arXiv:1708.01690v1 [math.OC] 5 Aug 2017

Efficient Rank Minimization to Tighten Semidefinite Programming for Unconstrained Binary Quadratic Optimization

Roman Pogodin[1], Mikhail Krechetov[2] and Yury Maximov[3]

***Abstract*— We propose a method for low-rank semidefinite programming in application to the semidefinite relaxation of unconstrained binary quadratic problems. The method improves an existing solution of the semidefinite programming relaxation to achieve a lower rank solution. This procedure is computationally efficient as it does not require projecting on the cone of positive-semidefinite matrices. Its performance in terms of objective improvement and rank reduction is tested over multiple graphs of large-scale Gset graph collection and over binary optimization problems from the Biq Mac collection.**

I. Introduction

Binary quadratic optimization is a classical combinatorial optimization problem, which finds a wide range of applications in computer vision [1], [2], circuit layout design [3], [4], computing ground states of Ising model [5], as well as a number of combinatorial favors [6], [7], [8]. For comprehensive list of applications we refer to [9], [10].

A special case of this problem is unconstrained binary quadratic programming (UBQP). It is a classical NP-hard problem, hardly possible to be solved exactly in polynomial time. A number of relaxation techniques substituting the original problem to a convex one has been proposed. Linear, second-order cone and semidefinite relaxations are among them.

Semidefinite programming relaxation (SDP) has been shown to lead to tighter approximation than other relaxation methods for many combinatorial optimization problems including binary quadratic optimization ones ([11] and [12]). Still, SDP reduces the problem to convex optimization over the cone of positive semidefinite matrices and outputs a full-rank matrix requiring to be rounded to obtain a vector valued solution.

In this paper we address the question of low-rank semidefinite programming, which is aimed at strengthening results of the standard SDP relaxation. While a number of methods has been proposed (see [13] for a survey), we discuss direct rank minimization of an obtained SDP solution. We use smoothed rank approximations in order to reduce the rank of the SDP solution without significant loss in the optimal value. For this purpose we propose an efficient first-order optimization procedure which does not require projecting on the feasible set. This will potentially lead to more accurate rounding procedure allowing to obtain a better vector solution.

In the rest of the paper we discuss problems of the form

$$\begin{aligned} &\max\ x^\top A x, \\ &\text{s.t. } x \in \{-1, 1\}^n, \end{aligned} \tag{1}$$

where A is an arbitrary symmetric $n \times n$ matrix.

There are many well-known problems that can be naturally written in this form: the maximum cut problem, the 0-1 knapsack problem, the linear quadratic regulator and many others.

Not all of these formulations appear in the form 1. Instead, some problems might have a linear term of the from $b^\top x$ and a 0-1 constraint, that is $x \in \{0, 1\}^n$. Nevertheless, such problems might be converted to the form 1, as showed in [14]. The corresponding derivation is showed in the appendix.

For instance, in the maximum cut (max-cut) problem we want to find a partition of graph's vertices into two disjoint sets such that the sum of edges between these sets is maximal. This problem is NP-hard [15], however, it can be solved approximately. For small instances (up to 50 vertices) the maximum cut may be solved efficiently with the branch-and-bound algorithm [16], [17]. For bigger problems (around 1000 vertices) the semidefinite relaxation discussed below gives the best known approximations. It is crucial for many applications to improve existing approximations or to extend them on large-scale instances.

Quadratic boolean programming (1) is a particular case of quadratically constrained quadratic problems (QCQP), so general heuristics for this class of problems may be applied. See for example the recent paper [18].

A. Semidefinite Relaxation.

Standard semidefinite (SDP) relaxation leads to the following matrix problem:

$$\begin{aligned} &\max\ \mathrm{Tr}\,(AX), \\ &\text{s.t. } \mathrm{diag}\, X = 1_n, \\ &\quad X^\top = X,\ X \succeq 0. \end{aligned} \tag{2}$$

This problem is convex and thus could be efficiently solved (we will discuss the particular method below).

[1]Center for Energy Systems, Skolkovo Institute of Science and Technology Skolkovo Innovation Center, Building 3, Moscow 143026 Russia R.Pogodin@skoltech.ru

[2]Center for Energy Systems, Skolkovo Institute of Science and Technology Skolkovo Innovation Center, Building 3, Moscow 143026 Russia M.Krechetov@skoltech.ru

[3]Center for Energy Systems, Skolkovo Institute of Science and Technology, and Theoretical Division T-4 and CNLS, Los Alamos National Laboratory MS-B258, Los Alamos, 87545 NM, USA yury@lanl.gov

To get a binary solution of the initial problem 1, we decompose the solution of the SDP relaxation $X = V^\top V$ (via Cholesky decomposition), then take a unit vector with uniformly distributed direction r. For each column of V, which is v_i, we take $x_i = \operatorname{sign} v_i^\top r$. If $A \succeq 0$, the mean result of this procedure is not worse than $2/\pi$ of the maximum value [19]. In a special case when A is a Laplacian of a graph with non-negative weights this bound can be further improved to ≈ 0.878 [11]. Note that this famous results cannot be improved if the Unique Games Conjecture is true [12].

