Computing Strong Bounds in Combinatorial Optimization

Hans D Mittelmann

School of Mathematical and Statistical Sciences
Arizona State University

Numerical Analysis and Optimization IV
Sultan Qaboos University
Muscat, Oman
January 2017

Outline

Computing lower bounds for QAPs

- The basic method and QAPLIB results
- Make method more efficient
- Solving a real-life problem from communications

Computing upper bounds for kissing numbers and binary codes

- High precision SDP bounds for kissing numbers
- High precision SDP bounds for binary codes

Outline

Computing lower bounds for QAPs

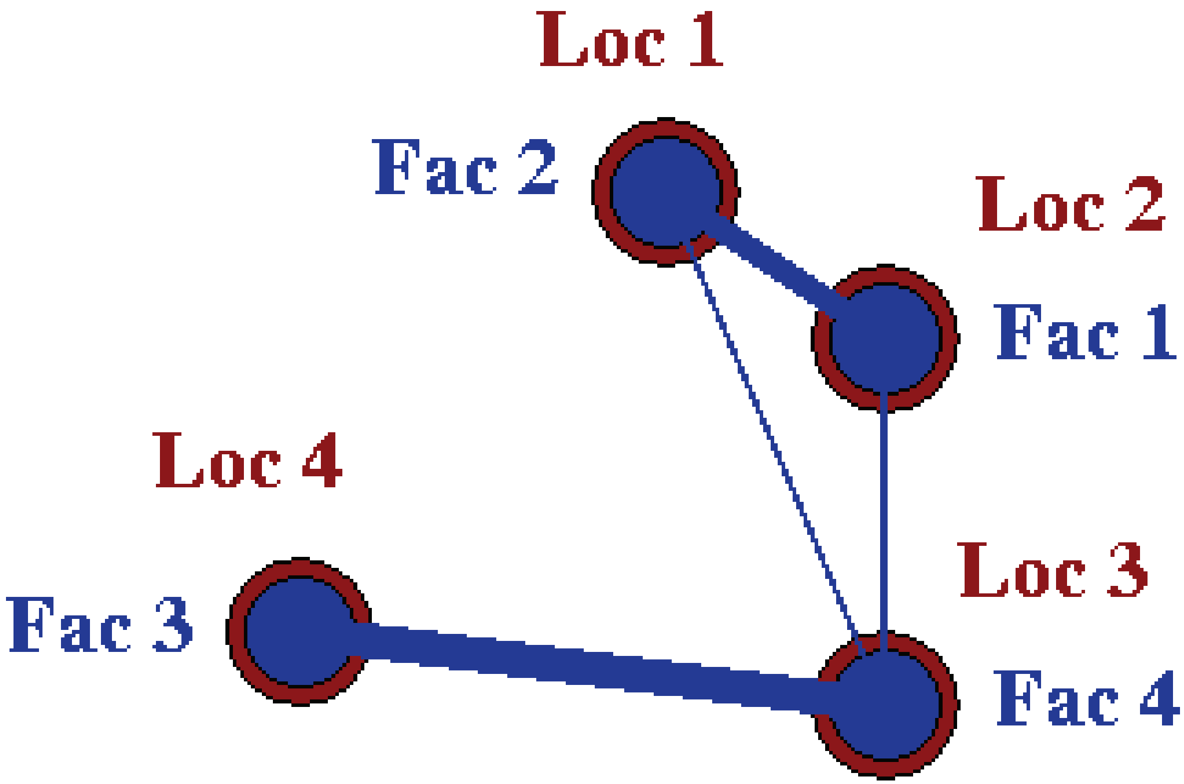
- The basic method and QAPLIB results
- Make method more efficient
- Solving a real-life problem from communications

Computing upper bounds for kissing numbers and binary codes

- High precision SDP bounds for kissing numbers
- High precision SDP bounds for binary codes

A QAP with four locations and facilities

Thickness of connections indicates level of flow



Optimal permutation: 2 1 4 3

A Mathematical Formulation of the Quadratic Assignment Problem

Mathematically, we can formulate the problem by defining two n by n matrices:

- a *flow matrix* F whose (i,j) element represents the flow between facilities i and j,
- and a *distance matrix* D whose (i,j) element represents the distance between locations i and j.

