Quantum Networks 2017

Viability of semidefinite program solutions in quantum information

Denis Rosset

August 4, 2017

Quantum Foundations, Nonlocality and Information Group
National Cheng Kung University, Taiwan

Fonds national suisse
Schweizerischer Nationalfonds
Fondo nazionale svizzero
Swiss National Science Foundation

Collaboration

Quantum Nonlocality, Foundations & Information group

Yeong-Cherng Liang
Denis Rosset
Pei-Sheng Lin

NCKU, Taiwan

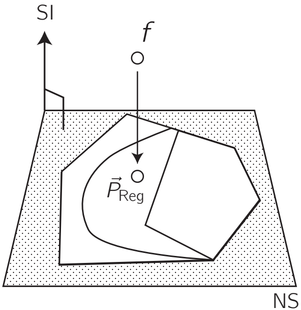
in collaboration with

Jean-Daniel Bancal, Uni. Basel, Switzerland
Yanbao Zhang, NTT, Japan

→

- Taming finite statistics for DI
  arXiv:1705.09245

  (Lin, *Rosset*, Zhang, Bancal, Liang)



References

- Copy of the slides: tinyurl.com/oxfordsdp
- Characterization of correlations in quantum networks

  PhD thesis (2015), Rosset

- Complete family of separability..., PRA 69 022308 (2004)

  Doherty, Parrilo, Spedali
- A convergent hierarchy ..., NJP 10 07013 (2008)

  Navascués, Pironio, Acín

- CVX / YALMIP / VSDP / INTLAB
- MOSEK / SCS / SDPA / SDPNAL+ / SDPT3 / SeDuMi

Summary

Conic programming (LP/SDP) in 40% of the talks in Natal.

Own research goal: ready-to-use tools for experimental tests.

Point estimates of correlations from experimental data (Lin 2017), MDIEWs for entanglement quantification (WIP)

Can we trust the numerical results?

Key messages

- Don't trust the primal-dual gap
- Don't trust blindly middle layers (CVX, YALMIP)
- Start with: MOSEK, CVX + VSDP if you can
- Download the slides for later use

tinyurl.com/oxfordsdp

Warning

Don't trust the primal-dual gap.

- Certifies an interval *if* both solutions feasible
- Primal/dual barrier methods sols. are slightly infeasible.
- Sometimes: weak duality does not hold (ex: SeDuMi).
- If all error measures $\leq \epsilon$, does not mean precision $\leq \epsilon$.
- State of the art: enclose primal/dual feasible solutions to certify bounds.

Problem amplified by...

> *Programs must be written for people to read, and only incidentally for machines to execute.*
>
> Harold Abelson

- Gap between published description and the canonical form used by solvers.
- Complicated by middle layers (YALMIP, CVX).
- Alleviated by good preprocessing (for linear programs).
- Bigger impact on semidefinite programs.

Plan

- Under the hood (canonical form, what do YALMIP/CVX do)
- Bad formulations
- Solver benchmark
- Towards robust semidefinite programming

Under the hood

How do solvers lie?

Linear programming:

$$\begin{array}{llll} \text{minimize} & \vec{f}^\top \cdot \vec{x} & & \\ \text{over} & \vec{x} \in \mathbb{R}^n & & \\ \text{subject to} & A\vec{x} \leqslant \vec{b} & \vec{x} \geqslant \vec{\ell} \\ & A_{\text{eq}}\vec{x} = \vec{b}_{\text{eq}} & \vec{x} \leqslant \vec{u} \end{array}$$

In reality:

$$\begin{array}{llll} & \text{Primal}(=) & & \text{Dual}(\geqslant) \\ \text{minimize} & \vec{c}^\top \cdot \vec{x} & \text{maximize} & \vec{b}^\top \cdot \vec{y} \\ \text{over} & \vec{x} \in \mathbb{R}^n & \text{over} & \vec{y} \in \mathbb{R}^m \\ \text{subject to} & A\vec{x} = \vec{b} & \text{subject to} & \vec{c} - A^\top \vec{y} \geqslant 0 \\ & \vec{x} \geqslant 0 & & \end{array}$$

