The improvements in MOSEK version 5. Technical report: 1-2007

Erling D. Andersen Bo Jensen Rune Sandvik Ulf Worsøe *

September 25, 2007

Abstract

MOSEK optimization tools is a software package for solution of large-scale mathematical optimization problems. The first release of MOSEK appeared in 1999 and the latest version in July 2007. In the present report we discuss the new features and improvements in the most recent version 5 of MOSEK.

1 Introduction

MOSEK optimization tools is a software package for solution of large-scale optimization problems. MOSEK can solve optimization problems of the following types:

- linear,
- quadratic,
- conic,
- nonlinear, and
- mixed-integer.

MOSEK optimization tools also include interfaces that makes it easy to deploy the functionality of MOSEK from programming languages such as C, C++, Java, .NET, and Python.

The major new improvements and features in MOSEK version 5 are:

- Improved speed and stability of the interior-point optimizer.
- Improved speed and stability of the simplex optimizers.
- Added a special optimizer for network flow problems.
- Added capabilities for solving conic mixed integer optimization problems.
- Added a Python interface.

Subsequently, we will discuss in some detail the improvements appearing in MOSEK version 5.

*MOSEK ApS, Fruebjergvej 3 Box 16, 2100 Copenhagen, Denmark

2 Improvements in the optimizers

2.1 The simplex optimizers

The main focus areas for the simplex optimizers have been:

- Numerical stability. The simplex optimizers are much less likely to terminate due to numerical problems in version 5 compared to version 4.
- Hot-start efficiency. In version 4 no presolve is applied when hot-starting. In version 5 a limited presolve is applied leading to a speed up in many cases.
- Degeneracy handling. Both in the primal and dual simplex optimizers the degeneracy handling has been improved.
- Performance improvements.

Beyond those general improvements the performance of the primal simplex optimizer on slim problems, i.e. problems with many more variables than constraints, has been improved dramatically. In addition the steepest-edge and Devex variable selection schemes are now available in the primal simplex optimizer.

On average the primal and the dual simplex optimizers are approximately 25% and 40% respectively faster than the version 4 implementation.

2.2 The network flow optimizer

Linear network flow optimization problems is a special class of linear optimization problems for which it is known that a special implementation of the simplex algorithm is much faster than a general purpose implementation. The specialized network flow optimizer can in many cases solve a pure network flow optimization problem one or two order of magnitudes faster than the standard simplex optimizer. See Section 3.1.4 for computational results comparing the network optimizer to the general purpose simplex optimizers.

2.3 The interior-point optimizer

The interior-point optimizers in MOSEK has been improved. Notably we have improved:

- The stopping criterion. For both the linear and conic interior-point optimizer the stopping criterion has been changed, so that a possible infeasible status is detected more reliably.
- Numerical stability. The numerical stability of the linear and conic interior-point optimizers have been improved.
- Speed improvements. The linear interior-point optimizer is on average 10% faster for large scale linear problems.

3 Computational results

This section will demonstrate the capabilities of the optimizers in MOSEK 5.

The computational results will be presented using a number of tables that usually has the format of the example Table 1. Subsequently we will explain Table 1 in details.

Table 1 shows a fictive run comparing version 4 and version 5 of an optimizer. Hence, a number of test problems has been solved using version 4 and version 5 of an optimizer e.g. the primal simplex optimizer. We categorize the test problems into small, medium, and large sized problems. A problem belong to the class of small sized problems if the fastest optimizer solves a problem in less than 1 second. Whereas if the fastest optimizer solve a problem in less than than 120 seconds then it belongs to the class of medium sized problems. The remaining problems belong to the class of large sized problems.

In Table 1 we report how many problems that where solved by each optimizer in the row named "Num." for each class of problems. Next we report how many times an optimizer is first in the row "Firsts". More precisely we say an optimizer is first on a problem if the optimizer solve a problem within a time that is at most 1% slower than the fastest optimizer.

