}_{d-1}$ and act similarly (upon rank reduction) if $n \in \mathcal{I}_{d-1}$.

The algorithm for the construction of $\mathcal{X}(\mathbf{V}_{N\times D})$ is provided at http://www.telecom.tuc.gr/∼karystinos. The algorithm visits independently the $|\mathcal{X}(\mathbf{V}_{N\times D})| = \mathcal{O}(N^{D-1})$ intersections and computes the candidate binary vector for each intersection. The calculation of the zero left singular vector of $\mathbf{V}_{\mathcal{I}_{d-1},1:d}$ costs $\mathcal{O}(d^2)$ while the operation $\text{sgn}(\mathbf{V}_{n,1:d}\mathbf{u})$ costs $\mathcal{O}(d)$. Since $\mathbf{u}'$ is computed for each $n \in \mathcal{I}_{d-1}$, the cost of the algorithm for each combination $\mathcal{I}_{d-1}$ is $\mathcal{O}(d^2)+(N-d+1)\mathcal{O}(d)+(d-1)\left(\mathcal{O}(d^2)+\mathcal{O}(d)\right) = \mathcal{O}(d^3+Nd)$. Therefore, the overall complexity of the algorithm for the computation of $\mathcal{X}(\mathbf{V}_{N\times D})$ with fixed $D \leq N-1$ becomes $\mathcal{O}(N^{D-1})\mathcal{O}(N) = \mathcal{O}(N^D)$. We recall that the corresponding complexity of the reverse search [6] is $\mathcal{O}(N^D\text{LP}(N,D))$ where $\text{LP}(N,D)$ denotes the time to solve a linear programming (LP) optimization problem with N inequalities and D variables. Provided that the complexity of $\text{LP}(N,D)$ is linear in N [12], it turns out that the reverse search costs $\mathcal{O}(N^{D+1})$ calculations. Therefore, the proposed algorithm in this present work is at least N times faster than reverse search. In addition, the computation of the candidate vectors of $\mathcal{X}(\mathbf{V}_{N\times D})$ is performed independently from cell to cell, which means that the proposed algorithm is fully parallelizable and the memory utilization is efficiently minimized, in contrast to the incremental algorithm in [5] which is very complicated to implement due to its large memory requirement. Finally, the proposed method is rank-scalable and, due to its nature, can be appropriately modified to serve complex-domain rank-deficient quadratic form maximization [11].

As an illustration, we revisit the familiar CDMA multiuser detection problem, convert the detection rule into a maximization of a full-rank quadratic form, and approximate the form with a reduced-rank one by keeping its D strongest principal compoments. The spreading gain is $L = 16$ and $K = 10$ users transmit synchronously and with identical powers. In Fig. 1, we plot the average bit error rate (BER) as a function of the user SNR, when $D = 1,\ldots,5$. As a reference, we plot the BER of the optimal multiuser detector. We observe that a rank-4 approximation of the rank-11 quadratic form is enough for attaining practically ML performance.

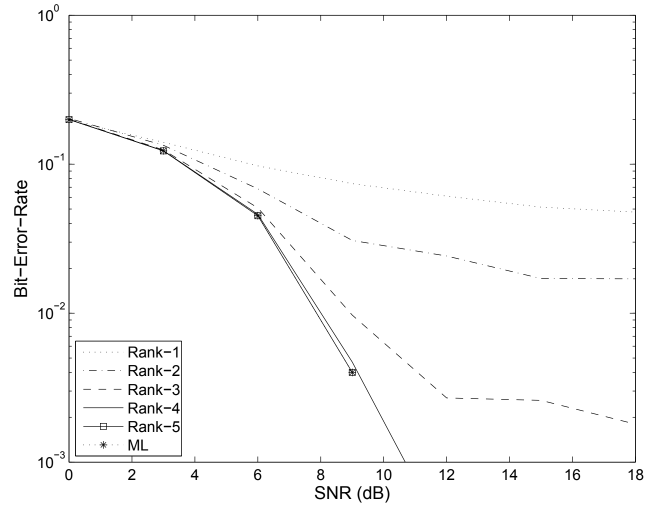

Fig. 1. BER versus SNR for reduced-rank and exact ML multiuser detection.

4. REFERENCES

[1] M. Grotschel, L. Lovsz, and A. Schriver, *Geometric Algorithms and Combinatorial Optimization*, 2nd ed. New York: Springer-Verlag, 1993.

[2] M. Ajtai, "The shortest vector problem in L is NP-hard for randomized reductions," in *Proc. 30th Ann. ACM Symp. Theory Comput.*, 1998, pp. 1019.

[3] G. N. Karystinos and D. A. Pados, "Rank-2-optimal adaptive design of binary spreading codes," *IEEE Trans. Inform. Theory*, vol. 53, pp. 3075-3080, Sept. 2007.

[4] G. N. Karystinos and A. P. Liavas, "Efficient computation of the binary vector that maximizes a rank-3 quadratic form," in *Proc. 2006 Allerton Conf. Commun., Control, and Computing*, Allerton House, Monticello, IL, Sept. 2006, pp. 1286-1291.

[5] H. Edelsbrunner, J. O'Rouke, and R. Seiel, "Constructing arrangements of lines and hyperplanes with applications," *SIAM J. Computing*, vol. 15, pp. 341-363, 1986.

[6] D. Avis and K. Fukuda, "Reverse search for enumeration," *Discrete Applied Mathematics*, vol. 65, pp. 21-46, 1996.

[7] G. Manglani and A. K. Chaturvedi, "Application of computational geometry to multiuser detection in CDMA," *IEEE Trans. Commun.*, vol. 54, pp. 204-207, Feb. 2006.

[8] V. Pauli, L. Lampe, R. Schober, and K. Fukuda, "Multiple-symbol differential detection based on combinatorial geometry," in *Proc. IEEE International Conference on Communications (ICC 2007)*, Glasgow, Scotland, June 2007, pp. 827 - 832.

[9] I. Motedayen, A. Krishnamoorthy, and A. Anastasopoulos, "Optimal joint detection/estimation in fading channels with polynomial complexity," *IEEE Trans. Inform. Theory*, vol. 53, pp. 209 - 223, Jan. 2007.

[10] G. N. Karystinos and A. P. Liavas, "Efficient computation of the binary vector that maximizes a rank-deficient quadratic form," in preparation.

[11] D. S. Papailiopoulos and G. N. Karystinos, "Efficient maximum-likelihood noncoherent MPSK detection in SIMO wireless systems," in preparation.

[12] Y. Ye, *Interior Point Algorithms: Theory & Analysis*, Wiley, 1997.