We represent an assignment by the vector p, which is a *permutation* of the numbers 1, 2, ... , n . p(j) is the location to which facility j is assigned.

With these definitions, the QAP can be written as

$$\min_{p \in \Pi} \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{p(i)p(j)}$$

Computing Lower Bounds for QAPs

via SDP relaxations and matrix splitting

Summary of results from

H. D. Mittelmann, J. Peng, *Estimating Bounds for Quadratic Assignment Problems Associated with the Hamming and Manhattan Distance Matrices based on Semidefinite Programming*, SIAM J. Optim.20, 3408-3426 (2010)

J. Peng, H. D. Mittelmann, X. Li, *A New Relaxation Framework for Quadratic Assignment Problems based on Matrix Splitting*, Math. Prog. Comp. 2, 59-77 (2010)

X, Wu, H. D. Mittelmann, X. Wang, and J. Wang, *On Computation of Performance Bounds of Optimal Index Assignment*, IEEE Trans. Info. Theory 59, 3229-3233 (2011)

Work partly supported by AFOSR under grant FA9550-12-1-0153.

Quadratic Assignment Problem (QAP)

$$\text{(QAP)} \qquad \min_{X \in \Pi} Tr(AXBX^T)$$

Π : the set of permutation matrices.

Both A and B are symmetric with nonnegative elements

- First introduced by Koopmans and Beckmann [1957]
- Many applications from various fields: facility location, communication…[QAPLib]
- Hundreds of papers dedicated to QAPs as listed in a recent survey by Hahn et al. [2007]

Existing approaches for QAPs

- Heuristics [Hahn et al. 07]
  - Genetic algorithm, tabu search, simulated annealing
- Exact methods
  - Branch&bound, cutting planes [Pardalos et al. 97, Brixius and Anstreicher 01, Hahn et al. 01,02]
  - Needs to solve some relaxed problem in the process to get a lower bound
  - The efficacy of the relaxation model plays a crucial role

Cheap Relaxations of QAPs

- Cheap relaxations that can be solved quickly
  - GLB reformulation [Gilmore 62 and Lawler 63],
  - QP relaxation [Anstreicher and Brixius 01],
  - Spectral bound based on eigenvalues and projection [Hadley-Rendl-Wolkowicz 92]
  - Weak bounds have been observed, especially when n becomes large
  - Resulting in a huge number of nodes in a B&B approach

Expensive QAP Relaxations

- LP relaxation based on ILP reformulation [Adams and Sherali 86, 90, Hahn et al. 98,01]
  - $z_{ijkl} = x_{ik} x_{jl}$ with extra constraints on z
- SDP relaxation based on matrix vectorization and Kronecker product [Zhao et al.98, Rendl et al.03]

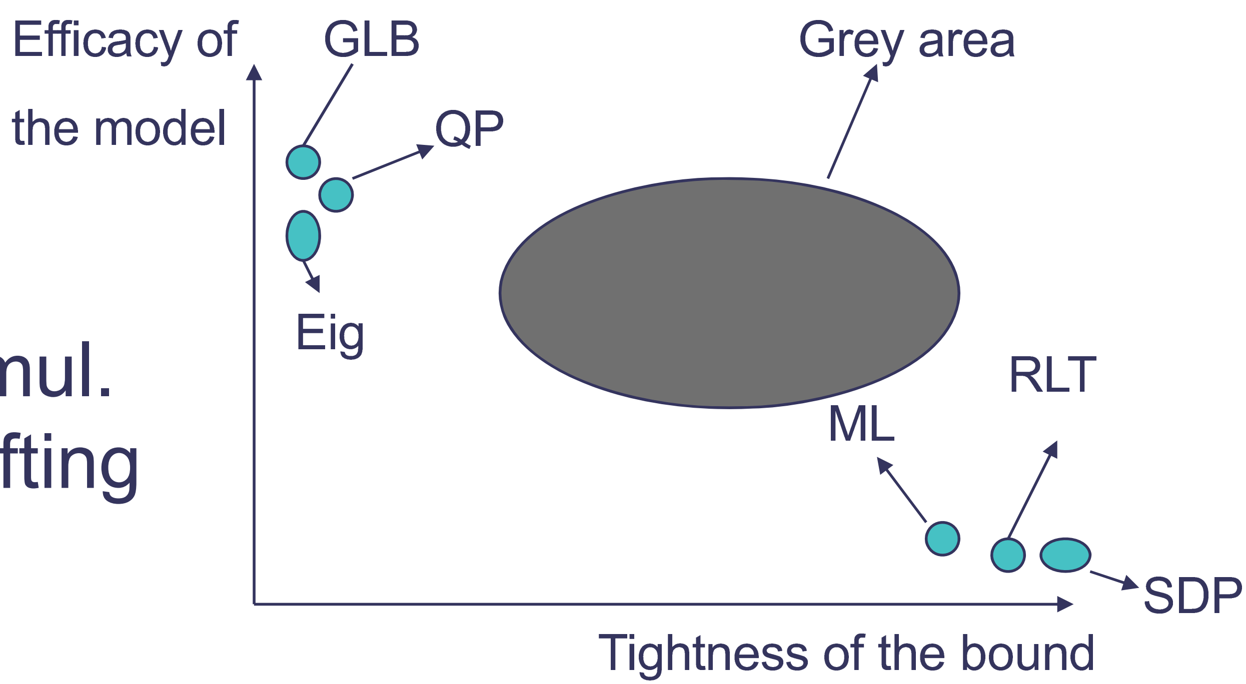
  Let $x = vector(X)$ *and* apply the standard SDP relaxation

