Enhancing low-rank solutions in semidefinite relaxations of Boolean quadratic problems

V. Cerone, S. M. Fosson*, D. Regruto

March 20, 2020

Abstract

Boolean quadratic optimization problems occur in a number of applications. Their mixed integer-continuous nature is challenging, since it is inherently NP-hard. For this motivation, semidefinite programming relaxations (SDR’s) are proposed in the literature to approximate the solution, which recasts the problem into convex optimization. Nevertheless, SDR’s do not guarantee the extraction of the correct binary minimizer. In this paper, we present a novel approach to enhance the binary solution recovery. The key of the proposed method is the exploitation of known information on the eigenvalues of the desired solution. As the proposed approach yields a non-convex program, we develop and analyze an iterative descent strategy, whose practical effectiveness is shown via numerical results.

1 Introduction

Boolean quadratic optimization refers to those minimization problems with quadratic cost functional and binary variables. It is a long-time problem with a number of applications in different scientific areas. To mention some examples, it is encountered in maximum cut (MAX-CUT) problems in graphs, see, [23], knapsack problems, see [13], hybrid model predictive control, see [21], sensor selection, see [27, 2], medical imaging, see [26, 30], and binary compressed sensing, see [8].

Boolean quadratic optimization is challenging as it is NP-hard, even when the cost functional is convex, due to the integer nature of the variables. To overcome this drawback, different relaxations are proposed in the literature that approximate the correct solution. The most popular approach consists in the semidefinite programming relaxation (SDR), known also as Shor relaxation, introduced by [28] and [19]. In a nutshell, the key idea at the basis of SDR is to embed the variable matrix \(xx^T\), \(x \in \mathbb{R}^n\), into the space of \(n \times n\) symmetric positive semidefinite matrices. The rationale is that a complete description of the cone of symmetric positive semidefinite matrices is available, while this does not hold for the convex polytope of matrices \(xx^T\), as illustrated, e.g., by [22]. The tightness of SDR is analyzed, e.g., in [10, 22].

More recently, a substantial evolution on SDR has been provided by the Lasserre’s theory, which states that the global optimum of a polynomial optimization problem can be achieved by solving a hierarchy of SDR’s; we refer
the reader to [14, 16] for details, and to [15] for the specific case of quadratic functionals. Theorem 4.2 in [14] and further elaborations in Chapter 6 of [16] provide sufficient conditions to get the global minimum and a global minimizer of the original problem from an SDR of a certain relaxation order. As to the Shor relaxation (or equivalently, SDR of order 1) of a Boolean quadratic problem, if the SDR solution is 1-rank, then the global minimum is achieved and the desired binary minimizer can be extracted, as discussed in [8]. For this motivation, methods to minimize or reduce the rank in semidefinite programming are relevant for SDR; we refer to [15] for a complete overview on the topic. Since the rank minimization is an NP-hard problem, the minimization of the nuclear norm is often exploited, which provides the tightest convex relaxation of the problem, as proven by [23]. As to symmetric positive semidefinite matrices, the nuclear norm is equal to the trace, and in turn to the sum of the eigenvalues: then, the minimization of the nuclear norm has a sparsifying effect on the eigenvalues vector, as the \( \ell_1 \)-norm is a suitable convex relaxation of the sparsity a vector, see, e.g., [29]. Then, a nuclear norm penalty encourages a low-rank solution. Nevertheless, this method does not apply to some Boolean quadratic problems, e.g., MAX-CUT, see [17], where the trace is constant. For these problems, an effective method is the log-det heuristic, proposed by [3].

In this paper, we propose a novel method to enhance SDR for Boolean quadratic optimization, by supporting 1-rank solutions. The key idea is to exploit the knowledge of the eigenvalues of the desired solution are known (in particular, only one eigenvalue is non-null). Then, we promote 1-rank solutions by maximizing the energy of the eigenvalues vector, which we prove to have a sparsifying effect on the eigenvalues, when the trace is constant. We illustrate the proposed approach for two SDR’s, with slightly different spectral properties. Furthermore, as the method yields a non-linear and non-convex program, a suitable iterative descent algorithm is developed to search the global minimum. The overall strategy is analyzed, and its effectiveness is illustrated through numerical simulations.

