Efficient Computation of the Binary Vector That Maximizes a Rank-Deficient Quadratic Form

George N. Karystinos, *Member, IEEE*, and Athanasios P. Liavas, *Member, IEEE*

***Abstract*—The maximization of a full-rank quadratic form over the binary alphabet can be performed through exponential-complexity exhaustive search. However, if the rank of the form is not a function of the problem size, then it can be maximized in polynomial time. By introducing auxiliary spherical coordinates, we show that the rank-deficient quadratic-form maximization problem is converted into a double maximization of a linear form over a multidimensional continuous set, the multidimensional set is partitioned into a polynomial-size set of regions which are associated with distinct candidate binary vectors, and the optimal binary vector belongs to the polynomial-size set of candidate vectors. Thus, the size of the candidate set is reduced from exponential to polynomial. We also develop an algorithm that constructs the polynomial-size candidate set in polynomial time and show that it is fully parallelizable and rank-scalable. Finally, we demonstrate the efficiency of the proposed algorithm in the context of adaptive spreading code design.**

***Index Terms*—Binary sequences, code-division multiple-access (CDMA), code-division multiplexing, maximization of quadratic forms, optimization, signal waveform design.**

I. Introduction

The maximization of a positive (semi)definite quadratic form that consists of a *matrix parameter* and a *vector argument* is a common design problem in communication systems that appears at both the transmitter (signal design) and the receiver (signal processing) end. The complexity of such an optimization is determined by the characteristics of the matrix parameter (whose rank determines the rank of the quadratic form) as well as the alphabet of the vector argument. For example, if the alphabet of the vector argument is unconstrained, then the quadratic form is maximized by the maximum-eigenvalue eigenvector of the matrix parameter. However, maximization of a full-rank quadratic form over the binary[1] 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 [3]–[5].[2] Specifically, based on results from computational geometry (CG), it was identified in [5] that the maximization of any rank-deficient quadratic form over the binary field is equivalent to rank-deficient maximization over the $0/1$ field [3]. The latter can be attained in polynomial time through a variety of CG algorithms, such as the incremental algorithm for cell enumeration in arrangements [8], [9] and the reverse search [4], [10]. It should be noted that although the incremental algorithm in [8], [9] is optimal in terms of speed, it is not parallelizable and may be very complicated to implement due to its large memory requirement. On the other hand, the highly parallelizable reverse search [4], [10] is speed and memory efficient and, as a result, has been utilized for the maximization of a rank-deficient quadratic form over the $0/1$ field [3].[3]

From a different perspective, in [6], [7] the authors present an algorithm which computes with log-linear complexity the binary vector that maximizes a rank-2 quadratic form. In [13], 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 [13] reduces the size of the candidate vector set from exponential to quadratic.

In the present work, we generalize the approach in [6], [7], [13] and build an efficient algorithm for the computation of the binary vector that maximizes a rank-deficient quadratic form. Specifically, we introduce as many auxiliary spherical coordinates as the rank of the problem reduced by one and partition

Manuscript received June 07, 2008; revised March 01, 2010. Current version published June 16, 2010. This work was supported in part by the Sixth Framework Program of the European Commission under Grant MIRG-CT-2007-46563. This paper was presented in part at the 2008 IEEE International Conference on Acoustics, Speech, and Signal Processing, Las Vegas, NV, April 2008, and in part at the 2008 IEEE International Symposium on Information Theory, Toronto, ON, Canada, July 2008.

The authors are with the Department of Electronic and Computer Engineering, Technical University of Crete, Chania, 73100, Greece (e-mail: karystinos@telecom.tuc.gr; liavas@telecom.tuc.gr).

Communicated by H. Bölcskei, Associate Editor for Detection and Estimation.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2010.2048450

[1] In this work, a vector is called binary if and only if each element of it equals $+1$ or -1. Contrarily, if each element of it equals 0 or 1, then the vector is said to belong to the $0/1$ field.

[2] A straightforward example is the rank-1 quadratic form maximization problem whose optimal solution is the hard-limiter output when applied to the maximum-eigenvalue eigenvector of the matrix parameter (i.e., the norm-constrained, alphabet-unconstrained quadratic form maximization solution) [6], [7].

[3] The reverse-search-based maximization over the $0/1$ field [3] has been used for maximum-likelihood (ML) block noncoherent detection of binary and quadrature phase-shift keying signals [11] and near-ML multiuser detection [5] while the incremental algorithm [8] has been identified as a tool for ML block noncoherent detection of M-ary phase-shift keying (MPSK) signals [12].

the multidimensional space into a polynomial-size set of regions. Each region is associated with a distinct binary vector. The set of binary vectors that we obtain has the same size as the set produced by the reverse search [4]. However, in the proposed approach the set is constructed in a completely different manner resulting in time and memory savings. We prove that the proposed algorithm is fully parallelizable and rank-scalable. Finally, due to its nature, it can be appropriately modified to serve (not, necessarily, constant-modulus) complex-domain optimization problems such as, for example, ML noncoherent single-input multiple-output (SIMO) detection of arbitrary-order MPSK [14], ML noncoherent pulse-amplitude modulation (PAM) or quadrature amplitude modulation (QAM) detection, and sparse rank-deficient variance maximization (in the context of sparse principal component analysis). Work on the latter two subjects is currently in progress.

The rest of the paper is organized as follows. Section II is devoted to the problem formulation. The proposed method for the maximization of a rank-deficient quadratic form with a binary vector is described in Section III. The performance of the proposed algorithm is tested through simulations in Section IV. A few concluding remarks are drawn in Section V.

Notation: Vectors and matrices are denoted by small and capital, respectively, bold letters, that is, $\mathbf{x}$ and $\mathbf{A}$. Their elements are denoted as x_i and $A_{i,j}$, respectively. $\mathbf{A}_{i:j,k:l}$ follows a MATLAB-like notation that denotes the submatrix of $\mathbf{A}$ that consists of the ith up to jth rows and kth up to lth columns of it. When the size of the $m \times n$ matrix $\mathbf{A}$ matters we denote it by $\mathbf{A}_{m\times n}$; otherwise, we denote it by $\mathbf{A}$.

II. 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

$$\mathbf{A} = \sum_{n=1}^{N} \lambda_n \mathbf{q}_n \mathbf{q}_n^T, \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$$
$$\|\mathbf{q}_n\| = 1, \mathbf{q}_n^T\mathbf{q}_k = 0, n \neq k, n,k = 1,2,\ldots,N \tag{2}$$

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

$$\mathbf{x}_{\rm opt} \triangleq \arg\max_{\mathbf{x}\in\{\pm 1\}^N} \mathbf{x}^T\mathbf{A}\mathbf{x}. \tag{3}$$

Without loss of generality (w.l.o.g.) we assume that $\lambda_N = 0$. Indeed, if $\lambda_N \neq 0$, then $\mathbf{A}$ can be substituted by $\mathbf{A} - \lambda_N\mathbf{I}$ so that the quadratic forms $\mathbf{x}^T(\mathbf{A} - \lambda_N\mathbf{I})\mathbf{x} = \mathbf{x}^T\mathbf{A}\mathbf{x} - N\lambda_N$ and $\mathbf{x}^T\mathbf{A}\mathbf{x}$ are maximized by the same binary vector and the minimum eigenvalue of $\mathbf{A} - \lambda_N\mathbf{I}$ equals zero. Therefore, in the following, w.l.o.g. $\mathbf{A}$ is assumed positive semidefinite 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

$$\mathbf{v}_n \triangleq \sqrt{\lambda_n}\mathbf{q}_n, n = 1,2,\ldots,D \tag{4}$$

and the corresponding $N \times D$ matrix

$$\mathbf{V} \triangleq [\mathbf{v}_1\mathbf{v}_2\ldots\mathbf{v}_D] \tag{5}$$

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

$$\mathbf{x}_{\rm opt} = \arg\max_{\mathbf{x}\in\{\pm 1\}^N} \{\mathbf{x}^T\mathbf{V}\mathbf{V}^T\mathbf{x}\}. \tag{6}$$

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

If $D = N-1$, then the computation of $\mathbf{x}_{\rm opt}$ is NP-hard [1], [2] and can be implemented by exhaustive search among all elements of $\{\pm 1\}^N$ with complexity $\mathcal{O}(2^N)$ since $|\{\pm 1\}^N| = 2^N$, an approach that becomes intractable even for moderate values of N.[4] However, if D is not a function of N, then lower-complexity solutions are available for the maximization problem in (6). For example, if $D = 1$, i.e., $\mathbf{V} = \sqrt{\lambda_1}\mathbf{q}_1$, then $\mathbf{x}_{\rm opt}$ in (6) can be derived by inspection[5] and is given by $\mathbf{x}_{\rm opt} = \mathrm{sgn}(\mathbf{q}_1)$ where $\mathrm{sgn}(\cdot)$ denotes the vector sign operation.[6] If, on the other hand, $D = 2$ (hence, $\mathbf{V}$ has size $N \times 2$), then it has been shown that there exists a set $\mathcal{X}(\mathbf{V}_{N\times 2}) \subset \{\pm 1\}^N$ which has cardinality $|\mathcal{X}(\mathbf{V}_{N\times 2})| = N$ and is constructed with complexity $\mathcal{O}(N\log N)$ [6], [7] such that $\mathbf{x}_{\rm opt}$ can be efficiently computed by numerical comparison of $\mathbf{x}^T\mathbf{A}\mathbf{x}$ among the elements of $\mathcal{X}(\mathbf{V}_{N\times 2})$. Therefore, maximization of a rank-2 quadratic form $\mathbf{x}^T\mathbf{A}\mathbf{x}$ with a binary argument $\mathbf{x}$ is efficiently achieved with log-linear complexity.

Special emphasis for the case $D = 3$ was given recently in [13] where an efficient algorithm for the computation of $\mathbf{x}_{\rm opt}$ was developed. The algorithm in [13] utilizes auxiliary spherical coordinates and partitions the three-dimensional space into a quadratic-size set of regions. Each region corresponds to a distinct binary vector and the set $\mathcal{X}(\mathbf{V}_{N\times 3})$ that contains all binary vectors associated with regions has cardinality $|\mathcal{X}(\mathbf{V}_{N\times 3})| = \binom{N}{2} + 1 = \mathcal{O}(N^2)$, is constructed with complexity $\mathcal{O}(N^2\log N)$, and contains the optimal vector $\mathbf{x}_{\rm opt}$ in (6).

From a different perspective, several works in the area of computational geometry have treated the equivalent problem of maximization of a rank-D quadratic form $\mathbf{b}^T\mathbf{Q}\mathbf{b}$ over the $0/1$ field, i.e., when $\mathbf{Q}$ is a matrix of rank D and $\mathbf{b} \in \{0,1\}^N$. They do so by identifying a subset of $\{0,1\}^N$ that contains

[4]In fact, since opposite arguments $\mathbf{x}$ and $-\mathbf{x}$ result in the same metric $\mathbf{x}^T\mathbf{A}\mathbf{x} = (-\mathbf{x})^T\mathbf{A}(-\mathbf{x})$, we can ignore half of the elements of $\{\pm 1\}^N$ and reduce the complexity of the quadratic form maximization to $\mathcal{O}(2^{N-1})$ which is still intractable.

[5]If $D = 1$, then w.l.o.g. we assume that $\mathbf{V} = \sqrt{\lambda_1}\mathbf{q}_1$ has only nonzero elements, i.e., $V_{n,1} \neq 0, n = 1,2,\ldots,N$. Indeed, if there exists an $n \in \{1,2,\ldots,N\}$ such that $V_{n,1} = 0$, then neither $x_n = +1$ or $x_n = -1$ have an effect on $\mathbf{V}^T\mathbf{x}$ in (6), implying that we can ignore the corresponding element $V_{n,1}$, assign an arbitrary value to $x_n = \pm 1$, and reduce the size of the original problem from N to $N-1$.