However, this relaxation is exact, when $\operatorname{rank} X = 1$. Moreover, low-rank solutions lead to a fewer number of possible binary sets in the rounding procedure described above. This idea becomes more clear if one considers half-space classifiers in $\mathbb{R}^r$ and n points. The maximum number of different labellings is controlled by Sauer's lemma, and grows as $(n+1)^r$. A more accurate discussion of Sauer's lemma can be found in [20]. Hence, if we lie in the vicinity of a correct solution, we would get the correct result from the rounding procedure more likely. This motivation brings us to the idea of low-rank semidefinite programming.

B. Related Work.

The existence of low-rank solutions of the problem 2 is a fundamental fact discussed in [21] and [22]. From these works we know that for such problems there exist solutions of the rank at most r, where $r(r+1) \leq 2n$.

Knowing about the existence of low-rank solutions, we may discuss popular approaches in low-rank semidefinite programming. This section is mostly based on the book [13].

Firstly, the existence of low-rank solutions is used in Burer and Monteiro method [23]. It is based on the factorization $X = VV^\top$, where V is an arbitrary matrix of size $n \times r$. This problem is non-convex, though it requires much less computations and performs well in practice. However, finding the minimum rank solution of multiple runs of the algorithm with different r (one increments r until resulting point satisfies particular conditions) might be computationally ineffective.

Another approach implies relaxation of the equality constraint [24]. It can be written as following:

$$\operatorname{Tr}(AX) = b \Rightarrow \beta b \leq \operatorname{Tr}(AX) \leq \alpha b,$$

where α and β control the rank of the solution. In our case the problem 2 has n equality constraints of the form $\operatorname{Tr}(XE_{ii}) = 1$, where E_{ii} is a zero matrix with unit in the position i, i. All n constraints together form $\operatorname{diag} X = 1$. Though this approach allows to reduce the rank, the resulting solution satisfies our constraints only approximately.

The next approach implies that we have already got a solution of the problem 2. This allows us to reduce its rank via some kind of rank minimization procedure keeping the value of $\operatorname{Tr}(AX)$ optimal. Note that rank minimization is NP-hard, so this formulation needs to be further relaxed to be efficiently solvable. This approach has been discussed in the literature, but we have not found a comparison of different ways to minimize the rank in application to boolean quadratic programming.

In our work we propose an efficient first-order algorithm, which performs rank minimization. It starts from the solution of the SDP relaxation. This solution is provided by either CVX interior-point algorithm [25], [26] or Burer-Monteiro low-rank procedure, implemented in SDPLR [23].

II. Rank minimization

In this section we introduce the problem of rank minimization for the SDP relaxation. After that we describe existing approaches for rank minimization, and discuss their pros and cons.

A. Problem.

Let X^* be a solution of 2 and let SDP be its value and W^* be the value of the binary solution obtained by the rounding procedure. Starting from X^*, we want to solve the following problem:

$$\begin{aligned} &\min \operatorname{rank} X \\ &\text{s.t. } \operatorname{diag} X = 1_n, \\ &\quad X^\top = X,\ X \succeq 0, \\ &\quad \operatorname{Tr}(AX) = SDP, \end{aligned} \tag{3}$$

B. Objective function's relaxations.

However, minimizing the rank is NP-hard. In order to solve the problem 3, we replace $\operatorname{rank} X$ with a smooth surrogate, usually non-convex. There is a number of rather popular ways to do that, discussed below.

First of all, the so-called trace norm (or nuclear), defined as

$$\|X\|_* = \sum_i \sigma_i, \tag{4}$$

where σ_i is the i-th singular value [13]. In our case, every feasible point of the problem has $\operatorname{Tr} X = \|X\|_* = n$, hence this relaxation does not make sense.

Next, the so-called log-det heuristic is to replace $\operatorname{rank} X$ with the concave function

$$\log\det(X + \varepsilon I). \tag{5}$$

This heuristic is discussed, for example, in [13], [27]. Though it performs well in practice, it requires an iterative procedure (described in [27]) with an SDP problem on each iteration. This problem allows to use Burer-Monteiro method [23], but it still needs several runs of this algorithm (at least one for each iteration), which is compatible with rank increment in the original Burer-Monteiro procedure.

The next two relaxations are singular value-based, and in the next section we show that, in fact, they allow an efficient first-order procedure, that does not require projections on the semidefinite cone and hence is computationally efficient.