Conic programming

$$
\begin{array}{rlcrl}
 & \text{Primal}(=) & & & \text{Dual}(\geqslant) \\
\text{minimize} & \vec{c}^\top \cdot \vec{x} & & \text{maximize} & \vec{b}^\top \cdot \vec{y} \\
\text{over} & \vec{x} \in \mathcal{K} = \mathbb{R}^n_+ & & \text{over} & \vec{y} \in \mathbb{R}^m \\
\text{subject to} & A\vec{x} = \vec{b} & & \text{subject to} & \vec{c} - A^\top \vec{y} \in \mathcal{K}^* = \mathbb{R}^n_+
\end{array}
$$

$\mathcal{K}$ is a convex cone: $x, y \in \mathcal{K}$, $\alpha \in \mathbb{R}^+ \Rightarrow \quad \alpha(x + y) \in \mathcal{K}$

Standard cones

	Cone	Dual cone
Nonnegative	$\mathcal{K}_\mathsf{L} = \mathbb{R}^n_+$	$\mathcal{K}^*_\mathsf{L} = \mathcal{K}_\mathsf{L} = \mathbb{R}^n_+$
Second-order	$\mathcal{K}_\mathsf{Q} = \{(t, \vec{x}) \text{ s.t. } \|\vec{x}\|_2 \leqslant t\}$	$\mathcal{K}^*_\mathsf{Q} = \mathcal{K}_\mathsf{Q}$
SDP (real)	$\mathcal{K}_\mathsf{S} = \left\{\vec{x} \in \mathbb{R}^{(n+1)n/2} \text{ s.t. } \underset{\rightarrow}{\mathrm{SymMat}(\vec{x})} \geqslant 0\right\}$	
Free	$\mathcal{K}_\mathsf{F} = \mathbb{R}^n$	$\mathcal{K}^*_\mathsf{F} = \{0\}$

Canonical form cone $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times \ldots$

```
model =
    At: [5568x304 double]
     b: [304x1 double]
     c: [5568x1 double]
     K: [1x1 struct]
```

```
model.K =
    f: 60
    l: 0
    q: []
    r: []
    s: [54 36 36]
```

Some problems map directly

Inflation: maps to primal (=)
$(\gamma_1 \to A_1 \leftarrow \beta_1 \to C_1 \leftarrow \alpha_1 \to B_1 \leftarrow \gamma_2)$

$$
\begin{aligned}
P'_{A_1C_1}(ac) &= P(ac) \\
P'_{B_1C_1}(bc) &= P(bc) \\
P'_{A_1B_1}(ab) &= P(a)P(b) \\
P'_{A_1B_1C_1}(abc) &\geqslant 0
\end{aligned}
$$

NPA hierarchy: maps to dual ($\geqslant$)

$$
\sum_{abxy} P(ab|xy)\vec{f}_{abxy} + \sum_i y_i \vec{g}_i \quad \in \quad \mathcal{K}_{\mathrm{S}(\mathbb{R})}
$$

Toolbox

Most people use a toolbox (YALMIP, CVX). Transformations:

- Complex to real:

$$X \geqslant 0 \qquad \Leftrightarrow \qquad \begin{pmatrix} \operatorname{Re} X & -\operatorname{Im} X \\ \operatorname{Im} X & \operatorname{Re} X \end{pmatrix} \geqslant 0$$

- In the $(=)$ form, if you have $x \in \mathbb{R}$:

$$x_+, x_- \in \mathbb{R}_+, \qquad x = x_+ - x_-$$

- In the $(\geqslant)$ form, if you have $\vec{a}^\top \cdot \vec{x} = b$:

$$\vec{a}^\top \cdot \vec{x} \leqslant b, \qquad \vec{a}^\top \cdot \vec{x} \geqslant b$$

- Limited variable elimination (CVX), don't hold your breath

YALMIP: native form $(\geqslant)$. Switch to $(=)$ with **dualize** .
CVX: decides $(=)$ or $(\geqslant)$ (most often). Force with **cvx_dualize** .

Bad formulations

Avoid redundant constraints (I)

$$\begin{array}{ccccc} x & + & y & = & b \\ x & + & y & = & b+\varepsilon \end{array}$$

(How large should ε be such that the solver raises infeasibility?)
Example: inflation

$$\sum_{ac} P'_{A_1C_1}(ac) = \sum_{ac} P(ac) = 1$$

$$\sum_{bc} P'_{B_1C_1}(bc) = \sum_{bc} P(bc) = 1$$

$$\sum_{ab} P'_{A_1B_1}(ab) = \sum_{ab} P(a)P(b) = 1$$

Avoid redundant constraints (II)

NPA hierarchy

$$\sum_{abxy} P(ab|xy)\vec{f}_{abxy} + \sum_i y_i \vec{g}_i \quad \in \mathcal{K}_{\mathrm{S}(\mathbb{R})}$$

$$P(ab|xy) \geqslant 0, \qquad \forall abxy$$

Do not add constraints to force P to be nonnegative.
Postprocess the result (e.g. add white noise).