[6]For any $\mathbf{x} \in \mathbb{R}^N$ with $x_n \neq 0, n = 1,2,\ldots,N, \mathbf{y} = \mathrm{sgn}(\mathbf{x})$ is an $N \times 1$ vector with $y_n = \begin{cases} -1, & x_n < 0, \\ 1, & x_n > 0, \end{cases} n = 1,2,\ldots,N.$

$\sum_{d=0}^{D-1}\binom{N-1}{d}$ vectors among which one is optimal. The subset of interest is constructed by the incremental algorithm [8], [9] or the reverse search [3], [4], [10]. The incremental algorithm [8], [9] is theoretically faster but very complicated to implement due to its large memory requirement while the reverse search is simpler to implement and constructs the set of $\sum_{d=0}^{D-1}\binom{N-1}{d}$ candidate vectors with complexity $\mathcal{O}(\mathrm{LP}(N,D)\cdot N^D)$ where $\mathrm{LP}(N,D)$ denotes the time to solve a linear programming (LP) optimization problem with N inequalities and D variables [3], [4], [10].

In the next section, we generalize the approach of [6], [7], [13] to treat the problem of quadratic-form maximization in (6) for any $D \leq N-1$. Specifically, we introduce $D-1$ auxiliary spherical coordinates and show that there exists a set $\mathcal{X}(\mathbf{V}_{N\times D}) \subset \{\pm 1\}^N$ which has cardinality $|\mathcal{X}(\mathbf{V}_{N\times D})| = \sum_{d=0}^{D-1}\binom{N-1}{d}$, can be computed in polynomial time, and contains the optimal vector $\mathbf{x}_{\mathrm{opt}}$ in (6). We also develop an algorithm that constructs $\mathcal{X}(\mathbf{V}_{N\times D})$ with computational complexity $\mathcal{O}(N^D)$ and show that it is fully parallelizable and rank-scalable.

III. Efficient Maximization of a Rank-Deficient Quadratic Form With a Binary Vector Argument

A. Theoretic Developments

Since $\mathbf{x}^T\mathbf{V}\mathbf{V}^T\mathbf{x} = \|\mathbf{V}^T\mathbf{x}\|^2$, our optimization problem (6) becomes

$$\mathbf{x}_{\mathrm{opt}} = \arg\max_{\mathbf{x}\in\{\pm1\}^N} \|\mathbf{V}^T\mathbf{x}\|. \tag{7}$$

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., $\mathbf{V}_{n,1:D} \neq \mathbf{0}_{1\times D}, n = 1,2,\ldots,N$. Indeed, if there exists an $n \in \{1,2,\ldots,N\}$ such that $\mathbf{V}_{n,1:D} = \mathbf{0}_{1\times D}$, then neither $x_n = +1$ nor $x_n = -1$ have an effect on $\mathbf{V}^T\mathbf{x}$ in (7), implying that we can ignore the corresponding row of $\mathbf{V}$, assign an arbitrary value to $x_n = \pm 1$, and reduce the size of the original problem from N to $N-1$. In addition, w.l.o.g. we assume that $V_{n,1} \neq 0, n = 1,2,\ldots,N$, because for any $\mathbf{V}\in\mathbb{R}^{N\times D}$ there exists an orthogonal matrix $\mathbf{U}\in\mathbb{R}^{D\times D}$ such that $\|\mathbf{V}^T\mathbf{x}\| = \|(\mathbf{V}\mathbf{U})^T\mathbf{x}\|$ and the $N\times D$ matrix $\mathbf{V}\mathbf{U}$ contains no zero in its first column, i.e., $[\mathbf{V}\mathbf{U}]_{n,1} \neq 0, n = 1,2,\ldots,N$, as the following proposition states. The proof is provided in the Appendix.

Proposition 1: For any $N\times D$ matrix $\mathbf{V}$ there exists an orthogonal $D\times D$ matrix $\mathbf{U}$ such that the matrix $\mathbf{V}\mathbf{U}$ does not contain any zero in its first column. □

To develop an efficient method for the maximization in (7), we introduce the spherical coordinates $\phi_1 \in (-\pi,\pi], \phi_2,\ldots,\phi_{D-1} \in (-\frac{\pi}{2},\frac{\pi}{2}]$ and define the spherical coordinate vector

$$\boldsymbol{\phi}_{i:j} \triangleq [\phi_i, \phi_{i+1}, \ldots, \phi_j]^T \tag{8}$$

and the hyperpolar vector

$$\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) \triangleq \begin{bmatrix} \sin\phi_1 \\ \cos\phi_1\sin\phi_2 \\ \cos\phi_1\cos\phi_2\sin\phi_3 \\ \vdots \\ \cos\phi_1\cos\phi_2\ldots\sin\phi_{D-1} \\ \cos\phi_1\cos\phi_2\ldots\cos\phi_{D-1} \end{bmatrix}. \tag{9}$$

A critical equality for our subsequent developments is

$$\max_{\mathbf{x}\in\{\pm1\}^N} \|\mathbf{V}^T\mathbf{x}\| = \max_{\mathbf{x}\in\{\pm1\}^N} \max_{\boldsymbol{\phi}_{1:D-1}\in(-\pi,\pi]\times(-\frac{\pi}{2},\frac{\pi}{2}]^{D-2}} \{\mathbf{x}^T\mathbf{V}\mathbf{c}(\boldsymbol{\phi}_{1:D-1})\} \tag{10}$$

which results from Cauchy–Schwartz Inequality, since for any $\mathbf{a}\in\mathbb{R}^D$

$$\mathbf{a}^T\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) \leq \|\mathbf{a}\|\underbrace{\|\mathbf{c}(\boldsymbol{\phi}_{1:D-1})\|}_{=1} \tag{11}$$

with equality if and only if $\phi_1,\ldots,\phi_{D-1}$ are the spherical coordinates of $\mathbf{a}$. We interchange the maximizations in (10) and obtain the equivalent problem

$$\max_{\boldsymbol{\phi}_{1:D-1}\in(-\pi,\pi]\times(-\frac{\pi}{2},\frac{\pi}{2}]^{D-2}} \sum_{n=1}^{N} \max_{x_n=\pm1}\{x_n \times\mathbf{V}_{n,1:D}\mathbf{c}(\boldsymbol{\phi}_{1:D-1})\}. \tag{12}$$

For a given point $\boldsymbol{\phi}_{1:D-1}\in(-\pi,\pi]\times(-\frac{\pi}{2},\frac{\pi}{2}]^{D-2}$, the maximizing argument of each term of the sum in (12) depends *only* on the corresponding row of $\mathbf{V}$ and is determined by

$$x_n = \arg\max_{x=\pm1}\{x\,\mathbf{V}_{n,1:D}\,\mathbf{c}(\boldsymbol{\phi}_{1:D-1})\}, \quad n = 1,2,\ldots,N. \tag{13}$$

Then, according to (12), the optimal vector $\mathbf{x}_{\mathrm{opt}}$ in (7) is met if we scan the entire set $(-\pi,\pi]\times(-\frac{\pi}{2},\frac{\pi}{2}]^{D-2}$ and collect the binary vector

$$\mathbf{x}(\mathbf{V}_{N\times D};\boldsymbol{\phi}_{1:D-1}) \triangleq \begin{bmatrix} \arg\max_{x=\pm1}\{x\,\mathbf{V}_{1,1:D}\,\mathbf{c}(\boldsymbol{\phi}_{1:D-1})\} \\ \arg\max_{x=\pm1}\{x\,\mathbf{V}_{2,1:D}\,\mathbf{c}(\boldsymbol{\phi}_{1:D-1})\} \\ \vdots \\ \arg\max_{x=\pm1}\{x\,\mathbf{V}_{N,1:D}\,\mathbf{c}(\boldsymbol{\phi}_{1:D-1})\} \end{bmatrix} \tag{14}$$

for every point $\boldsymbol{\phi}_{1:D-1}\in(-\pi,\pi]\times(-\frac{\pi}{2},\frac{\pi}{2}]^{D-2}$. However, as it is explained later on in this section, we may ignore all points in $(-\pi,\pi]\times(-\frac{\pi}{2},\frac{\pi}{2}]^{D-2}$ which result in an ambiguous decision[7] and restrict our search to

$$\bar{\Phi}(\mathbf{V}_{N\times D}) \triangleq \left\{\boldsymbol{\phi}_{1:D-1}\in(-\pi,\pi]\times\left(-\frac{\pi}{2},\frac{\pi}{2}\right]^{D-2} : \mathbf{V}_{n,1:D}\,\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) \neq 0, n = 1,2,\ldots,N\right\}. \tag{15}$$

[7]Point $\boldsymbol{\phi}_{1:D-1}$ results in an ambiguous decision if $\mathbf{V}_{n,1:D}\,\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) = 0$ for some $n = 1,2,\ldots,N$.

Then, for the given $N \times D$ matrix $\mathbf{V}$, each point $\boldsymbol{\phi}_{1:D-1} \in \bar{\Phi}(\mathbf{V}_{N\times D})$ is mapped to a candidate binary vector

$$\mathbf{x}(\mathbf{V}_{N\times D};\boldsymbol{\phi}_{1:D-1}) = \begin{bmatrix} \mathrm{sgn}(\mathbf{V}_{1,1:D}\,\mathbf{c}(\boldsymbol{\phi}_{1:D-1})) \\ \mathrm{sgn}(\mathbf{V}_{2,1:D}\,\mathbf{c}(\boldsymbol{\phi}_{1:D-1})) \\ \vdots \\ \mathrm{sgn}(\mathbf{V}_{N,1:D}\,\mathbf{c}(\boldsymbol{\phi}_{1:D-1})) \end{bmatrix} = \mathrm{sgn}(\mathbf{V}_{N\times D}\mathbf{c}(\boldsymbol{\phi}_{1:D-1})) \quad (16)$$

and the optimal vector $\mathbf{x}_{\mathrm{opt}}$ in (7) belongs to $\bigcup_{\boldsymbol{\phi}_{1:D-1}\in\bar{\Phi}(\mathbf{V}_{N\times D})} \mathbf{x}(\mathbf{V}_{N\times D};\boldsymbol{\phi}_{1:D-1})$.

We note 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 $\boldsymbol{\phi}_{1:D-1} \in \bar{\Phi}(\mathbf{V}_{N\times D})$. Since opposite binary vectors $\mathbf{x}$ and $-\mathbf{x}$ result in the same metric value in (7), we ignore the values of ϕ_1 in $(-\pi, -\frac{\pi}{2}] \cup (\frac{\pi}{2}, \pi]$ and rewrite the optimization problem in (12) as

$$\max_{\boldsymbol{\phi}_{1:D-1}\in\Phi(\mathbf{V}_{N\times D})} \sum_{n=1}^{N} \max_{x_n=\pm 1} \{x_n \mathbf{V}_{n,1:D}\mathbf{c}(\boldsymbol{\phi}_{1:D-1})\},$$
$$\Phi(\mathbf{V}_{N\times D}) \stackrel{\triangle}{=} \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]^{D-1} \cap \bar{\Phi}(\mathbf{V}_{N\times D}). \quad (17)$$

Finally, we collect all candidate binary vectors into set

$$\boldsymbol{\mathcal{X}}(\mathbf{V}_{N\times D}) \stackrel{\triangle}{=} \bigcup_{\boldsymbol{\phi}_{1:D-1}\in\Phi(\mathbf{V}_{N\times D})} \{\mathbf{x}(\mathbf{V}_{N\times D};\boldsymbol{\phi}_{1:D-1})\}$$
$$= \{\bar{\mathbf{x}} \in \{\pm 1\}^N : \exists\ \boldsymbol{\phi}_{1:D-1} \in \Phi(\mathbf{V}_{N\times D}) \text{ such that } \mathbf{x}(\mathbf{V}_{N\times D};\boldsymbol{\phi}_{1:D-1}) = \bar{\mathbf{x}}\}$$
$$\subseteq \{\pm 1\}^N \quad (18)$$

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

$$\mathbf{x}_{\mathrm{opt}} = \arg\max_{\mathbf{x}\in\mathcal{X}(\mathbf{V})}\{\mathbf{x}^T\mathbf{V}\mathbf{V}^T\mathbf{x}\}. \quad (19)$$

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

We begin by observing that the decision in (13) determines a hypersurface that partitions the $(D-1)$-dimensional hypercube $(-\frac{\pi}{2}, \frac{\pi}{2}]^{D-1}$ into two regions; one corresponds to $x_n = +1$ and the other corresponds to $x_n = -1$. The following proposition presents the details of such a partition. The proof is provided in the Appendix.