  to the matrix xx^T with extra constraints on the matrix elements.

- Tight bound but involves intensive computation
  - Out of the question for QAP instances of size n=50

Efficacy VS Tightness

- RLT: reformul. based on lifting
- ML: matrix lifting



Motivation and Observation

- **Motivation:**
  - find cheap relaxations that yield strong bounds.
- **Observations:**
  - **Most relaxations are based on the binary structure of the matrix elements, not the algebraic feature of the permutation matrix itself!**
  - Specific QAPs arising from data mining have positive semidefinite matrices and large scale problems (n=1000s) have been solved based on SDP approaches

$$B \succ 0 \quad \Leftrightarrow \quad XBX^T \succ 0$$

Matrix Splitting

- **What to do when *B* is not PSD?**
  - Split the matrix into two parts OH,AH

$$B = B^+ - B^-, \quad B^+, B^- \succeq 0.$$

Both XB^+X^T **and** XB^-X^T **are positive semidefinite. Let** $Y^+ = XB^+X^T, Y^- = XB^-X^T$**, we have**

$$Y^+, Y^- \succeq 0.$$

New SDP relaxations

- Let *e* be the all 1 vector and *min(B)* the minimum element of B. Using the properties of *X*, we derive the following relaxation

$$
\begin{aligned}
\min \quad & \mathbf{Tr}(A(Y^+ - Y^-)) && (1)\\
s.t. \quad & Y^+ e = XB^+ e, \quad Y^- e = XB^- e; \\
& \mathbf{diag}(Y^+) = X\mathbf{diag}(B^+), \quad Y^+ \geq \min(B^+); \\
& \mathbf{diag}(Y^-) = X\mathbf{diag}(B^-), \quad Y^- \geq \min(B^-); \\
& Y^+ - XB^+X^T \succeq 0, \quad Y^- - XB^-X^T \succeq 0; \\
& Xe = X^T e = e, \quad X \geq 0.
\end{aligned}
$$

One Theorem

- **Theorem:** The lower bound provided by the new SDP relaxation is always tighter than the bound derived by SDP relaxation based on matrix lifting in [Ding and Wolkowicz 06].
  - As observed in Ding-Wolkowicz paper, such a bound is comparable to the strongest SDP bounds.

Improvement and simplification

- We could swap A and B to derive a more complex SDP relaxation;
- Symmetries can be explored to improve the model;
- We could simplify the SDP constraints to

$$Y^+, Y^- \succeq 0.$$

  - Leading to certain speedup in the solving process, while without much loss of tightness of the bound

New Splitting Schemes: I

- **Definition:** We call the matrix splitting $B=B^+-B^-$ an orthogonal splitting if B^+ and B^- are orthogonal to each other
  - Can be derived by using the singular value decomposition of B directly
  - Additional constraints can be added based on the orthogonality

