

Computer programs for linear SDP

Michal Kočvara

Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic

and

Czech Technical University

kocvara@utia.cas.cz

<http://www.utia.cas.cz/kocvara>

Available codes

Primal-dual IP codes (available):

CSDP-4.9 <http://www.nmt.edu/~borchers/csdp.html>

very good overall

SDPA-6.00 <http://grid.r.dendai.ac.jp/sdpa/>

SDPT3-3.1 <http://www.math.nus.edu.sg/~mattohkc/>

very good overall

SeDuMi-1.05 <http://fewcal.kub.nl/sturm/software/sedumi.html>

very robust, slow on large-scale problems

Dual IP code:

DSDP-5.6 <http://www-unix.mcs.anl.gov/~benson/>

very good new version

Available codes

Augmented Lagrangian code:

PENSDP-2.0 <http://www.penopt.com/>

First order code:

SDPLR-1.01 <http://dollar.biz.uiowa.edu/~burer/software/SDPLR/>
very good for certain large-scale problems; may be very slow,
limited accuracy

Other codes:

SBmethod-1.1.2

<http://www-user.tu-chemnitz.de/~helmberg/SBmethod/>
very efficient; limited applicability

Few benchmarks (SDPLIB)

problem	CSDP	DSDP	SDPA	SDPT3	SeDuMi	PENNON	SBmeth	SDPLR
control7	33	18	40	91	42	62		>3000
control11	242	131	316	151	389	740		>3000
equalG11	80	60	63	84	1042	66		13!
equalG51	207	112	127	157	1847	235		64!
gpp500-4	34	10	18	27	152	16		8!
maxG32	391	119	1378	428	9405	185	45	40
maxG51	105	34	83	70	1337	127	8	19
mcp500-1	11	4	13	11	162	5		1
qap9	2	8	2	3	13	7		175!
qap10	4	15	5	5	32	11		385!
qpG11	314	50	332	214	4562	55		17
qpG51	459	204	693	428	12113	218		4
theta3	4	6	5	6	27	11		49!
theta5	57	73	68	51	474	165		358!
theta6	152	147	180	136	1440	499		554!
thetaG51	1614	2553	1455	1681	8087	5183	13975	3890
truss8	3	9	17	8	5	10		3205

Few benchmarks (SDPLIB + dimensions)

problem	dimensions		PENSDP		PEN-PCG(BFGS)			PEN-A-PCG(BFGS)		
	n	m	CPU	CPU/it	CPU	CPU/it	CG/it	CPU	CPU/it	C
control7	666	105	63	0.79	120	1.71	162	164	2.31	
control11	1596	165	937	5.42	1421	12.69	502	1870	19.68	
equalG51	1001	1001	296	3.65	295	6.56	3	438	9.73	
gpp500-4	501	500	26	0.67	32	1.10	5	51	1.89	
maxG32	2000	2000	108	4.70	185	5.97	8	476	14.00	
maxG51	1000	1000	115	2.45	204	3.82	4	343	6.86	
mcp500-1	500	500	5	0.16	7	0.18	5	11	0.30	
qap9	748	82	3	0.09	4	0.07	7	4	0.06	
qap10	1021	101	8	0.22	11	0.16	7	8	0.10	
qpG51	1000	2000	172	5.93	241	7.53	3	400	11.76	
ss30	132	426	10	0.28	13	0.32	4	7	0.18	
theta3	1106	150	9	0.26	7	0.23	4	4	0.08	
theta5	3028	250	93	3.32	43	2.29	5	16	0.39	
theta6	4375	300	366	9.15	139	4.80	5	41	0.67	
truss8	496	628	6	0.13	55	0.70	157	11	0.22	