Proposition 2: Let $\mathbf{v} \in \mathbb{R}^D, v_1 \neq 0$, and $\boldsymbol{\phi}_{1:D-1} \in \{\boldsymbol{\phi}_{1:D-1} \in (-\frac{\pi}{2}, \frac{\pi}{2}]^{D-1} : \mathbf{v}^T\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) \neq 0\}$. Then, the decision rule $x(\mathbf{v}^T;\boldsymbol{\phi}_{1:D-1}) \stackrel{\triangle}{=} \mathrm{sgn}(\mathbf{v}^T\mathbf{c}(\boldsymbol{\phi}_{1:D-1}))$ is equivalent to (20), shown at the bottom of the page. □

As seen in Proposition 2, for any $\mathbf{v} \in \mathbb{R}^D$ with $v_1 \neq 0$, the function $\phi_1 = \tan^{-1}(-\frac{\mathbf{v}_{2:D}^T\mathbf{c}(\boldsymbol{\phi}_{2:D-1})}{v_1})$ is equivalent to $\mathbf{v}^T\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) = 0$ and determines a hypersurface $S(\mathbf{v}^T)$ which partitions $(-\frac{\pi}{2}, \frac{\pi}{2}]^{D-1}$ into two regions that correspond to the 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 partition the hypercube $(-\frac{\pi}{2}, \frac{\pi}{2}]^{D-1}$ into K cells $C_1, C_2, \ldots, C_K$ such that $\bigcup_{k=1}^{K} C_k = \Phi(\mathbf{V}_{N\times D}), C_k \cap C_j = \emptyset$ if $k \neq j$, and each cell C_k corresponds to a *distinct* $\mathbf{x}_k \in \{\pm 1\}^N$ in the sense that $\mathbf{x}(\mathbf{V}_{N\times D};\boldsymbol{\phi}_{1:D-1}) = \mathbf{x}_k$ for any $\boldsymbol{\phi}_{1:D-1} \in C_k$ and $\mathbf{x}_k \neq \mathbf{x}_j$ if $k \neq j, k, j \in \{1, 2, \ldots, K\}$. The excluded points $\boldsymbol{\phi}_{1:D-1} \in (-\frac{\pi}{2}, \frac{\pi}{2}]^{D-1} - \Phi(\mathbf{V}_{N\times D})$ result in ambiguous decisions, since for such points $\mathbf{V}_{n,1:D}\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) = 0$ for one or more coordinates $n \in \{1, 2, \ldots, N\}$. However, for every such a point we may assign a candidate binary vector which is affiliated with a neighboring cell without losing optimality in the original problem in (7).

To illustrate such a partition, we set $D = 3$ and $N = 8$, generate a rank-3 matrix $\mathbf{V}_{8\times 3}$ with $V_{n,1} \neq 0, n = 1, 2, \ldots, 8$, and plot in Fig. 1(a) the curve $\phi_1 = \tan^{-1}(-\frac{\mathbf{V}_{1,2:3}\mathbf{c}(\phi_2)}{V_{1,1}}) = \tan^{-1}(-\frac{V_{1,2}\sin\phi_2 + V_{1,3}\cos\phi_2}{V_{1,1}})$ that originates from the first row of $\mathbf{V}_{8\times 3}$. In Fig. 1(b) and 1(c), we add the curves that originate from the second and third, respectively, rows of $\mathbf{V}_{8\times 3}$. We generalize in Fig. 1(d) which includes all eight curves of the form $\phi_1 = \tan^{-1}(-\frac{\mathbf{V}_{n,2:3}\mathbf{c}(\phi_2)}{V_{n,1}}) = \tan^{-1}(-\frac{V_{n,2}\sin\phi_2 + V_{n,3}\cos\phi_2}{V_{n,1}})$, $n = 1, 2, \ldots, 8$. We observe that the two-dimensional set $(-\frac{\pi}{2}, \frac{\pi}{2}]^2$ is partitioned into regions (cells); each cell corresponds to a distinct binary vector according to (20). We repeat our example for $D = 4$, generate a rank-4 matrix $\mathbf{V}_{8\times 4}$, and plot in Fig. 2 the four surfaces that correspond to $\phi_1 = \tan^{-1}(-\frac{\mathbf{V}_{n,2:4}\mathbf{c}(\boldsymbol{\phi}_{2:3})}{V_{n,1}}) = \tan^{-1}(-\frac{V_{n,2}\sin\phi_2 + V_{n,3}\cos\phi_2\sin\phi_3 + V_{n,4}\cos\phi_2\cos\phi_3}{V_{n,1}})$, $n = 1, 2, 3, 4$. Again, if we consider *all* eight surfaces, then the 3-D cube $(-\frac{\pi}{2}, \frac{\pi}{2}]^3$ is partitioned into regions (cells) and distinct binary vectors are associated with each cell according to (20).[8] Our objective in the sequel is to *efficiently identify these candidate binary vectors, since one of them is the optimal binary vector in* (7).

Several properties of the resulting partition that are very important for our subsequent developments are presented in the following proposition. The proof is provided in the Appendix.

[8]For visualization purposes, we do not plot the complete partition.

$$x(\mathbf{v}^T;\boldsymbol{\phi}_{1:D-1}) = \begin{cases} -\mathrm{sgn}(v_1), & \phi_1 \in \left(-\frac{\pi}{2}, \tan^{-1}\left(-\frac{\mathbf{v}_{2:D}^T\mathbf{c}(\boldsymbol{\phi}_{2:D-1})}{v_1}\right)\right) \\ \mathrm{sgn}(v_1), & \phi_1 \in \left(\tan^{-1}\left(-\frac{\mathbf{v}_{2:D}^T\mathbf{c}(\boldsymbol{\phi}_{2:D-1})}{v_1}\right), \frac{\pi}{2}\right] \end{cases} \quad (20)$$

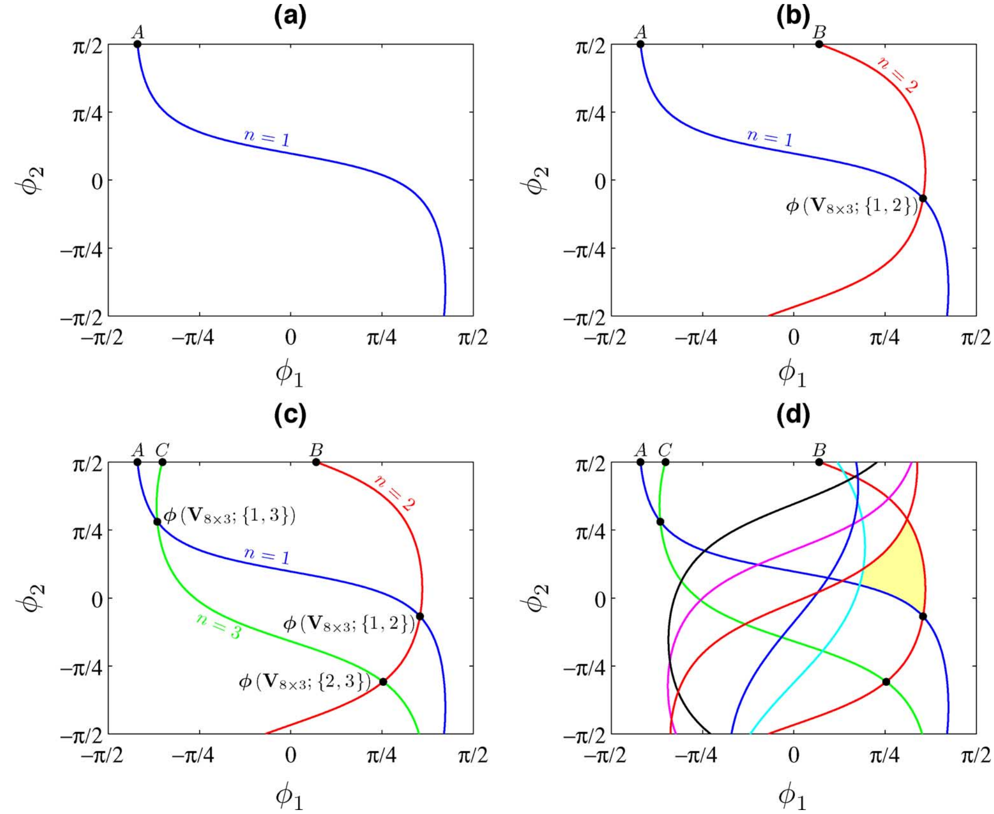

Fig. 1. Partition of $(-\frac{\pi}{2},\frac{\pi}{2}]^2$ into cells.

Proposition 3: Let $\mathbf{V}_{N\times D}$ be a rank-D matrix and $V_{n,1}\neq 0, n=1,2,\ldots,N$. The following hold true.

a) Each subset of $\{S(\mathbf{V}_{1,1:D}),S(\mathbf{V}_{2,1:D}),\ldots,S(\mathbf{V}_{N,1:D})\}$ that consists of $D-1$ hypersurfaces has either a single intersection or uncountably many intersections in $(-\frac{\pi}{2},\frac{\pi}{2}]^{D-1}$.

b) For any $\phi_1,\phi_2,\ldots,\phi_{D-1}\in\Phi(\mathbf{V}_{N\times D})$:
   i) $\mathbf{x}(\mathbf{V}_{N\times D};\boldsymbol{\phi}_{1:D-2},\frac{\pi}{2})=\mathbf{x}(\mathbf{V}_{N\times(D-1)};\boldsymbol{\phi}_{1:D-2})$.
   ii) $\mathbf{x}(\mathbf{V}_{N\times D};\boldsymbol{\phi}_{1:D-2},-\frac{\pi}{2})$ $=-\mathbf{x}(\mathbf{V}_{N\times D};-\boldsymbol{\phi}_{1:D-2},\frac{\pi}{2})$.
   iii) $\mathbf{x}(\mathbf{V}_{N\times D};\boldsymbol{\phi}_{1:D-3},\frac{\pi}{2},\phi_{D-1})$ $=\mathbf{x}(\mathbf{V}_{N\times(D-2)};\boldsymbol{\phi}_{1:D-3})$.
   iv) $\mathbf{x}(\mathbf{V}_{N\times D};\boldsymbol{\phi}_{1:D-3},-\frac{\pi}{2},\phi_{D-1})$ $=-\mathbf{x}(\mathbf{V}_{N\times D};-\boldsymbol{\phi}_{1:D-3},\frac{\pi}{2},\phi'_{D-1})$, $\forall\phi'_{D-1}\in\Phi(\mathbf{V}_{N\times D})$.
   v) $\mathbf{x}(\mathbf{V}_{N\times D};\boldsymbol{\phi}_{1:D-3},\pm\frac{\pi}{2},\phi_{D-1})$ $=\mathbf{x}(\mathbf{V}_{N\times D};\boldsymbol{\phi}_{1:D-3},\pm\frac{\pi}{2},\phi'_{D-1})$, $\forall\phi'_{D-1}\in\Phi(\mathbf{V}_{N\times D})$. □