CVX script for basic model

```
e = ones(n,1);
E = e*e';
I = eye(n);
[V,D] = eig(B);
Dp = max(D, zeros(n));
Dm = max(Dp - D,zeros(n));
Bp = V*Dp*V';
Bm = V*Dm*V';
Dp = sqrtm(Dp);
Dm = sqrtm(Dm);
Rp = V*Dp;
Rm = V*Dm;
```

```
cvx_begin
    variable X(n,n)
    variable Yp(n,n) symmetric
    variable Ym(n,n) symmetric
    variable Zp(n,n)
    variable Zm(n,n)
    minimize( trace(A*(Yp-Ym)) )
    subject to
       diag(Yp) == X*diag(Bp);
       diag(Ym) == X*diag(Bm);
       Yp*e == X*Bp*e;
       Ym*e == X*Bm*e;
       tril(Yp,-1) - tril(Ym,-1) >= min(min(B));
       tril(Yp,-1) >= min(min(tril(Bp,-1)));
       tril(Ym,-1) >= min(min(tril(Bm,-1)));
```

```
%        norm(Yp + Ym,'fro') <= norm(B,'fro');
        Zp == X*Rp;
        Zm == X*Rm;
        lambda_min([I,Zp';Zp,Yp]) >= 0;
        lambda_min([I,Zm';Zm,Ym]) >= 0;
%        lambda_min(Yp) >= 0;
%        lambda_min(Ym) >= 0;
        X    >= 0;
        sum(X)  == 1;
        sum(X') == 1;
cvx_end
```

Numerical Results: I

Prob	$QAP_{L\text{-}S1}$	CPU	$QAP_{S\text{-}L2}$	CPU	Model-1	CPU
Nug25	18%	4665	3%	8914	11%	23s
Nug30	22%	11321	1%	26347	10%	72s
Tail30b	78%	12172	18%	50582	15%	161s
Tail35b	65%	24440	15%	141300	22%	322s
Tail40b	74%	43181	15%	330773	16%	763s

The relative gap is listed for comparison on tightness

Numerical Results: 2

Problem	Tail50b	Tail60b	Tail64C	Tail80b	Tail100b	Tail150b
New gap	18.4%	22.2%	2.4%	18.4%	22%	13.6%
Old gap	91.2%	91.8%	51.7%	89.1%	86.3%	87.4%

All the above problems have been solved within 40 minutes

Strong bounds have been obtained for QAPs of size up to n=256

From QAPLIB: E.D. Taillard [Taillard:91,Taillard:94]

name	n	feas.sol.		permutation/bound		gap
Tai40b	40	637250948	(Ro-TS)	544404685	(SDRMS)	14.57 %
Tai50a	50	4938796	(ITS)	4390920	(L&P)	11.09 %
Tai50b	50	458821517	(Ro-TS)	381474057	(SDRMS)	16.86 %
Tai60a	60	7205962	(TS-2)	5555095	(GLB)	22.91 %
Tai60b	60	608215054	(Ro-TS)	494776302	(SDRMS)	18.65 %
Tai64c	64	1855928	(Ro-TS)	1812779	(SDRMS)	2.32 %
Tai80a	80	13515450	(ITS)	10329674	(GLB)	23.57 %
Tai80b	80	818415043	(Ro-TS)	683526345	(SDRMS)	16.48 %
Tai100a	100	21052466	(ITS)	15824355	(GLB)	24.86 %
Tai100b	100	1185996137	(Ro-TS)	961844607	(SDRMS)	18.90 %
Tai150b	150	498896643	(GEN-3)	435738380	(SDRMS)	12.66 %
Tai256c	256	44759294	(ANT)	43849646	(SDRMS)	2.03 %

Outline

Computing lower bounds for QAPs

- The basic method and QAPLIB results
- Make method more efficient
- Solving a real-life problem from communications

Computing upper bounds for kissing numbers and binary codes

- High precision SDP bounds for kissing numbers
- High precision SDP bounds for binary codes

Desirable Improvements of Our Approach

there are four major ones

- Reduce memory consumption
- Cut CPU time for very large problems
- Make bounds guaranteed lower bounds
- Solve large, real-life problem

All issues were addressed successfully.

Reduce memory consumption

w/o sacrificing quality of bounds

Reason for the large memory consumption:

- Solution of the SDP relaxations via interior point methods that employ direct numerical algebra (Cholesky decomposition)
- Problems are not necessarily sparse

Reduce memory consumption

go iterative!

