A new library of structured semidefinite programming instances

Etienne de Klerk Renata Sotirov

September 2, 2008

Abstract

Solvers for semidefinite programming (SDP) have evolved a great deal in the last decade, and their development continues. In order to further support and encourage this development, we present a new test set of SDP instances. These instances arise from recent applications of SDP in coding theory, computational geometry, graph theory and structural design. Most of these instances have a special structure that may be exploited during a pre-processing phase, e.g. algebraic symmetry, or low rank in the constraint matrices.

1 Introduction

Since interior point algorithms were extended from linear to semidefinite programming (SDP) in the early 1990's by Nesterov and Nemirovski [25] and others, several interior point codes for SDP have been developed and maintained. The list includes DSDP [4], CSDP [6], SDPA [31], SDPT3 [30], and SeDuMi [29].

For SDP instances that are too large for the interior point approach, there are also lower order algorithms available, including PENSDP [18], SBmethod [11], and SDPLR [9].

Sets of SDP benchmark problems are instrumental to the development of these codes. There currently exist three standard test sets of SDP instances, namely SDPLIB [7] and the DIMACS test sets [26], and an extended set of SDP benchmark problems maintained by Hans Mittelmann at `http://plato.asu.edu/ftp/sparse_sdp.html`. The main SDP codes are regularly benchmarked on the web site of Hans Mittelmann with respect to these test sets; see [22] as well as `http://plato.asu.edu/bench.html`.

The goal of this paper is to augment these test sets with some recent problems from interesting applications of SDP. The new problems mostly have additional structure that could potentially be exploited by a new generation of solvers.

These instances were chosen with the following guidelines:

- all instances correspond to (relaxations of) optimization problems of independent interest, i.e. no random instances;

- several instances have special structure that may be exploited during pre-processing;
- the sizes range from moderate to large-scale. Thus the difficulty does not always lie in the size of the instances, but may also be due to poor numerical conditioning.

Thus the goal is to stimulate further development of codes for semidefinite programming, including more sophisticated pre-processing techniques for problems with special structure. Moreover, we list the SDP models explicitly where possible, or supply detailed references to enable researchers to investigate better modeling of the SDP problems in cases of numerical ill-conditioning.

Our instances come from the following, recent applications of SDP:

- Estimation of the crossing number in graphs (see §2);
- Computing the Lovász ϑ-function of certain graphs with large symmetry groups, e.g. the Erdös-Renyi graphs (see §3);
- Computing bounds in coding theory (see §4);
- Computing bounds on kissing numbers in various dimensions (see §5);
- SDP relaxations of quadratic assignment problems (QAP's) (see §6);
- SDP relaxations of the traveling salesman problem (see §7);
- Two types of structural optimization problems, namely truss topology design problems (see §8.1) and free material optimization problems (see §8.2).

SDP and structures in the data

We will consider SDP problems in the standard form:

$$\min_{X \succeq 0} \operatorname{trace}(A_0 X) \ \text{ subject to } \ \operatorname{trace}(A_k X) = b_k \ \ (k = 1, \ldots, m_{SDP}), \qquad (1)$$

where the data matrices $A_i \in \mathbb{R}^{n_{SDP} \times n_{SDP}}$ are symmetric, and $X \succeq 0$ means X is positive semi-definite.

This standard form is used by most SDP input formats, including the sparse SDPA input format (see [31] for details), that we have used for the new problem library.

There are currently three types of structures (apart from general sparsity) that may be exploited in SDP.

Chordal structure

Here the matrices A_i $(i = 0, \ldots, m_{SDP})$ have a common sparsity pattern, and this pattern is the same as the sparsity pattern of the adjacency matrix of some chordal graph. (Recall that a graph is called chordal if it does not contain a cycle of length 4 or more as an induced subgraph.) This structure is already exploited by the solver SDPA-C [24].

Low rank

Here the matrices A_i ($i = 0, \ldots, m_{SDP}$) have low rank. This structure is already exploited by the solvers DSDP [4] and the latest version of SDPT3 [30].

Algebraic symmetry

Here the matrices A_i ($i = 0, \ldots, m_{SDP}$) belong to a matrix $*$-algebra of low dimension. (Recall that a matrix $*$-algebra (over $\mathbb{R}$) is a subspace of $\mathbb{R}^{n \times n}$ that is also closed under matrix multiplication and taking transposes.)

This structure is not yet exploited in any SDP software, but some preprocessing techniques have been suggested in [23].

Several instances in the new library have algebraic symmetry, in order to encourage the development of techniques to exploit this structure.

Location of the problem libary

The sparse SDPA input files for the new library are available from the web site:

`http://lyrawww.uvt.nl/~sotirovr/library/`

2 Bounding the crossing number of complete bipartite graphs

The *crossing number* $\mathrm{cr}(G)$ of a graph G is the minimum number of intersections of edges (at a point other than a vertex) in a drawing of G in the plane.

In the earliest known instance of a crossing number question, Paul Turán raised the problem of calculating the crossing number of the complete bipartite graph $K_{m,n}$.