The paper is organized as follows. In Sec. 2 we formally state the problem. In Sec. 3 we present and analyze the proposed approach. In Sec. 4 we show some numerical results; finally, we draw some conclusions in Sec. 5.

2 Problem Statement

Let \( \mathbb{S}^n \) be the space of \( n \times n \) symmetric matrices, and \( \mathbb{S}^n_+ \subset \mathbb{S}_n \) the subspace of positive semidefinite matrices. We consider Boolean quadratic problems of the kind

\[
\min_{x \in \{0,1\}^n} x^T C x + 2d^T x \quad \text{s. t.} \quad Ax = b
\]  

(1)

where \( C \in \mathbb{S}^n \), \( d \in \mathbb{R}^n \), \( A \in \mathbb{R}^{m,n} \), and \( b \in \mathbb{R}^m \). As in [17], we exploit the Lagrangian formulation

\[
\min_{x \in \{0,1\}^n} x^T C x + 2d^T x + \mu \|Ax - b\|_2^2
\]  

(2)

where \( \mu > 0 \). If \( \mu \) is sufficiently large, problems (1) and (2) are equivalent, see [17]. By adding a slack variable \( x_0 = 1 \) and by redefining \( x = (1, x_1, \ldots, x_n)^T \),
we rewrite (2) as the augmented problem

\[
\min_{x \in \{0,1\}^{n+1}} x^T Q x
\]

(3)

where \(Q = \begin{pmatrix} 0 & d^T \\ d & C \end{pmatrix} + \mu (-b|A)^T (-b|A)\). The solution to this NP-hard problem can be searched by solving the associated SDR, see [22] for details:

\[
\min_{X \in S^{n+1}_+} \langle Q, X \rangle \\
\text{s.t.} \quad X_{0,0} = 1; \quad X_{i,i} = X_{0,i}, \quad i = 1, \ldots, n.
\]

(4)

The constraint \(X_{i,i} = X_{0,i}\) represents the constraint \(x_i = x^2_i\), which holds for any \(x_i \in \{0,1\}\).

An alternative, MAX-CUT approach is studied in [17]: given \(1 = (1, \ldots, 1)^T \in \mathbb{R}^{n+1}\), by substituting \(z = 2x - 1 \in \{-1,1\}^{n+1}\) in (3), we obtain \(x^T Q x = \frac{1}{4}(z^2 + 1)^T Q (z + 1)\). Then, problem (3) is equivalent to

\[
\min_{z \in \{-1,1\}^{n+1}} z^T R z
\]

(5)

where \(R = Q + \begin{pmatrix} 0 & 1^T Q \\ Q^T & 0 \end{pmatrix}\). Problem (5) is a MAX-CUT problem, and the associated SDR is

\[
\min_{Z \in S^{n+1}_+} \langle R, Z \rangle \\
\text{s.t.} \quad Z_{i,i} = 1, \quad i = 0, \ldots, n.
\]

(6)

The constraint \(Z_{i,i} = 1\) represents the fact that \(z^2_i = 1\) whenever \(z_i \in \{-1,1\}\).

The aim of this paper is the recovery of the correct binary minimizers of problems (3) and (5) by starting from their SDR’s (4) and (6). As described, e.g., in [17] and [18], the correct binary minimizers can be extracted if and only if the SDR solutions are 1-rank. Therefore, we develop a strategy to reduce, possibly minimize, the rank, by exploiting specific features of the Boolean setting.

### 3 Concave penalization exploiting information on the eigenvalues

In this section, we develop the proposed strategy to promote 1-rank solutions to SDR’s (4) and (6). Specifically, we propose a suitable cost functional, and we illustrate the algorithms used to minimize it.