The first one is the non-convex function of the following form

$$\Phi(X, \varepsilon) = (1 + \varepsilon^q) \operatorname{Tr}\left(X^\top (XX^\top + \varepsilon I)^{-1} X\right). \tag{6}$$

for $q \in \mathbb{Q} \cap [0, 1]$. Its properties are discussed in [28]. An important fact is that this relaxation is quite close to the rank:

$$|\operatorname{rank} X - \Phi(X, \varepsilon)| \leq$$
$$\leq \varepsilon^q \max \left\{ \operatorname{rank} X, \sum_{i=1}^{\operatorname{rank} X} \left| \frac{\varepsilon^{1-q}}{\sigma_i^2(X)} - 1 \right| \right\}.$$

The second relaxation utilizes so-called smoothed Schatten p-norms. They are defined as following:

$$\|X\|_{S_{p,\varepsilon}}^p = \sum_{i \geq 1} \left(\sigma_i^2 + \varepsilon\right)^{\frac{p}{2}} = \operatorname{Tr}\left(X^\top X + \varepsilon I\right)^{\frac{p}{2}}. \quad (7)$$

With $p \to 0$ and $\varepsilon = 0$ we get the rank function exactly. Note that for $p < 1$ this function is non-convex, and for $p = 1$ it is identical to the nuclear norm. Applications of this relaxation can be found in [29] and [30]. Both papers introduce an iteratively reweighted least squares (IRLS) algorithm in order to solve this problem. However, in our case it requires solving of an SDP problem with quadratic objective function at each iteration. Hence, it cannot be applied to large-scale problems.

C. Constraint relaxation.

If we omit the last constraint in the problem 3, which is $\operatorname{Tr}(AX) = SDP$, rank minimization might occasionally converge to a solution of rank one. Moreover, it may converge to a low-rank vicinity of such solution. If the SDP relaxation is not tight, then this constraint prevents the procedure from such behavior.

We can also obtain a binary solution after solving the SDP relaxation. If we denote the objective value at this point W^*, then this value would be a natural lower bound on $\operatorname{Tr}(AX)$. It means that all solutions of rank one above this value are actually better, than the one we got.

This motivation allows us to relax the problem further, and solve (along with rank approximations) the following one:

$$\begin{aligned} &\min \operatorname{rank} X \\ &\text{s.t. } \operatorname{diag} X = 1_n, \\ &\quad X^\top = X,\ X \succeq 0, \\ &W^* \leq \operatorname{Tr}(AX) \leq SDP. \end{aligned} \quad (8)$$

Obviously, the binary solution, that gives W^*, is also a solution of the last problem. However, the typical procedure for solving 8 would start from the SDP matrix in order to improve the resulting cut. Since the rank of this matrix is not unit in general, we need to optimize the objective function further.

An important consequence of this relaxation will be clear in the next section as it allows to avoid projecting on the feasible set.

For completeness we emphasize that our approach cannot be generalized to QCQP problems. To avoid projecting on the set, it relies significantly on the special structure of constraints that occur in the problem 8.

III. Main result

In this section we introduce an efficient first-order procedure that solves the problem 8 without projecting on the positive-semidefinite cone. It can be applied to the singular value relaxation 6 and smoothed Schatten p-norm 7.

A. Efficient first-order procedure.

The problem 8 allows a natural reparametrization to a vector problem of dimension $n(n-1)/2$. To do that, we consider the upper triangular part of the matrix:

$$X = \begin{pmatrix} 1 & x_1 & \dots & x_{n-1} \\ x_1 & 1 & x_n & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix}. \quad (9)$$

In this case for indices i, j we get $X_{i,j} = x_d$, where $d = \sum_{k=1}^{i-1}(n-k) + j - i$. This satisfies two constraints immediately: $X^\top = X$ and $\operatorname{diag} X = 1_n$.

Such reparametrization changes the gradient of the matrix function $f(X)$:

$$\frac{\partial f(X(x))}{\partial x_{(d)}} = \sum_{i,j} \frac{\partial f(X)}{\partial x_{i,j}} \frac{\partial x_{i,j}}{\partial x_{(d)}} = \frac{\partial f(X)}{\partial x_{i',j'}} \frac{\partial x_{i',j'}}{\partial x_{(d)}} +$$
$$+ \frac{\partial f(X)}{\partial x_{j',i'}} \frac{\partial x_{j',i'}}{\partial x_{(d)}} = \frac{\partial f(X)}{\partial x_{i',j'}} + \frac{\partial f(X)}{\partial x_{j',i'}},$$

where i', j' relate to the vector of index d.

Obviously, the upper bound $\operatorname{Tr}(AX) \leq SDP$ is always satisfied. Moreover, violation of the lower one $W^* \leq \operatorname{Tr}(AX)$ implies that we got the point that could not improve our binary solution, hence we need to stop.