Zarankiewicz published a paper [32] in 1954, in which he claimed that $\mathrm{cr}(K_{m,n}) = Z(m,n)$ for all positive integers m, n, where

$$Z(m,n) = \left\lfloor \frac{m-1}{2} \right\rfloor \left\lfloor \frac{m}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{n}{2} \right\rfloor. \tag{2}$$

However, several years later it became clear that there was a mistake in Zarankiewicz's argument. A comprehensive account of the history of the problem, including a discussion of the gap in Zarankiewicz's argument, is given by Guy [10].

Figure 2 shows a drawing of $K_{4,5}$ with 8 crossings. As Zarankiewicz observed, such a drawing strategy can be naturally generalized to construct, for any positive integers m, n, drawings of $K_{m,n}$ with exactly $Z(m,n)$ crossings. This implies that

$$\mathrm{cr}(K_{m,n}) \leq Z(m,n).$$

No one has yet exhibited a drawing of any $K_{m,n}$ with fewer than $Z(m,n)$

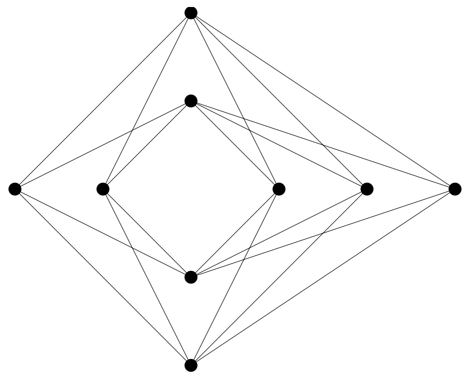

Figure 1: A drawing of $K_{4,5}$ with 8 crossings. A similar strategy can be used to construct drawings of $K_{m,n}$ with exactly $Z(m,n)$ crossings.

crossings, and the following is therefore commonly known as *Zarankiewicz's Crossing–Number Conjecture*:

$$\mathrm{cr}(K_{m,n}) \stackrel{?}{=} Z(m,n), \qquad \text{for all positive integers } m,n.$$

2.1 A lower bound on cr($K_{m,n}$) via SDP

De Klerk et al. [13] showed that one may obtain a lower bound on $\mathrm{cr}(K_{m,n})$ via the optimal value of a suitable SDP problem.

$$\begin{aligned} & \mathrm{cr}(K_{m,n}) \\ \geq \; & \frac{n}{2}\left(n \min\{x^TQx \;\mid\; x \in \mathbb{R}_+^{(m-1)!}, e^Tx = 1\} - \left\lfloor \frac{m}{2} \right\rfloor \left\lfloor \frac{m-1}{2} \right\rfloor\right) \\ \geq \; & \frac{n}{2}\left(n \min_{X \geq 0,\, X \succeq 0}\{\mathrm{trace}(QX) \;\mid\; \mathrm{trace}(JX) = 1\} - \left\lfloor \frac{m}{2} \right\rfloor \left\lfloor \frac{m-1}{2} \right\rfloor\right), \end{aligned}$$

where Q is a certain (given) matrix of order $(m-1)!$, and J is the all-ones matrix of the same size.

De Klerk and al. [13] could solve the SDP problem for $m = 7$ by exploiting the algebraic structure of the matrix Q, to obtain the bound:

$$\mathrm{cr}(K_{7,n}) > 2.1796n^2 - 4.5n.$$

Using an averaging argument, the bound for $\mathrm{cr}(K_{7,n})$ implies the following asymptotic bound on $\mathrm{cr}(K_{m,n})$:

$$\lim_{n\to\infty} \frac{\mathrm{cr}(K_{m,n})}{Z(m,n)} \geq 0.83\frac{m}{m-1}.$$

Thus loosely speaking, asymptotically, $Z(m, n)$ and $\mathrm{cr}(K_{m,n})$ do not differ by more than 17%.

In subsequent, related work, De Klerk, Schrijver and Pasechnik [14] improved the constant 0.83 to 0.859 by solving the SDP for $m = 9$. This was possible by exploiting the algebraic structure of Q in a more sophisticated way. Nevertheless, the final SDP formulation had more than 4×10^7 nonzero data entries.

Instances in the test set

The test set instances have names of the form `crossing_K_mn` for $m = 7, 8, 9$. The optimal values correspond to the values

$$\min_{X \geq 0,\ X \succeq 0} \{\mathrm{trace}(QX) \ \mid \ \mathrm{trace}(JX) = 1\}$$

mentioned above, and the formulations of the SDP instances are those described in [14].

The largest instance `crossing_K_9n` is truly large-scale, and all three instances have algebraic symmetry that may be exploited during pre-processing.