From Table 1 it is seen that version 5 solved three out of four problems faster than version 4 in the group of small problems. Moreover, for the small problems version 5 and version 4 requires 40.30 secs. and 80.34 secs. respectively to solve all the small problems.

	small		medium		large	
	5	4	5	4	5	4
Num	4	4	20	20	13	13
Firsts	3	1	15	5	10	3
Total time	40.30	80.34	210.23	703.45	3789.68	10456.78
G. avg.	1.45	2.35	4.55	7.02	245.46	463.78

Table 1: Example Table

In order to obtain the test results we employ public available test problems such as the NETLIB and the Kennington problems as well as problems collected from our users. Finally, all results are obtained using the two computers:

- A Dual Core server with 4GB RAM running Windows 2003 (Intel CPU).
- A Quad Core server with 8GB RAM running Windows 2003 (Intel CPU).

All results presented in one table is obtained using one of the two computers only.

3.1 The simplex optimizers

As mentioned in Section 2.1 the simplex optimizers have been improved. In the following sections we will show test results for version 4 and 5 optimizers on a number of test problems.

3.1.1 The primal simplex optimizer

In Table 2 the performance of the version 4 and version 5 primal simplex optimizer is compared. Independent of the problem size the version 5 optimizer has the most firsts, the lowest total time, and the lowest geometric average time. For the large problems the solution time is approximately reduced by 25%.

	small		medium		large	
	5	4	5	4	5	4
Num.	399	399	148	148	30	30
Firsts	329	245	91	62	22	11
Total time	100.40	101.66	2425.34	8962.31	29905.15	39333.17
G. avg.	0.06	0.07	7.49	9.24	591.39	746.01

Table 2: Performance of the version 4 and version 5 primal simplex optimizer.

On problems with many more variables than constraints the version 5 of the primal simplex optimizer show much more improvement than for general problems as documented in Table 3. Table 3 demonstrates for a class of long slim problems version 5 primal simplex optimizer may reduce the solution with up to 75% for large instances. It should be mentioned that a large part of the problem set employed to produce Table 3 are relaxation of set covering, set partition and set packing problems. Moreover, the problems range from having 5 to 3000 times more variable than constraints.

	small		medium		large	
	5	4	5	4	5	4
Num.	49	49	36	36	5	5
Firsts	35	45	20	16	4	1
Total time	7.51	4.96	460.76	761.98	1890.98	13545.38
G. avg.	0.04	0.03	7.70	8.53	362.70	1331.27

Table 3: Performance of the version 4 and 5 of the primal simplex optimizer on slim problems.

3.1.2 The dual simplex optimizer

In Table 4 the performance of the version 4 and version 5 dual simplex optimizer is compared. The version 5 optimizer is comparable to version 4 optimizer on the small sized problems. For medium and large sized problems the average time is reduced by approximately 40%.

	small		medium		large	
	5	4	5	4	5	4
Num.	412	412	150	150	21	21
Firsts	198	286	133	22	18	5
Total time	84.79	106.42	1852.95	7611.27	23678.99	38994.33
G. avg.	0.10	0.08	4.65	8.70	544.44	1065.24

Table 4: Performance of the version 4 and version 5 dual simplex optimizer

3.1.3 Numerical difficult problems

The numerical stability of the simplex optimizers have also been improved in version 5. The simplex optimizers are now able to solve significantly more instance from our collection of difficult test problems.

In Table 5 the performance of the primal simplex optimizer on a small set of numerical difficult problems is shown.

By numerical difficulty we mean problems that are badly scaled or problem where MOSEK experience:

- Repeated loss of feasibility.
- Repeated basis singularity occurs.

Table 5 documents that the primal simplex in version 5 is slightly faster for solving these numerically difficult problems . Version 4 fails on 6 instances. A failure means the optimizer did not report an optimal solution or did not terminate within one hour. If none of the optimizers can not solve a problem,that problem is excluded from the results.

	small		medium		large	
	5	4	5	4	5	4
Num.	9	9	19	19	2	2
Firsts	5	5	13	6	2	0
Total time	2.73	2.75	235.93	319.63	1297.76	1503.34
G. avg.	0.19	0.18	7.19	9.54	413.26	464.04
Fails	0	0	0	3	0	3

Table 5: Performance of the version 4 and 5 of the primal simplex optimizer on numerical problems.