Key paper for both resource issues:

X.Y. Zhao, D.F. Sun, and K.C. Toh, *A Newton-cg Augmented Lagrangian method for Semidefinite Programming*, SIAM J. Optimization 20, 1737-1765 (2010)

Code available from K.C. Toh

This needed to be combined with CVX in place of SDPT3/SeDuMi

Cut CPU time for very large problems

go iterative!

The Newton-cg method is also faster, requiring fewer and cheaper iterations compared with the IPM

Example to be given with real-life problem in next section

Compute guaranteed bounds
in case SDP solved incompletely

A key paper is here:

C. Jansson, D. Chaykin, and C. Keil, *Rigorous error bounds for the optimal value in semidefinite programming*,
SIAM J. Numerical Analysis, 46 (2007/08), 188-200.

- Both rigorous lower and upper bounds computed
- Employs interval-arithmetic ideas
- Minor post-processing effort

Outline

- Computing lower bounds for QAPs
  - The basic method and QAPLIB results
  - Make method more efficient
  - Solving a real-life problem from communications
- Computing upper bounds for kissing numbers and binary codes
  - High precision SDP bounds for kissing numbers
  - High precision SDP bounds for binary codes

A real-life problem from communications
optimal index assignment

- In communication systems index assignment is the problem of labeling source codewords by binary integer numbers (channel codewords)
- For a source code of fixed integer rate n, there are $\frac{2^n!}{2^n \times n!} = \frac{(2^n-1)!}{n!}$ distinct index assignments
- In the presence of channel errors the overall system performance does depend on the index assignment
- Channel-optimized index assignment of source codewords is the simplest way of improving the system error resilience

A real-life problem from communications

optimal index assignment

- Source codewords that are distant from one another in code space should be indexed by binary numbers of large Hamming distances
- A basic element of a signal compression and communication system is the quantizer Q, either scalar or vector.
- We focus on index assignment of vector quantizers (VQ) for their superior source coding performance
- A vector quantizer $Q : \Re^d \to \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_N\}$ maps a continuous source vector $\mathbf{x} \in \Re^d$ to a codeword $\mathbf{c}_i \in \Re^d$ in the VQ codebook $\mathbb{C} = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_N\}$ by the nearest neighbor rule.
- The index i rather than the codeword $\mathbf{c}_i$ itself is transmitted via the channel. Upon receiving i correctly, the VQ decoder can reconstruct $\mathbf{x}$ from $\mathbf{c}_i$ by inverse quantizer mapping Q^{-1}

A real-life problem from communications

optimal index assignment

- Typically, the size N of the codebook $\mathbb{C}$ is made an integer power of two, $N = 2^n$ so that the codeword index i is a binary number of n bits. An index assignment of $\mathbb{C}$ is a bijection map $\pi : \mathbb{C} \leftrightarrow \{0, 1\}^n$.
- If in the event of a transmission error an index $\pi(\mathbf{c}_i)$ is received as $\pi(\mathbf{c}_j)$, an input vector $\mathbf{x}$ such that $w_i = Q(\mathbf{x})$ will be reconstructed as w_j, incurring an extra channel distortion $d(\mathbf{c}_i, \mathbf{c}_j)$ that does depend on index assignment π.
- Let $P(\jmath|\imath)$ be the probability of transmitting index $\imath$ but receiving index $\jmath$, and $P(\mathbf{c}_i)$ be the prior probability of the codeword $\mathbf{c}_i$.

A real-life problem from communications

optimal index assignment

- Given an index assignment π, the expected channel distortion is

$$\bar{d}_\pi = \sum_{i=1}^{N} P(\mathbf{c}_i) \sum_{j=1}^{N} P(\pi(w_j)|\pi(\mathbf{c}_i)) d(\mathbf{c}_i, \mathbf{c}_j)$$

- Adopting the common probability model of binary symmetric channel (BSC), we have

$$P(\pi(w_j)|\pi(\mathbf{c}_i)) = (1-p)^{n-h(\pi(w_j),\pi(\mathbf{c}_i))} p^{h(\pi(w_j),\pi(\mathbf{c}_i))}$$

  p the BSC crossover probability, and $h(\cdot,\cdot)$ Hamming distance.
- To minimize the expected channel distortion $\bar{d}_\pi$ one needs to find an index assignment defined by the objective function

$$\pi_* = \arg\min_\pi \sum_{i=1}^{N} P(\mathbf{c}_i) \sum_{j=1}^{N} (1-p)^{n-h(\pi(w_j),\pi(\mathbf{c}_i))} p^{h(\pi(w_j),\pi(\mathbf{c}_i))} d(\mathbf{c}_i, \mathbf{c}_j).$$