Symmetric extensions

$$\rho_{\mathrm{AB}'} = \mathrm{tr}_{\mathrm{B}'}[\tau_{\mathrm{ABB}'}], \qquad \tau_{\mathrm{ABB}'} \geqslant 0$$

we have $\rho_{\mathrm{AB}'} \geqslant 0$ for free.
Tricky: the constraint $\rho_{\mathrm{AB}'} \geqslant 0$ is useless in the ($\geqslant$) form — but useful in the ($=$) form.

Never solve feasibility problems

Do not write:

$$\max_{\vec{P}'} 0 \qquad M \cdot \vec{P}' = \vec{q}, \qquad \vec{P}' \geqslant 0$$

Infeasibility = yes/no, different solvers = different thresholds.

Write instead:

$$\max_{\vec{P}', t} t \qquad M \cdot \vec{P}' = \vec{q}, \qquad \vec{P}' \geqslant t.$$

Value t can be compared across different solvers.

- Try to use a physical figure of merit for the slack t (ex: visibility under addition of white noise).
- Use reports of infeasibility as sign of serious numerical issues.
- Always check the error code of the solver, not of the toolbox.

Use a good basis for objects

Many objects obey linear relations. Example:

$$\sum_b P(ab|xy) - P(ab|xy') = 0 \qquad \forall axyy'$$

Goal: find a basis (ideally sparse, simple coefficients) that satisfies the constraint.

Tricks

Trick #1

- Find the symmetry group that preserves the constraints.
- Find the basis that splits the group representation into irreps.

Example: qubit-qubit density matrix ρ under $\mathcal{U}(2)$

$$\rho = \nu\mathbb{1} + (\vec{\alpha}\cdot\vec{\sigma})\otimes\mathbb{1} + \mathbb{1}\otimes(\vec{\beta}\cdot\vec{\sigma}) + \sum_{ij} t_{ij}(\sigma_i\otimes\sigma_j)$$

Trick #2

In marginal problems, use a “Russian doll” basis like Collins-Gisin. (Solves the problem of redundant constraints in inflation.)

Have the solver formulation in mind

Example: elements of a POVM

$$A_1, A_2, A_3 \geqslant 0, \qquad A_1 + A_2 + A_3 = \mathbb{1}$$

In the (=) form, three semidefinite variables, OK.
In the ($\neq$) form, better to write:

$$A_1, A_2 \geqslant 0, \qquad A_3 = \mathbb{1} - A_1 - A_2 \geqslant 0,$$

where A_3 is not a problem variable but an expression of variables.

In CVX: use `expression`, not `variable`
In YALMIP: assign to new variable, or overwrite the contents of a previously declared `sdpvar`.

Free variables / equality constraints

Solver authors don't like them (numerically unstable).
Automatic variable elimination (YALMIP's `removeequalities`)?

- Gaussian elimination not stable.
- QR decomposition kills sparsity and introduces nasty floating-point approximations.

CVX eliminates *simple* equality constraints. YALMIP does not.
Some solvers have special support for equality constraints (SeDuMi, SDPT3, etc...) as they are common.

Solver benchmark

Sample problem

(Doherty 2004)

Entangled PPT state due to Horodecki:

$\boxed{\text{Separable for } \alpha \in [2,3].}$

$$\rho_{AB} = \frac{1}{7}[2|\psi_+\rangle\langle\psi_+| + \alpha\sigma_+ + (5-\alpha)V\cdot\sigma_+\cdot V]$$

$$|\psi_+\rangle = \frac{1}{\sqrt{3}}\sum_{i=0}^{2}|ii\rangle, \qquad \sigma_+ = \frac{1}{3}[|01\rangle\langle 01| + |12\rangle\langle 12| + |20\rangle\langle 20|]$$

Our test problem: find $\min\alpha$ such that $\rho_{AB} \in \mathbf{Sep}(\mathbb{C}^3 : \mathbb{C}^3)$.

