11 Oct 2002 ==

Mixed Integer Linear Programming Benchmark (commercial codes)

==========

H. Mittelmann (mittelmann@asu.edu)

The URL of this page is ftp://plato.la.asu.edu/pub/milpc.txt
Previous version at: ftp://plato.la.asu.edu/pub/milpc.txt.old
Logfiles for these runs at: ftp://plato.la.asu.edu/pub/milp_logs

The MPS-datafiles for all testcases are in one of (see column "src")
- http://www.caam.rice.edu/~bixby/miplib/miplib.html [1]
- ftp://plato.la.asu.edu/pub/fctp/ [2]
- ftp://ftp.mcs.anl.gov/neos/mip-bench/ [3]
- ftp://plato.la.asu.edu/pub/milp/ [4]
- http://www.ps.uni-sb.de/~walser/acc/acc.html [5]

The following codes were run in default mode except for rel/abs "mipgap" set to .05/1e-6 (1) resp. 1e-5/1e-6 (2). P4 (2.5 GHz, 2GB RDRAM, Linux-2.4.19)

- CPLEX-8.0: http://www.cplex.com/
- XPRESS-MP-13.26: http://www.dashoptimization.com/
- MOSEK-2.5.1: http://www.mosek.com (8/14/02)
- GLPK-3.2.1: http://www.gnu.org/software/glpk/glpk.html

Times given are user times in seconds. No paging due to sufficient memory.
For comparison results for a free code with default options are included.

src	problem	CPLX-1	XPRS-1	CPLX-2	XPRS-2	MOSK-1	MOSK-2	GLPK
1	10teams	8	11	7	>10000	237	>10000	
	air04	16	18	58	98	34	5679	
	air05	15	7	47	481	681	>10000	
	bell5	1	1	1	>10000	1	25	5
	cap6000	1	52	30	63	3	9	
	l152lav	1	1	2	26	2	84	39
	misc07	99	45	112	47	1188	1203	365
	mod011	405	43	1749	284	1716	5478	
	nw04	5	45	17	154	31	70	929
	pk1	180	755	197	635	1486	1540	
	qiu	185	289	209	304	>10000		
	stein45	16	13	20	23	111	94	196
2	ran10x26	6	54	62	6218	230	990	
	ran12x21	38	55	336	2132	117	1205	
	ran13x13	19	70	121	353	124	551	
	ran14x18_1	1803	>12000			>10000		
3	binkar10_1	1	1	53	860	2	>10000	
	eilD76	222	28	281	32	2003	2008	3939
	irp	1	7	6	12	3	21	240
	mas284	1	1	23	38	5	206	
	prod1	93	>12000	192	>12000	1389	>10000	
4	bc1	464	2501	534	2655	4661	4780	
	bienst1	741	817	872	889	>10000		997
	bienst2	4062	8080	4464	8413	>10000		16716
	dano3_3	261	62	484	75	1378	2255	173
	dano3_4	258	73	376	132	1672	2267	212
	dano3_5	271	85	1292	2023	1400	>10000	

markshare1_1	276	5421	276	4511	1	126	
markshare2_1	234	>20000	234	>10000	1355	>10000	
mkc1	6	1	526	>10000	48	>10000	
neos1	17895	>15000	>30000	>10000	>10000		
neos2	193	18984	209	>10000	2550	2852	
neos3	3838	>15000	3859	>10000	>10000		
neos4	3	233	23	263	130	149	fail
neos5	3	35	19	89	126	132	197
neos6	40	35	139	6099	103	>10000	
neos7	3	2	994	>10000	37	>10000	fail
neos8	145	222	189	222	42	45	13733
neos9	48	219	>30000	>10000	>10000		
neos10	576	262	609	321	284	360	12760
neos11	120	581	121		>10000		5356
neos12	1105	>10000	1105		>10000		
neos13	923	>60000	926		>75000		
neos14	53	>10000	1193				
neos15	13536	>60000					
nug08	10	7	52	7	74	7361	18
qap10	139	87	423	87	2893	4666	743
seymour1	6	14	1232	>10000	13	>10000	
swath2	631	1&	809	1024&	4032	5371	3338
swath3	3790	1&			>10000		>10000
5 acc0	1	111	1	112	23	23	26
acc1	5	232	5	234	173	173	332
acc2	25	237	25	237	469	474	880
acc3	99	649	99	649	7128	7225	
acc4	765	453	762	454	>10000		
acc5	2168	65	2174	65	>10000		
acc6	189	>7700	189	>10000	>10000		