Let $\mathcal{I}_{D-1}\triangleq\{i_1,i_2,\ldots,i_{D-1}\}\subset\{1,2,\ldots,N\}$ denote the subset of $D-1$ indices that correspond to hypersurfaces $S(\mathbf{V}_{i_1,1:D})$, $S(\mathbf{V}_{i_2,1:D})$, $\ldots$, $S(\mathbf{V}_{i_{D-1},1:D})$ and $\boldsymbol{\phi}(\mathbf{V}_{N\times D};\mathcal{I}_{D-1})\in(-\frac{\pi}{2},\frac{\pi}{2}]^{D-1}$ be the vector of spherical coordinates of their intersection. If $\boldsymbol{\phi}(\mathbf{V}_{N\times D};\mathcal{I}_{D-1})$ is uniquely determined according to Proposition 3, Part (a), then it "leads" a cell, say $C(\mathbf{V}_{N\times D};\mathcal{I}_{D-1})$, associated with a distinct binary vector $\mathbf{x}(\mathbf{V}_{N\times D};\mathcal{I}_{D-1})$ in the sense that $\mathbf{x}(\mathbf{V}_{N\times D};\boldsymbol{\phi}_{1:D-1})=\mathbf{x}(\mathbf{V}_{N\times D};\mathcal{I}_{D-1})$ for all $\boldsymbol{\phi}_{1:D-1}\in C(\mathbf{V}_{N\times D};\mathcal{I}_{D-1})$ and $\boldsymbol{\phi}(\mathbf{V}_{N\times D};\mathcal{I}_{D-1})=\arg\inf_{\phi_{D-1}}(\{\boldsymbol{\phi}_{1:D-1}\in C(\mathbf{V}_{N\times D};\mathcal{I}_{D-1})\})$. For example, in the partition of $(-\frac{\pi}{2},\frac{\pi}{2}]^2$ presented in Fig. 1, we observe that every pair of curves intersect once and each intersection determines a cell that lies "above" it. Each cell is associated with a distinct binary vector. We collect all such vectors into

$$J(\mathbf{V}_{N\times D})\triangleq\bigcup_{\mathcal{I}_{D-1}\subset\{1,\ldots,N\}}\{\mathbf{x}(\mathbf{V}_{N\times D};\mathcal{I}_{D-1})\}\tag{21}$$

and observe that $J(\mathbf{V}_{N\times D})\subseteq\{\pm1\}^N$ and $|J(\mathbf{V}_{N\times D})|=\binom{N}{D-1}$.[9] In other words, $J(\mathbf{V}_{N\times D})$ contains $\binom{N}{D-1}$ binary vectors; each vector is associated with a cell in $(-\frac{\pi}{2},\frac{\pi}{2}]^{D-1}$ that starts, with respect to variable ϕ_{D-1}, from 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, according to Proposition 3, Part (b.*ii*), 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 the initial cell contains a point with $\phi_{D-2}=\pm\frac{\pi}{2}$, as Proposition 3, Part (b.*v*) mentions. For example, in Fig. 1 such cells are identified at the bottom of the plane, that is, for $\phi_2=-\frac{\pi}{2}$. We observe that the vectors that are associated with these cells are opposite to the vectors that are associated with the cells that are identified at

[9]In general, $|J(\mathbf{V}_{N\times D})|\leq\binom{N}{D-1}$ with equality if and only if the $\binom{N}{D-1}$ intersections of hypersurfaces are distinct. In the sequel, we consider the most computationally demanding case of distinct intersections.

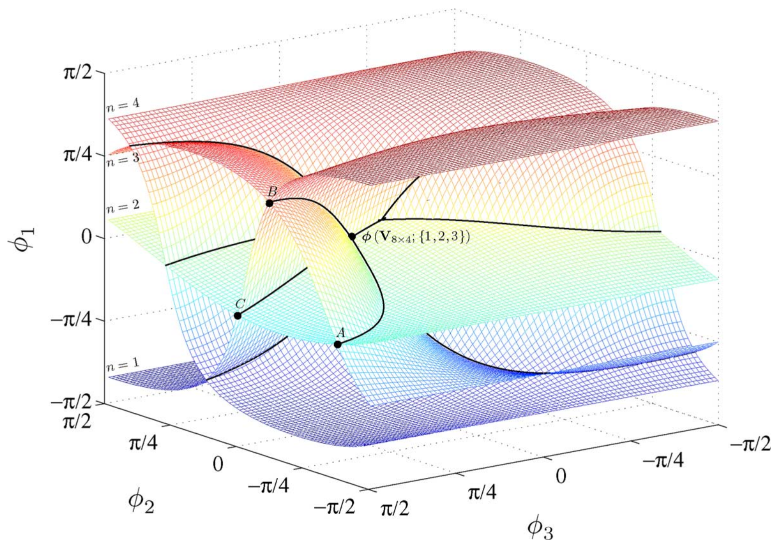

Fig. 2. Partition of $(-\frac{\pi}{2}, \frac{\pi}{2}]^3$ into cells.

the top of the plane. Therefore, the former ones can be ignored. Similarly, in Fig. 2 the binary vectors that are determined for $\phi_3 = -\frac{\pi}{2}$ are opposite to the vectors determined for $\phi_3 = \frac{\pi}{2}$; hence, the former ones can again be ignored.

Finally, 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 $(-\frac{\pi}{2}, \frac{\pi}{2}]^{D-3} \times \{\pm\frac{\pi}{2}\} \times \{\phi'_{D-1}\}$. In Fig. 2, such cells are identified for $\phi_2 = \frac{\pi}{2}$ and $\phi_2 = -\frac{\pi}{2}$. In addition, due to Proposition 3, Part (b.*iv*), the cells that are defined when $\phi_{D-2} = -\frac{\pi}{2}$ are associated with vectors which 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, according to Proposition 3, Part (b.*iii*), identify the cells that are determined by the *reduced-size* matrix $\mathbf{V}_{N\times(D-2)}$ over the hypercube $(-\frac{\pi}{2}, \frac{\pi}{2}]^{D-3}$. As an example, in Fig. 2 we set $\phi_2 = \phi_3 = \frac{\pi}{2}$ and examine the cells that appear on the leftmost vertical edge of the cube.

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,

$$\mathcal{X}(\mathbf{V}_{N\times d}) = J(\mathbf{V}_{N\times d}) \cup \mathcal{X}(\mathbf{V}_{N\times(d-2)}) \quad d = 3,4,\ldots,D \tag{22}$$

which implies that

$$\mathcal{X}(\mathbf{V}_{N\times D}) = J(\mathbf{V}_{N\times D}) \cup J(\mathbf{V}_{N\times(D-2)}) \cup \ldots \cup J\left(\mathbf{V}_{N\times(D-2\lfloor\frac{D-1}{2}\rfloor)}\right) = \bigcup_{d=0}^{\lfloor\frac{D-1}{2}\rfloor} J(\mathbf{V}_{N\times(D-2d)}) \tag{23}$$

since $\mathcal{X}(\mathbf{V}_{N\times1}) = J(\mathbf{V}_{N\times1})$ with $|\mathcal{X}(\mathbf{V}_{N\times1})| = |J(\mathbf{V}_{N\times1})| = 1$ and $\mathcal{X}(\mathbf{V}_{N\times2}) = J(\mathbf{V}_{N\times2})$ with $|\mathcal{X}(\mathbf{V}_{N\times2})| = |J(\mathbf{V}_{N\times2})| = N$ [6], [7]. As a result, the cardinality of $\mathcal{X}(\mathbf{V}_{N\times D})$ is

$$\begin{aligned}|\mathcal{X}(\mathbf{V}_{N\times D})| &= |J(\mathbf{V}_{N\times D})| + |J(\mathbf{V}_{N\times(D-2)})| + \cdots + \left|J\left(\mathbf{V}_{N\times(D-2\lfloor\frac{D-1}{2}\rfloor)}\right)\right| \\ &= \binom{N}{D-1} + \binom{N}{D-3} + \ldots + \binom{N}{D-1-2\lfloor\frac{D-1}{2}\rfloor} \\ &= \sum_{d=0}^{\lfloor\frac{D-1}{2}\rfloor}\binom{N}{D-1-2d} = \sum_{d=0}^{D-1}\binom{N-1}{d}.\end{aligned} \tag{24}$$

To summarize the developments in this subsection, we have utilized $D-1$ auxiliary spherical coordinates, partitioned the hypercube $(-\frac{\pi}{2}, \frac{\pi}{2}]^{D-1}$ into $\sum_{d=0}^{D-1}\binom{N-1}{d}$ cells that are associated with distinct binary vectors which constitute $\mathcal{X}(\mathbf{V}_{N\times D}) \subseteq \{\pm1\}^N$, and proved that $\mathbf{x}_{\text{opt}} \in \mathcal{X}(\mathbf{V}_{N\times D})$. Therefore, the initial problem in (7) 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 $\mathcal{O}(\sum_{d=0}^{D-1}\binom{N-1}{d}) = \mathcal{O}(N^{D-1})$ comparisons upon construction of $\mathcal{X}(\mathbf{V}_{N\times D})$. An efficient algorithm for the construction of $\mathcal{X}(\mathbf{V}_{N\times D})$ is developed in the next subsection.

B. Algorithmic Developments

Let $\mathbf{V}_{N\times D}$ be a real matrix that satisfies the assumptions made in the beginning of Section III. According to (23), the construction of $\mathcal{X}(\mathbf{V}_{N\times D})$ reduces to the construction of $J(\mathbf{V}_{N\times D}), J(\mathbf{V}_{N\times(D-2)}), \ldots, J(\mathbf{V}_{N\times2})$ if D is even and $J(\mathbf{V}_{N\times D}), J(\mathbf{V}_{N\times(D-2)}), \ldots, J(\mathbf{V}_{N\times1})$ if D is odd which, as seen in (21), can be computed *independently* and *in parallel*. Recall that $J(\mathbf{V}_{N\times1}), J(\mathbf{V}_{N\times2})$, and $J(\mathbf{V}_{N\times3})$ can be obtained with complexity $\mathcal{O}(N), \mathcal{O}(N\log N)$, and $\mathcal{O}(N^2\log N)$, respectively [6], [7], [13]. Therefore, it remains to describe a way to construct $J(\mathbf{V}_{N\times d})$ for any $d > 3$. Interestingly, from (21), we observe that the construction of $J(\mathbf{V}_{N\times d})$ can also be fully parallelized since the candidate vector $\mathbf{x}(\mathbf{V}_{N\times d};\mathcal{I}_{d-1})$ can be computed *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 $\mathbf{x}(\mathbf{V}_{N\times d};\mathcal{I}_{d-1})$ $\forall\, \mathcal{I}_{d-1} \subset \{1,2,\ldots,N\}, d \in \{3,4,\ldots,D\}$.