The last constraint is $X \succeq 0$. We show that proper choice of the gradient step results in a feasible point in case of singular value relaxation and Schatten norms.

First of all, the gradient of the singular value relaxation is

$$\frac{\partial \Phi(X, \varepsilon)}{\partial X} = 2\varepsilon(1 + \varepsilon^q)\left(XX^\top + \varepsilon I\right)^{-2} X. \quad (10)$$

For Schatten p-norms we have

$$\frac{d\|X\|_{S_{p,\varepsilon}}^p}{dX} = pX\left(X^\top X + \varepsilon I\right)^{\frac{p-2}{2}}. \quad (11)$$

If X is symmetric and PSD, then from SVD factorization both gradients are symmetric. Thus in vector parametrization we simply need to multiply the gradient by 2, and then force diagonal elements to be unit.

For further convenience we denote a symmetric matrix with unit diagonal, upper triangular part of which is constructed from the vector x, as $X(x)$. Finally, we show the following:

Theorem 1: Let $f(x)$ be the vector-parametrized singular value relaxation 6 of the matrix $X(x) \succeq 0$. Then for $\alpha \leq \frac{\varepsilon}{4(1+\varepsilon^q)}$ we get $X(x - \alpha\nabla f(x)) \succeq 0$.

Proof: The gradient step in upper-triangular parametrization is equivalent to the ordinary gradient step (multiplied by 2), and then substituting diagonal elements to units. We are going to show that the first step results in a PSD matrix and then the second step keeps matrix PSD.

Consider a symmetric PSD point and its SVD decomposition $X = USU^\top$. Hence for a step α and $C = 2\varepsilon(1+\varepsilon^q)$ the new point is

$$\begin{aligned} &X - 2\alpha C(XX^\top + \varepsilon I)^{-2}X = \\ &= USU^\top - 2\alpha C\left(US^2U^\top + \varepsilon I\right)^{-2}USU^\top = \\ &= U\left(S - 2\alpha C\left(S^2 + \varepsilon I\right)^{-2}S\right)U^\top. \end{aligned}$$

Hence the positive-semidefiniteness of the resulting matrix is equivalent to such characteristic of the expression in brackets. It is a diagonal matrix.

If $S_{ii} = 0$, then the corresponding diagonal elements are obviously zero. Otherwise we need it to be positive:

$$\begin{aligned} &S_{ii} - 2\alpha C\left(S_{ii}^2 + \varepsilon I\right)^{-2}S_{ii} \geq 0 \Rightarrow \\ &\Rightarrow \alpha \leq \frac{\left(S_{ii}^2 + \varepsilon I\right)^2}{2C} = \frac{\left(S_{ii}^2 + \varepsilon I\right)^2}{4\varepsilon(1+\varepsilon^q)}. \end{aligned}$$

Therefore, it is enough to take

$$\alpha \leq \frac{\varepsilon}{4(1+\varepsilon^q)}.$$

Now we want to show that substituting diagonal elements with units is equivalent to adding a diagonal matrix with non-negative entries. In this case, a sum of two PSD matrices is PSD.

It is also equivalent to the fact that all diagonal elements of the gradient are non-negative. This is always true for a symmetric and PSD matrix $X = USU^\top$:

$$\begin{aligned} &\left((XX^\top + \varepsilon I)^{-2}X\right)_{ii} = \left(U\left(S^2 + \varepsilon I\right)^{-2}SU^\top\right)_{ii} = \\ &= \sum_j U_{ij}U_{ij}\left(S_{jj}^2 + \varepsilon I\right)^{-2}S_{jj} = \\ &= \sum_j U_{ij}^2\left(S_{jj}^2 + \varepsilon I\right)^{-2}S_{jj} \geq 0. \end{aligned}$$

This observation completes the proof. ■

The same technique allows us to get a similar result for smoothed Schatten p-norms:

Theorem 2: Let $f(x)$ be vector-parametrized smoothed Schatten p-norm of a matrix $X(x) \succeq 0$. Then for $\alpha \leq \frac{1}{2p}\varepsilon^{(2-p)/p}$ we get $X(x - \alpha\nabla f(x)) \succeq 0$.

For completeness, its proof is done in the appendix.

In practice, the singular value relaxation bound is much better, since the step bound tends to be larger.

B. Algorithm.

The results above allow us to introduce a gradient descent method, summarized in the algorithm 1 (for singular values relaxation). In this algorithm an abstract procedure ChooseStep returns the appropriate step, which is less or equal to $\frac{\varepsilon}{4(1+\varepsilon^q)}$. The second procedure RoundSolution corresponds to the rounding method, described in the introduction.