3 The Lovász ϑ and Schrijver ϑ' numbers

The Lovász ϑ number of a graph $G = (V, E)$, introduced in [21],

$$\left.\begin{array}{rcl} \vartheta(G) & := & \max \mathrm{tr}(JX) \\ & & \text{s.t. } X_{ij} = 0, \quad \{i,j\} \in E \ (i \neq j) \\ & & \quad\ \mathrm{tr}(X) = 1 \\ & & \quad\ X \succeq 0, \end{array}\right\} \tag{3}$$

gives an upper bound on the stability number $\alpha(G)$ of G. The related Schrijver ϑ'–number [27] is defined as:

$$\left.\begin{array}{rcl} \vartheta'(G) & := & \max \mathrm{tr}(JX) \\ & & \text{s.t. } \mathrm{tr}((A+I)X) = 1 \\ & & \quad\ X \geq 0 \\ & & \quad\ X \succeq 0. \end{array}\right\} \tag{4}$$

Clearly one has

$$\alpha(G) \leq \vartheta'(G) \leq \vartheta(G).$$

These SDP problems have an interesting structure if the graph G has a large automorphism group.

An example of such a graph is the Erdös–Rényi graph $ER(q)$ (where $q > 2$ is a given prime number), that has $q^2 + q + 1$ vertices; $q + 1$ of the vertices have degree q and the remaining vertices have degree $q + 1$; for more details, see [17].

Instances in the test set

The test set includes the SDP problems to calculate the ϑ and ϑ' values of $ER(q)$ for various prime values of q up to 31.

The following instances compute $-\vartheta(ER(q))$, where the value of q is clear from the name of the instance: `ThetaER23_red`, `ThetaER23_full`, `ThetaER29_red`, `ThetaER29_full`, `ThetaER31_red`, and `ThetaER31_full`.

All of these instances have algebraic symmetry that may be exploited. The symmetry in the instances with names ending in '`red`' have already been partially exploited; details of the reduced formulations are given in [17].

Similarly, the analogous instances for computing $-\vartheta'(ER(q)$ have names of the form `ThetaPrimeERq_red`, or `ThetaPrimeER2q_full`, where q again takes the values 23, 29, or 31.

4 Bounds from binary codes

A classical problem in coding theory is to find the largest set (or *library*) of binary words with n letters, such that the Hamming distance between two words is at least some given value d.

One may reformulate this as a maximum stable set problem by defining the Hamming graph $H(n, d)$ on 2^n vertices indexed by the binary words of length n, and connecting two vertices if their Hamming distance is less than d. Note that the maximum stable set in $H(n, d)$ corresponds to the largest possible library, and $\alpha(H(n, d))$ is the number of words in this library. It is usual in coding theory to use the notation $A(n, d)) := \alpha(H(n, d))$.

Thus $\vartheta'(H(n, d))$ gives a lower bound on $A(n, d)$, but the resulting SDP reduces to the well-known Delsarte LP bound, as shown by Schrijver in the seminal work [27].

More recently, Schrijver [28] suggested a stronger SDP-based bound for minimal distance codes, that is at least as good as the ϑ' bound, and still of size polynomial in n. Like the ϑ' bound, it is given as the optimal value of an SDP problem.

In order to introduce this bound we require some notation.

For $i, j, t \in \{0, 1, \ldots, n\}$, and $X, Y \in \{0, 1\}^n$ define the matrices

$$\left(M^t_{i,j}\right)_{X,Y} = \begin{cases} 1 & \text{if } |X| = i,\ |Y| = j,\ d_H(X, Y) = n - t \\ 0 & \text{otherwise,} \end{cases}$$

where d_H denotes the Hamming distance.

The upper bound is given as the optimal value of the following semidefinite program:

$$A(n, d) \leq \max \sum_{i=0}^{n} \binom{n}{i} x^0_{i,0}$$

subject to

$$
\begin{aligned}
x_{0,0}^0 &= 1 \\
0 &\le x_{i,j}^t \le x_{i,0}^0 \text{ for all } i,j,t \in \{0,\ldots,n\} \\
x_{i,j}^t &= x_{i',j'}^{t'} \text{ if } \{i',j',i'+j'-2t'\} \text{ is a permutation of } \{i,j,i+j-2t\} \\
x_{i,j}^t &= 0 \text{ if } \{i,j,i+j-2t\} \cap \{1,\ldots,d-1\} \neq \emptyset,
\end{aligned}
$$

as well as

$$\sum_{i,j,t} x_{i,j}^t M_{i,j}^t \succeq 0, \ \sum_{i,j,t} \left(x_{i+j-2t,0}^0 - x_{i,j}^t\right) M_{i,j}^t \succeq 0.$$

The matrices $M_{i,j}^t$ are of order 2^n and the SDP problem is therefore too large to solve if, say $n \ge 10$. Schrijver pointed out that these matrices form a basis of the Terwilliger algebra of the Hamming scheme, and worked out the details for computing the irreducible block diagonalization of this (non-commutative) matrix algebra of dimension $O(n^3)$.

Thus, the constraint $\sum_{i,j,t} x_{i,j}^t M_{i,j}^t \succeq 0$ is replaced by

$$\sum_{i,j,t} x_{i,j}^t Q^T M_{i,j}^t Q \succeq 0$$

where Q is an orthogonal matrix that gives the irreducible block diagonalization. For details the reader is referred to Schrijver [28]. Since SDP solvers can exploit block diagonal structure, this reduces the sizes of the matrices in question to the extent that computation is possible in the range $n \le 32$.