Doherty hierarchy: symmetric extensions

Doherty (2004)

SDP constraints defining a cone $\mathcal{K} \supseteq \mathsf{Sep}(\mathbb{C}^3 : \mathbb{C}^3)$.

For a number of copies $n = 2$:

$$\rho_{AB} = \mathrm{tr}_{B'}[\tau_{ABB'}], \qquad \tau_{ABB'} \geqslant 0$$

$$\tau_{ABB'} = \Pi \tau_{ABB'} \Pi^*, \qquad \Phi^k_{PT}(\tau_{ABB'}) \geqslant 0$$

permutation partial transpose

Option: $\tau_{ABB'} = \Pi \tau_{ABB'}$ (Bose sym).

Three formulations:

- As prescribed in Doherty 2004, without symmetry (DNS)
- As prescribed in Doherty 2004, with Bose symmetry (DS)
- Formulation (≈QETLAB), with full Bose symmetry (QVX)

Formulations, CVX vs YALMIP

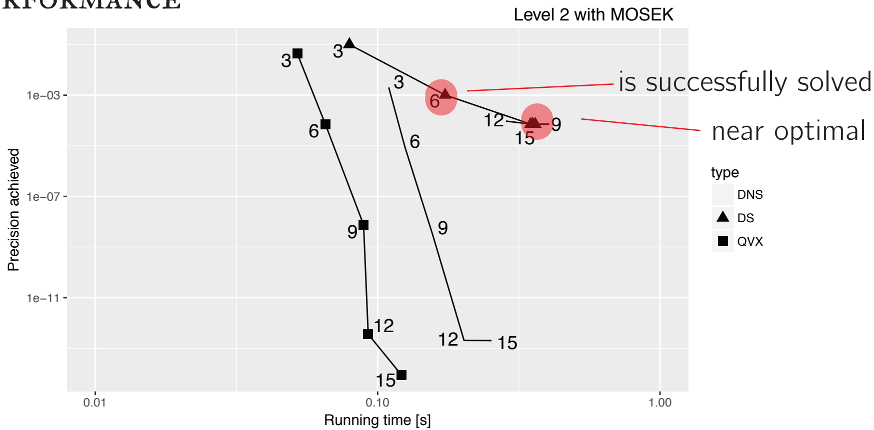
Level 2

	DNS	DS	QVX	QVX
			YALMIP	CVX
$\mathcal{K}_S$	54,54,54	54,36,36	54,36,36	54,36,36
$\mathcal{K}_F$	0	0	1188	60
n	8748	5508	6696	5568
m	325	325	1378	304

Level 3

	DNS	DS	QVX	QVX
			YALMIP	CVX
$\mathcal{K}_S$	162,162,162,162	108,108,60,60	108,108,60,60	108,108,60,60
$\mathcal{K}_F$	0	0	6960	81
n	104976	30528	37488	30609
m	1405	1405	7633	901

Performance



Solution status

- optimal (problem solved within tolerances)
- near-optimal / solver stalls (cannot make progress)
- serious problems

Interpreted then by the toolbox (a *lot* is lost in translation)

Here: we investigate solution quality, not the info returned

Solvers

	SDPA	SDPT3	SeDuMi	Mosek	SCS	SDPNAL+
License	GPL	GPL	GPL	Prop.	MIT	GPL?
Language	C/C++	Matlab/C	Matlab/C		C	Matlab/C
Version	7.3.8	4.0 †	1.32 †	8.0 †	1.26	0.5
Algo	pdbm	pdbm	pdbm	pdbm	admm	aug. $\mathcal{L}$
Complex SDP	no	yes*	depends*	no	no	no
Maximum m	10^3-10^4	10^3	10^3	$\geqslant 10^3$	10^6	10^6