Algorithm 1 Gradient descent for singular values relaxation

$X_0 = X_{SDP}$
$K = \left\{X \,\middle|\, X = X^\top,\ X \succeq 0,\ W \leq \operatorname{Tr}(AX) \leq SDP\right\}$
for $n = 1$:max_iter **do**
  $\alpha = \text{ChooseStep}(X_n)$
  $X_n = X_{n-1} - 4\alpha\varepsilon(1+\varepsilon^q)\left(XX^\top + \varepsilon I\right)^{-2}X$
  $(X_n)_{ii} = 1$
  if $X_n \notin K$ **then**
    break
  end if
  if $\|4\alpha\varepsilon(1+\varepsilon^q)\left(X_nX_n^\top + \varepsilon I\right)^{-2}X_n\|_F < \text{tol}$ **then**
    break
  end if
end for
$\mathbf{x}, W^{**} = \text{RoundSolution}(A, X_n)$
return $\mathbf{x}, W^{**}$

IV. Experiments

A. Setup

We have tested both CVX and SDPLR solvers followed by our algorithm on Gset graphs collection `https://web.stanford.edu/~yyye/yyye/Gset/` (originally introduced in [31]). We also tested the Biq Mac library `http://biqmac.uni-klu.ac.at/biqmaclib.html` (namely on Beasley instances [32]). The latter is a collection of $\{0,1\}$ problems, which are converted to $\{-1,1\}$ ones as discussed in the appendix.

For each solver we first applied it to the problem. Then we computed the maximum cut based on the 1×10^5 rounding operation. The number of rounding operations was chosen to be completely sure that the solution provides better results compared to others. After that we chose $\varepsilon = 0.005$ (smaller values led to ill-conditioned gradients), $q = 0.8$ for the singular values relaxation (this value is used by the authors of [28]), $p = 0.1$ and $p = 0.01$ for Schatten norms. The stopping criteria for the gradient descent were 100 iterations and Frobenius norm of the gradient (less than 1×10^{-5}). After that a new cut was obtained after 1×10^5 rounding operations. Rank tolerance was chosen to be 1×10^{-4}. For Schatten norms the step size was at most ε, which is larger than the theoretical value. Nevertheless, for such steps we have not observed any violations of the PSD constraint.

We tested all methods on the first 21 Gset graphs. These graphs have 800 nodes and are solvable with CVX. Another 10 graphs of size 2000 were tested with SDPLR only.

We also tested all methods on the Beasley instances from the Biq Mac library. We chose relatively large $\{0,1\}$ problems with 250 and 500 nodes. They were tested for CVX only.

B. Discussion

Results are shown in tables I for SDPLR and II for CVX.

Our approach outperforms both solvers in terms of the maximum cut on approximately half of the graphs (bold

	SDP		singular values		Schatten p=0.1		Schatten p=0.01	
#	rank	cut	rank	cut	rank	cut	rank	cut
1	14	11466	13	11448	15	11451	13	11459
2	15	11436	13	11438	13	**11456**	14	11430
3	14	11446	14	11445	461	**11455**	26	11453
4	14	11487	14	11475	319	**11511**	14	11497
5	13	11462	12	11462	18	11451	12	**11471**
6	13	2026	13	1989	105	2013	13	2012
7	12	1833	12	1821	12	**1834**	12	1822
8	12	1834	11	1833	11	**1840**	12	1831
9	12	1879	12	1872	12	1869	12	1875
10	12	1841	12	1829	12	1818	12	1820
11	22	538	138	538	718	538	273	536
12	13	532	27	**536**	190	534	41	534
13	11	562	8	560	24	562	8	562
14	14	2994	13	2995	22	2992	14	**2999**
15	16	2979	99	2982	173	**2986**	122	2983
16	16	2978	14	2981	195	**2984**	15	**2984**
17	16	2978	13	2975	152	2974	15	2974
18	11	924	10	**930**	12	921	11	929
19	10	850	9	846	9	**854**	9	851
20	9	888	8	882	8	**889**	8	884
21	10	868	33	864	52	862	41	863
22	18	13008	18	13006	18	13003	18	**13025**
23	20	13010	52	12985	78	13010	55	13004
24	19	13000	18	13004	22	13005	19	**13010**
25	19	13006	19	12988	196	**13026**	19	12987
26	19	12971	132	12985	181	12969	120	**12990**
27	18	2988	17	2988	17	**3027**	18	2989
28	20	2956	20	2948	54	2947	20	2956
29	18	3044	17	**3056**	23	3050	17	3038
30	17	3076	17	**3081**	17	3076	17	3067
31	19	2947	18	2955	18	**2959**	18	2958