A real-life problem from communications

optimal index assignment

For convenience, we rewrite in matrix form. Let

- $\mathbf{W} = \text{diag}(P(\mathbf{c}_1), P(\mathbf{c}_2), \cdots, P(\mathbf{c}_N))$ be the diagonal matrix consisting of prior probabilities of the VQ codewords,
- $\mathbf{B} = \{(1-p)^{n-h(i,j)} p^{h(i,j)}\}_{1 \le i \le N, 1 \le j \le N}$ be the symmetric matrix whose elements $B(i,j)$ are the codeword transition probabilities $P(\pi(w_j)|\pi(\mathbf{c}_i))$ due to BSC bit errors of probability p,
- $\mathbf{D} = \{d(\mathbf{c}_i, \mathbf{c}_j)\}_{1 \le i \le N, 1 \le j \le N}$ be the symmetric distance matrix between pairs of codewords, and
- $\mathbf{X}$ be the $N \times N$ permutation matrix to specify π.

A real-life problem from communications

optimal index assignment

- Then,

$$\begin{aligned}\bar{d}_\pi &= \sum_{i=1}^{N} P(\mathbf{c}_i) \sum_{j=1}^{N} \{\mathbf{XBX}^T\}_{i,j} d(\mathbf{c}_i, \mathbf{c}_j) \\ &= \mathrm{trace}(\mathbf{WXBX}^T\mathbf{D}) \\ &= \mathrm{trace}(\mathbf{DWXBX}^T)\end{aligned}$$

Symmetrizing

$$\tilde{\mathbf{D}} = \mathbf{DW} + \mathbf{D}^T\mathbf{W}^T.$$

we finally have

$$\bar{d}_\pi = \frac{1}{2}\mathrm{trace}(\tilde{\mathbf{D}}\mathbf{XBX}^T)$$

A quadratic assignment problem!

A real-life problem from communications

optimal index assignment

- Due to the wide use of VQ in image coding, we present a case study on image VQ index assignment. A training set of 18 natural images is used to design 16-dimensional vector quantizers of various fixed integer rates n.
- We consider the general case (multiple bit errors) or we assume the BSC channel crossover probability p to be sufficiently small and consider only one bit errors This simplifies the codeword transition probability expression to

$$P(\pi(w_j)|\pi(\mathbf{c}_i)) = (1-p)^{n-1}p$$

and consequently,

$$\mathbf{B} = (1-p)^{n-1}p\mathbf{A}$$

where $\mathbf{A}$ is the adjacency matrix of the n-dimensional hypercube.

Table: Different lower and upper bounds

One bit errors

n	PB	GLB	SDP	ILS	gap	NBC	gap
5	102156	84784	304551	358984	0.71 dB	470147	1.88 dB
6	42807	58617	289883	349337	1.06 dB	530994	2.62 dB
7	< 0	45503	242657	334360	1.39 dB	592534	3.89 dB
8	< 0	43942	199959	294756	1.69 dB	650936	5.14 dB
9	< 0	38156	193271	291314	1.78 dB	719776	5.71 dB

Multiple bit errors

n	PB	GLB	SDP	ILS	gap	NBC	gap
5	1146	884	2891	3608	0.96 dB	4685	2.10 dB
6	614	819	2834	3734	1.20 dB	5273	2.70 dB
7	< 0	484	2412	3347	1.42 dB	5871	3.86 dB
8	< 0	465	2020	2984	1.69 dB	6424	5.02 dB
9	< 0	416	1931	2937	1.82 dB	7072	5.64 dB

Outline

Computing lower bounds for QAPs

- The basic method and QAPLIB results
- Make method more efficient
- Solving a real-life problem from communications

Computing upper bounds for kissing numbers and binary codes

- High precision SDP bounds for kissing numbers
- High precision SDP bounds for binary codes

Kissing Number Problem

- In geometry, the *kissing number* τ_n is the maximum number of spheres of radius 1 that can simultaneously touch the unit sphere S^{n-1} in n-dimensional Euclidean space.

