Running Time and Accuracy Comparison Between Miles, Cplex, Gurobi, and Mosek

COMP 400 Report

Xueyang Zhang

260886286

xueyang.zhang@mail.mcgill.ca

Abstract

There are four software, Miles, Cplex, Gurobi, and Mosek, being able to solve ILS, MILS, BILS. I run a numerical test for them on MATLAB and find that Miles is way much faster than others and is 100% correct. Cplex and Gurobi are a lot slower than Miles, but their answers are accurate as well. Between them two, Cplex is good for difficult problems, and Gurobi is good for easier problems. Mosek is discouraged because it makes mistakes in finding the optimal solution, and is the slowest.

1. Introduction

A standard integer least square(ILS) problem is to solve

$$min_{x\in B}\|\mathrm{y}-\mathrm{Ax}\|_2^2$$

For some arbitrary $y \in \mathbb{R}^n$, $x \in \mathbb{Z}^m$, $A \in \mathbb{R}^{m\times n}$, and $B = \mathbb{Z}^n$. For box constraint integer least square problem (BILS), we change the B to $\{x \in \mathbb{Z}^n : l \le x \le u\}$. Another type of problem, standard mixed-integer least squares problem(MILS) is to solve

$$min_{x_1\in\mathbb{R}^k, x\in\mathbb{Z}^n}\|y - Ax_1 - Bx\|_2^2$$

where $A \in \mathbb{R}^{m\times k}$, $B \in \mathbb{R}^{m\times n}$, and $y \in \mathbb{R}^n$. These are the three problems we are interested in this paper.

ILS problem has huge amount applications in industry, for example, GPS data. However, it is an NP-hard problem: basically, if we believe in exponential time hypothesis (ETH), then the ILS problem is an exponential-time algorithm, so it is exponentially costly as we increase input size; that's why a lot of effort is in to make more efficient algorithm, and this report is to test how efficient each developed softare is and compare them.

2. Test Cases

We will test on Miles, Cplex, Gurobi, and Mosek on these three problems on MATLAB. We select n to be 10, 20, 30, and 40. Without losing generality, we make $m = n$, $n = 2k$. We generate x by MATLAB command randi([0, 1], n, 1); in other words, we generate an n-by-1 integer matrix from range 0 to 1. We make a standard normally distributed error vector z with $\sigma = 0.05$ and 0.4, and it is different for each test; we achieve that by generating different x vectors from different random seeds. From z, we construct y by

$$y = Ax + z$$

For BILS, we will test two kings of bounds: $l_1 = [-1, -1, \dots, -1]^T$ and $u_1 = [1,1 \dots,1]^T$, and $l_{10} = [-10, -10, \dots, -10]^T$ and $u_{10} = [10,10 \dots,10]^T$. For simplicity, we will refer to l_1 and u_1 as B1, and refer to l_{10} and u_{10} as B10.

3. Test method

Miles is already a MATLAB file code, so we convert it by codegen command in MATLAB to make it a C program to run faster and test in MATLAB as mex files. Other three software has MATLAB connector, so we test through this connector in MATLAB. They specify on their official websites that these integer programming problems are time-consuming and sub-optimal solutions are often enough. However, since Miles finds an optimal solution, we need to make that three software get optimal solutions as well for fairness. Cplex and Gurobi allow changing settings to return optimal solution, but Mosek can only increase optimality and cannot give optimal solution. As a result, the result from Mosek may be incorrect, and it indeed is as we can see in the error table in Appendix. Cplex has a direct ILS connector, but Gurobi and Mosek only have generic quadratic integer optimization as follow:

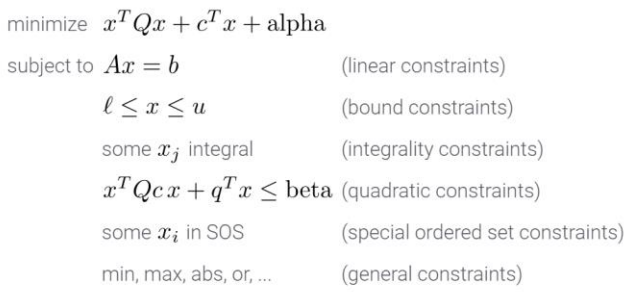

Figure 1 Gurobi_problem_formulation

Mosek has the same functionality but in a different form, so we need to convert our problem to this.

$$\begin{aligned}\|\mathrm{y} - \mathrm{Ax}\|_2^2 &= (y^T - x^T A^T)(y - Ax) \\ &= y^T y - 2y^T Ax + x^T A^T Ax \\ &= x^T (A^T A)x + (-2y^T A)x + y^T y\end{aligned}$$

As a result, we can calculate $A^T A$, $-2y^T A$, and $y^T y$ and plug these into the parameters to solve the ILS problem. We can see it provides the l and u parameters so we can write those in directly.

For each trial, we not only measure its running time and record it, but see if it is correct. We test Miles first and take that as the correct answer(this correctness is guaranteed). After Cplex, Gurobi, and Mosek return their answer, they will be compared with the correct answer. For real value answers, like in MILS, to check equality is inappropriate because of numerical issues, so we take the difference of the returned answer and the correct answer and calculate the norm of the difference. If the norm is more than 0.1, then they are different answers, otherwise, we count them as correct. In this case, this threshold should be appropriate because the real value vector only depends on the integer value vector returned. If two integer vectors are identical, then they should get at least nearly identical real value vectors because they are calculated in the exact same way. If two integer vectors are different, then they will already be judged to be different from the integer vector. As a result, we can measure the correctness of the answer for three software in the method above.

However, for MILS, because there are 2 vectors to output and the variable **x** in figure 1 only supports vector instead of a matrix, we technically cannot solve MILS by Cplex, Gurobi, and Mosek. However, we observe that in the Miles package, MILS modifies the input and uses the ILS program, so we replace the ILS of Miles with ILS of Cplex, Gurobi, and Mosek so that they can solve MILS as well.

Because of the shortage of time and computation power, we cannot run a full test. We will set a time limit to 5000s, which is already very unreasonably long. Besides, we run a 3-trial on each set up first; if the time on 1 more software all exceeds the time limit, we will only run 20 trials, otherwise 100 trials. You can see a few stars (*) in the table for each scenario in the Appendix; those setups of parameters are only tested 20 times, and others are tested 100 times.

4. Notes on packages

Miles is developed by Prof X.-W. Chang and T. Zhou and available in www.cs.mcgill.ca/~chang. It contains direct functions to solve ILS, MILS, and BILS.

ILOG CPLEX Optimization Studio(Cplex) is developed by IBM and available at https://www.ibm.com/analytics/cplex-optimizer. Only versions before and including 12 have a connector to MATLAB so we installed version 12.10.0. Cplexlsqmilp is the MATLAB interface to solve ILS and BILS; it essentially creates a problem object and passes this object to ILOG CPLEX Optimization Studio to solve it.

GUROBI OPTIMIZATION(Gurobi) is developed by Gurobi and available in https://www.gurobi.com/products/gurobi-optimizer/. It similarly creates a problem object and passes this object to Gorobi Interactive Shell to solve it.