TABLE I: SDPLR improvement on Gset graphs

	SDP		singular values		Schatten p=0.1		Schatten p=0.01	
#	rank	cut	rank	cut	rank	cut	rank	cut
1	13	11462	13	11448	15	11451	13	11456
2	13	11436	13	11438	13	**11456**	13	11433
3	14	11446	14	11445	502	11446	28	**11453**
4	14	11487	14	11471	304	**11511**	14	11497
5	12	11462	12	11464	18	11451	12	**11471**
6	13	2024	13	1994	108	2013	13	2016
7	13	1833	12	1821	12	1828	12	1822
8	12	1835	12	**1856**	109	1846	13	1839
9	12	1879	12	1872	12	1869	12	1875
10	12	1841	12	1825	12	1820	12	1836
11	10	534	6	534	6	534	7	534
12	9	532	8	534	29	**536**	8	**536**
13	8	562	8	560	76	562	8	560
14	13	2994	13	2995	22	2994	13	**2999**
15	13	2979	13	2982	51	**2987**	13	2981
16	14	2982	14	2981	589	2979	61	**2986**
17	13	2978	13	2978	439	2973	24	2976
18	10	924	10	**930**	11	920	10	929
19	9	847	9	846	9	**850**	9	846
20	9	882	9	887	222	**888**	20	886
21	9	862	9	865	9	**867**	9	865

TABLE II: CVX improvement on Gset graphs

	SDP		singular values		Schatten p=0.1		Schatten p=0.01	
#	rank	cut	rank	cut	rank	cut	rank	cut
1	6	45369	6	45369	60	45369	173	45369
2	6	44579	6	44513	7	44571	47	44515
3	6	48857	6	48857	13	48833	81	48857
4	7	41094	7	**41116**	17	41094	88	**41116**
5	5	47685	5	**47738**	16	47679	75	47685
6	7	40519	7	**40545**	9	40475	70	40469
7	6	46605	6	46563	6	46659	38	**46671**
8	7	35076	7	35000	8	35076	75	**35079**
9	6	48454	6	**48570**	9	48447	59	48364
10	6	39944	6	**39990**	15	39974	77	39944

TABLE III: CVX improvement on the Biq Mac graph collection (250 nodes)

	SDP		singular values		Schatten p=0.1		Schatten p=0.01	
#	rank	cut	rank	cut	rank	cut	rank	cut
1	9	114540	9	114440	37	113880	246	114100
2	8	127280	8	127230	8	127080	16	**127370**
3	9	129080	9	129210	17	129130	148	**129400**
4	9	128000	9	128040	43	127940	273	**128170**
5	8	123570	8	**123860**	8	123430	35	123480
6	8	119770	8	**120350**	8	119750	8	119590
7	9	120210	9	119920	15	120100	177	120070
8	9	121940	9	121650	12	**121980**	101	121920
9	9	118700	9	118570	26	118360	209	118420
10	7	129220	7	129100	7	129120	26	129040

TABLE IV: CVX improvement on the Biq Mac graph collection (500 nodes)

numbers). Moreover, for the first 21 graphs, it outperforms them on almost the same set of graphs. Among tested rank relaxations the best performance was shown by Schatten norm relaxation with $p = 0.1$. Probably, $p = 0.01$ was a worse choice as it resulted in bigger influence of the smoothing parameter ε as it is included in the gradient in order of $\varepsilon^{(p-2)/2}$.

The same result is observed for the $\{0, 1\}$ problems, but the singular values relaxation showed the best performance.

All methods performed well for rank reducing. However, some runs resulted in extremely large ranks. It show, that the resulting point has a lot of relatively small singular values, which are however not thresholded by 1×10^{-4} (note that all singular values sum to the graph's size). This behavior might be considered a drawback of the Schatten norms relaxation and might be caused by the minimization of the sum of singular values (compared to the sum of fractions in the singular values relaxation). Note that small singular values have little effect in the rounding procedure, so this drawback does not have great influence on the method's performance.

V. Conclusion

We developed an efficient first-order procedure which is aimed at improvement of SDP relaxation solutions for quadratic binary programming. We relax rank minimization in terms of the objective function and the linear constraint that controls optimality of the initial SDP objective. Rank function relaxation is performed by either singular value

relaxation or Schatten p-norms. The latter approach showed the best performance on Gset graphs, while the singular values relaxation performed best on the Beasley instances.

REFERENCES

[1] P. Wang, C. Shen, A. van den Hengel, and P. H. Torr, "Large-scale binary quadratic optimization using semidefinite relaxation and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 3, pp. 470–485, 2017.

[2] J. Ren, X. Jiang, J. Yuan, and G. Wang, "Optimizing lbp structure for visual recognition using binary quadratic programming," *IEEE Signal Processing Letters*, vol. 21, no. 11, pp. 1346–1350, 2014.

[3] F. Barahona, "An application of combinatorial optimization to statistical physics and circuit layout design." *Operations Research*, pp. 493–513, 1988.

[4] J. M. Kleinhans, G. Sigl, F. M. Johannes, and K. J. Antreich, "Gordian: Vlsi placement by quadratic programming and slicing optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, no. 3, pp. 356–365, 1991.