#### 3.1 Proposed cost functional

Let us consider SDR (4), and let us call \(Z^* \in S^{n+1}_+\) the desired 1-rank solution. The rank of \(Z^*\) corresponds to the number of non-null eigenvalues; thus, the eigenvalues vector of \(Z^*\), denoted as \(v\), is 1-sparse, i.e., \(v = (v_0, 0, \ldots, 0), v_0 > 0\). We also remark that \(Z^*\) necessarily has all the components in \([-1,1]\), and \(\frac{1}{2}((Z^*_{0,1}, \ldots, Z^*_{0,n})^T + 1)\) is the exact minimizer of (2). [25, 18] show that a
sparsifying effect on \(v\) can be obtained by penalizing the trace of the variable \(Z\), hereafter denoted as \(\text{tr}(Z)\), which corresponds to the \(\ell_1\)-norm of the eigenvalues, and, in turn, the \(\ell_1\)-norm is a suitable convex relaxation of the sparsity of a vector. Nevertheless, this approach is not beneficial for SDR (6), as \(\text{tr}(Z) = n + 1\) by construction; therefore, it makes no sense to penalize a constant quantity.

However, we observe that the information \(\text{tr}(Z^*) = \sum_{i=0}^{n} v_i = n + 1\) can be exploited to state that \(v_0 = n + 1\). In other terms, not only we know that \(v\) is 1-sparse, but also that its components belong to the binary alphabet \(\{0, n + 1\}\). Then, we wonder how to force the solution to have \(v\) exploited to state that

\[
\begin{align*}
\text{min} & \quad \langle R, Z \rangle - \lambda \langle Z, Z \rangle \\
\text{s. t.} & \quad Z_{i,1} = 1, \quad i = 0, \ldots, n
\end{align*}
\]

where \(\lambda > 0\) is a design parameter that can be assessed by cross-validation. Interestingly, if a global minimizer of (6) is binary, then, for any \(\lambda > 0\), the global minimizer of (7) is exact, as illustrated in the following proposition.

**Proposition 1** Let \(\tilde{z} \in \{-1, 1\}^{n+1}\) be the correct solution to problem (5). Let us assume that SDR (4) has the desired binary, 1-rank solution \(\tilde{z}z^T\) among its global minimizers. Then, the minimizer of (7) is \(Z^* = \tilde{z}z^T\), for any \(\lambda > 0\). Moreover, \(\tilde{z} = (1, Z_{1,0}^*, \ldots, Z_{n,0}^*)^T\).

**Proof** Since \(\text{tr}(Z) = n + 1\), then \(\langle Z, Z \rangle \leq (n + 1)^2\). In particular, the maximum \((n + 1)^2\) is achieved if and only if the eigenvalues of \(Z\) are \(v = (n + 1, 0, \ldots, 0)\). In this case, \(Z\) is 1-rank, and necessarily \(Z = \tilde{z}z^T\). Moreover, since by assumption \(\tilde{z}z^T\) minimizes \(\langle R, Z \rangle\), then it is the unique global minimizer of (7).

A similar approach can be applied to SDR (4), with a considerable difference: \(\text{tr}(X)\) is not priorly set by construction. Then, we proceed by considering two possible settings.

In the first setting, we assume to know the sparsity level \(k\) of the true solution of problem (3). This implies that the desired solution to SDR (4) has trace equal to \((k + 1)^2\) and eigenvalues \(v = (k + 1, 0, \ldots, 0)\). The sparsity level is known in many applications, e.g., in compressed sensing, see [9, Sec. III], or in sensor selection, where \(k\) is the number of used sensors, see [27]. In other cases, an unknown \(k\) can be estimated through ad hoc techniques, see [24]. Given \(k\), we propose to modify SDR (4) as follows to leverage the binary nature of the eigenvalues:

\[
\begin{align*}
\min & \quad \langle Q, X \rangle + \lambda [(k + 1)\text{tr}(X) - \langle X, X \rangle] \\
\text{s. t.} & \quad X_{0,0} = 1; \quad X_{i,i} = X_{0,i}, \quad i = 1, \ldots, n.
\end{align*}
\]