MOSEK is developed by Mosek ApS and available at https://www.mosek.com/downloads/. It does the same process as Gurobi and Cplex when using it in MATLAB. We will see that it performs very badly in our tests because it is designed for large-scale sparse systems, which is not our case. (recall that we generate A and B by rand and randn command, so they are dense matrices)

5. Test Results and discussions

(1) scenario1(ILS), $n = 20$, $\sigma = 0.4$, $A = randn(n, n)$

σ=0.4, n=20, randn	Miles	Cplex	Gurobi	Mosek
MAX	0.000471	0.584472	0.053710	0.424823
MIN	0.000223	0.099851	0.022890	0.053255
AVG	0.000275	0.134309	0.037003	0.208379
Num. Outliers	15	11	0	4
Q1	0.000242	0.107434	0.033025	0.171692
Q2	0.000251	0.116590	0.035905	0.202475
Q3	0.000286	0.129689	0.041360	0.236958

Table1: statistics of time for scenario1

Miles	Cplex	Gurobi	Mosek
0%	0%	0%	0%

Table2: error rate for scenario1

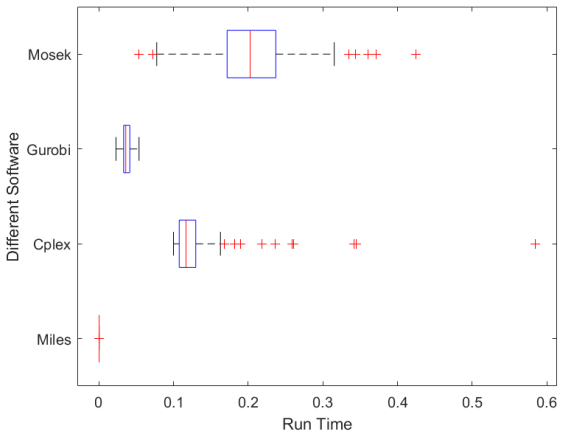

Figure 2 box plot for senario1

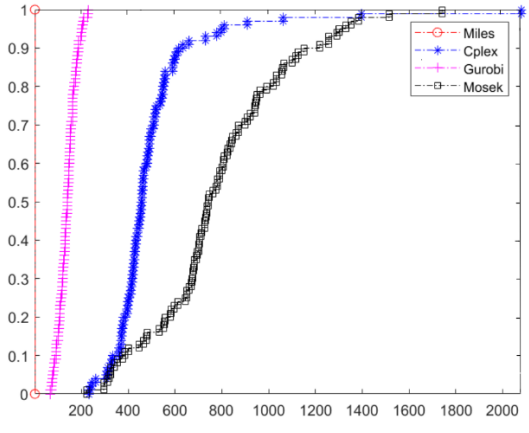

Figure 3 performance profile for senario1

Above are all related statistics for the running time for this scenario. The top-left corner of table 1 is the summary of this scenario, and it will be specified in every table in appendix 1, and this is one of them. Table 2 is from Appendix 2 with some modification: because we measure that there is no error for Miles, Cplex, and Gurobi for any circumstance that we tested, so we only show the nontrivial error rate, so only the error rate for Mosek is shown in the Appendix. For clarity, we show the error rate for all 4 software in a row in the discussion part. Also, there will be two figures for each scenario as well, box plot and performance profile. Box plot gives a direct presentation of the dispersity of data by only showing general statistics of most of the data and directly plot outliers. Meanwhile, the performance profile is a "normalized" data summary: since the same test cases have been run on all four software, some of them are more difficult than others, therefore measuring the speed ratio to the fastest among them makes more sense. Performance is a representation of these speed ratios.

The running time for all software is less than 1 second, so this scenario is easy to compute (or simply refer to easy or difficult later in the paper), and none of them makes a mistake. In terms of average speed, the speed of Miles is 100 times faster than the other 3 software. There are 15 outliers in running time, meaning there is a relatively large fluctuation, but it does not matter much because of its superior speed to others. Gurobi is in second place, and it is 4 times faster than Cplex. Its running times are very concentrated to the median, as we can see there are no outliers in table 1 and boxplot. Cplex is the third fastest in terms of average speed, but as we can see in the performance profile and box plot, there are many outliers to the right whereas most of the running time is relatively concentrated, so Cplex is not very stable in terms of speed in solving a problem like scenario1. Mosek is the slowest, being about 50% slower than Cplex, and its running time varies a lot as we can see from the box plot. Taking into account that Mosek is designed for sparse matrices, this result explains some reason Mosek being the worst. Overall, when solving senario1, the first option is Miles. If it is no handy, you can use Gurobi, Cplex, and Mosek at last.

(2) scenario2(ILS), $n = 40$, $\sigma = 0.4$, $A = rand(n, n)$

σ=0.4, n=40, rand (*)	Miles	Cplex	Gurobi	Mosek
MAX	0.245755	295.342904	1512.434353	5063.346127
MIN	0.055700	41.371151	387.706056	5001.975342
AVG	0.121784	132.104977	953.280466	5025.120338
Num. Outliers	2	2	0	0
Q1	0.092615	84.325502	501.287797	5005.354789
Q2	0.119719	93.298170	967.868239	5021.237640
Q3	0.149128	194.368711	1406.709330	5037.967842

Table3: statistics of time for scenario2

Miles	Cplex	Gurobi	Mosek
0%	0%	0%	60%

Table4: error rate for scenario2

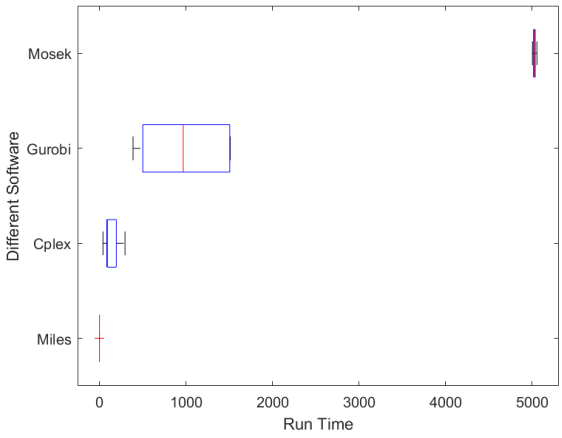

Figure 4 box plot for scenario2

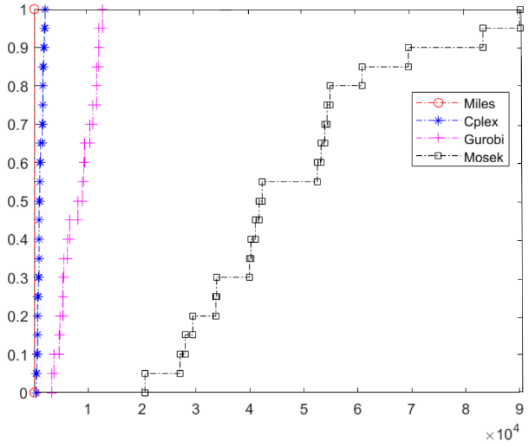

Figure 5 performance profile for scenario2

We can see from table3, that the running time for each software increases, meaning this probably is a more difficult problem than scenario1. Indeed, as we can see through appendix1, as dimension increases, the running time generally increases by several times if other options(like σ, type of A, etc) hold the same, and it is expected because by Exponential Time Hypothesis, running time is exponential in length of problem input. The magnitude of σ is also a crucial factor. Recall that we generate y by $y = Ax + z$, and each element of z is generated by a standard normal distribution with specified σ. As a result, a smaller sigma means a smaller magnitude of z, so the bond between y and x is tighter, so the search process when solving the problem is easier. That is the reason we normally see an increase in running time when we increase σ. Besides, the type of matrix influences the running time a lot as well. It is hard to exactly explain especially when we do not know the computation method of Cplex, Gurobi, and Mosek, but we can see that the Frobenius norm of randn(n,n) type matrix is larger than that of rand(n,n), so the same amount of change in one element in x is expected to change more on y, so y is more sensitive to a change in x, and this makes search processes easier by eliminating more candidate values to check.