Few benchmarks (vibra)

```
=====
problem  PENSDP  SeDuMi  SDPT3  CSDP  DSDP  SDPA  SDPLR
=====
buck-3      30    1414     30    89    32    24    443
buck-4     174   14544    130   536   223   248   3411
buck-5    2445      m   1350a  5005  3964  4960  57018
mater-3      5     17     26    12    16   1044  >26000
mater-4     23     50    129   290   102  138452
mater-5     60    137    291  3647   412      m
mater-6    178    281    919      m  1765      m
shmup3     301   10280   247  1498   232   419   209
shmup4    1655  109670  1148  3774  1643  1988  1866
shmup5   10994      m   8740  52295  16812      m  61794
trto-3     14   1171    15    19     9    12   1872
trto-4     83  11260    64   238    86   125  13168
trto-5    1149      m   694  2671  1194  1963
vibra-3     27   1641    32    69    36    28   1091
vibra-4    148  16199   150   774   303   269  11206
vibra-5    2107      m   2269  5984  4396  4740 >153000
=====
```

Few benchmarks (vibra dimensions)

problem	variables	constraints	nonzeros	blocks
buck-3	544	1,186	7,831	4
buck-4	1,200	2,546	17,509	4
buck-5	3,280	6,802	48,385	4
mater-3	1,439	3,588	45,189	329
mater-4	4,807	12,498	157,779	1139
mater-5	10,143	26,820	338,757	2441
mater-6	20,463	56,311	690,149	4969
shmup3	420	2,642	21,600	4
shmup4	800	4,962	36,477	4
shmup5	1,800	11,042	82,317	4
trto-3	544	866	3,902	3
trto-4	1,200	1,874	8,734	3
trto-5	3,280	5,042	24,158	3
vibra-3	544	1,186	7,831	4
vibra-4	1,200	2,546	17,509	4
vibra-5	3,280	6,802	48,385	4

SDP input

MATLAB most popular; most codes have their own MATLAB interfaces;
gathered in YALMIP

`http://control.ee.ethz.ch/~joloef/yalmip.msql`

ASCII file in SDPA sparse format; good for testing

C/Fortran available by few codes; best for heavy-duty applications

YALMIP example

Lyapunov stability:

$$P \succeq 0$$
$$A^T P + P A \preceq 0$$

```
A = [-1 2; -3 -4];  
P = sdpvar(2,2);
```

Looking for feasible solutions only

```
F = set(P > 0) + set(A'*P+P*A < 0);  
F = F + set(trace(P) == 1); %to avoid zero/unbounded solutions  
solvesdp(F);
```

Minimizing top-left element of P

```
F = set(P > 0) + set(A'*P+P*A < 0);  
solvesdp(F, P(1,1), sdpsettings('solver', 'sedumi'));
```

SDPA example

```
min 10 x1 + 20 x2
st  X = F1 x1 + F2 x2 - F0
    X >= 0
```

```
F1=[1 0 0 0   F2=[1 0 0 0   F3=[0 0 0 0
    0 2 0 0     0 1 0 0     0 1 0 0
    0 0 3 0     0 0 0 0     0 0 5 2
    0 0 0 4]   0 0 0 0]   0 0 2 6]
```

A sample problem.

```
2 =mdim           ...cont...
2 =nblocks       1 1 1 1 1.0
{2, 2}          1 1 2 2 1.0
10.0 20.0 2     1 2 2 1.0
0 1 1 1 1.0     2 2 1 1 5.0
0 1 2 2 2.0     2 2 1 2 2.0
0 2 1 1 3.0     2 2 2 2 6.0
0 2 2 2 4.0
...cont...
```

Useful WWW sites

<http://www-neos.mcs.anl.gov/neos/>
Server for optimization. All codes mentioned here are implemented.
SDPA and Matlab binary input.

<http://plato.la.asu.edu/bench.html>
Benchmarks of latest versions of all available codes.

YALMIP to SDPA

1. create a YALMIP model
2. solve the problems with

```
solvesdp(F,x(1),...  
    sdpsettings('solver','penbmi','savedebug',1));
```

this creates file `penbmimdebug.mat` in the working directory

3. `load penbmimdebug.mat`; you get a structure `pen`
4. run `pen2sdpa(pen,'problem.dat')`; this creates an SDPA file `problem.dat` in the working directory