&: function value very large

Problem Statistics

problem	rows	col	integer	nonzero
10teams	231	2025	1600	14175
air04	824	8904	8904	81869
air05	427	7195	7195	59316
bell3a	124	133	71	347
bell5	92	104	58	266
cap6000	2177	6000	6000	54238
l152lav	98	1989	1989	11911
misc07	213	260	259	8620
mod011	4482	10958	96	37425
nw04	37	87482	87482	636666
pk1	46	86	55	915
qiu	1193	840	48	3744
stein45	332	45	45	1079
ran10x26	297	520	259	1560
ran12x21	286	504	251	1512
ran13x13	196	338	168	1014
ran14x18_1	285	504	252	1512
binkar10_1	1027	2298	170	6238
eilD76	76	1898	1898	21009
irp	40	20315	20315	118569

mas284	69	151	150	9782
prod1	209	250	149	5351
bc1	1914	1751	252	227073
bienst1	577	505	28	2185
bienst2	577	505	35	2072
dano3_3	3203	13873	423	79656
dano3_4	3203	13873	569	79656
dano3_5	3203	13873	708	79656
markshare1_1	7	62	45	324
markshare2_1	8	74	54	448
mkc1	3412	5325	3082	19984
neos1	5021	2112	2112	21600
neos2	1104	2101	1040	7330
neos3	1443	2747	1360	9585
neos4	38578	22884	17136	116040
neos5	36703	21124	17136	109068
neos6	1037	8768	8409	252169
neos7	1995	1556	454	5472
neos8	46325	23228	23228	313212
neos9	31601	81408	2099	264192
neos10	46794	23489	23489	251230
neos11	2702	1220	900	9540
neos12	8318	3983	3136	26229
neos13	20853	1827	1815	253854
neos14	553	792	136	2318
neos15	553	792	160	2318
nug08	913	1632	1632	8304
qap10	1821	4150	4150	20810
seymour1	4945	1372	372	34921
swath2	885	6805	2213	34966
swath3	885	6805	2706	34966
acc0	1738	1620	1620	7291
acc1	2287	1620	1620	12979
acc2	2521	1620	1620	15328
acc3	3250	1620	1620	24913
acc4	3286	1620	1620	17074
acc5	3053	1339	1335	16135
acc6	3048	1335	1335	16109

18 Oct 2002

Mixed Integer Quadratic Programming Benchmark

H. Mittelmann (mittelmann@asu.edu)

The URL of this page is ftp://plato.la.asu.edu/pub/miqp.txt
Logiles at ftp://plato.la.asu.edu/pub/miqp_logs

The MPS-datafiles for all testcases are in
ftp://plato.la.asu.edu/pub/miqp/

The following codes were run in default mode on a Pentium 4 (2.5 GHz,
2GB RDRAM, Linux-2.4.19). For accuracy reached, see logfiles.

CPLEX-8.0: http://www.cplex.com/
XPRESS-MP-13.26: http://www.dashoptimization.com/
MOSEK-2.5 http://www.mosek.com (to be completed)

Times given are user times in seconds. An * indicates some failure.

problem	CPLEX	XPRESS	MOSEK
iair04	1542	>10000	
iair05	346	>10000	
ibc1	423	*	
ibell3a	11	320	
ibienst1	1371	*	
icap6000	15	1874	
ieilD76	883	1905	
imas284	37	*	
imisc07	333	3456	
imod011	2174	*	
ineos4	3808	>10000	
ineos5	3703	>10000	
inug08	106	234	
iportfolio	1467	>6000	>6000
iqap10	3180	1776	
iqiu	353	729	
iran13x13	186	1805	
iran8x32	36	1801	
iswath2	346	293	
itointqor	435	*	>6000
ivalues	*	153	387

Problem Statistics

problem	rows	col	integer	nonzero	quad elts
iair04	823	8904	8904	81869	17807
iair05	426	7195	7195	59316	14389
ibc1	1913	1751	252	276843	503
ibell3a	104	122	60	390	119
ibienst1	576	505	28	2185	55
icap6000	2171	6000	6000	24238	11999
ieilD76	75	1898	1898	21009	3795
imas284	68	151	150	9782	299
imisc07	212	260	259	8620	517

imod011	4480	10957	97	29838	191
ineos4	36702	21124	17136	109073	34271
ineos5	36702	21124	17136	109068	34271
inug08	912	1632	1632	8304	3263
iportfolio	202	1200	967	202200	200
iqap10	1820	4150	4150	20810	8299
iqiu	1192	840	48	3744	95
iran13x13	195	338	169	1014	337
iran8x32	296	512	256	1536	511
iswath2	483	6404	2213	34085	4425
itointqor	1	50	50	48	280
ivalues	2	202	202	404	7442