Laurent [20] suggested a refinement of the Schrijver relaxation that takes the following form:

$$A(n,d) \le \max 2^n x_{0,0}^0$$

subject to

$$
\begin{aligned}
0 &\le x_{i,j}^t \le x_{i,0}^0 \text{ for all } i,j,t \in \{0,\ldots,n\} \\
x_{i,j}^t &= x_{i',j'}^{t'} \text{ if } \{i',j',i'+j'-2t'\} \text{ is a permutation of } \{i,j,i+j-2t\} \\
x_{i,j}^t &= 0 \text{ if } \{i,j,i+j-2t\} \cap \{1,\ldots,d-1\} \neq \emptyset,
\end{aligned}
$$

as well as

$$\sum_{i,j,t} x_{i,j}^t M_{i,j}^t \succeq 0$$

and

$$\begin{pmatrix} 1 - x_{0,0}^0 & c^T \\ c & \sum_{i,j,t} \left(x_{i+j-2t,0}^0 - x_{i,j}^t\right) M_{i,j}^t \end{pmatrix} \succeq 0,$$

where $c := \sum_{i=0}^n \left(x_{0,0}^0 - x_{0,i}^0\right) \chi_i$, and χ_i is defined by

$$(\chi_i)_X := \begin{cases} 1 & \text{if } |X| = i \\ 0 & \text{else.} \end{cases}$$

This SDP problem may be block-diagonalised as before to obtain an SDP of size $O(n^3)$.

Instances in the test set

The test set contains the SDP relaxations of both the Schrijver and Laurent for the values $A(19,6)$, $A(26,10)$, $A(28,8)$, $A(37,15)$ (Schrijver relaxation only), $A(40,15)$ (Schrijver relaxation only), $A(48,15)$, $A(50,15)$, and $A(50,23)$.

To obtain the upper bounds on $A(n,d)$, the optimal values of the `Laurent` instances have to be multiplied by -2^n. For the `Schrijver` instances, one should add $+1$ to minus the optimal value.

These instances are badly conditioned numerically, and it is difficult to obtain their optimal values to high accuracy.

5 SDP bounds on kissing numbers

The kissing number of $\mathbb{R}^n$ is defined as the maximum number of unit balls that can simultaneously touch a unit ball centered at the origin, without any overlap.

Thus the kissing number of $\mathbb{R}^2$ is 6 and in $\mathbb{R}^3$ it is 12. (There was a famous disagreement between Newton and Gregory on whether the correct answer is 12 or 13).

Not much is known about kissing numbers for general values of n, and it is interesting to compute upper bounds for fixed n.

In a seminal paper by Bachoc and Vallentin [2], the authors introduce new SDP relaxations of this problem, and succeeded to compute the improved upper bounds on the kissing number in the dimensions $n = 5, 6, 7, 9$ and 10.

The statement of the SDP problem may be found in Theorem 4.2 in [2]; we do not reproduce it here because of the amount of extra notation required.

Instances in the test set

The names of the test set instances are of the form `kissing_n_d_N`, where n indicates the dimension. The parameters d and N are used in constructing the SDP relaxation and their meaning is explained in §5 of [2]. An upper bound on the kissing number of $\mathbb{R}^n$ may be obtained from the optimal value of each instance by multiplying it by -1 and adding one to the answer.

The instances are all badly conditioned numerically, and it is therefore difficult to obtain accurate solutions. The last two instances are unsolved at the time of writing. Solving them may well provide improved upper bounds on the kissing numbers of $\mathbb{R}^{11}$ and $\mathbb{R}^{12}$ respectively.

6 SDP relaxation of the quadratic assignment problem

The quadratic assignment problem (QAP) may be stated in the following form:

$$\min_{X \in \Pi_n} \operatorname{trace}(AXBX^T) \tag{5}$$

where A and B are given symmetric $n \times n$ matrices, and Π_n is the set of $n \times n$ permutation matrices.

The QAP has many applications in facility location, circuit design, graph isomorphism and other problems, but is NP-hard in the strong sense, and hard to solve in practice for $n \geq 30$; for a review, see Anstreicher [1].

An SDP relxation of (QAP) from [33] and [16] takes the form:

$$\left.\begin{array}{rl} \min & \operatorname{trace}(B \otimes A)Y \\ \text{subject to} & \\ & \operatorname{trace}((I \otimes (J-I))Y + ((J-I) \otimes I)Y) = 0 \\ & \operatorname{trace}(Y) - 2e^T y = -n \\ & \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix} \succeq 0, \;\; Y \geq 0. \end{array}\right\} \tag{6}$$

It is easy to verify that this is indeed a relaxation of QAP, by noting that setting $Y = \operatorname{vec}(X)\operatorname{vec}(X)^T$ and $y = \operatorname{diag}(Y)$ gives a feasible solution if $X \in \Pi_n$.