(8)
Proposition 2 Let $\tilde{x} \in \{0,1\}^{n+1}$ be the correct solution to problem (3). Let us assume that SDR (4) has the desired binary, 1-rank solution $\tilde{x}\tilde{x}^T$ among its global minimizers. Then, the minimizer of problem (8) is $X^* = \tilde{x}\tilde{x}^T$, for any $\lambda > 0$. Moreover, $\tilde{x} = (1, X^*_1, 0, \ldots, X^*_n, 0)^T$.

Proof If $v$ is the eigenvalues vector of $X$, we have

$$\text{tr}(X) - \langle X, X \rangle = \sum_{i=1}^{n} [(k+1)v_i - v_i^2] \geq 0 \quad (9)$$

since $v_i \in [0, k+1]$ for each $i = 1, \ldots, n$. Moreover, $\text{tr}(X) - \langle X, X \rangle = 0 \iff v_i \in [0, k+1]$ for each $i = 1, \ldots, n$. Then, we can conclude that the global minimum is achieved for $v \in \{0, k+1\}^{n+1}$. Eventually, since $\sum_{i=1}^{n} v_i = k+1$, we have $v = (k+1, 0, \ldots, 0)$, which implies the 1-rank solution, and $X^* = \tilde{x}\tilde{x}^T$. □

In the second setting, we assume that $k$ is unknown. In this case, we propose to replace $k$ by $n \geq k$. This approximate procedure takes a larger weight on the term $\text{tr}(X)$, which might be advantageous when $k \ll n$: in fact, in this setting, not only $\text{tr}(X^*) = \sum_{i=1}^{n+1} v_i$, but also $\text{tr}(X^*) = 1 + \sum_{i=1}^{n} x_i^*$, where $x^* \in [0,1]^n$ is the final estimation of $\tilde{x}$, extracted from the diagonal of $X^* = (X^*_1, \ldots, X^*_n)^T$. In conclusion, by penalizing $\text{tr}(X)$ we obtain a sparsifying effect both on the eigenvalues and on the solution; for this motivation, we expect better performance when $k \ll n$.

Remark 1 The considered problem bears some similarities with phase retrieval from Fourier transform magnitude, see, e.g., [11, 12]. An effective approach to phase retrieval is the embedding of the unknown vector $x$ into a higher dimensional space by the transformation $X = xx^T$. This approach, called lifting, shares the same principle of the SDR’s illustrated in this paper. Similarly to the MAX-CUT problem, the penalization of the trace is not effective for phase retrieval, as the energy of $x$ is fixed, see [11, Sec. III]. In [11], this issue is overcome by penalizing the term $\log \det(X + \epsilon I)$, where $\epsilon > 0$ is a small design parameter necessary for boundedness. This term is a concave surrogate of the rank, whose effectiveness is discussed in [3]. Our approach consists in a concave penalization as well. However, differently from the log-det heuristic, it exploits the known binary eigenvalues; moreover, its practical implementation is computationally less complex, as illustrated in Remark 2.

3.2 Descent algorithms

Problems (8) and (7) are well posed, that is, their unique global minima are the correct binary solutions. However, they introduce the concave term $-\langle X, X \rangle$, which makes the problem non-linear and non-convex. For this motivation, we propose an iterative descent algorithm to search the minimum, which, although sub-optimal, is effective in practice. The idea is to replace the concave term $\langle X, X \rangle$ with $\langle G, X \rangle$, where $G$ is an available estimate of $X$. By assuming $G$ fixed, the penalty is linear in $X$, then the whole cost functional is linear. Then, we propose an iterative procedure: we start from an initial estimate of the solution, denoted by $X_0$, which can be assessed, for example, by solving the non-penalized problems (4) and (6); in turn, we solve the penalized problems.
until convergence is reached. The overall procedure is summarized in algorithms 1 and 2 for problems (8) and (7), respectively. In Algorithm 1, \( h = k + 1 \) when \( k \) is known, and \( n + 1 \) otherwise.