Math. Program., Ser. B (2002)

Digital Object Identifier (DOI) 10.1007/s10107-002-0355-5

H. D. Mittelmann

An independent benchmarking of SDP and SOCP solvers

Received: March 27, 2001 / Accepted: April 5, 2002
Published online: ■ – © Springer-Verlag 2002

Abstract. This work reports the results of evaluating all computer codes submitted to the Seventh DIMACS Implementation Challenge on Semidefinite and Related Optimization Problems. The codes were run on a standard platform and on all the benchmark problems provided by the organizers of the challenge. A total of ten codes were tested on fifty problems in twelve categories. For each code the most important information is summarized. Together with the tabulated and commented benchmarking results this provides an overview of the state of the art in this field.

Key words. semidefinite programming – second order cone programming – optimization software – performance evaluation

1. Introduction

1.1. The problems solved

The primal and dual pair of conic optimization problems over a self-dual cone are defined as

$$\begin{array}{lll} & \min \ \langle c, x\rangle & \max \quad b^T y \\ (P) & \text{s.t.} \ \ x \in K & s.t. \quad z \in K \qquad\qquad (D) \\ & \quad \mathcal{A}x = b & \quad \mathcal{A}^* y + z = c \end{array}$$

where

- K is a closed, convex cone in a euclidean space X.
- $\mathcal{A} : X \to \mathbb{R}^m$ is a linear operator, and $\mathcal{A}^*$ is its adjoint.
- $b \in \mathbb{R}^m$, and $c \in X$.

In the case of a semidefinite-quadratic-linear program these are defined as follows:

- **The space X:** $x \in X \Leftrightarrow x = (x_1^s, \ldots, x_{n_s}^s, x_1^q, \ldots, x_{n_q}^q, x^\ell)$, where
  - $x_1^s, \ldots, x_{n_s}^s$ are symmetric matrices (possibly of various sizes).
  - $x_1^q, \ldots, x_{n_q}^q$ are vectors (again, possibly of various sizes).
  - x^ℓ is a vector.

H.D. Mittelmann: Department of Mathematics and Statistics, Arizona State University, Box 871804, Tempe, AZ 85287-1804, e-mail: mittelmann@asu.edu

This work is supported in part by grant NSF-CISE-9981984

2002/10/25 ==========

Newer SDP/SOCP-codes on the 7th DIMACS Challenge problems

==========

Hans D. Mittelmann (mittelmann@asu.edu)

The URL of this page is ftp://plato.la.asu.edu/pub/dimacs_sdp.txt

CSDP-4.1 (SDPA):	www.nmt.edu/~borchers/csdp.html
PENNON-2.01 (SDPA):	www2.am.uni-erlangen.de/~kocvara/pennon/
SDPT3-3.01 (SeDuMi):	www.math.nus.edu.sg/~mattohkc/
SeDuMi-1.05 (SeDuMi):	fewcal.kub.nl/sturm/software/sedumi.html
DSDP-4.5 (SDPA):	www-unix.mcs.anl.gov/~benson/
SDPA-6.0 (SDPA):	www.is.titech.ac.jp/~yamashi9/sdpa/
SDP-LR_130301 (graph):	dollar.biz.uiowa.edu/~burer/SDP-LR/
SBmethod-1.1.1 (graph):	www.mathematik.uni-kl.de/~helmberg/SBmethod/
LOQO-6.03 (AMPL):	orfe.princeton.edu/~loqo/

These codes (input formats) were used to solve the SDP/SOCP problems from
http://dimacs.rutgers.edu/Challenges/Seventh/Instances/
This is meant to update information made available in the original
Challenge benchmark: http://plato.asu.edu/dimacs/ The codes were run
in default mode on a Pentium 4 (2.5GHz, 2GB, Linux-2.4.19). Given are
total CPU seconds. "m" memory exceeded, "na" not applicable, $ eps=1-4