By table4, we know only Mosek makes lots of mistakes. As a result, unless you do not care about accuracy, Mosek already has the worst performance. By table3, we know Miles has a decisive advantage over the other three, having an average running time of only 0.12 seconds and having no outliers. In terms of average running time, it is 1100 times shorter than the second place, Cplex. If you have a problem like scenario2, then use Miles. Although Cplex is slower than Gurobi in scenario1, it is better in scenario2. It needs 132 seconds to solve a problem in scenario2, which is still reasonable. Compare to the scenario1, we see Cplex may have some expensive data structure to make their running time shorter for difficult problems, but the complicated data structure becomes a heavy overhead when the problem is easy. As a result, when solving difficult ILS problems like scenario2, if Miles is not handy, you can use Cplex. Gurobi, meanwhile, becomes slower in this more difficult scenario, being 7 times slower than Cplex. It also has lots of variance in running time as we see in figure 4. Lastly, Mosek, not only making lots of mistakes but all of them exceed the time limit as we can see in the box plot. Note that the time limit is 5000s, and it needs some time to halt the process and return data; this process costs some time especially for a program ran for a long time, so we see that all time in table3 for Mosek is more than 5000s. The actual time it will run does not matter, because 5000 seconds is already unreasonably long so people will consider new methods of solving it. As a result, in a more difficult scenario, the best is Miles, and then Cplex, Gurobi, and Mosek.

(3) scenario3(MILS), $n = 20, \sigma = 0.05, A = rand(n, n)$

σ=0.05, n=20, rand	Miles	Cplex	Gurobi	Mosek
MAX	0.002030	0.616464	0.166520	5.273527
MIN	0.000825	0.207273	0.069284	0.929038
AVG	0.001020	0.338409	0.114903	2.582232
Num. Outliers	14	2	0	1
Q1	0.000916	0.264742	0.102142	2.014623
Q2	0.000949	0.304508	0.112251	2.552927
Q3	0.001018	0.393782	0.129246	3.031747

Table5: statistics of time for scenario3

Miles	Cplex	Gurobi	Mosek
0%	0%	0%	25%

Table6: error rate for scenario3

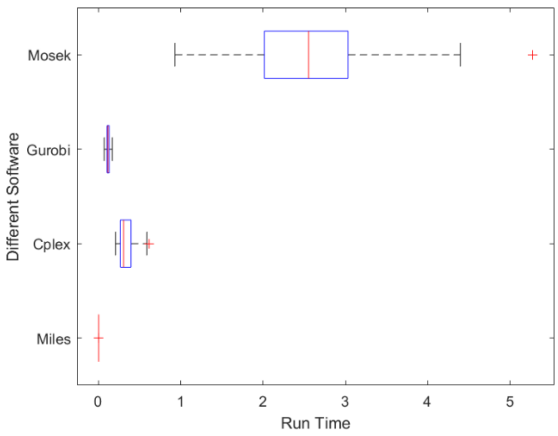

Figure 6 box plot for scenario3

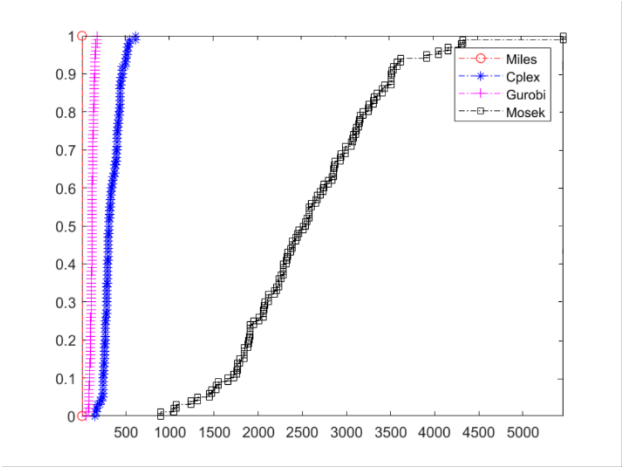

Figure 7 performance profile for scenario3

As we can see in table6, similarly, only Mosek has mistakes, so arguably Mosek is already the worst. This scenario is relatively easy as we can see it has a dimension of only 20, and the average time for all software solving it being short. Miles is still the quickest, having 0.001s on average; it has lots of outliers, so its running time may be unstable, but it does not matter considering it is 100 times faster than the best of the other three, so it is decisively the best. Again, we see Gurobi performs better for simpler problems. It is three times faster than cplex, with little variance in running time as we can see in the box plot and performance profile. Similarly, Cplex is not performing very well for a simple problem and has lots of variances as we can see in Figures 6 and 7. Mosek is the slowest and has a big variance, as usual. It even has a large variance in the speed ratio with "standard time" from Miles, varying from 100 times to 5000. Similarly, for simpler problems, Miles is the best, and then Gurobi, Cplex, and Mosek.

(4) scenario4(MILS), $n = 40$, $\sigma = 0.05$, $A = randn(n, n)$

σ=0.05, n=40, randn (*)	Miles	Cplex	Gurobi	Mosek
MAX	0.516101	728.243709	5030.737643	5011.447547
MIN	0.375896	248.242465	5018.311371	5005.983521
AVG	0.451914	409.892016	5025.446931	5008.857153
Num. Outliers	0	1	0	0
Q1	0.420661	319.515819	5022.313210	5007.930878
Q2	0.451048	352.155332	5024.803892	5008.810561
Q3	0.481932	496.253763	5029.287487	5010.187636