These SDP problems form challenging test instances, since the matrix variable Y is nonnegative and of order n^2.

Instances in the test set

The SDP instances in the test set correspond to the SDP relaxation (6) of two QAP instances from the QAPLIB [8] library, namely `Esc16e` and `Esc64a`. The names of the SDP instances start with `QAP` and if the name ends with `red` it means that the algebraic symmetry has been (partially) exploited (see [16] for details).

7 The traveling salesman problem

It is well-known that the QAP contains the symmetric traveling salesman problem (TSP) as a special case. To show this, we denote the complete graph on n vertices with edge lengths (weights) $D_{ij} = D_{ji} > 0$ $(i \neq j)$, by $K_n(D)$, where D is called the matrix of edge lengths (weights). The TSP is to find a Hamiltonian circuit of minimum length in $K_n(D)$. The n vertices are often called *cities*, and the Hamiltonian circuit of minimum length the *optimal tour*.

To see that TSP is a special case of QAP, let C_1 denote the adjacency matrix of the standard circuit on n vertices:

$$C_1 := \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \\ 0 & & & & 0 & 1 \\ 1 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix}.$$

Now the TSP problem is obtained from the QAP problem (5) by setting $A = \frac{1}{2}D$ and $B = C_1$. To see this, note that every Hamiltonian circuit in a complete graph has adjacency matrix XC_1X^T for some $X \in \Pi_n$. Thus we may concisely state the TSP as

$$\min_{X \in \Pi_n} \operatorname{trace}\left(\frac{1}{2}DXC_1X^T\right).$$

It was shown by De Klerk et al. [15] that the SDP relaxation (6) reduces to the following problem for the special case of TSP.

$$\min \frac{1}{2} \operatorname{trace}\left(DX^{(1)}\right)$$

subject to

$$\left.\begin{array}{rcll} X^{(k)} & \geq & 0, & k = 1, \ldots, d \\ \sum_{k=1}^d X^{(k)} & = & J - I, & \\ I + \sum_{k=1}^d \cos\left(\frac{2\pi ik}{n}\right) X^{(k)} & \succeq & 0, & i = 1, \ldots, d \\ X^{(k)} & \in & \mathcal{S}^n, & k = 1, \ldots, d, \end{array}\right\} \qquad (7)$$

where $d = \lfloor \frac{1}{2}n \rfloor$.

Note that this problem only involves matrix variables $X^{(1)}, \ldots, X^{(d)}$ of order n as opposed to the matrix variable of order n^2 in (6), i.e. the problem size is reduced by a factor n in this sense. Nevertheless, these SDP problems are of a challenging size if $n \geq 30$.

Instances in the test set

The SDP relaxation (7) of the TSPlib[1] instances `eil51` and `bays29` are given by the SDP instances `TSPeil51` and `TSPbays29` respectively.

8 Structural optimization

8.1 Truss topology design problems

We consider a truss defined by a ground structure of nodes and bars. Let m be the number of bars.

Let $b \in \mathbb{R}^m$ be the vector of bar lengths, and $z \in \mathbb{R}^m$ the vector of cross-sectional areas. A specific topology optimization problem, introduced in [12], is to find a truss of minimum volume such that the fundamental frequency of vibration is higher than some prescribed critical value :

$$\begin{array}{lll} \text{(TOP)} & \min & \sum_{i=1}^m b_i z_i \\ & \text{s.t.} & S = \sum_{i=1}^m (K_i - \bar{\Omega} M_i) z_i - \bar{\Omega} M_0 \\ & & z_i \geq 0 \quad i = 1, \ldots, m \\ & & S \succeq 0, \end{array}$$

[1] `http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/`

where $\bar{\Omega}$ is a lower bound on the (squared) fundamental frequency of vibration of the truss, and M_0 the so-called non-structural mass matrix. The matrices $x_i K_i$ and $x_i M_i$ are called the stiffness and mass matrices of bar i respectively. The order of these matrices is the number of free nodes in the structure times the degrees of freedom.

If the ground structure of nodes and bars has isometries, then the SDP problem has algebraic symmetry that may be exploited. An example of such a truss is shown in Figure 2.

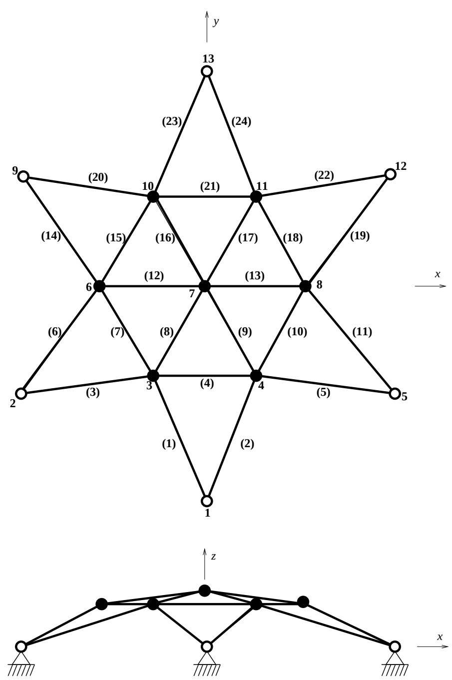

Figure 2: Top and side views of a spherical lattice dome with D_6 symmetry. The black nodes are free and the white nodes fixed.