[5] G. R. Frauke Liers, Michael Jnger and G. Rinaldi, "Computing exact ground states of hard ising spin glass problems by branch-and-cut," *New optimization algorithms in physics*, pp. 47–69, 2004.

[6] R. G. Parker and R. L. Rardin, *Discrete optimization*. Elsevier, 2014.

[7] M. R. Garey and D. S. Johnson, *Computers and intractability*. wh freeman New York, 2002, vol. 29.

[8] T. H. Cormen, *Introduction to algorithms*. MIT press, 2009.

[9] M. M. Deza and M. Laurent., *Geometry of cuts and metrics*. Springer, 2009.

[10] G. Kochenberger, "The unconstrained binary quadratic programming problem: a survey." *Journal of Combinatorial Optimization*, pp. 58–81, 2014.

[11] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, no. 6, pp. 1115–1145, Nov. 1995.

[12] S. Khot, "On the power of unique 2-prover 1-round games," in *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, ser. STOC '02. New York, NY, USA: ACM, 2002, pp. 767–775.

[13] A. Lemon, A. M.-C. So, and Y. Ye, "Low-Rank Semidefinite Programming: Theory and Applications," *Foundations and Trends® in Optimization*, vol. 2, no. 1-2, pp. 1–156, 2016.

[14] C. Helmberg, S. Poljak, and F. Rendl, "Combining semidefinite and polyhedral relaxations for integer programs," *Integer Programming and Combinatorial Optimization*, pp. 124–134, 1995. [Online]. Available: http://www.springerlink.com/index/F377K847M974R422.pdf

[15] R. M. Karp, *Reducibility among Combinatorial Problems*. Boston, MA: Springer US, 1972, pp. 85–103.

[16] J. M. Krislock, Nathan and F. Roupin., "Improved semidefinite bounding procedure for solving max-cut problems to optimality." *Mathematical Programming*, pp. 61–86, 2014.

[17] G. R. Rendl, Franz and A. Wiegele, "Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations," *Mathematical Programming*, pp. 307–335, 2010.

[18] J. Park and S. Boyd., "General heuristics for nonconvex quadratically constrained quadratic programming." *arXiv preprint arXiv:1703.07870*, 2017.

[19] Y. Nesterov, "Semidefinite relaxation and nonconvex quadratic optimization," *Optimization Methods and Software*, vol. 9, no. 1-3, pp. 141–160, 1998.

[20] P. L. Bartlett, "Growth function and VC-dimension," 2011. [Online]. Available: https://courses.cs.washington.edu/courses/cse522/11wi/scribes/lecture10.pdf

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

[22] A. I. Barvinok, "Problems of distance geometry and convex properties of quadratic maps," *Discrete & Computational Geometry*, vol. 13, no. 2, pp. 189–202, 1995.

[23] S. Burer and R. D. C. Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," vol. 357, pp. 329–357, 2003.

[24] A. M.-C. So, Y. Ye, and J. Zhang, "A unified theorem on sdp rank reduction," *Mathematics of Operations Research*, vol. 33, no. 4, pp. 910–920, 2008.

[25] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[26] ——, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences. Springer-Verlag Limited, 2008, pp. 95–110.

[27] M. Fazel, H. Hindi, and S. Boyd, "Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices," in *Proceedings of the 2003 American Control Conference, 2003.*, vol. 3, 2003, pp. 2156–2162.

[28] C. Li, "Approximation of Matrix Rank Function and Its Application to Matrix Rank Minimization," *Journal of Optimization Theory and Applications*, pp. 569–594, 2014.

[29] F. Nie, H. Huang, and C. Ding, "Low-rank matrix recovery via efficient schatten p-norm minimization," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, ser. AAAI'12. AAAI Press, 2012, pp. 655–661.

[30] K. Mohan and M. Fazel, "Iterative reweighted algorithms for matrix rank minimization," *Journal of Machine Learning Research*, vol. 13, no. Nov, pp. 3441–3473, 2012.

[31] C. Helmberg and F. Rendl, "A Spectral Bundle Method for Semidefinite Programming," *SIAM Journal on Optimization*, vol. 10, no. August, pp. 673–696, 1997.

[32] J. E. Beasley, "Heuristic algorithms for the unconstrained binary quadratic programming problem," Management School, Imperial College, London, UK, Tech. Rep. December, 1998.

VI. APPENDIX

A. Correspondence between {0,1} and {-1, 1} problems

This part mostly copies derivations from the paper [14] and also clarifies some of the steps. It is included for completeness.