Table7: statistics of time for scenario4

Miles	Cplex	Gurobi	Mosek
0%	0%	0%	5%

Table8: error rate for scenario4

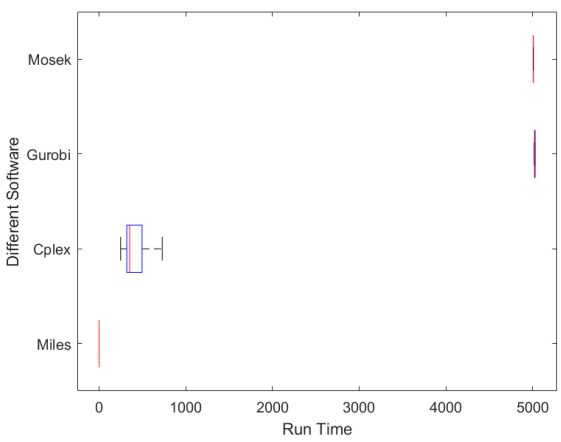

Figure 8 performance profile for scenario4

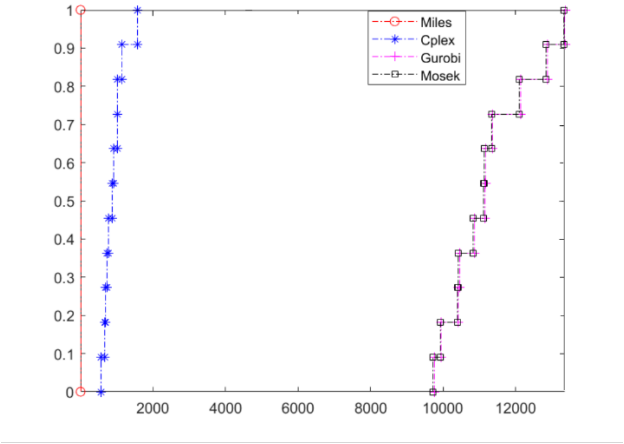

Figure 9 box plot for scenario4

From table8, we see, still, only Mosek makes an error, although it is not much error than itself, but still comparatively unacceptable. It should have been an easy problem as we analyzed in scenario2, that a smaller σ and a $randn(n,n)$ type matrix would make the problem easier, but it is not the case. For Miles, the reason is that it does a transformation of matrix and then solves a standard ILS, so the "good properties" that we analyzed do not work in MILS. As we said in the Test method part, Cplex, Gurobi, and Mosek technically do not have the functionality to solve MILS, and we observe that the main part of Miles is to do a standard ILS, so we do the same thing on three software to make them work for MILS as well, so the same reason applies to three software as well to make this scenario hard. Miles still has the best running time, 0.45 second on average, and 900 times faster than second-place Cplex. As we can see from table7, Q1, Q2, and Q3 of Miles are very close, and there is no outlier, so Miles has a stable run time in scenario4. Still, we see Cplex being second best for difficult problems: it solves the problem still in a reasonable amount of time. It has a relatively large variance as we can see in Q1, Q2, and Q3 in table7, and it has one outlier. However, Gurobi and Mosek both fail to solve scenario4 within the time limit. We can see that Gurobi and Mosek are not good for difficult problems. As a result, when solving scenario4, the first option is still Miles, and then Cplex, Gurobi, and Mosek, because this scenario is difficult.

(5) scenario5(BILS), $n = 30, \sigma = 0.4, A = rand(n,n), B = 1$

σ=0.4, n=30, rand, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.004488	0.460267	0.799035	3.900129
MIN	0.000602	0.150377	0.107815	0.123742
AVG	0.001349	0.235663	0.369915	0.787350
Num. Outliers	11	11	0	6
Q1	0.000831	0.199160	0.268293	0.379753
Q2	0.001110	0.220851	0.355547	0.628807
Q3	0.001591	0.251742	0.469565	0.967842

Table9: statistics of time for scenario5

Miles	Cplex	Gurobi	Mosek
0%	0%	0%	14%

Table10: statistics of error for scenario4

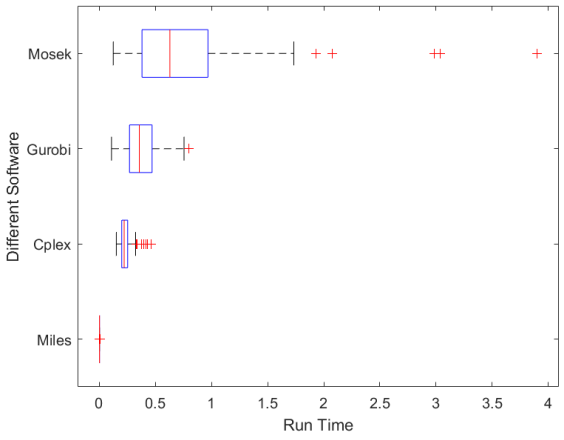

Figure 10 box plot for scenario5

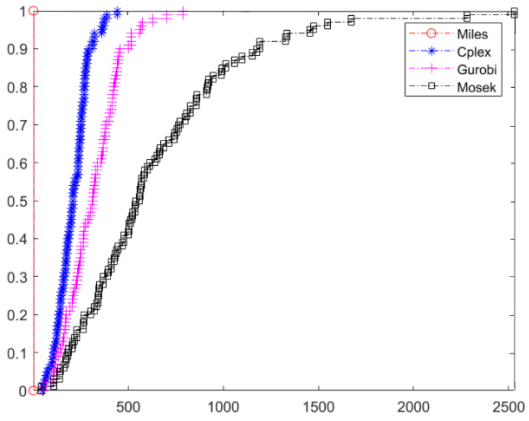

Figure 11 performance profile for scenario5

Conceptually, BILS is very similar to ILS. The only difference is that BILS specifies the range of the solution, but ILS does not. To constrain solution normally makes solving this problem easier. This scenario has bound on x with lower bound $= [-1, -1, \dots, -1]^T$, and upper bound = $[1,1, \dots, 1]^T$. You can compare the time with that with ILS with the same parameter selection and it is indeed the case. Under the same parameter selection, this problem is hard, but not anymore with specifying a range of solution: the maximum time for any software solving this problem is only 3 seconds. We can see through Table10 that only Mosek still makes mistakes, so it is still discouraged to use. Miles is still the quickest with 200 times faster than the second quickest, Cplex. The speed ratio between Cplex and Miles is relatively low, because Cplex is robust in problem specification, meaning it uses the same procedure to solve ILS and BILS, whereas there is a new program solving BILS. As a result, BILS for Miles is not necessarily easy because this new constraint may make the program more complicated, so there is not that much of a difference between the two in this scenario. In contrast, though this problem is easy, Gurobi is slower than Cplex by 50% and has a large variance in run time as we can see in the box plot, but it is still acceptable. As usual, Mosek is the rightmost part of the performance profile and box plot, so it is both slow and unstable in running time, but this time, under this scenario, the difference is smaller. However, it still has errors, so it is still unacceptable.

(6) scenario6(BILS), $n = 30$, $\sigma = 0.4$, $A = rand(n, n)$, $B = 10$

σ=0.4, n=30, rand, B=10	Miles	Cplex	Gurobi	Mosek
MAX	2.662336	18.644138	77.207288	1978.809075
MIN	0.021084	3.604093	12.066648	273.597023
AVG	0.133567	8.786895	38.547147	1080.113244
Num. Outliers	2	2	0	0
Q1	0.057762	6.118268	28.947506	653.054608
Q2	0.078205	8.484645	36.950025	929.496845
Q3	0.111072	10.545104	48.986172	1551.102561