1694

Sir Isaac Newton

12 or 13?

David Gregory

What is known in low dimensions?

- $\tau_1 = 2$, $\tau_2 = 6$ is trivial.
- $\tau_3 = 12$, Schütte and van der Waerden (1953).
- $\tau_8 = 240$, $\tau_{24} = 196560$, Odlyzko, Sloane, and Levenshtein (1979).
- $\tau_4 = 24$, Musin (2003).
- *Goal:*
  Find good upper bounds using semidefinite programming.

Which SDP bounds have been found?

- Previous best upper bounds from 1979-2007
- In 2008 paper by C. Bachoc and F. Vallentin with moderately improved bounds up to dimension 10.
- Use of regular (double precision) computation prevented better results.

- New paper:
  H.D. Mittelmann and F. Vallentin,
  High Accuracy Semidefinite Programming Bounds for Kissing Numbers,
  Exp. Math. 19, 174-179 (2010)

- Use of multiple precision SDP solvers (SDPA/CSDP)
- Computations tedious/tricky

Which SDP needs to be solved?

- kissing number is stability number of infinite graph
- stability number is bounded by Lovasz theta number
- Lovasz theta number is solution of SDP
- graph $\Gamma(S^{n-1}, (0, \pi/3))$ on vertex set $S^{n-1} = \{x \in \mathbb{R}^n : x \cdot x = 1\}$
- edges when angular distance < $\pi/3$, inner product > $1/2$
- bounds from this SDP strengthened using symmetries and Lasserre hierarchy

$$\vartheta'(\Gamma(S^{n-1}, (0, \pi/3))) = \inf \Big\{ \lambda : \begin{array}{l} K \in \mathcal{C}(S^{n-1} \times S^{n-1})_{\succeq 0}, \\ K(x, x) = \lambda - 1, \text{ for all } x \in S^{n-1}, \\ K(x, y) \leq -1, \text{ for all } x, y \in S^{n-1} \\ \quad \text{with } x \cdot y \leq 1/2 \Big\}, \end{array}$$

- $\mathcal{C}(S^{n-1} \times S^{n-1})_{\succeq 0}$ cone of positive definite Hilbert-Schmidt kernels