**Algorithm 1** Descent algorithm for Problem (8)

**Input:** \( Q, \lambda > 0 \);

1: \( X_0 = \) solution of (4)

2: for all \( t = 1, \ldots, T \) do

3: \( X_t = \arg\min_{X \in S_n^{+1}} \langle Q, X \rangle + \lambda [h \text{tr}(X) - \langle X_{t-1}, X \rangle], \) s. t. \( X_{0,0} = 1, \ X_{i,i} = X_0, \ i = 1, \ldots, n \)

4: end for

**Algorithm 2** Descent algorithm for Problem (7)

**Input:** \( R, \lambda > 0 \);

1: \( X_0 = \) solution of (6)

2: for all \( t = 1, \ldots, T \) do

3: \( X_t = \arg\min_{X \in S_n^{+1}} \langle R, X \rangle - \lambda \langle X_{t-1}, X \rangle, \) s. t. \( X_{i,i} = 1, \ i = 0, \ldots, n \)

4: end for

In this way, we solve a sequence of semidefinite programming problems. This is not guaranteed to get the global minimum, while it is guaranteed to provide a non-increasing cost functional sequence.

**Proposition 3** Let us define \( f(X) := \langle Q, X \rangle + \lambda [h \text{tr}(X) - \langle X_{t-1}, X \rangle], \) and \( F(X_t) := f(X_t) - \frac{\lambda}{2} \langle X_t, X_t - 1 \rangle. \)

By applying Algorithm 1 (respectively, Algorithm 2), \( F(X_t) \) is a non-increasing function.

**Proof** Since \( X_t \) is the minimizer,

\[
f(X_t) - \lambda \langle X_{t-1}, X_t \rangle \leq f(X_{t-1}) - \lambda \langle X_{t-1}, X_{t-1} \rangle. \tag{10}
\]

On the other hand, if \( A, B \in S_n^+ \), then \( 2 \langle A, B \rangle \leq \langle A, A \rangle + \langle B, B \rangle \), see, e.g., [31]. Therefore, for any \( t \), \( 2 \langle X_{t-1}, X_t \rangle - \langle X_{t-1}, X_{t-1} \rangle \leq \langle X_t, X_t \rangle \), which implies

\[
F(X_t) \leq f(X_t) - \lambda \langle X_{t-1}, X_t \rangle + \frac{\lambda}{2} \langle X_{t-1}, X_{t-1} \rangle \\
\leq f(X_{t-1}) - \lambda \langle X_{t-1}, X_{t-1} \rangle + \frac{\lambda}{2} \langle X_{t-1}, X_{t-1} \rangle \\
\leq F(X_{t-1}) - \frac{\lambda}{2} \langle X_{t-1}, X_{t-1} \rangle + \frac{\lambda}{2} \langle X_{t-1}, X_{t-1} \rangle \\
= F(X_{t-1}).
\]

Furthermore, if the proposed approach achieves a binary solution, by uniqueness it is guaranteed that this is exactly the desired global minimum. Conversely,
when the obtained solution is not 1-rank, the final solution is not binary and it is guaranteed that it is not the correct solution. This awareness about achieving the correct solution provides the possibility of running again the algorithm by suitably changing the initialization, which may yield a better solution.

Remark 2 The proposed descent strategy can be interpreted as a reweighting algorithm. In [11, 3], the reweighting algorithm is derived as iterative minimization of the local linearization of the concave log-det term. The convergence of the log-det method to a local minimum is discussed in [3]. We mention that the proposed descent algorithm can be described under the reweighting viewpoint as well, and results from [3, 5] might be leverage to rigorously prove its convergence. The convergence analysis and a complete comparison to log-det heuristic are beyond the scope of this paper; however, some numerical comparisons are proposed in Sec. 4.