Table11: statistics of time for scenario6

Miles	Cplex	Gurobi	Mosek
0%	0%	0%	26%

Table12: statistics of error for scenario6

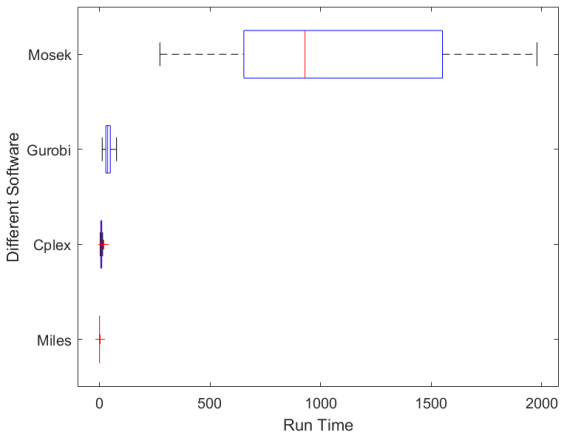

Figure 12 box plot for scenario6

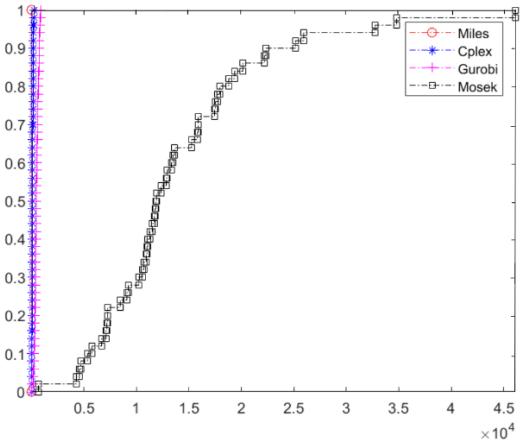

Figure 13 performance profile for scenario6

This is the exact same scenario as the previous one, except that the bound is bigger to lower bound $= [-10, -10, \ldots, -10]^T$, and upper bound = $[10,10, \ldots, 10]^T$. As we analyzed previously, the running time of this scenario should be larger than previous scenario, and it is indeed the case. Similarly, in theory, it should be smaller than that of ILS because the range of searching is smaller. Cplex, Gurobi, and Mosek satisfy this, but not for Miles, because it uses two different program to solve this two problems. For Miles, the reason ILS runs faster may be that the ILS program takes advantage of the infinite bound so that it is free to pick any candidate, so this new constraint becomes a issue, not a boost, for solving BILS. However, Miles is still the fastest of all, being 0.13s in average and 65 times faster than second place Cplex, although this ratio is the smallest through out the paper. From this fact, you can see the superiority for Miles to solve these three problems. The running time for Miles has lots of variability, because we can see the average(0.13) is even more than the third quantile(0.11), which means there is some cases that Miles feels difficult. Cplex becomes even more faster than Gurobi compare to the previous scenario, so Cplex is resilient in changing parameters and solves problems relatively fast. Gurobi is not ideal but still acceptable so solve scenario6. Mosek still makes mistakes, unsurprisingly it makes more mistakes than previous easier scenario. What's more, as we can see from performance profile an box plot, it takes way more time than other three softwares as we can see the difference in box and that we cannot clearly see lines for three other software. As a result, Mosek is still discouraged to use, and Miles is the best, followed by Cplex and Gurobi.

6. Conclusion

We can see that Miles is the best by any means. Under all scenarios, including those that we did not show in the paper, Miles has a 100% accuracy, and its average speed is at least several tens' times faster than the second place, probably hundreds and thousands of times. It sometimes varies a lot, but it does not matter because of its absolute superiority in speed. Miles designs targeted programs to solve different problems. This could be the reason it is so much faster than others. The other three software run a general big program to take care and satisfy every option possible, so they can solve ILS problems with more constraints, like linear and quadratic constraints as we see in figure 1.

Cplex is generally the second-best in solving the above problems. It has 100% accuracy, and it is relatively good to solve difficult problems by its high robustness. It sometimes varies a lot and runs slower than Gurobi especially when the problem is easy to solve. However, all four software can solve easy problems quickly, probably less than several seconds, so this drawback can sometimes be ignored. Its reasonable run time for difficult problems can be appreciated.

Gurobi has 100% accuracy, and is good for smaller problems, but it struggles in time when dimension increases, and the problem becomes more difficult. Maybe it lacks some optimization for higher dimensions.

Unfortunately, Mosek is inarguably the worst software to solve these three problems. Not only it makes mistakes for about 20% of the problems it solves, but it is slow by several times or even tens of times than the second-worst. As a result, it is not a good choice to solve ILS, MILS, and BILS problems by Mosek.

7. Appendix

(1) ILS time tables

σ=0.05, n=10, rand	Miles	Cplex	Gurobi	Mosek
MAX	0.000928	0.503089	0.033432	0.184986
MIN	0.000125	0.043612	0.014710	0.024626
AVG	0.000187	0.071826	0.019005	0.038472
Num. Outliers	24	19	7	20
Q1	0.000131	0.049460	0.017124	0.028061
Q2	0.000140	0.053371	0.018153	0.030439
Q3	0.000184	0.065286	0.019269	0.035549

σ=0.05, n=10, randn	Miles	Cplex	Gurobi	Mosek
MAX	0.000221	0.156290	0.026742	0.044067
MIN	0.000106	0.042936	0.014040	0.024037
AVG	0.000124	0.049287	0.017421	0.028396
Num. Outliers	21	8	7	6
Q1	0.000109	0.044849	0.016330	0.027314
Q2	0.000112	0.046547	0.016965	0.028038
Q3	0.000125	0.048804	0.018062	0.028933

σ=0.4, n=10, rand	Miles	Cplex	Gurobi	Mosek
MAX	0.000394	0.205619	0.038543	0.209837
MIN	0.000123	0.073309	0.015812	0.034148
AVG	0.000163	0.109525	0.024554	0.096073
Num. Outliers	14	10	3	5
Q1	0.000132	0.102919	0.022454	0.076293
Q2	0.000142	0.105835	0.024250	0.092025
Q3	0.000175	0.109724	0.026501	0.109915

σ=0.4, n=10, randn	Miles	Cplex	Gurobi	Mosek
MAX	0.000330	0.220746	0.037044	0.082966
MIN	0.000105	0.044430	0.014431	0.026264
AVG	0.000129	0.091284	0.019149	0.044316
Num. Outliers	19	39	6	0
Q1	0.000109	0.053198	0.017407	0.033684
Q2	0.000115	0.104913	0.018573	0.043534
Q3	0.000131	0.108305	0.019907	0.051490

σ=0.05, n=20, rand	Miles	Cplex	Gurobi	Mosek
MAX	0.792704	0.567832	0.079137	0.323134
MIN	0.000322	0.039956	0.020051	0.042606
AVG	0.008317	0.103546	0.028317	0.148517
Num. Outliers	12	26	6	0
Q1	0.000344	0.091325	0.024973	0.059288
Q2	0.000363	0.099485	0.026831	0.178873
Q3	0.000399	0.110511	0.029035	0.200904

σ=0.05, n=20, randn	Miles	Cplex	Gurobi	Mosek
MAX	0.000399	0.100727	0.046181	0.072724
MIN	0.000226	0.030247	0.019280	0.039246
AVG	0.000258	0.038938	0.025369	0.044435
Num. Outliers	17	6	2	2
Q1	0.000232	0.034731	0.023532	0.042156
Q2	0.000241	0.036828	0.025435	0.043992
Q3	0.000258	0.039618	0.026812	0.045447

σ=0.4, n=20, rand	Miles	Cplex	Gurobi	Mosek
MAX	0.001126	0.698194	0.716195	21.689515
MIN	0.000366	0.118315	0.059471	1.237625
AVG	0.000469	0.223883	0.163236	8.023438
Num. Outliers	7	16	5	3
Q1	0.000407	0.166648	0.123035	5.452168
Q2	0.000407	0.190216	0.147045	7.864821
Q3	0.000489	0.223949	0.175972	9.977286