Table 6 presents the performance of the version 4 and 5 of the dual simplex optimizer. The table shows that version 5 wins on 24 out of the 34 problems and has both a much better total running time and geometric average, except for medium sized problems where one outlier dominates the version 5 results.

	small		medium		large	
	5	4	5	4	5	4
Num.	11	11	19	19	4	4
Firsts	7	6	13	6	4	0
Total time	3.94	6.58	3198.33	345.92	4736.27	12820.55
G. avg.	0.24	0.31	8.44	9.67	802.24	2525.35
Fails	0	0	0	1	0	1

Table 6: Performance of the version4 and 5 dual simplex optimizer on numerical difficult problems.

To summarize the stability of the simplex optimizers has been improved significantly in version 5 compared to version 4.

3.1.4 The network flow optimizer

MOSEK version 5 includes a new and extremely fast optimizer for network flow problems. The network optimizer is a specialized implementation of the primal simplex algorithm that exploits the network structure.

Table 7 compare the performance of the network optimizer with the general purpose implementation of the primal simplex and dual simplex optimizer. The network test instances are of various sizes and structure. The largest test problem has more than 8 million variables and is solved in less than 200 seconds with the network optimizer. In the table "netw", "psim", and "dsim" denote the network, primal simplex and the dual simplex optimizer respectively.

The network optimizer is faster than the standard primal and dual simplex optimizer on all instances. The geometric average for the standard primal and dual simplex optimizer is about 15 and 3 times larger than for network simplex. The total running time show similar results.

	small			medium			large		
	netw	psim	dsim	netw	psim	dsim	netw	psim	dsim
Num.	30	30	30	43	43	43	2	2	2
Firsts	30	0	1	43	0	0	2	0	0
Total time	13.75	114.83	27.81	589.95	10676.60	3015.24	366.30	2905.82	968.86
G. avg.	0.39	2.42	0.70	6.30	91.74	19.70	182.98	1115.71	468.76

Table 7: Performance of the network flow, primal simplex and dual simplex optimizer on pure network problems.

The version 5 simplex optimizers are much faster than the version 4 simplex optimizers on network problems, which is demonstrated in Table 8.

	small		medium		large	
	5	4	5	4	5	4
Num.	9	9	43	43	23	23
Firsts	8	1	43	0	23	0
Total time	6.96	10.18	943.61	4824.87	12202.67	49982.06
G. avg.	0.75	1.05	9.68	35.76	364.17	1697.86

Table 8: Performance of the version 4 and 5 of the primal simplex optimizer on pure network problems.

For completeness Table 9 demonstrates that the version 5 dual simplex optimizer has better performance than the version 4 dual simplex optimizer on network flow problems.

	small		medium		large	
	5	4	5	4	5	4
Num.	20	20	44	44	11	11
Firsts	12	10	39	5	11	0
Total time	9.90	10.96	772.08	1369.56	3238.97	10348.34
G. avg.	0.45	0.48	7.02	9.35	228.11	463.91

Table 9: Performance of the version 4 and 5 of the dual simplex optimizer in on network problems.

3.2 The interior-point optimizer

Table 10 shows results of the interior-point optimizer when applied to a test library of approximately 1000 linear optimization problems. For small sized problems version 4 and and 5 is approximately equally fast whereas for medium and large sized problems version 5 is 10% faster than version 4. Moreover, for the medium and large sized problems the version 5 optimizer has the most firsts.

	small		medium		large	
	5	4	5	4	5	4
Num.	714	714	286	286	25	25
Firsts	534	565	192	98	19	6
Total time	137.01	138.85	5502.88	6039.54	9691.92	12016.65
G. avg.	0.07	0.07	7.38	7.87	320.89	371.92

Table 10: Performance on the version 4 and 5 of the interior-point optimizer.

4 Conclusion

This report has discussed the performance improvements gained from version 4 to version 5. For medium to large sized problems the primal simplex, the dual simplex and the interior-point optimizer is approximately 25%, 40% and 10% faster respectively in version 5. Moreover, we have documented that for slim problems the version 5 primal simplex optimizer may be up to 75% faster.

Finally, we have shown that the new network flow optimizer introduced in version 5 can solve network flow problems much faster than the general purpose simplex optimizer. The network optimizer generally solves network flow problems 50% to 90% faster then the general purpose simplex optimizers.