† Version bundled with CVX

* but CVX does not use the feature

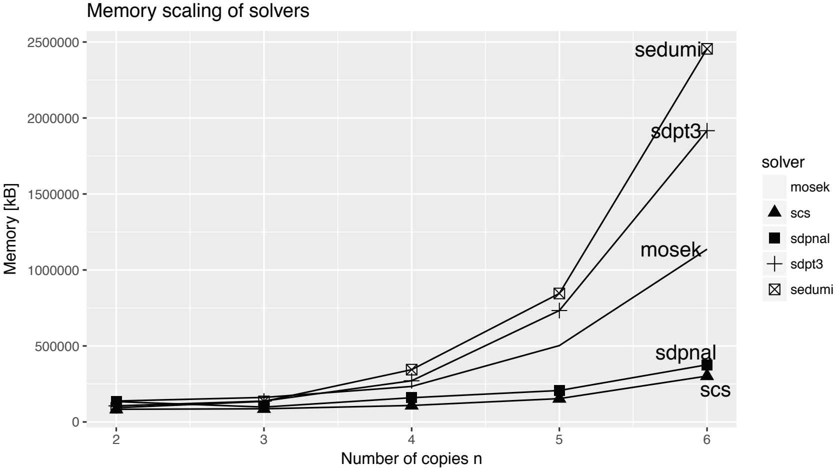
Benchmark details

Problem formulated with CVX, canonical form given to YALMIP.
Default settings of solver, large iter. limit (beware of YALMIP defaults, e.g. SeDuMi).
Precision options as done by CVX.
Computer: Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz, 32 Gb RAM
Time: wall time

Memory: max of RSS(MATLAB + YALMIP + solver) - RSS(MATLAB + YALMIP)

Memory usage

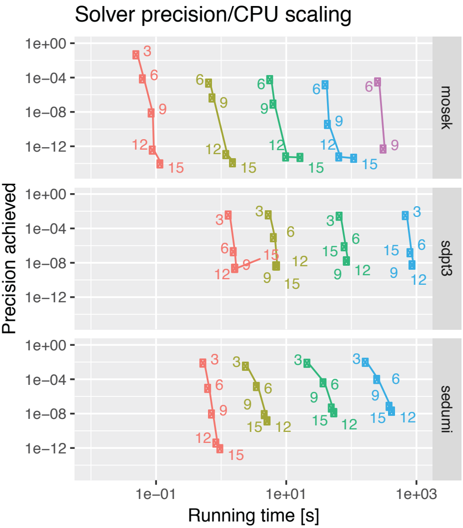
Hard limit: Desktop: 32 - 64 Gb. Amazon EC2: 512 - 2048 Gb

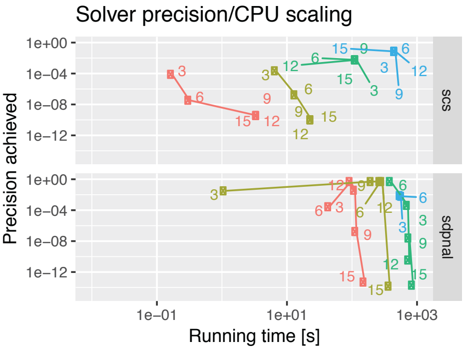


Primal-dual barrier methods: $\mathcal{O}(\overbrace{m^2}^{\text{S.C.}} + \overbrace{mn^2}^{\text{Data upper bound}})$

(except when Schur complement matrix is sparse)

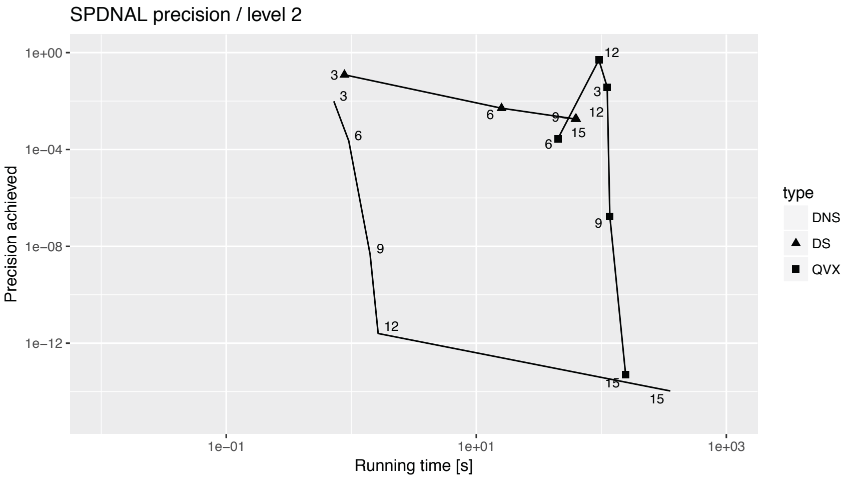
CPU TIME





$$\text{Iteration p.d.b.m. :} \quad \overbrace{\mathcal{O}(mn^3 + m^2n^2)}^{\text{Computation S.C.}} + \underbrace{\mathcal{O}(m^3)}_{\text{Cholesky S.C.}} + \overbrace{O(n^3)}^{\text{Various}}$$