σ=0.4, n=20, randn	Miles	Cplex	Gurobi	Mosek
MAX	0.000471	0.584472	0.053710	0.424823
MIN	0.000223	0.099851	0.022890	0.053255
AVG	0.000275	0.134309	0.037003	0.208379
Num. Outliers	15	11	0	4
Q1	0.000242	0.107434	0.033025	0.171692
Q2	0.000251	0.116590	0.035905	0.202475
Q3	0.000286	0.129689	0.041360	0.236958

σ=0.05, n=30, rand	Miles	Cplex	Gurobi	Mosek
MAX	0.001932	0.358849	0.147173	3.871621
MIN	0.000757	0.111917	0.031492	0.121504
AVG	0.000939	0.159201	0.064077	0.750322
Num. Outliers	17	9	3	4
Q1	0.000819	0.127653	0.046855	0.531467
Q2	0.000854	0.142637	0.056555	0.681504
Q3	0.000914	0.164057	0.075767	0.813515

σ=0.05, n=30, randn	Miles	Cplex	Gurobi	Mosek
MAX	0.001289	0.422777	0.115965	0.542207
MIN	0.000611	0.040541	0.028125	0.065006
AVG	0.000734	0.094365	0.040386	0.193003
Num. Outliers	14	4	6	37
Q1	0.000642	0.051760	0.034305	0.078033
Q2	0.000683	0.095262	0.036990	0.086196
Q3	0.000744	0.110403	0.042174	0.354507

σ=0.4, n=30, rand (*)	Miles	Cplex	Gurobi	Mosek
MAX	0.019092	62.911906	348.988998	5004.592225
MIN	0.002822	4.975625	24.052834	316.7936931
AVG	0.010379	33.529773	185.603990	4928.253426
Num. Outliers	0	0	0	3
Q1	0.006910	22.969903	129.715620	5000.013640
Q2	0.010510	29.596569	164.233725	5000.541972
Q3	0.013320	45.710977	236.770992	5000.627255

σ=0.4, n=30, randn	Miles	Cplex	Gurobi	Mosek
MAX	0.003793	0.599470	1.446578	10.898088
MIN	0.000650	0.164433	0.110619	0.783768
AVG	0.000930	0.279305	0.338099	2.813416
Num. Outliers	10	6	5	4
Q1	0.000696	0.204368	0.161011	1.368793
Q2	0.000741	0.235521	0.233891	1.944502
Q3	0.000849	0.305961	0.397220	3.470903

σ=0.05, n=40, rand	Miles	Cplex	Gurobi	Mosek
MAX	0.002976	0.525913	0.073278	1.130010
MIN	0.000937	0.055153	0.030632	0.113451
AVG	0.001083	0.127591	0.04081	0.339849
Num. Outliers	10	25	5	47
Q1	0.000966	0.119678	0.035723	0.143668
Q2	0.001013	0.127061	0.039090	0.175806
Q3	0.001113	0.133202	0.044012	0.523663

σ=0.05, n=40, randn	Miles	Cplex	Gurobi	Mosek
MAX	0.001609	0.139681	0.074032	0.213327
MIN	0.000628	0.042991	0.028628	0.101933
AVG	0.000713	0.053251	0.034992	0.120870
Num. Outliers	9	10	6	7
Q1	0.000648	0.047504	0.032075	0.113380
Q2	0.000680	0.049399	0.033646	0.117676
Q3	0.000738	0.053208	0.035435	0.122198

σ=0.4, n=40, rand (*)	Miles	Cplex	Gurobi	Mosek
MAX	0.245755	295.342904	1512.434353	5063.346127
MIN	0.055700	41.371151	387.706056	5001.975342
AVG	0.121784	132.104977	953.280466	5025.120338
Num. Outliers	2	2	0	0
Q1	0.092615	84.325502	501.287797	5005.354789
Q2	0.119719	93.298170	967.868239	5021.237640
Q3	0.149128	194.368711	1406.709330	5037.967842

σ=0.4, n=40, randn	Miles	Cplex	Gurobi	Mosek
MAX	0.001921	0.326567	0.373813	1.155376
MIN	0.000696	0.166552	0.062588	0.165783
AVG	0.000886	0.251574	0.162844	0.607466
Num. Outliers	2	5	2	5
Q1	0.000765	0.244752	0.116323	0.501305
Q2	0.000808	0.256133	0.149623	0.592978
Q3	0.000913	0.269272	0.198961	0.666021

(2) MILS time tables

σ=0.05, n=10, rand	Miles	Cplex	Gurobi	Mosek
MAX	0.001193	0.827448	0.136895	0.236411
MIN	0.000296	0.134256	0.017539	0.048175
AVG	0.000629	0.241887	0.031190	0.109148
Num. Outliers	22	6	11	2
Q1	0.000542	0.176718	0.022960	0.083959
Q2	0.000565	0.209095	0.025984	0.104129
Q3	0.000651	0.262349	0.033388	0.126335

σ=0.05, n=10, randn	Miles	Cplex	Gurobi	Mosek
MAX	0.001623	0.610520	0.074238	0.221242
MIN	0.000510	0.137756	0.014317	0.035296
AVG	0.000661	0.278518	0.024391	0.076423
Num. Outliers	19	2	6	3
Q1	0.000548	0.188627	0.019842	0.055799
Q2	0.000576	0.255796	0.022540	0.068311
Q3	0.000712	0.350776	0.025923	0.092915

σ=0.4, n=10, rand	Miles	Cplex	Gurobi	Mosek
MAX	0.001642	1.017951	0.299200	0.301618
MIN	0.000517	0.141630	0.018915	0.043841
AVG	0.000736	0.332149	0.038818	0.133376
Num. Outliers	6	3	6	1
Q1	0.000563	0.208662	0.025566	0.097583
Q2	0.000672	0.290222	0.032259	0.129996
Q3	0.000672	0.409876	0.040758	0.163255

σ=0.4, n=10, randn	Miles	Cplex	Gurobi	Mosek
MAX	0.002502	0.669969	0.043189	0.147028
MIN	0.000188	0.126095	0.014140	0.028488
AVG	0.000613	0.238739	0.023874	0.071203
Num. Outliers	17	6	3	3
Q1	0.000527	0.174580	0.020100	0.052747
Q2	0.000555	0.207545	0.023072	0.068351
Q3	0.000617	0.281055	0.026716	0.082162

σ=0.05, n=20, rand	Miles	Cplex	Gurobi	Mosek
MAX	0.002030	0.616464	0.166520	5.273527
MIN	0.000825	0.207273	0.069284	0.929038
AVG	0.001020	0.338409	0.114903	2.582232
Num. Outliers	14	2	0	1
Q1	0.000916	0.264742	0.102142	2.014623
Q2	0.000949	0.304508	0.112251	2.552927
Q3	0.001018	0.393782	0.129246	3.031747

σ=0.05, n=20, randn	Miles	Cplex	Gurobi	Mosek
MAX	0.002077	0.461855	0.197817	2.122921
MIN	0.000798	0.172451	0.056588	0.541641
AVG	0.001014	0.286637	0.077607	1.013477
Num. Outliers	26	0	6	3
Q1	0.000836	0.220934	0.06853	0.846765
Q2	0.000861	0.273933	0.073311	1.015356
Q3	0.001053	0.341594	0.081312	1.143412