We consider a certain value of $d \in \{3, 4, \ldots, D\}$ and a certain set of indices $\mathcal{I}_{d-1} \subset \{1, 2, \ldots, N\}$ such that the $d-1$ hypersurfaces $S(\mathbf{V}_{i_1,1:d}), S(\mathbf{V}_{i_2,1:d}), \ldots, S(\mathbf{V}_{i_{d-1},1:d})$ intersect at a single point $\boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$. Cell $C(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ that is "led" by $\boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ is associated with the binary vector $\mathbf{x}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ [see, for example, the highlighted cell in Fig. 1(d)]. To identify $\mathbf{x}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$, we consider its N elements separately and observe the following.

i) For any $n \in \{1, 2, \ldots, N\} - \mathcal{I}_{d-1}$, the corresponding element of $\mathbf{x}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ maintains its value at $\boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$; hence, it is determined by

$$x_n(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1}) = x(\mathbf{V}_{n,1:d}; \boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})). \tag{25}$$

ii) For any $n \in \mathcal{I}_{d-1}$, say $n = i_k$, the corresponding element of $\mathbf{x}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ *cannot* be determined at $\boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$. However, it maintains its value at the intersection of the remaining $d-2$ hypersurfaces $S(\mathbf{V}_{i_1,1:d-1})$, $S(\mathbf{V}_{i_2,1:d-1}), \ldots, S(\mathbf{V}_{i_{k-1},1:d-1})$, $S(\mathbf{V}_{i_{k+1},1:d-1})$, $S(\mathbf{V}_{i_{k+2},1:d-1}), \ldots, S(\mathbf{V}_{i_{d-1},1:d-1})$; hence, it is determined by

$$x_n(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1}) = x(\mathbf{V}_{n,1:d-1}; \boldsymbol{\phi}(\mathbf{V}_{N\times (d-1)}; \mathcal{I}_{d-1} - \{i_k\})). \tag{26}$$

Equations (25) and (26) suggest the following construction of $\mathbf{x}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$. If the $\binom{N}{d-1}$ intersections of hypersurfaces are distinct, then 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 polarity of $\boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ with respect to $S(\mathbf{V}_{n,1:d})$ is the same as the polarity of any point of the cell of interest $C(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ with respect to the same hypersurface. As a result, 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, as (25) states. For example, in Fig. 1, $x_3(\mathbf{V}_{8\times 3}; \{1, 2\})$ is well determined at $\boldsymbol{\phi}(\mathbf{V}_{8\times 3}; \{1, 2\})$ through (25) and maintains its value in the associated cell $C(\mathbf{V}_{8\times 3}; \{1, 2\})$ which is highlighted in Fig. 1(d) for illustration purposes. 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 $x(\mathbf{V}_{n,1:d}; \boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})) = \pm 1$. For example, in Fig. 1, curves $\mathcal{S}(\mathbf{V}_{1,1:3})$ and $\mathcal{S}(\mathbf{V}_{2,1:3})$ pass through $\boldsymbol{\phi}(\mathbf{V}_{8\times 3}; \{1, 2\})$ leading to ambiguous decisions of $x(\mathbf{V}_{1,1:3}; \boldsymbol{\phi}(\mathbf{V}_{8\times 3}; \{1, 2\}))$ and $x(\mathbf{V}_{2,1:3}; \boldsymbol{\phi}(\mathbf{V}_{8\times 3}; \{1, 2\}))$. In such a case, ambiguity is resolved if we exclude $S(\mathbf{V}_{n,1:d})$ and consider the intersection of the remaining $d-2$ hypersurfaces at $\phi_{d-1} = \frac{\pi}{2}$. Indeed, the polarity of any point of the cell of interest $C(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ with respect to $S(\mathbf{V}_{n,1:d})$ is the same as the polarity of $\boldsymbol{\phi}(\mathbf{V}_{N\times (d-1)}; \mathcal{I}_{d-1} - \{i_k\})$ with respect to the same hypersurface. Therefore, the sign of the corresponding binary element $x_n(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ is well determined through (26). In Fig. 1, the ambiguity with respect to $x_1(\mathbf{V}_{8\times 3}; \{1, 2\})$ at intersection $\boldsymbol{\phi}(\mathbf{V}_{8\times 3}; \{1, 2\})$ is resolved through (26) at $B = \boldsymbol{\phi}(\mathbf{V}_{8\times 2}; \{2\})$ and the ambiguity with respect to $x_2(\mathbf{V}_{8\times 3}; \{1, 2\})$ is resolved through (26) at $A = \boldsymbol{\phi}(\mathbf{V}_{8\times 2}; \{1\})$.

Similarly, in Fig. 2, $x_4(\mathbf{V}_{8\times 4}; \{1, 2, 3\})$ is well determined at $\boldsymbol{\phi}(\mathbf{V}_{8\times 4}; \{1, 2, 3\})$ through (25) and maintains its value in the associated cell $C(\mathbf{V}_{8\times 4}; \{1, 2, 3\})$. Conversely, surfaces $\mathcal{S}(\mathbf{V}_{1,1:4}), \mathcal{S}(\mathbf{V}_{2,1:4})$, and $\mathcal{S}(\mathbf{V}_{3,1:4})$ pass through $\boldsymbol{\phi}(\mathbf{V}_{8\times 4}; \{1, 2, 3\})$ leading to ambiguous decisions of $x(\mathbf{V}_{1,1:4}; \boldsymbol{\phi}(\mathbf{V}_{8\times 4}; \{1, 2, 3\})), x(\mathbf{V}_{2,1:4}; \boldsymbol{\phi}(\mathbf{V}_{8\times 4}; \{1, 2, 3\}))$, and $x(\mathbf{V}_{3,1:4}; \boldsymbol{\phi}(\mathbf{V}_{8\times 4}; \{1, 2, 3\}))$. Ambiguity is resolved through (26) at $A = \boldsymbol{\phi}(\mathbf{V}_{8\times 3}; \{2, 3\}), B = \boldsymbol{\phi}(\mathbf{V}_{8\times 3}; \{1, 3\})$, and $C = \boldsymbol{\phi}(\mathbf{V}_{8\times 3}; \{1, 2\})$, respectively.

Finally, if there exist common intersections, then for the intersections that belong to more than $d-1$ hypersurfaces we follow the above procedure of setting $\phi_{d-1} = \frac{\pi}{2}$ to resolve the ambiguity with respect to the index that does not belong to the set of $d-1$ indices that we examine.

It remains to describe how the vector of spherical coordinates $\boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ is computed efficiently. Recall that $\boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ represents the 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 solution of

$$\mathbf{V}_{\mathcal{I}_{d-1},1:d}\mathbf{c}(\boldsymbol{\phi}_{1:d-1}) = \mathbf{0}_{(d-1)\times 1}. \tag{27}$$

According to the proof of Proposition 3, Part (a), for a full-rank $(d-1)\times d$ real matrix, (27) has a unique solution $\boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1}) \in (-\frac{\pi}{2}, \frac{\pi}{2}]^{d-1}$ which consists of the spherical coordinates of the zero right singular vector of $\mathbf{V}_{\mathcal{I}_{d-1},1:d}$. Therefore, to obtain $\boldsymbol{\phi}(\mathbf{V}_{N\times d}; \mathcal{I}_{d-1})$ we just need to compute the zero right singular vector of $\mathbf{V}_{\mathcal{I}_{d-1},1:d}$ and calculate its spherical coordinates.

The complete algorithm for the construction of $\mathcal{X}(\mathbf{V}_{N\times D})$ is provided in Table I. The input is matrix $\mathbf{V}_{N\times D}$. Function `compute_candidates(V)` first computes $J(\mathbf{V}_{N\times D})$ and continues by calling itself with a reduced-size matrix $\mathbf{V}_{N\times (D-2)}$ as (22) indicates. The function output is set $\mathcal{X}(\mathbf{V}_{N\times D})$. After consecutive rank reductions, it ends up with $\mathbf{V}_{N\times 2}$ or $\mathbf{V}_{N\times 1}$. Then, the rank-1-optimal or rank-2-optimal solution is obtained according to the work in [6], [7].

The algorithm visits independently the $|\mathcal{X}(\mathbf{V}_{N\times D})| = \mathcal{O}(N^{D-1})$ intersections and computes the candidate binary vector associated with each intersection. We notice that the algorithm avoids the calculation of the Cartesian coordinates of each intersection. Function `find_intersection(V)` calculates the Cartesian coordinates of the intersection of the hypersurfaces that correspond to the rows of its input `V` within a sign ambiguity by the singular value decomposition, according to the proof of Proposition 3, Part (a). Then, the conversion into spherical coordinates is only necessary to resolve the sign ambiguity and is performed by function `determine_sign(c)`. The calculation of the zero right singular vector of $\mathbf{V}_{\mathcal{I}_{d-1},1:d}$ costs $\mathcal{O}(d^2)$, the conversion into spherical coordinates costs $\mathcal{O}(d)$, and the operation $\mathrm{sgn}(\mathbf{V}_{n,1:d}\mathbf{u})$ costs $\mathcal{O}(d)$ for any $\mathbf{u} \in \mathbb{R}^d$. Since $\mathbf{u}' \in \mathbb{R}^{d-1}$ is computed for each $n \in \mathcal{I}_{d-1}$, the cost of the algorithm for each combination $\mathcal{I}_{d-1}$ is $\mathcal{O}(d^2) + (N-d+1)\mathcal{O}(d) + (d-1)(\mathcal{O}(d^2) + \mathcal{O}(d)) = \mathcal{O}(d^3 + Nd)$. 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 $\mathcal{O}(N^{D-1})\mathcal{O}(N) = \mathcal{O}(N^D)$.

Finally, we study the characteristics of the proposed algorithm and compare it with the CG-based methods in [3], [4], [8]–[10]. Our interest is directed towards computational complexity, memory requirement, parallelizability, scalability, and applicability.

a) Computational Complexity: As described above, the overall complexity of the proposed algorithm is $\mathcal{O}(N^D)$ when a rank-D order-N problem is considered. In addition, further improvements in terms of complexity are allowed. For example, when $D = 2$ the complexity becomes $\mathcal{O}(N \log N)$ instead of $\mathcal{O}(N^2)$ [6], [7] while when $D = 3$ it becomes $\mathcal{O}(N^2 \log N)$ instead of $\mathcal{O}(N^3)$ [13]. We recall that the complexity of the reverse search [4], [10] is $\mathcal{O}(N^D \mathrm{LP}(N, D))$.

b) Memory Requirement: We recall that the computation of the candidate vectors of $\mathcal{X}(\mathbf{V}_{N\times D})$ is performed independently from cell to cell, which implies that there is *no need to store the data* that have been used for each candidate and we only have to store in the memory the "best" vector (in terms of the metric of interest (7)) that has been met. The memory utilization of the proposed method is, therefore, reduced, in contrast to the incremental algorithm in [8], [9] which is very complicated to implement due to its large memory requirement.

c) Parallelizability: As mentioned above, the $|\mathcal{X}(\mathbf{V}_{N\times D})| = \sum_{d=0}^{D-1}\binom{N-1}{d}$ cells are visited *independently* of each other so that the candidate vectors of $\mathcal{X}(\mathbf{V}_{N\times D})$ are computed independently of each other. Hence, the proposed algorithm is *fully parallelizable*.

d) Scalability: If the initial problem is of a high rank that makes the optimization intractable, then matrix $\mathbf{A}$ in (1) can be approximated by keeping its D strongest principal components. In such a case, as seen in (23) the proposed method is rank-scalable. The optimization begins with rank $D = 1$ or 2 and additional principal components are introduced to increase D and, hence, expand $\mathcal{X}(\mathbf{V}_{N\times D})$ until a satisfactory reduced-rank approximation is reached.

e) Applicability: Whenever the quadratic form under consideration is of (nearly) low rank, we expect that the proposed method will produce a (nearly) optimal binary vector. In fact, there might be applications where the quadratic form has exactly a low rank. Such a situation is met in semidefinite relaxation (SDR) [15]–[18] where it has been empirically observed that the returned matrix argument has rank 2 or 3. Since the Frobenius norm of the difference between the outer product of a binary vector and the returned matrix is minimized if and only if a corresponding low-rank quadratic form is maximized over a binary vector argument, our proposed method can serve as the rounding step of SDR, as well. Another example is ML noncoherent SIMO [11], [12], [14] or space-time block coded multiple-input multiple-output [19] detection where the rank of the quadratic form turns out to equal twice the rank of the covariance matrix of the vectorized channel matrix. Finally, as mentioned in Section I, the proposed method can be appropriately modified to serve (not, necessarily, constant-modulus) complex-domain optimization problems such as ML noncoherent SIMO detection of arbitrary-order MPSK [14], ML noncoherent PAM or QAM detection, and sparse rank-deficient variance maximization.

IV. Application Example

In Section III, we developed an algorithm that computes with polynomial complexity the binary vector which maximizes a rank-deficient quadratic form. To illustrate the applicability of the proposed algorithm and justify the complexity gain it offers, we consider an example drawn from recent literature on code-division multiple-access (CDMA) where -to obtain an efficient approach- the optimization problem is *approximated* by a rank-deficient quadratic form maximization.