4 Numerical results

In this section, we illustrate some numerical results, that support the effectiveness of the proposed approach. We consider the following problem, as presented in [8]: we aim to solve the underdetermined system

\[ b = Ax, \quad x \in \{0,1\}^n, \quad b \in \mathbb{R}^m, \quad A \in \mathbb{R}^{m,n}, \quad m < n \]  

under the assumption that the solution is unique in \( \{0,1\}^n \). This linear problem is encountered in a number of applications, ranging from compressed sensing, see [4, 6, 7] to tomography, see [26, 30].

In [8], problem (11) is tackled in case of sparse \( x \), which recasts into binary compressed sensing, and a Shor SDR is proposed to solve it. In previous works, other methods for binary compressed sensing are proposed, namely, relaxation over the convex hull in [4], \( \ell_1 \)-reweighting algorithms in [6], alternating direction method of multipliers in [7], and difference of convex functions in [26, 30]. SDR is shown to achieve better accuracy in [8].

Here, we recast problem (11) into Boolean quadratic optimization by replacing the constraint \( Ax = b \) with the the cost functional \( \|Ax - b\|_2^2 \). As we know that \( \tilde{x} \in \{0,1\}^n \) is solution of \( Ax = b \), the global minimum is null. Given, the cost functional \( \|Ax - b\|_2^2 \), we homogenize it by adding the slack variable \( x_0 = 1 \), so that \( \|Ax - x_0b\|_2^2 = x^TQx \) with \( Q = (-b|A)^T(-b|A) \), and \( x = (x_0, x_1, \ldots, x_n)^T \). Then, we can apply the approach developed in Sec. 3.

Specifically, we compare the recovery accuracy of the proposed “known binary eigenvalues” (KBE) approach to the following known methods: SDR (4) and SDR (6), nuclear norm heuristic, and log-det heuristic, see [8, 18]. The implemented nuclear norm and log-det algorithms read as follows, respectively:

\[
\min_{X \in \mathbb{R}^{n+1}^+} (Q, X) + \lambda \text{tr}(X)
\]

s. t. \( X_{0,0} = 1, \quad X_{i,i} = X_{0,i}, \quad i = 1, \ldots, n \)

and, for \( t = 1, \ldots, T, \)

\[
X_t = \arg\min_{X \in \mathbb{R}^{n+1}^+} (Q, X) + \lambda ((X_{t-1} + \epsilon I)^{-1}, X)
\]

s. t. \( X_{0,0} = 1, \quad X_{i,i} = X_{0,i}, \quad i = 1, \ldots, n. \)
The considered setting is as follows. We take binary vectors \( \tilde{x} \in \{0, 1\}^n \), with \( n = 50 \), sparsity level \( k \in [5, 45] \), and uniformly distributed support. \( A \) is a random Gaussian matrix with \( m \) rows, with \( m \in [14, 34] \). For all the methods, \( \lambda = 10^{-4} \), and \( T = 3 \). Algorithms 1 and 2, and log-det heuristic (13) are initialized with the solutions of SDR’s (4) and (6). If a binary solution is not achieved, i.e., the algorithm stops in a local minimum or saddle point, we randomly reinitialize for a maximum of 5 times. In (13), we set \( \epsilon = 10^{-6} \). To solve the involved SDR’s, we use the Mosek C++ Fusion API, see [20], which guarantees fast solutions even for quite large dimensional problems. For reasons of space, a complete analysis of the complexity and large-scale experiments are left for future work. We just remark that the matrix inversion in the log-det heuristic yields an higher computational complexity when compared to Algorithms 1 and 2, and might be prohibitive in large-scale problems. The considered algorithms are compared in terms of exact recovery rate, i.e., the number of experiments where a perfect recovery is achieved. Results are averaged over 200 random runs. As SDR’s (4) and (6) are different formulations of the same problem, they are conveyed in the same graph.