Instances in the test set

The test set includes instances that generalize the truss in Figure 2 to have D_n symmetry, where D_n is the dihedral symmetry group of a regular n-gon. Thus the outer ring of nodes form a regular n-gon as opposed to the hexagon in the figure. Thus these instances have algebraic symmetry. The names of the instances start with `Truss` followed by the value of n (502 or 1002). The algebraic symmetry has been partially exploited for the instances with names ending in `no_blocks`, as described in [3].

8.2 Free material optimization

Free material optimization refers to structural design problems where the material properties are allowed to change continuously in the final design.

A recent "real world application" of free material optimization was in the conceptual design phase of the leading edge rib of the Airbus A380. This project is reviewed in the recent paper [19], and is illustrated in Figure 3.

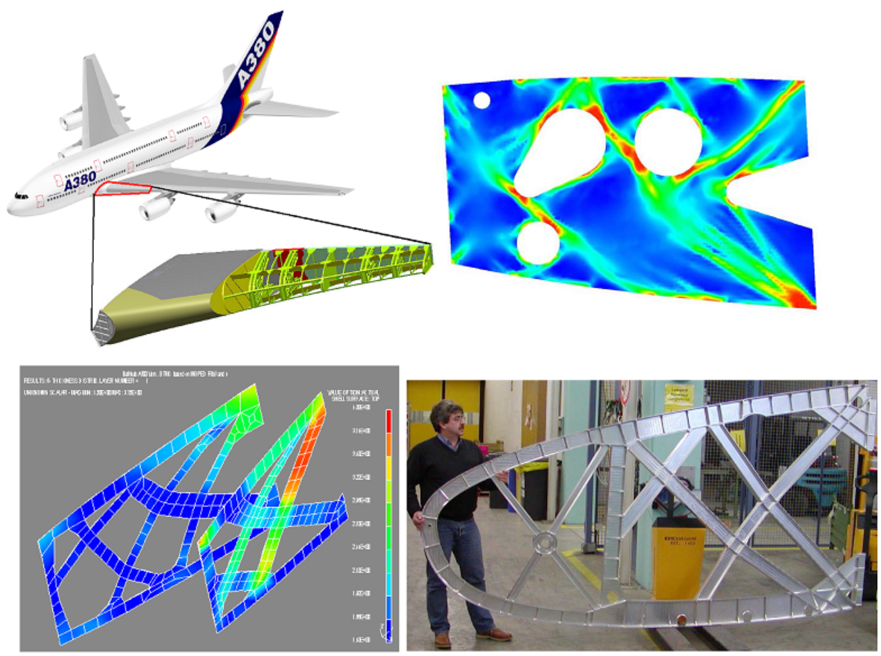

Figure 3: Design stages of a leading edge rib of the Airbus A380: the wing leading edge, sample free material optimization result, post-optimization design incorporating technological constraints, and the final product.

The SDP model that was used is not reproduced here due to the amount of notation required, but may be found in Theorem 3.1 in [5].

Instances in the test set

The instances `r3_l`, `r3_m`, and `r3_s` correspond to three different versions (discretizations and loading scenarios) of worst-case multiple load scenarios. These instances have a structured block sparsity that may be extended to a chordal sparsity structure.

9 Summary of the instances

In Table 1 we provide detailed information on the new SDP instances. The column 'size' refers to the size of the (uncompressed) sparse SDPA input file of the instance. The column 'optimal value' lists the optimal value to the accuracy within which it is known for each instance.

The values n_{SDP} and m_{SDP} refer to parameters for the standard form SDP (1), and $n_{\max}$ refers to the largest block in the matrix variable in case of block diagonal structure. The final column 'Structure' indicates known structure(s) in the problem data, e.g. algebraic symmetry or low rank in the data matrices. Thus, for example, 'symmetry' means that it is possible to reduce the problem size by exploiting algebraic symmetry.

Acknowledgements

The authors would like to thank Lex Schrijver, Dima Pasechnik, and Frank Vallentin for making their problem generators available for the instances in §2, §4, and §5 respectively. The authors are also indebted to Michael Stingl and Michael Kočvara for supplying the free material optimization instances, and the pictures in Figure 3. The authors would also like to thank Hans Mittelmann for valuable comments, and for providing more accurate optimal values for several of the SDP instances in the new library.

References

[1] K.M. Anstreicher. Recent advances in the solution of quadratic assignment problems. *Mathematical Programming, Ser. B*, **97**: 27–42, 2003.

[2] C. Bachoc and F. Vallentin. New upper bounds for kissing numbers from semidefinite programming. *Journal of the AMS*, **21**, 909-924, 2008.

[3] Y.-Q. Bai, E. de Klerk, D.V. Pasechnik, and R. Sotirov. Exploiting group symmetry in truss topology optimization. *Optimization and Engineering* (to appear).