We consider a synchronous direct-sequence CDMA system with processing gain $N = 16$ where the user of interest with a normalized binary spreading code $\mathbf{x} \in \{\pm\frac{1}{\sqrt{N}}\}^N$ transmits over an additive noise channel in the presence of K interfering users. The received signal vector is

$$\mathbf{y} = b\sqrt{P}\mathbf{x} + \mathbf{z} \tag{28}$$

where $b \in \{\pm 1\}$ is a uniformly distributed bit random variable, $P > 0$ is the collected energy per bit, and

$$\mathbf{z} = \sum_{k=1}^{K} b_k\sqrt{P_k}\mathbf{x}_k + \mathbf{n} \tag{29}$$

where $b_k \in \{\pm 1\}, P_k > 0$, and $\mathbf{x}_k \in \{\pm\frac{1}{\sqrt{N}}\}^N$ are the uniformly distributed user bit, received energy per bit, and normalized binary spreading code of the kth interferer, $k = 1, \ldots, K$, and $\mathbf{n}$ represents additive zero-mean channel noise. The total disturbance vector $\mathbf{z}$ is zero-mean with positive definite autocovariance matrix

$$\mathbf{R} \stackrel{\triangle}{=} \mathcal{E}\{\mathbf{z}\mathbf{z}^T\} = \sum_{k=1}^{K} P_k\mathbf{x}_k\mathbf{x}_k^T + \sigma^2\mathbf{I} \tag{30}$$

where σ^2 is the additive noise variance.

For an arbitrary spreading code $\mathbf{x} \in \{\pm\frac{1}{\sqrt{N}}\}^N$, the linear receiver $\mathbf{w}$ that exhibits maximum signal-to-noise ratio (SNR) at its output has the form

$$\mathbf{w}(\mathbf{x}) = c\mathbf{R}^{-1}\mathbf{x}, \quad c > 0 \tag{31}$$

and the maximum SNR value is

$$\mathrm{SNR}(\mathbf{x}) = \frac{\mathcal{E}\{(\mathbf{w}^T(\mathbf{x})b\sqrt{P}\mathbf{x})^2\}}{\mathcal{E}\{(\mathbf{w}^T(\mathbf{x})\mathbf{z})^2\}} = P\mathbf{x}^T\mathbf{R}^{-1}\mathbf{x}. \tag{32}$$

Therefore, optimization of the binary code $\mathbf{x}$ in the maximum $\mathrm{SNR}(\mathbf{x})$ sense is equivalent to maximization of a full-rank quadratic form with matrix parameter $\mathbf{R}^{-1}$ and binary vector argument $\mathbf{x} \in \{\pm 1\}^N$, i.e.,

$$\mathbf{x}_{\mathrm{opt}} \stackrel{\triangle}{=} \frac{1}{\sqrt{N}} \operatorname*{arg\,max}_{\mathbf{x}\in\{\pm 1\}^N} \mathbf{x}^T\mathbf{R}^{-1}\mathbf{x}. \tag{33}$$

In our study, we set the received SNR of the user of interest, $\mathrm{SNR} \stackrel{\triangle}{=} \frac{P}{\sigma^2}$, to 10 dB and the received SNRs of the K interferers, $\mathrm{SNR}_k \stackrel{\triangle}{=} \frac{P_k}{\sigma^2}, k = 1, \ldots, K$, uniformly spaced between 8 and 11

TABLE I
MATLAB CODE OF THE PROPOSED ALGORITHM

```
function X=compute_candidates(V)

% X=compute_candidates(V) returns a matrix whose columns are the corresponding
% binary vector candidates x for the maximization of x'*V*V'*x

[N D]=size(V);
if D>2
    combinations=nchoosek(1:N,D-1);
    X=zeros(N,size(combinations,1));
    for i=1:length(combinations)
        I=combinations(i,:);
        VI=V(I,:);
        c=find_intersection(VI);
        c=c*determine_sign(c);
        X(:,i)=sign(V*c);
        for d=1:D-1
            c=find_intersection([VI([1:d-1 d+1:D-1],1:D-1)]);
            c=c*determine_sign(c);
            X(I(d),i)=sign(VI(d,1:end-1)*c);
        end
    end
    X=[X compute_candidates(V(:,1:D-2))];
elseif D==1
    X=sign(V);
else
    phi_crosses=atan(-V(:,2)./V(:,1));
    [phi_sort,phi_ind]=sort(phi_crosses);
    X(phi_ind,1:N+1)=(repmat(-sign(V(phi_ind,1)),[1 N+1])).*(2*tril(ones(N,N+1))-1);
end
```

```
function c=find_intersection(V)

% c=find_intersection(V) returns the zero right singular vector of V

[junk1,junk2,C]=svd(V);
c=C(:,end);
```

```
function sign_c=determine_sign(c)

% sign_c=determine_sign(c) returns the correct sign of the hyperpolar vector c

D=length(c);
phi=zeros(D-1,1);
for phi_ind=1:D-1
    phi(phi_ind)=asin(c(phi_ind)/prod(cos(phi(1:phi_ind-1))));
end
if (phi(D-1)*c(D-1))==0
    sign_c=1;
else
    sign_c=sign(tan(phi(D-1))*c(D-1)*c(D));
end
```

dB. The interfering spreading codes are randomly generated. We compare the output SNR performance of *(i)* the *optimal* binary spreading code $\mathbf{x}_{\rm opt}$ of (33) obtained through exhaustive search over all possible N-bit combinations, *(ii)* the *rank-1-optimal* binary spreading code $\mathbf{x}_1$ obtained by applying the sign operator on the maximum-eigenvalue eigenvector of the inverse interference-plus-noise autocovariance matrix $\mathbf{R}^{-1}$ [20], [21], *(iii)* the *rank-2-optimal* binary spreading code $\mathbf{x}_2$ (which is optimal under a rank-2 approximation of $\mathbf{R}^{-1}$) obtained with complexity $\mathcal{O}(N\log N)$ by the procedure developed in [6] and [7], and *(iv)* the *rank-D-optimal* binary spreading code $\mathbf{x}_D$ (which is optimal under a rank-D approximation of $\mathbf{R}^{-1}$) obtained with complexity $\mathcal{O}(N^D), D=3,4$, by the procedure developed in Section III. For comparison purposes, we evaluate the output SNR loss, $\mathrm{SNR}(\mathbf{x}_{\rm opt})-\mathrm{SNR}(\mathbf{x}_D)$, of $\mathbf{x}_D, D=1,2,3,4$, with respect to the output SNR of the optimal binary spreading code $\mathbf{x}_{\rm opt}$. The results that we present are averages over 2 000 randomly generated interference signature-set realizations.

In Fig. 3, we plot the output SNR loss of the rank-D-optimal, $D=1,2,3,4$, binary spreading codes as a function of the number of interferers K. We are particularly interested in overloaded systems and vary K from 16 to 40 interferers. We observe that for $D\geq 3$ the proposed rank-D-optimal spreading code exhibits less than .01 dB performance loss which is significantly lower than the performance loss of the rank-1-optimal and rank-2-optimal codes (interestingly, all four loss values decrease as K increases).

In Fig. 4, we plot the probability of global, full-rank, optimality $\Pr\{\mathbf{x}_D=\mathbf{x}_{\rm opt}\}$ for the rank-D-optimal, $D=1,2,3,4$, binary spreading codes as a function of the number of interferers

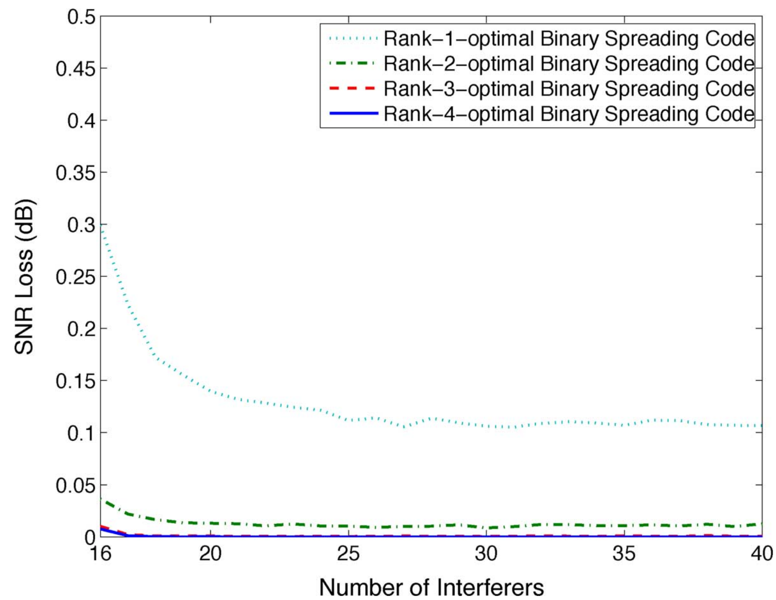

Fig. 3. SNR loss of rank-D-optimal, $D = 1, 2, 3, 4$, binary spreading code designs versus number of interferers.

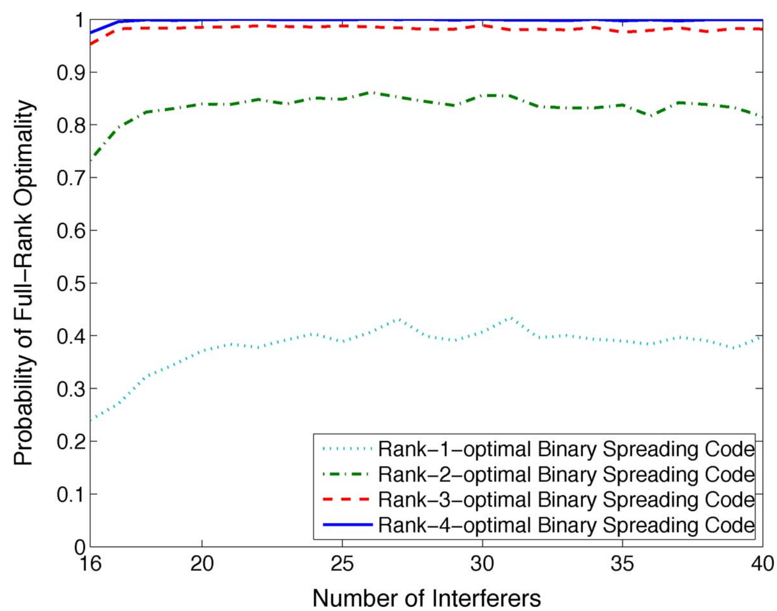

Fig. 4. Probability of full-rank optimality of rank-D-optimal, $D = 1, 2, 3, 4$, binary spreading code designs versus number of interferers.

K. We observe that $\Pr\{\mathbf{x}_4 = \mathbf{x}_{\rm opt}\} \simeq 1$ for almost all values of K between 16 and 40 interferers. Therefore, with the proposed optimization of the binary spreading code under the rank-4 approximation of $\mathbf{R}^{-1}$, we have significantly increased the probability that the designed spreading code is full-rank optimal with only $\mathcal{O}(N^4)$ additional computational cost.

V. Conclusion

We considered the problem of identifying the binary vector that maximizes a rank-deficient quadratic form. We introduced auxiliary spherical coordinates and proved that there exists a polynomial-size set of candidate binary vectors that is constructed in polynomial time and contains the optimal vector. The size of the set depends strictly on the parameter vector length and the rank of the quadratic form. When the rank of the form is constant, the size of the candidate vector set is a polynomial function of the vector length. We continued by developing an algorithm that computes the polynomial-size set of candidate vectors in polynomial time. Detailed examination of the properties of the proposed method revealed that it is time and memory efficient, fully parallelizable, and rank-scalable. Consequently, without loss of optimality, the proposed algorithm serves as an efficient alternative approach to exhaustive search and computational-geometry inspired methods. In terms of performance evaluation, we considered the optimization problem of adaptive design of binary spreading codes and showed that under reduced-rank approximation of the corresponding quadratic form the proposed algorithm attains nearly optimal performance with polynomial complexity.