In Fig. 1 we show the case of unknown \( k \). We observe that Algorithm 2 increases the success rate with respect to SDR’s (4)-(6). Specifically, KBE - Algorithm 2 has a transition phase from low to high recovery rate (namely, from 60% to 90%, highlighted by red color in the figure) with less measurements than SDR’s (4)-(6): \( m \in [24, 26] \) instead of \( m \in [26, 28] \). We remark that these two methods have approximately constant performance in \( k \). In contrast, as discussed above, the nuclear norm and log-det heuristics as well as KBE - Algorithm 1 generally achieve the best performance. In conclusion KBE - Algorithm 1 is the most reliable approach in non sparse problems, while KBE - Algorithm 2 is the most reliable approach for sparse problems. This attests that the KBE methodology is effective.

In Fig. 2 the same experiment is reported in case of known \( k \). As discussed in Sec. 3 Algorithm 1 can specifically exploit the information on \( k \). Moreover, the equation \( \tilde{x}^T x = k \) is added to \( Ax = b \) for all the methods. In Fig. 2 we observe that, as expected, all the methods take advantage of the knowledge of \( k \), in particular when \( k \) is either very small or very large \( k \). Also in this experiment, we observe that the proposed KBE Algorithms 1-2 are more accurate than the other methods. In particular, they are successful in at least 99% of runs when \( m \geq 30 \), which is not achieved by the competitors.

Finally, we observe that, in our experiments, the run-time is between 0.1 and 0.8 seconds for KBE Algorithms 1-2 and between 0.2 and 1.4 seconds for the log-det heuristic.

5 Conclusion

In this paper, we propose a method to enhance the semidefinite programming relaxations of Boolean quadratic problems. These relaxations provide the right minimizer if a 1-rank solution is found; nevertheless, enforcing low-rank solutions is NP-hard. The proposed strategy overcomes this drawback by leveraging known information on the eigenvalues. The global minimum of the proposed
cost functional is proven to be the exact solution. Since the problem is non-
convex, a low-complex descent algorithm is developed, and shown to be effective 
through numerical results. The accuracy is enhanced with respect to state-of-
the-art methods. Future work will envisage a formal proof of the convergence of 
the proposed algorithms, and the development of different descent techniques, 
with particular focus on large scale problems.

References

[1] Alberto Bemporad and Vihangkumar V. Naik. A numerically robust mixed-
integer quadratic programming solver for embedded hybrid model predic-
tive control. In Proc. IFAC NMPC, number 20, pages 412–417, 2018.

[2] A. Carlini and P. Gurfil. Sensor selection via compressed sensing. Automatica, 49(11):3304–3314, 2013.

[3] M. Fazel, H. Hindi, and S. Boyd. Log-det heuristic for matrix rank mini-
mization with applications to Hankel and Euclidean distance matrices. In Proc. IEEE ACC, volume 3, pages 2156–2162, 2003.

[4] A. Flinth and G. Kutyniok. PROMP: A sparse recovery approach to lattice-valued signals. Appl. Comput. Harmon. Anal., 45(3):668–708, 2018.

[5] S. M. Fosson. A biconvex analysis for Lasso $\ell_1$ reweighting. IEEE Signal Process. Lett., 25(12):1795–1799, 2018.

[6] S. M. Fosson. Non-convex approach to binary compressed sensing. In Asilomar Conf. Signals Syst. Comput., pages 1959–1963, 2018.

[7] S. M. Fosson. Non-convex Lasso-kind approach to compressed sensing for finite-valued signals. ArXiv, 2019.

[8] S. M. Fosson and M. Abuabiah. Recovery of binary sparse signals from compressed linear measurements via polynomial optimization. IEEE Signal Process. Lett., 26(7):1070–1074, 2019.