[4] S. J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM J. Optim.*, 10(2):443–461 (electronic), 2000.

[5] A. Ben-Tal, M. Kočvara, A. Nemirovski and J. Zowe. Free material optimization via semidefinite programming: the multiload case with contact conditions. *SIAM Review*, 42(4): 695 – 715, 2000.

[6] B. Borchers. CSDP, a C library for semidefinite programming. *Optimization Methods and Software*, **11/12**(1-4):613–623, 1999.

[7] B. Borchers. SDPLIB 1.2, A Library of Semidefinite Programming Test Problems. *Optimization Methods and Software*, **11**(1):683–690, 1999.

[8] R.E. Burkard, S.E. Karisch, and F. Rendl. QAPLIB — a quadratic assignment problem library. *Journal on Global Optimization*, **10**: 291–403, 1997; see also `http://www.seas.upenn.edu/qaplib/`.

[9] S. Burer and R.D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming Series B*, **95**(2):329–357, 2003.

[10] R.K. Guy, The decline and fall of Zarankiewicz's Theorem. *Proof Techniques in Graph Theory* (Proc. Second Ann Arbor Graph Theory Conf., Ann Arbor, Mich., 1968), 63–69. Academic Press, New York.

[11] C. Helmberg. Numerical Evaluation of SBmethod. *Mathematical Programming*, **95**(2), 381–406, 2003.

[12] Y. Kanno, M. Ohsaki, K. Murota and N. Katoh. Group symmetry in interior-point methods for semidefinite program, *Optimization and Engineering*, **2**(3): 293–320, 2001.

[13] E. de Klerk, J. Maharry, D.V. Pasechnik, B. Richter and G. Salazar. Improved bounds for the crossing numbers of $K_{m,n}$ and K_n. *SIAM Journal on Discrete Mathematics* **20**:189–202, 2006.

[14] E. de Klerk, D.V. Pasechnik and A. Schrijver. Reduction of symmetric semidefinite programs using the regular *-representation. *Mathematical Programming B*, **109**(2-3):613-624, 2007.

[15] E. de Klerk, D.V. Pasechnik and R. Sotirov. On semidefinite programming relaxations of the traveling salesman problem. *SIAM Journal on Optimization*, to appear.

[16] E. de Klerk and R. Sotirov. Exploiting Group Symmetry in Semidefinite Programming Relaxations of the Quadratic Assignment Problem. CentER Discussion Paper 2007-44, Tilburg University, The Netherlands, 2007. Available at: `http://arno.uvt.nl/show.cgi?fid=60929`

[17] E. de Klerk, M.W. Newman, D.V. Pasechnik, and R. Sotirov. On the Lovász ϑ-number of almost regular graphs with application to Erdös-Rényi graphs. CentER Discussion paper 2006-93, Tilburg University, The Netherlands, September 2006.

[18] M. Kočvara and M. Stingl. On the solution of large-scale SDP problems by the modified barrier method using iterative solvers. *Mathematical Programming Series B*, **109**(2-3):413–444, 2007.

[19] M. Kočvara, M. Stingl and J. Zowe. Free material optimization: recent progress. *Optimization*, 57(1), 79 – 100, 2008.

[20] M. Laurent. Strengthened semidefinite bounds for codes. *Mathematical Programming*, **109**(2-3):239–261, 2007.

[21] L. Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information theory*, **25**:1–7, 1979.

[22] H.D. Mittelmann, An Independent Benchmarking of SDP and SOCP solvers, *Mathematical Programming*, **95**, 407–430, 2003.

[23] K. Murota, Y. Kanno, M. Kojima and S. Kojima, A Numerical Algorithm for Block-Diagonal Decomposition of Matrix *-Algebras, Technical report B-445, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, September 2007 (Revised June 2008).

[24] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota. Exploiting sparsity in semidefinite programming via matrix completion II: Implementation and numerical results," *Mathematical Programming* Series B, **95**, 303–327, 2003.

[25] Yu. Nesterov and A.S. Nemirovski. *Interior point polynomial algorithms in convex programming.* SIAM Studies in Applied Mathematics, Vol. 13. SIAM, Philadelphia, USA, 1994.

[26] G. Pataki and S. Schmieta. The DIMACS library of mixed semidefinite-quadratic-linear programs. Technical Report, Columbia University, 2002. Available at `http://dimacs.rutgers.edu/Challenges/Seventh/Instances/`

[27] A. Schrijver. A comparison of the Delsarte and Lovász bounds. *IEEE Transactions on Information Theory*, **25**:425–429, 1979.

[28] A. Schrijver. New code upper bounds from the Terwilliger algebra. *IEEE Transactions on Information Theory*, **51**:2859–2866, 2005.

[29] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, **11-12**:625–653, 1999.

[30] K. C. Toh, M. J. Todd, and R. H. Tütüncü. SDPT3—a MATLAB software package for semidefinite programming, version 1.3. *Optimization Methods and Software*, **11/12**(1-4):545–581, 1999.