σ=0.4, n=20, rand	Miles	Cplex	Gurobi	Mosek
MAX	0.002535	0.664769	0.190719	4.061664
MIN	0.000841	0.180671	0.041741	0.336322
AVG	0.001065	0.337188	0.096834	1.765275
Num. Outliers	17	4	2	4
Q1	0.000906	0.271318	0.078813	1.182456
Q2	0.000961	0.303928	0.093681	1.661305
Q3	0.001047	0.412760	0.109579	2.055240

σ=0.4, n=20, randn	Miles	Cplex	Gurobi	Mosek
MAX	0.001579	0.516549	0.227465	2.202074
MIN	0.000754	0.159788	0.037290	0.289504
AVG	0.000911	0.274002	0.073436	0.921726
Num. Outliers	14	1	5	3
Q1	0.000807	0.209458	0.060054	0.648098
Q2	0.000853	0.259952	0.068820	0.842354
Q3	0.000917	0.318945	0.076693	1.088739

σ=0.05, n=30, rand	Miles	Cplex	Gurobi	Mosek
MAX	0.005523	2.901601	25.227480	375.842835
MIN	0.002289	0.810301	9.938927	47.173375
AVG	0.003286	1.925461	16.331425	130.352474
Num. Outliers	2	0	0	1
Q1	0.002845	1.530404	13.393229	77.406259
Q2	0.002845	2.003119	16.201483	114.517962
Q3	0.003508	2.281303	18.813914	170.727315

σ=0.05, n=30, randn	Miles	Cplex	Gurobi	Mosek
MAX	0.006116	3.318812	10.984681	192.630694
MIN	0.003884	2.006763	7.330396	49.943532
AVG	0.004907	2.600960	9.140967	114.954567
Num. Outliers	2	1	0	0
Q1	0.004603	2.466759	8.640681	93.989179
Q2	0.004887	2.577473	9.074334	113.283163
Q3	0.005070	2.775028	9.614826	132.598015

σ=0.4, n=30, rand	Miles	Cplex	Gurobi	Mosek
MAX	0.005458	3.023398	29.469760	619.833292
MIN	0.001937	0.377212	0.423581	15.884672
AVG	0.003300	1.827074	16.447260	160.906530
Num. Outliers	0	0	1	2
Q1	0.002617	1.323778	12.886597	90.821595
Q2	0.003120	1.753353	16.116091	148.880107
Q3	0.003889	2.419575	19.618560	196.098610

σ=0.4, n=30, randn	Miles	Cplex	Gurobi	Mosek
MAX	0.008039	3.702028	13.964519	222.267921
MIN	0.002257	0.744942	2.691140	23.568421
AVG	0.004542	2.169681	7.794493	94.769169
Num. Outliers	0	0	2	1
Q1	0.003633	1.667701	6.673525	67.590384
Q2	0.004391	2.208108	7.788217	91.473823
Q3	0.005562	2.638735	9.249139	118.710332

σ=0.05, n=40, rand (*)	Miles	Cplex	Gurobi	Mosek
MAX	5.599239	223.820327	1952.486820	5008.331100
MIN	0.195099	46.356740	332.650490	3433.622872
AVG	1.168566	119.870675	989.383762	4862.402544
Num. Outliers	3	0	1	1
Q1	0.328651	82.714136	521.034085	5003.200088
Q2	0.356178	108.351000	707.983639	5004.044636
Q3	0.659639	168.005266	1621.531700	5006.813754

σ=0.05, n=40, randn (*)	Miles	Cplex	Gurobi	Mosek
MAX	0.516101	728.243709	5030.737643	5011.447547
MIN	0.375896	248.242465	5018.311371	5005.983521
AVG	0.451914	409.892016	5025.446931	5008.857153
Num. Outliers	0	1	0	0
Q1	0.420661	319.515819	5022.313210	5007.930878
Q2	0.451048	352.155332	5024.803892	5008.810561
Q3	0.481932	496.253763	5029.287487	5010.187636

σ=0.4, n=40, rand (*)	Miles	Cplex	Gurobi	Mosek
MAX	0.616967	785.764907	5017.306867	5010.758456
MIN	0.198668	204.099579	719.458345	5003.809195
AVG	0.387814	447.532446	2584.644061	5007.140669
Num. Outliers	0	0	0	0
Q1	0.319231	248.642834	1776.028370	5005.588102
Q2	0.334591	342.775219	2420.821879	5007.346900
Q3	0.483849	665.573463	3267.933111	5008.330272

σ=0.4, n=40, randn (*)	Miles	Cplex	Gurobi	Mosek
MAX	0.606901	785.764907	5050.369481	5011.523522
MIN	0.341954	204.099579	5014.940342	5006.963722
AVG	0.477724	447.532446	5030.381311	5009.009294
Num. Outliers	0	0	0	0
Q1	0.400778	248.642834	5019.362704	5007.967842
Q2	0.468001	342.775219	5033.602589	5008.777802
Q3	0.566076	665.573463	5039.046498	5009.909406

(3) BILS time table

σ=0.05, n=10, rand, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.000823	0.244454	0.416307	0.091778
MIN	0.000142	0.053468	0.023975	0.030027
AVG	0.000179	0.084591	0.047442	00.030027
Num. Outliers	22	13	8	16
Q1	0.000147	0.070008	0.031470	0.033194
Q2	0.000152	0.075864	0.041979	0.034800
Q3	0.000174	0.087344	0.045712	0.038118

σ=0.05, n=10, randn, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.000320	0.158125	0.052067	0.059129
MIN	0.000136	0.049109	0.021628	0.030048
AVG	0.000158	0.066636	0.032014	0.035323
Num. Outliers	16	4	1	4
Q1	0.000142	0.054687	0.025116	0.033579
Q2	0.000145	0.061333	0.029653	0.034730
Q3	0.000157	0.074345	0.039757	0.036088

σ=0.4, n=10, rand, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.000368	0.203298	0.094487	0.080328
MIN	0.000138	0.054826	0.022979	0.030600
AVG	0.000166	0.141676	0.046581	0.049807
Num. Outliers	14	18	1	0
Q1	0.000148	0.136759	0.037158	0.042743
Q2	0.000153	0.152589	0.046005	0.049615
Q3	0.000160	0.164715	0.056484	0.058005

σ=0.4, n=10, randn, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.000638	0.691864	0.299284	0.086116
MIN	0.000136	0.053005	0.022032	0.031089
AVG	0.000176	0.118757	0.046822	0.042009
Num. Outliers	21	6	5	14
Q1	0.000146	0.074997	0.030598	0.034889
Q2	0.000152	0.093881	0.042679	0.037428
Q3	0.000170	0.141512	0.049926	0.045437

σ=0.05, n=10, rand, B=10	Miles	Cplex	Gurobi	Mosek
MAX	0.000350	0.235270	0.117280	0.125635
MIN	0.000136	0.065603	0.031078	0.030366
AVG	0.000179	0.102015	0.050794	0.050702
Num. Outliers	21	28	16	29.000000
Q1	0.000148	0.075140	0.039957	0.036118
Q2	0.000157	0.081921	0.043635	0.039476
Q3	0.000194	0.130113	0.051819	0.066229