APPENDIX A
PROOF OF PROPOSITION 1

Let $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N]^T$, where $\mathbf{v}_n \in \mathbb{R}^D$ and $\mathbf{v}_n^T$ denotes the nth row of $\mathbf{V}, n = 1, 2, \ldots, N$. In addition, let $\mathcal{N}(\mathbf{A})$ denote the nullspace of an $m \times k$ matrix $\mathbf{A}$, i.e.,

$$\mathcal{N}(\mathbf{A}) \stackrel{\triangle}{=} \{\mathbf{x} \in \mathbb{R}^k : \mathbf{A}\mathbf{x} = \mathbf{0}_{m\times 1}\} \tag{34}$$

and $\dim \mathcal{V}$ denote the dimension of a vector space $\mathcal{V}$. Since $\mathcal{N}(\mathbf{v}_n^T) \subset \mathbb{R}^D$ and $\dim \mathcal{N}(\mathbf{v}_n^T) = D - 1, n = 1, 2, \ldots, N$, it is implied that [22] $\bigcup_{n=1}^N \mathcal{N}(\mathbf{v}_n^T)$ is a proper subset of $\mathbb{R}^D$. Therefore, there exists a nonzero $D \times 1$ vector $\mathbf{u}$ that does not belong to $\bigcup_{n=1}^N \mathcal{N}(\mathbf{v}_n^T)$; hence, $\mathbf{v}_n^T\mathbf{u} \neq 0, \forall n = 1, 2, \ldots, N$.

Let $\{\frac{\mathbf{u}}{\|\mathbf{u}\|}, \mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{D-1}\}, \mathbf{t}_i \in \mathbb{R}^D, i = 1, 2, \ldots, D-1$, be an orthonormal basis for $\mathbb{R}^D$. Then, the $D \times D$ matrix $\mathbf{U} \stackrel{\triangle}{=} [\frac{\mathbf{u}}{\|\mathbf{u}\|}, \mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{D-1}]$ is orthonormal and $[\mathbf{V}\mathbf{U}]_{n,1} = [\mathbf{V}]_{n,:}[\mathbf{U}]_{:,1} = \mathbf{v}_n^T \frac{\mathbf{u}}{\|\mathbf{u}\|} \neq 0, \forall n = 1, 2, \ldots, N$. □

APPENDIX B
PROOF OF PROPOSITION 2

Since

$$\begin{aligned}
&\mathbf{v}^T\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) \\
&\quad= [v_1, v_2, \ldots, v_D] \\
&\qquad\times [\sin\phi_1, \cos\phi_1 \sin\phi_2, \cos\phi_1\cos\phi_2\sin\phi_3, \ldots, \\
&\qquad\cos\phi_1\cos\phi_2\ldots\sin\phi_{D-1}, \\
&\qquad\cos\phi_1\cos\phi_2\ldots\cos\phi_{D-1}]^T \\
&\quad= v_1\sin\phi_1 + [v_2, v_3, \ldots, v_D][\sin\phi_2, \cos\phi_2\sin\phi_3, \\
&\qquad\cos\phi_2\cos\phi_3\sin\phi_4, \ldots, \cos\phi_2\cos\phi_3\ldots\sin\phi_{D-1}, \\
&\qquad\cos\phi_2\cos\phi_3\ldots\cos\phi_{D-1}]^T\cos\phi_1 \\
&\quad= v_1\sin\phi_1 + \mathbf{v}_{2:D}^T\mathbf{c}(\boldsymbol{\phi}_{2:D-1})\cos\phi_1
\end{aligned} \tag{35}$$

the decision rule $x(\mathbf{v}^T;\boldsymbol{\phi}_{1:D-1}) = \operatorname{sgn}(\mathbf{v}^T\mathbf{c}(\boldsymbol{\phi}_{1:D-1}))$ becomes

$$v_1\sin\phi_1 + \mathbf{v}_{2:D}^T\mathbf{c}(\boldsymbol{\phi}_{2:D-1})\cos\phi_1 \underset{x(\mathbf{v}^T;\boldsymbol{\phi}_{1:D-1})=-1}{\overset{x(\mathbf{v}^T;\boldsymbol{\phi}_{1:D-1})=1}{\gtrless}} 0. \tag{36}$$

Recall that we have assumed w.l.o.g. that $v_1 \neq 0$. If $\phi_1 = \frac{\pi}{2}$, then (36) becomes

$$v_1 \underset{x(\mathbf{v}^T;\boldsymbol{\phi}_{1:D-1})=-1}{\overset{x(\mathbf{v}^T;\boldsymbol{\phi}_{1:D-1})=1}{\gtrless}} 0 \Leftrightarrow x(\mathbf{v}^T;\boldsymbol{\phi}_{1:D-1}) = \operatorname{sgn}(v_1). \tag{37}$$

Otherwise, $-\frac{\pi}{2} < \phi_1 < \frac{\pi}{2}$; hence, $\cos\phi_1 > 0$ and (36) becomes

$$\begin{aligned}
&v_1\tan\phi_1 \underset{x(\mathbf{v}^T;\boldsymbol{\phi}_{1:D-1})=-1}{\overset{x(\mathbf{v}^T;\boldsymbol{\phi}_{1:D-1})=1}{\gtrless}} -\mathbf{v}_{2:D}^T\mathbf{c}(\boldsymbol{\phi}_{2:D-1}) \\
&\overset{v_1\neq 0}{\Longleftrightarrow} \tan\phi_1 \underset{x(\mathbf{v}^T;\boldsymbol{\phi}_{1:D-1})=-\operatorname{sgn}(v_1)}{\overset{x(\mathbf{v}^T;\boldsymbol{\phi}_{1:D-1})=\operatorname{sgn}(v_1)}{\gtrless}} -\frac{\mathbf{v}_{2:D}^T\mathbf{c}(\boldsymbol{\phi}_{2:D-1})}{v_1} \\
&\Longleftrightarrow \phi_1 \underset{x(\mathbf{v}^T;\boldsymbol{\phi}_{1:D-1})=-\operatorname{sgn}(v_1)}{\overset{x(\mathbf{v}^T;\boldsymbol{\phi}_{1:D-1})=\operatorname{sgn}(v_1)}{\gtrless}} \tan^{-1}\left(-\frac{\mathbf{v}_{2:D}^T\mathbf{c}(\boldsymbol{\phi}_{2:D-1})}{v_1}\right) \\
&\Longleftrightarrow x(\mathbf{v}^T;\boldsymbol{\phi}_{1:D-1}) \\
&= \begin{cases} -\operatorname{sgn}(v_1), & -\frac{\pi}{2} < \phi_1 < \tan^{-1}\left(-\frac{\mathbf{v}_{2:D}^T\mathbf{c}(\boldsymbol{\phi}_{2:D-1})}{v_1}\right) \\ \operatorname{sgn}(v_1), & \tan^{-1}\left(-\frac{\mathbf{v}_{2:D}^T\mathbf{c}(\boldsymbol{\phi}_{2:D-1})}{v_1}\right) < \phi_1 < \frac{\pi}{2}. \end{cases}
\end{aligned} \tag{38}$$

□

APPENDIX C
PROOF OF PROPOSITION 3, PART (A)

Consider $\mathcal{I}_{D-1} = \{i_1, i_2, \ldots, i_{D-1}\}$ and the $D-1$ hypersurfaces $\mathcal{S}(\mathbf{V}_{i_1,1:D}), \mathcal{S}(\mathbf{V}_{i_2,1:D}), \ldots, \mathcal{S}(\mathbf{V}_{i_{D-1},1:D})$ that correspond to $D-1$ rows of $\mathbf{V}_{N\times D}$. Since each hypersurface $\mathcal{S}(\mathbf{V}_{i,1:D})$ is described by the equation $\mathbf{V}_{i,1:D}\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) = 0, i \in \{i_1, i_2, \ldots, i_{D-1}\}$, their intersection(s) will satisfy the system of equations

$$\left\{\begin{array}{c} \mathbf{V}_{i_1,1:D}\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) = 0 \\ \mathbf{V}_{i_2,1:D}\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) = 0 \\ \vdots \\ \mathbf{V}_{i_{D-1},1:D}\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) = 0 \end{array}\right\}. \tag{39}$$

The above system is rewritten as $\mathbf{V}_{\mathcal{I}_{D-1},1:D}\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) = \mathbf{0}$. Therefore, the solution $\boldsymbol{\phi}_{1:D-1}$ is such that $\mathbf{c}(\boldsymbol{\phi}_{1:D-1})$ belongs to the null space of $\mathbf{V}_{\mathcal{I}_{D-1},1:D}$ which is denoted by $\mathcal{N}(\mathbf{V}_{\mathcal{I}_{D-1},1:D})$ and has dimension greater than or equal to one, since $\operatorname{rank}(\mathbf{V}_{\mathcal{I}_{D-1},1:D}) \leq D-1$. Let $\mathbf{V}_{\mathcal{I}_{D-1},1:D} = \tilde{\mathbf{U}}_{(D-1)\times(D-1)}\boldsymbol{\Lambda}_{(D-1)\times D}\mathbf{U}_{D\times D}^T$ be the singular value decomposition of $\mathbf{V}_{\mathcal{I}_{D-1},1:D}$, where $\tilde{\mathbf{U}}$ and $\mathbf{U}$ are orthogonal matrices, $\boldsymbol{\Lambda} = [\operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{D-1}), \mathbf{0}_{(D-1)\times 1}]$, and w.l.o.g. $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{D-1} \geq 0$.

We consider two cases.

i) If $\lambda_{D-1} > 0$, then $\mathcal{N}(\mathbf{V}_{\mathcal{I}_{D-1},1:D}) = \{a\mathbf{U}_{1:D,D} : a \in \mathbb{R}\}$, which implies that $\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) = \frac{\mathbf{U}_{1:D,D}}{\|\mathbf{U}_{1:D,D}\|}$ or $\mathbf{c}(\boldsymbol{\phi}_{1:D-1}) = -\frac{\mathbf{U}_{1:D,D}}{\|\mathbf{U}_{1:D,D}\|}$. Since we require $\phi_1 \in (-\frac{\pi}{2}, \frac{\pi}{2}]$, only one solution $\pm\frac{\mathbf{U}_{1:D,D}}{\|\mathbf{U}_{1:D,D}\|}$ is valid and the spherical coordinate vector we look for is uniquely determined by the spherical coordinates of $\frac{\mathbf{U}_{1:D,D}}{\|\mathbf{U}_{1:D,D}\|}$ or $-\frac{\mathbf{U}_{1:D,D}}{\|\mathbf{U}_{1:D,D}\|}$.

ii) If $\lambda_{D-1} = 0$, then $\dim \mathcal{N}(\mathbf{V}_{\mathcal{I}_{D-1},1:D}) \geq 2$ which implies that there are uncountably many solutions for $\mathbf{c}(\boldsymbol{\phi}_{1:D-1})$ that satisfy the requirement $\phi_1 \in (-\frac{\pi}{2}, \frac{\pi}{2}]$. □