[31] M. Yamashita, K. Fujisawa, and M. Kojima. Implementation and evaluation of SDPA 6.0 (SemiDefinite Programming Algorithm 6.0), *Optimization Methods and Software* **18**, 491–505, 2003.

[32] K. Zarankiewicz, On a problem of P. Turán concerning graphs. *Fundamenta Mathematicae* **41**, 137–145, 1954.

[33] Q. Zhao, S.E. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite Programming Relaxations for the Quadratic Assignment Problem. *Journal of Combinatorial Optimization*, **2**:71–109, 1998.

Instance	Size (MB)	Optimal value	n_{SDP}	m_{SDP}	$n_{\max}$	Structure
crossing_K_7n	0.350	4.3593154965	135	56	78	symmetry
crossing_K_8n	14.689	5.8599856444	620	239	380	symmetry
crossing_K_9n	848.3	7.735212	3805	1366	2438	symmetry
ThetaER23_red	1.387	-96.240780	26555	57	57	symmetry
ThetaER23_full	10.934		13803	153181	553	symmetry
ThetaER29_red	2.739	-137.07352	52271	69	69	symmetry
ThetaER29_full	27.015		26973	379756	871	symmetry
ThetaER31_red	3.332	-151.953731	63563	73	73	symmetry
ThetaER31_full	35.052		32739	493521	993	symmetry
ThetaPrimeER23_red	0.292	-96.240038	116	57	57	symmetry
ThetaPrimeER23_full	25.049		306364	153181	553	symmetry
ThetaPrimeER29_red	0.559	-136.97842	140	69	69	symmetry
ThetaPrimeER29_full	62.695		759514	379756	871	symmetry
ThetaPrimeER31_red	0.676	-151.7024268	148	73	73	symmetry
Schrijver_A(19,6)	0.328	-1279.036270	632	156	20	
Schrijver_A(26,10)	0.743	-8.8e+02*	999	227	27	
Schrijver_A(28,8)	1.432	-3.215079e+04	1746	466	29	
Schrijver_A(37,15)	2.413	?*	2049	468	38	
Schrijver_A(40,15)	3.739	?*	2900	720	41	
Schrijver_A(48,15)	9.706	-2e+06*	6198	1723	49	
Schrijver_A(50,15)	11.872	-7.5e+06*	7278	2056	51	
Schrijver_A(50,23)	5.403	?*	3024	606	51	
Laurent_A(19,6)	0.364	-2.44151e-03	668	157	21	
Laurent_A(26,10)	0.807	-1.5e-05*	1045	228	28	
Laurent_A(28,8)	1.539	-1.1985e-04*	1798	467	30	
Laurent_A(48,15)	10.142	-1.0e-08*	6283	1724	50	
Laurent_A(50,15)	12.382	-7e-09*	7367	2057	52	
Laurent_A(50,23)	5.674	-1e-11*	3105	607	52	
kissing_3_5_5	0.231	-11.872060	220	297	56	
kissing_3_10_10	9.206	-11.4385*	1210	1792	286	
kissing_4_7_7	1.283	-23.579687	488	695	120	
kissing_4_10_10	9.209	-23.14*	1210	1792	286	
kissing_5_10_10	9.213	-44.15*	1210	1792	286	
kissing_6_10_10	9.213	-77.9*	1210	1792	286	
kissing_7_10_10	9.215	-134.3*	1210	1792	286	
kissing_8_10_10	9.217	-2.3e+02*	1210	1792	286	
kissing_9_10_10	9.218	-3e+02*	1210	1792	286	
kissing_10_10_10	9.219	-5e+02*	1210	1792	286	
kissing_11_10_10	9.220	-8.89745e+02*	1210	1792	286	
kissing_12_10_10	9.221	-1.369485e+03*	1210	1792	286	
QAP_Esc16e_red	0.79	26.33679747	179	90	17	symmetry
QAP_Esc16e_part_red	0.216	26.33679747	351	90	257	chordal
QAP_Esc16e_full	6.833		66053	33152	257	symmetry
QAP_Esc64a_red	2.745	97.74*	976	517	65	symmetry
QAP_Esc64a_part_red	25.090	97.749	4618	517	4097	symmetry
TSPbays29.txt	4.997	1999.7*	13862	6090	29	symmetry
TSPeil51.txt	42.522	?	71502	33150	51	symmetry
Truss502_no_blocks	0.634	1615235.6*	1512	3	1509	symmetry
Truss502_full	309.305		4557679	1141303	1509	symmetry low rank
Truss1002_no_blocks	1.286	9.82e+06*	3012	3	3009	symmetry
r3_l	37.279	-3.9587422e-01	72932	27555	11	chordal
r3_m	9.156	-3.4820148e-01	18493	7241	11	chordal
r3_s	4.634	-3.9876491e-01	11769	3621	7	chordal

Table 1: Details on the test set. A "*" indicates numerical difficulties, i.e. the optimal values thus indicated are uncertain or only known with low accuracy.