[9] S. Foucart and H. Rauhut. A Mathematical Introduction to Compressive Sensing. Springer, New York, 2013.

[10] M. X. Goemans and D. P. Williamson. Improved approximation algorithms 
for maximum cut and satisfiability problems using semidefinite program-
ning. J. ACM, 42(6):1115–1145, 1995.

[11] K. Jaganathan, S. Oymak, and B. Hassibi. Sparse phase retrieval: Convex 
algorithms and limitations. In Proc. IEEE Int. Symp. Inf. Theory (ISIT), 
pages 1022–1026, 2013.

[12] K. Jaganathan, S. Oymak, and B. Hassibi. Sparse phase retrieval: Uniqueness guarantees and recovery algorithms. IEEE Trans. Signal Process., 
65(9):2402–2410, 2017.

[13] H. Kellerer, H.K.U.P.D. Pisinger, U. Pferschy, and D. Pisinger. Knapsack 
Problems. Springer Nature Book Archives Millennium. Springer, 2004.
[14] J. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, 11(3):796–817, 2001.

[15] J. B. Lasserre. Convergent lmi relaxations for nonconvex quadratic programs. In *Proc. IEEE CDC*, pages 5041–5046, 2000.

[16] J.-B. Lasserre. An introduction to polynomial and semi-algebraic optimization. Cambridge Univ. Press, 2015.

[17] Jean B. Lasserre. A MAX-CUT formulation of 0/1 programs. *Oper. Res. Lett.*, 44(2):158–164, 2016.

[18] A. Lemon, A. M.-C. So, and Y. Ye. Low-rank semidefinite programming: Theory and applications. *Found. Trends Optim.*, 2(1-2):1–156, 2016.

[19] L. Lovász and A. Shrijver. Cone of matrices and setfunctions. and 0-1 optimization. *SIAM J. Optim.*, 1:166–190, 1990.

[20] MOSEK. *Fusion API for C++*, 2019.

[21] Vihangkumar V. Naik and Alberto Bemporad. Embedded mixed-integer quadratic optimization using accelerated dual gradient projection. In *Proc. IFAC World Congress*, pages 10723–10728, 2017.

[22] Y. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optim. Methods Softw.*, 9(1-3):141–160, 1998.

[23] L. Palagi, V. Piccialli, F. Rendl, G. Rinaldi, and A. Wiegele. *Computational Approaches to Max-Cut*, pages 821–847. Springer US, Boston, MA, 2012.

[24] Chiara Ravazzi, Sophie M. Fosson, Tiziano Bianchi, and Enrico Magli. Sparsity estimation from compressive projections via sparse random matrices. *EURASIP J. Adv. Signal Process.*, 2018(1), Sep 2018.

[25] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3):471–501, 2010.

[26] T. Schüle, C. Schnörr, S. Weber, and J. Hornegger. Discrete tomography by convex-concave regularization and D.C. programming. *Discrete Appl. Math.*, 151:229–243, 2005.

[27] S. Shekhar, V.M. Patel, N.M. Nasrabadi, and R. Chellappa. Joint sparse representation for robust multimodal biometrics recognition. *IEEE Trans. Patt. Ana. & Mach. Intel.*, 36(1):113–126, 2014.

[28] N.-Z. Shor. Quadratic optimization problems. *Sov. J. Comput. Syst. Sci.*, 26, 1987.

[29] R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. Royal. Statist. Soc. B.*, 58:267–288, 1996.

[30] S. Weber, T. Schüle, and C. Schnörr. Prior learning and convex-concave regularization of binary tomography. *Elec. Notes Discrete Math.*, 20:313–327, 2015.
[31] H. Zhou. On some trace inequalities for positive definite Hermitian matrices. *J. Inequal. Appl.*, 64(1), 2014.
Figure 1: Experiment 1 (unknown k): exact recovery rate.
Figure 2: Experiment 2 (known $k$): exact recovery rate.