SDP problem	PENNON	SeDuMi	SDPT3	CSDP	DSDP	SDPA	SDP-LR	SBmth
bm1	129	2133	217	231	1272	109	26	96$
biomed	m	m	m	m	m	m	1447	66822$
industry2	m	m	m	m	m	m	2618	713753$
copo14	31	7	9	8	3	63	na	na
copo23	2069	702	657	651	2897	6382	na	na
filter48	349	6	14	148	fail	418	na	na
filtinf	192	6	14	133	1422	480	na	na
ham_7_5_6	48	43	18	15	7	19	1	1
ham_9_8	144	224	59	65	10	33	7	1
ham_8_3_4	29274	m	m	m	20260	m	64	1
ham_9_5_6	m	m	m	m		m	92	1
ham_10_2	m	m	m	m	m	m	248	35
ham_11_2	m	m	m	m	m	m	951	77
torusg3-8	28	277	25	27	3	28	8	2
torusp*8*	20	262	24	27	3	20	7	1847
torusg3-15	4925	m	3156	3458	753	m	229	103
torusp*15*	3380	m	3124	3457	710	m	38	92

SOCP problem	MOSEK	SDPT3	SeDuMi	LOQO
nb	3	11	9	11
nb_L1	3	20	12	9
nb_L2	8	18	25	16
nb_L2_bessel	2	11	13	7
nql30	2	4	4	11
nql60	9	14	10	151
nql180	199	232	461	m
qssp30	2	6	4	15
qssp60	14	29	20	221
qssp180	211	504	730	m
sched_50_50_orig	1	6	6	5
sched_100_50_orig	2	17	13	22
sched_100_100_orig	4	28	29	94

```
sched_200_100_orig           10        95       139       409
sched_50_50_scaled            1         6         4         7
sched_100_50_scaled           2        13         9        28
sched_100_100_scaled          5        22        32       107
sched_200_100_scaled         10        68        76       445
=====================================================================
```

12 Nov 2002

Several SDP-codes on sparse SDP problems

Hans D. Mittelmann (mittelmann@asu.edu)

The URL of this page is ftp://plato.la.asu.edu/pub/sparse_sdp.txt

The following codes

CSDP-4.1:	http://www.nmt.edu/~borchers/csdp.html
PENNON-2.01:	http://www2.am.uni-erlangen.de/~kocvara/pennon/
SDPT3-3.01:	http://www.math.nus.edu.sg/~mattohkc/
SeDuMi-1.05R4:	http://fewcal.kub.nl/sturm/software/sedumi.html
DSDP-4.5:	http://www-unix.mcs.anl.gov/~benson/
SDPA-6.0	http://www.is.titech.ac.jp/~yamashi9/sdpa/index.html

were used to solve the SDP problems from

ftp://plato.la.asu.edu/pub/mater/
http://www2.am.uni-erlangen.de/~kocvara/pennon/problems.html

The codes were run in default mode on a Sun Ultra 60, Solaris 8, 450 MHz, 2GB except as indicated. Given are total CPU seconds.

problem	PENNON	s	SeDuMi	s	SDPT3	s	CSDP	s	DSDP	s	SDPA#	s
buck-3	249	8	8435	5	144	5	263	4	342	6	36	4
buck-4	1390	7	187586	8	846	6	5477	5	2417	5	369	5
buck-5	16127	7			9520	4	73631	1	72460	6	7305	5
mater-3	40	10	62	11	119	7	63	9	888	6	1442	7
mater-4	183	9	334	11	m		1565	8	20840	6	>30000	
mater-5	491	8	754	10	m		12832	8				
mater-6	1467	8	2116	8	m	&	201916&	8				
trto-3	106	10	5429	11	62	6	117	5	104	6	18	6
trto-4	646	8	180130	6	476	4	933	3	1336	6	186	6
trto-5	10109	7			4753	4	9907	4	34840	7	2808	5
vibra-3	217	8	7578	11	149	5	369	4	438	6	49	5
vibra-4	1186	7	196999	9	1033	5	3480	2	4156	5	399	6
vibra-5	13337	7			12301	5	64150	3	138600	6	7313	6

"s" correct digits in primary objective, #: run on 2.5GHz, 2GB P4
"m" available memory exceeded, "&" solved on 24GB Sun E6500, 400MHz
CSDP, PENNON: cc -fast -lsunperf; SDPT3: HKM direction.

Problem Statistics

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
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