Consider a following problem:

$$\begin{aligned} &\max\ x^\top A x + b^\top x, \\ &\text{s.t. } x \in \{0,1\}^n. \end{aligned}$$

Since it is a $\{0,1\}$-problem, we substitute $B = A + \operatorname{diag}(b)$ and obtain an equivalent formulation:

$$\begin{aligned} &\max\ x^\top B x, \\ &\text{s.t. } x \in \{0,1\}^n. \end{aligned}$$

To get a $\{-1,1\}$-problem, we denote $y = 2x - 1$:

$$\begin{aligned} &\max\ \frac{1}{4}(y + 1_n)^\top B (y + 1_n), \\ &\text{s.t. } y \in \{-1,1\}^n. \end{aligned}$$

Note, that:

$$(y + 1_n)^\top B (y + 1_n) = y^\top B y + 2 y^\top B 1_n + 1_n^\top B 1_n.$$

If we denote $B1_n = c \in \mathbb{R}^n,\ 1_n^\top B 1_n = d \in \mathbb{R}$, this problem might be expressed as

$$\begin{aligned} &\max\ \frac{1}{4} z^\top C z, \\ &\text{s.t. } z \in \{-1,1\}^{n+1}, \\ &z_{n+1} = 1, \end{aligned}$$

where the matrix C is a block matrix:

$$C = \begin{pmatrix} B & c \\ c^\top & d \end{pmatrix}.$$

We obtained an equivalent formulation of the initial problem. The last constraint of the final problem, which is $z_{n+1} = 1$, is not necessary. It does not affect on the objective since it

is a quadratic problem, hence one might flip the values of z as $z \to -z$ to get a solution of the initial $\{0,1\}$-problem in case of $z_{n+1} = -1$.

We are also interested in the case where $C \succeq 0$, which immediately provides guarantees for an SDP solution. Using Schur complement, we observe that

$$\begin{pmatrix} B & c \\ c^\top & d \end{pmatrix} \succeq 0 \Longleftrightarrow dB \succeq cc^\top.$$

If eigenspaces of B and $cc^\top$ are such that this condition might be satisfied for some d, we might change it (as it is actually a constant and does not affect maximization). However, an increased value of d would provide a $2/\pi$ bound for a different problem with bigger objective value. Hence, a true $2/\pi$ bound for the initial problem must be obtained without changes in the constant part of the objective.

B. Gradient descent step choice for Schatten norm relaxation

Theorem 2: Let $f(x)$ be vector-parametrized smoothed Schatten p-norm of a matrix $X(x) \succeq 0$. Then for $\alpha \leq \frac{1}{2p}\varepsilon^{(2-p)/p}$ we get $X(x - \alpha\nabla f(x)) \succeq 0$.

Proof: The gradient step in the upper-triangular parametrization is equivalent to the ordinary gradient step (multiplied by 2) with substitution of diagonal elements with units. We are going to show that the first step results in a PSD matrix and the second step keeps matrix PSD.

Consider a symmetric PSD point and its SVD decomposition $X = USU^\top$. Hence for the step α the new point is

$$\begin{aligned} &X - 2\alpha p X \left(X^\top X + \varepsilon I\right)^{(p-2)/2} = \\ &= USU^\top - 2\alpha p USU^\top \left(US^2U^\top + \varepsilon I\right)^{(p-2)/2} = \\ &= U\left(S - 2\alpha p S\left(S^2 + \varepsilon I\right)^{(p-2)/2}\right)U^\top. \end{aligned}$$

Hence positive-semidefiniteness of the resulting matrix is equivalent to such characteristic of the expression in brackets. It is a diagonal matrix.

If $S_{ii} = 0$, then the corresponding diagonal elements are obviously zero. Otherwise we need them to be positive:

$$\begin{aligned} &S_{ii} - 2\alpha p S_{ii}\left(S_{ii}^2 + \varepsilon I\right)^{(p-2)/2} \geq 0 \Rightarrow \\ &\Rightarrow \alpha \leq \frac{\left(S_{ii}^2 + \varepsilon I\right)^{(2-p)/2}}{2p}. \end{aligned}$$

Therefore, it suffices to take

$$\alpha \leq \frac{\varepsilon^{(2-p)/2}}{2p}.$$

Now we want to show that substituting diagonal elements with units is equivalent to adding a diagonal matrix with non-negative entries. In this case a sum of two PSD matrix is PSD.

It is also equivalent to the fact that all diagonal elements of the gradient are non-negative. This is always true for a symmetric and PSD matrix $X = USU^\top$:

$$\begin{aligned} &\left(X\left(X^\top X + \varepsilon I\right)^{(p-2)/2}\right)_{ii} = \left(US\left(S^2 + \varepsilon I\right)^{(p-2)/2}U^\top\right)_{ii} = \\ &= \sum_j U_{ij}U_{ij}S_{jj}\left(S_{jj}^2 + \varepsilon I\right)^{(p-2)/2} = \\ &= \sum_j U_{ij}^2 S_{jj}\left(S_{jj}^2 + \varepsilon I\right)^{(p-2)/2} \geq 0. \end{aligned}$$

This observation completes the proof. ■