SDPNAL+



SPDNAL precision / level 2
Precision achieved
Running time [s]
1e+00
1e−04
1e−08
1e−12
1e−01
1e+01
1e+03
type
DNS
DS
QVX
3
6
9
12
15

Future work

- Add SDPA (flexible family)
- Add PENLAB (support for general cones)
- Extend problem library

  (exact solution, integer coefficients, SDPA or SeDuMi format)
- Understand best parameters for SCS/SDPNAL+

Verified semidefinite programming

INTLAB and VSDP

- INTLAB: interval arithmetic for MATLAB (EUR 90)
- VSDP (2012): add-on for verified semidefinite programming
- Solves a series of perturbed problems using a regular solver
- Encloses feasible, near-optimal solutions for the primal&dual problems
- Each perturbed problem has same memory/iteration cost profile as original problem
- **Need robust problem data** (integer coefficients or enclosures)

On the separability problem

Test of state separable for $\alpha \in [2, 3]$:

$$\rho_{\mathrm{AB}} = \frac{1}{7}[2|\psi_+\rangle\langle\psi_+| + \alpha\sigma_+ + (5-\alpha)V \cdot \sigma_+ \cdot V]$$

Results for level $n = 2$

	Lower bound	Upper bound
Original, no symmetry	$2 - 2.6 \cdot 10^{-8}$	$+\infty$
Original, with symmetry	$2 - 4.1 \cdot 10^{-5}$	$+\infty$
QVX	$2 - 1.7 \cdot 10^{-7}$	$+\infty$

(using SeDuMi)

How-to

(missing: installation/initialization of VSDP, INTLAB, CVX)

Yalmip

```
model = export(cons, obj, sdpsettings('solver', 'sedumi'));
A = model.A; b = model.b; c = model.C; K = model.K;
```

CVX

```
cvx_solver sedumi % or use our diagout pseudo-solver
cvx_solver_settings('dumpfile', 'model.mat');
cvx_begin; ...; cvx_end
model = load(model.mat); A = model.At; b = model.b; c = model.c; K = model.K;
```

VSDP

```
[objt,xt,yt,zt,info] = mysdps(A,b,c,K);
% VSDP starts here
[fL,y,dl] = vsdplow(A,b,c,K,xt,yt,zt);
[fU,x,lb] = vsdpup(A,b,c,K,xt,yt,zt);
% [fL fU] is a certified bound
```

I_{3322}

Tsirelson bound, I_{3322} inequality, SDPA 7.3.1 Rosset 2015

Level	Symmetries	m	$K.s$	Time (s)	Memory (MB)
3	no	867	88	73.2	15
3	partial diag.	124	44,44	4.3	3
3	diag.	124	22,22,13,11,11,9	1.2	2

$$I_3^{\text{without-symmetrization}} \in [\mathbf{1.2508755}5, \mathbf{1.2508755}7]$$

$$I_3^{\text{with-symmetrization}} \in [\mathbf{1.25087556}1, \mathbf{1.25087556}8]$$

Values for higher levels: need VSDP + higher precision.

$$I_3 \cong 1.2508755620230350 \text{ (gap} \sim 10^{-31}),$$

$$I_4 \cong 1.2508753845139768 \text{ (gap} \sim 10^{-30}),$$

$$I_5 \cong 1.2508753845139766 \text{ (gap} \sim 10^{-21}).$$

Future ideas

TODO

- Interface VSDP with additional solvers
- Interface CVX with VSDP

Longer term

- Related work: Jean-Daniel Bancal, Refiner, YALMIP
- Better diagnostics; ex. behavioral measures (Freund 2007)
- Combine symbolic+numerical methods (better preprocessing)

Conclusion

Conclusion

tinyurl.com/oxfordsdp

- To start: use CVX + MOSEK
- Integer coefficients / simple problem structure? Try VSDP
- Ideally: open science, open source
- Keep intermediate data files, including solver input and output

- Join Github project github.com/denisrosset/quantumsdp
- Provide problem instances with exact solutions