Our improved bounds as recorded in Wikipedia

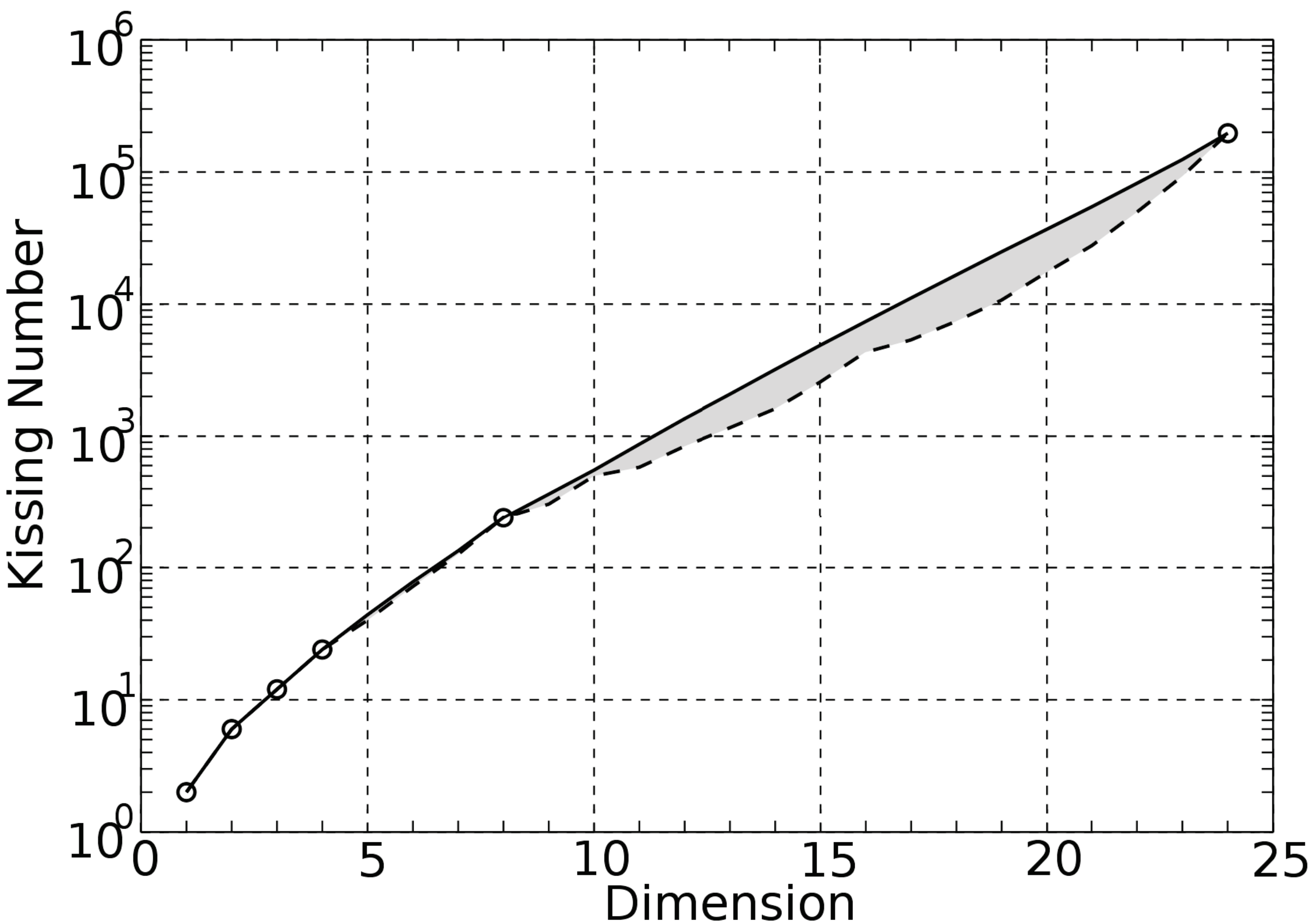
n	lower	old bd	year	our bd
5	40	45	2008	44
7	126	135	2008	134
9	306	366	2008	364
10	500	567	2008	554
11	582	915	1979	870
12	840	1416	1979	1357
13	1130	2233	1979	2069
14	1582	3492	1979	3183
15	2564	5431	1979	4866

Our improved bounds as recorded in Wikipedia

n	lower	old bd	year	our bd
16	4320	8312	2007	7355
17	5346	12210	2007	11072
18	7398	17877	1979	16572
19	10668	25900	1994	24812
20	17400	37974	1979	36764
21	27720	56851	1994	54584
22	49896	86537	1979	82340
23	93150	128095	1994	124416

From Wikipedia

Lower and our upper bounds



Outline

Computing lower bounds for QAPs

- The basic method and QAPLIB results
- Make method more efficient
- Solving a real-life problem from communications

Computing upper bounds for kissing numbers and binary codes

- High precision SDP bounds for kissing numbers
- High precision SDP bounds for binary codes

High precision SDP bounds for binary codes

D. C. Gijswijt, H. D. Mittelmann, and A. Schrijver,
Semidefinite code bounds based on quadruple distances,
IEEE Transactions on Information Theory 58(5), 2697-2705 (2012)

- A(n,d) is maximum number of binary words of length n, any two having Hamming distance at least d
- Classical Delsarte bound yields huge SDP which can be reduced to small LP
- In 2005 Schrijver generalized to SDPs of sets of size at most 3
- They can be reduced to small SDPs with block-diagonalization
- New work generalizes to quadruples of words. Reduced SDPs are still large
- They are ill-conditioned and require high precision

n	d	known lower bound	known upper bound	new upper bound	$A_4(n,d)$
18	6	512	680	673	673.005
19	6	1024	1280	1237	1237.939
20	6	2048	2372	2279	2279.758
23	6	8192	13766	13674	13674.962
19	8	128	142	135	135.710
20	8	256	274	256	256.000
25	8	4096	5477	5421	5421.499
26	8	4096	9672	9275	9275.544
27	8	8192	17768	17099	17099.644
21	10	42	48	47	47.007
22	10	64	87	84	84.421
24	10	128	280	268	268.812
25	10	192	503	466	466.809
26	10	384	886	836	836.669
27	10	512	1764	1585	1585.071
28	10	1024	3170	2817	2817.313
25	12	52	56	55	55.595
26	12	64	98	96	96.892

THE END

Thank you for your attention

Questions or Remarks?