APPENDIX D
PROOF OF PROPOSITION 3, PART (B)

i)

$$\begin{aligned}
&\mathbf{x}\left(\mathbf{V}_{N\times D}; \boldsymbol{\phi}_{1:D-2}, \frac{\pi}{2}\right)\\
&= \operatorname{sgn}\Big(\mathbf{V}_{N\times D}[\sin\phi_1, \cos\phi_1 \sin\phi_2, \ldots,\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-2} \sin\frac{\pi}{2},\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-2} \cos\frac{\pi}{2}\Big]^T\Big)\\
&= \operatorname{sgn}\Big(\mathbf{V}_{N\times(D-1)}[\sin\phi_1, \cos\phi_1 \sin\phi_2, \ldots,\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-2}]^T\Big)\\
&= \mathbf{x}(\mathbf{V}_{N\times(D-1)}; \boldsymbol{\phi}_{1:D-2}).
\end{aligned}$$

ii)

$$\begin{aligned}
&\mathbf{x}\left(\mathbf{V}_{N\times D}; \boldsymbol{\phi}_{1:D-2}, -\frac{\pi}{2}\right)\\
&= \operatorname{sgn}\Big(\mathbf{V}_{N\times D}\Big[\sin\phi_1, \cos\phi_1 \sin\phi_2, \ldots,\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-2} \sin\left(-\frac{\pi}{2}\right),\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-2} \cos\left(-\frac{\pi}{2}\right)\Big]^T\Big)\\
&= \operatorname{sgn}\Big(\mathbf{V}_{N\times D}\Big[-\sin(-\phi_1),\\
&\quad -\cos(-\phi_1)\sin(-\phi_2), \ldots,\\
&\quad -\cos(-\phi_1) \ldots \cos(-\phi_{D-2}) \sin\frac{\pi}{2},\\
&\quad -\cos(-\phi_1) \ldots \cos(-\phi_{D-2}) \cos\frac{\pi}{2}\Big]^T\Big)\\
&= -\mathbf{x}\left(\mathbf{V}_{N\times D}; -\boldsymbol{\phi}_{1:D-2}, \frac{\pi}{2}\right).
\end{aligned}$$

iii)

$$\begin{aligned}
&\mathbf{x}\left(\mathbf{V}_{N\times D}; \boldsymbol{\phi}_{1:D-3}, \frac{\pi}{2}, \phi_{D-1}\right)\\
&= \operatorname{sgn}\Big(\mathbf{V}_{N\times D}\Big[\sin\phi_1, \cos\phi_1 \sin\phi_2, \ldots,\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-3} \sin\frac{\pi}{2},\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-3} \cos\frac{\pi}{2} \sin\phi_{D-1},\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-3} \cos\frac{\pi}{2} \cos\phi_{D-1}\Big]^T\Big)\\
&= \operatorname{sgn}(\mathbf{V}_{N\times(D-2)}[\sin\phi_1,\\
&\quad \cos\phi_1 \sin\phi_2, \ldots, \cos\phi_1 \ldots \cos\phi_{D-3}]^T)\\
&= \mathbf{x}(\mathbf{V}_{N\times(D-2)}; \boldsymbol{\phi}_{1:D-3}).
\end{aligned}$$

iv)

$$\begin{aligned}
&\mathbf{x}\left(\mathbf{V}_{N\times D}; \boldsymbol{\phi}_{1:D-3}, -\frac{\pi}{2}, \phi_{D-1}\right)\\
&= \operatorname{sgn}\Big(\mathbf{V}_{N\times D}\Big[\sin\phi_1, \cos\phi_1 \sin\phi_2, \ldots,\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-3} \sin\left(-\frac{\pi}{2}\right),\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-3} \cos\left(-\frac{\pi}{2}\right) \sin\phi_{D-1},\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-3} \cos\left(-\frac{\pi}{2}\right) \cos\phi_{D-1}\Big]^T\Big)\\
&= \operatorname{sgn}\Big(\mathbf{V}_{N\times(D-2)}\Big[-\sin(-\phi_1),\\
&\quad -\cos(-\phi_1)\sin(-\phi_2), \ldots,\\
&\quad -\cos(-\phi_1) \ldots \cos(-\phi_{D-3})\Big]^T\Big)\\
&= \operatorname{sgn}\Big(\mathbf{V}_{N\times D}\Big[-\sin(-\phi_1),\\
&\quad -\cos(-\phi_1)\sin(-\phi_2), \ldots,\\
&\quad -\cos(-\phi_1) \ldots \cos(-\phi_{D-3}) \sin\frac{\pi}{2},\\
&\quad -\cos(-\phi_1) \ldots \cos(-\phi_{D-3}) \cos\frac{\pi}{2} \sin\phi'_{D-1},\\
&\quad -\cos(-\phi_1) \ldots \cos(-\phi_{D-3}) \cos\frac{\pi}{2} \cos\phi'_{D-1}\Big]^T\Big)\\
&= -\mathbf{x}\left(\mathbf{V}_{N\times D}; -\boldsymbol{\phi}_{1:D-3}, \frac{\pi}{2}, \phi'_{D-1}\right).
\end{aligned}$$

v)

$$\begin{aligned}
&\mathbf{x}\left(\mathbf{V}_{N\times D}; \boldsymbol{\phi}_{1:D-3}, \pm\frac{\pi}{2}, \phi_{D-1}\right)\\
&= \operatorname{sgn}\Big(\mathbf{V}_{N\times D}\Big[\sin\phi_1, \cos\phi_1 \sin\phi_2, \ldots,\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-3} \sin\left(\pm\frac{\pi}{2}\right),\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-3} \cos\left(\pm\frac{\pi}{2}\right) \sin\phi_{D-1},\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-3} \cos\left(\pm\frac{\pi}{2}\right) \cos\phi_{D-1}\Big]^T\Big)\\
&= \operatorname{sgn}(\mathbf{V}_{N\times(D-2)}[\sin\phi_1, \cos\phi_1 \sin\phi_2, \ldots,\\
&\quad \pm\cos\phi_1 \ldots \cos\phi_{D-3}]^T)\\
&= \operatorname{sgn}\Big(\mathbf{V}_{N\times D}\Big[\sin\phi_1, \cos\phi_1 \sin\phi_2, \ldots,\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-3} \sin\left(\pm\frac{\pi}{2}\right),\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-3} \cos\left(\pm\frac{\pi}{2}\right) \sin\phi'_{D-1},\\
&\quad \cos\phi_1 \ldots \cos\phi_{D-3} \cos\left(\pm\frac{\pi}{2}\right) \cos\phi'_{D-1}\Big]^T\Big)\\
&= \mathbf{x}(\mathbf{V}_{N\times D}; \boldsymbol{\phi}_{1:D-3}, \pm\frac{\pi}{2}, \phi'_{D-1}).
\end{aligned}$$

□

References

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

[2] M. Ajtai, "The shortest vector problem in L_2 is NP-hard for randomized reductions," in *Proc. 30th Annu. ACM Symp. Theory of Computing*, Dallas, TX, May 1998, pp. 10–19.

[3] K. Allemand, K. Fukuda, T. M. Liebling, and E. Steiner, "A polynomial case of unconstrained zero-one quadratic optimization," *Math. Programm. A*, vol. 91, pp. 49–52, Oct. 2001.

[4] J.-A. Ferrez, K. Fukuda, and T. M. Liebling, "Solving the fixed rank convex quadratic maximization in binary variables by a parallel zonotope construction algorithm," *Eur. J. Oper. Res.*, vol. 166, pp. 35–50, 2005.

[5] G. Manglani and A. K. Chaturvedi, "Application of computational geometry to multiuser detection in CDMA," *IEEE Trans. Commun.*, vol. 54, no. 2, pp. 204–207, Feb. 2006.

[6] G. N. Karystinos and D. A. Pados, "Rank-2-optimal binary spreading codes," in *Proc. Conf. Inform. Sci. and Syst.*, Princeton, NJ, Mar. 2006, pp. 1534–1539.

[7] G. N. Karystinos and D. A. Pados, "Rank-2-optimal adaptive design of binary spreading codes," *IEEE Trans. Inf. Theory*, vol. 53, no. 9, pp. 3075–3080, Sep. 2007.

[8] H. Edelsbrunner, J. O'Rourke, and R. Seidel, "Constructing arrangements of lines and hyperplanes with applications," *SIAM J. Comput.*, vol. 15, pp. 341–363, May 1986.

[9] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*. New York: Springer-Verlag, 1987.

[10] D. Avis and K. Fukuda, "Reverse search for enumeration," *Discrete Appl. Math.*, vol. 65, no. 3, pp. 21–46, Mar. 1996.

[11] V. Pauli, L. Lampe, R. Schober, and K. Fukuda, "Multiple-symbol differential detection based on combinatorial geometry," *IEEE Trans. Commun.*, vol. 56, no. 10, pp. 1596–1600, Oct. 2008.

[12] I. Motedayen-Aval, A. Krishnamoorthy, and A. Anastasopoulos, "Optimal joint detection/estimation in fading channels with polynomial complexity," *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 209–223, Jan. 2007.

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

[14] D. S. Papailiopoulos and G. N. Karystinos, "Polynomial-complexity maximum-likelihood block noncoherent MPSK detection," in *Proc. IEEE ICASSP*, Las Vegas, NV, Apr. 2008, pp. 2681–2684.

[15] P. H. Tan and L. K. Rasmussen, "The application of semidefinite programming for detection in CDMA," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 8, pp. 1442–1449, Aug. 2001.

[16] M. Abdi, H. El Nahas, A. Jard, and E. Moulines, "Semidefinite positive relaxation of the maximum-likelihood criterion applied to multiuser detection in a CDMA context," *IEEE Signal Process. Lett.*, vol. 9, no. 6, pp. 165–167, Jun. 2002.

[17] X. M. Wang, W.-S. Lu, and A. Antoniou, "A near-optimal multiuser detector for DS-CDMA systems using semidefinite programming relaxation," *IEEE Trans. Signal Process.*, vol. 51, no. 9, pp. 2446–2450, Sep. 2003.

[18] W.-K. Ma, B.-N. Vo, T. N. Davidson, and P.-C. Ching, "Blind ML detection of orthogonal space-time block codes: Efficient high-performance implementations," *IEEE Trans. Signal Process.*, vol. 54, no. 2, pp. 738–751, Feb. 2006.

[19] D. S. Papailiopoulos and G. N. Karystinos, "Maximum-likelihood noncoherent OSTBC detection with polynomial complexity," *IEEE Trans. Wireless Commun.* [Online]. Available: http://www.telecom.tuc.gr/~karystinos/paper_TW2.pdf

[20] T. F. Wong and T. M. Lok, "Transmitter adaptation in multicode DS-CDMA systems," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 1, pp. 69–82, Jan. 2001.

[21] C. W. Sung and H. Y. Kwan, "Heuristic algorithms for binary sequence assignment in DS-CDMA systems," in *Proc. IEEE Int. Symp. Personal, Indoor and Mobile Radio Commun.*, Lisbon, Portugal, Sep. 2002, vol. 5, pp. 2327–2331.

[22] S. Roman, *Advanced Linear Algebra*. New York: Springer-Verlag, 1992.

George N. Karystinos (S'98–M'03) was born in Athens, Greece, on April 12, 1974. He received the Diploma degree in computer science and engineering (five-year program) from the University of Patras, Patras, Greece, in 1997 and the Ph.D. degree in electrical engineering from the State University of New York at Buffalo, Amherst, in 2003.

In August 2003, he joined the Department of Electrical Engineering, Wright State University, Dayton, OH, as an Assistant Professor. Since September 2005, he has been an Assistant Professor with the Department of Electronic and Computer Engineering, Technical University of Crete, Chania, Greece. His current research interests are in the general areas of communication theory and adaptive signal processing with an emphasis on wireless and cooperative communications systems, low-complexity data detection, optimization with low complexity and limited data, spreading code and signal waveform design, and sparse principal component analysis.

Dr. Karystinos received a 2001 IEEE International Conference on Telecommunications best paper award and the 2003 IEEE Transactions on Neural Networks Outstanding Paper Award. He is a member of the IEEE Communications, Signal Processing, Information Theory, and Computational Intelligence Societies and a member of Eta Kappa Nu.

Athanasios P. Liavas (S'89–M'93) received the Diploma and Ph.D. degrees from the University of Patras, Greece, in 1989 and 1993, respectively.

From 1993 to 1995, he served in the Greek Army. From 1996 to 1998, he was a Marie Curie postdoctoral Fellow at the Institut National des Telecommunications, Evry, France. From 1999 to 2001, he was visiting an Assistant Professor at the Department of Informatics, University of Ioannina, Greece. In 2001, he became an Assistant Professor at the Department of Mathematics, University of the Aegean, Greece. In 2004, he joined the Department of Electronic and Computer Engineering, Technical University of Crete, Chania, Greece, as an Associate Professor. In September 2009, he became a Professor and Department Chair. His current research interests lie in the areas of signal processing for communications and information theory.

Dr. Liavas served as Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2005 to 2009. He is an elected member of the IEEE SPCOM Technical Committee (first election 2006, re-elected 2009).