σ=0.05, n=10, randn, B=10	Miles	Cplex	Gurobi	Mosek
MAX	0.000260	0.081018	0.049536	0.038075
MIN	0.000136	0.066760	0.033751	0.032482
AVG	0.000152	0.073190	0.040986	0.034782
Num. Outliers	14	1	2.000000	0.000000
Q1	0.000141	0.071203	0.039484	0.033877
Q2	0.000145	0.072782	0.040656	0.034608
Q3	0.000152	0.074687	0.042816	0.035567

σ=0.4, n=10, rand, B=10	Miles	Cplex	Gurobi	Mosek
MAX	0.000275	0.325647	0.110597	0.188619
MIN	0.000138	0.156414	0.031408	0.046770
AVG	0.000167	0.185741	0.073447	0.109508
Num. Outliers	15	7	4.000000	0.000000
Q1	0.000148	0.177223	0.066499	0.085075
Q2	0.000157	0.182830	0.072362	0.105086
Q3	0.000166	0.189223	0.084079	0.128558

σ=0.4, n=10, randn, B=10	Miles	Cplex	Gurobi	Mosek
MAX	0.000300	0.258421	0.084647	0.091637
MIN	0.000131	0.072546	0.032593	0.033283
AVG	0.000160	0.137938	0.048728	0.052967
Num. Outliers	18	0	8.000000	0.000000
Q1	0.000141	0.087713	0.039956	0.039525
Q2	0.000147	0.158082	0.045406	0.051676
Q3	0.000158	0.174290	0.052598	0.062728

σ=0.05, n=20, rand, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.000630	0.226646	0.135662	0.107607
MIN	0.000293	0.051562	0.027799	0.046702
AVG	0.000345	0.093573	0.047815	0.058326
Num. Outliers	16	1	3	7
Q1	0.000311	0.073443	0.033186	0.053553
Q2	0.000322	0.092142	0.046217	0.056293
Q3	0.000345	0.108924	0.055088	0.060487

σ=0.05, n=20, randn, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.000616	0.093415	0.076135	0.069192
MIN	0.000304	0.058206	0.026046	0.048436
AVG	0.000343	0.074319	0.043052	0.056559
Num. Outliers	15	1	0	1
Q1	0.000317	0.069532	0.032109	0.053741
Q2	0.000325	0.074217	0.041770	0.056513
Q3	0.000334	0.078100	0.053614	0.058897

σ=0.4, n=20, rand, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.000603	1.062884	0.196048	0.461618
MIN	0.000280	0.056679	0.033783	0.057095
AVG	0.000349	0.205997	0.089913	0.145926
Num. Outliers	13	6	2	4
Q1	0.000313	0.178146	0.063907	0.102418
Q2	0.000331	0.192406	0.088939	0.135589
Q3	0.000356	0.207175	0.105480	0.177822

σ=0.4, n=20, randn, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.000713	0.381372	0.182287	0.124072
MIN	0.000299	0.064134	0.027592	0.050557
AVG	0.000380	0.094809	0.063269	0.068008
Num. Outliers	21	14	7	13
Q1	0.000317	0.076061	0.050622	0.058462
Q2	0.000331	0.081433	0.058552	0.062627
Q3	0.000382	0.102132	0.069146	0.068671

σ=0.05, n=20, rand, B=10	Miles	Cplex	Gurobi	Mosek
MAX	0.000664	0.918636	0.152481	0.499459
MIN	0.000290	0.092182	0.055290	0.070262
AVG	0.000350	0.188737	0.078298	0.190602
Num. Outliers	15	25	2	1
Q1	0.000311	0.173456	0.068672	0.097143
Q2	0.000323	0.185734	0.075949	0.207829
Q3	0.000343	0.199710	0.084803	0.243829

σ=0.05, n=20, randn, B=10	Miles	Cplex	Gurobi	Mosek
MAX	0.000649	0.174677	0.191882	0.121830
MIN	0.000304	0.073027	0.058426	0.067313
AVG	0.000336	0.095265	0.078647	0.074623
Num. Outliers	10	1	3	4
Q1	0.000316	0.078242	0.070914	0.071656
Q2	0.000323	0.099180	0.075356	0.073949
Q3	0.000333	0.103468	0.080780	0.076297

σ=0.4, n=20, rand, B=10	Miles	Cplex	Gurobi	Mosek
MAX	0.003094	0.950524	1.884412	25.233772
MIN	0.000453	0.281717	0.158411	1.752953
AVG	0.001278	0.420958	0.275809	8.076511
Num. Outliers	4	7	3	3
Q1	0.000927	0.364753	0.222548	5.459248
Q2	0.001223	0.405807	0.253163	7.651014
Q3	0.001521	0.440676	0.296106	9.565521

σ=0.4, n=20, randn, B=10	Miles	Cplex	Gurobi	Mosek
MAX	0.000678	0.335181	0.630767	1.082696
MIN	0.000301	0.186676	0.071465	0.096341
AVG	0.000357	0.224234	0.138825	0.284078
Num. Outliers	11	4	3	6
Q1	0.000327	0.207382	0.114678	0.227896
Q2	0.000338	0.220732	0.129941	0.257148
Q3	0.000359	0.233394	0.154860	0.314165

σ=0.05, n=30, rand, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.414825	0.588529	0.376936	0.208997
MIN	0.000562	0.061339	0.060165	0.080845
AVG	0.004861	0.134987	0.120308	0.104636
Num. Outliers	16	6	4	10
Q1	0.000606	0.093120	0.087453	0.090894
Q2	0.000648	0.115349	0.116687	0.098631
Q3	0.000748	0.149526	0.132195	0.106861

σ=0.05, n=30, randn, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.001385	0.134106	0.142457	0.162684
MIN	0.000566	0.069885	0.060640	0.079274
AVG	0.000664	0.087798	0.094777	0.096322
Num. Outliers	11	9	10	3
Q1	0.000592	0.078241	0.088328	0.089724
Q2	0.000618	0.082833	0.092794	0.094017
Q3	0.000670	0.094984	0.098379	0.098853

σ=0.4, n=30, rand, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.004488	0.460267	0.799035	3.900129
MIN	0.000602	0.150377	0.107815	0.123742
AVG	0.001349	0.235663	0.369915	0.787350
Num. Outliers	11	11	0	6
Q1	0.000831	0.199160	0.268293	0.379753
Q2	0.001110	0.220851	0.355547	0.628807
Q3	0.001591	0.251742	0.469565	0.967842

σ=0.4, n=30, randn, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.001743	0.417077	0.261647	0.278847
MIN	0.000559	0.074858	0.083553	0.090612
AVG	0.000665	0.134967	0.115167	0.120524
Num. Outliers	10	4	10	10
Q1	0.000595	0.088762	0.099246	0.107857
Q2	0.000613	0.114617	0.108897	0.113057
Q3	0.000651	0.167778	0.118135	0.120542

σ=0.05, n=30, rand, B=10	Miles	Cplex	Gurobi	Mosek
MAX	0.023204	0.734529	0.490552	1.455291
MIN	0.000585	0.222717	0.093183	0.237371
AVG	0.002140	0.313461	0.181576	0.687109
Num. Outliers	17	7	5	6
Q1	0.000617	0.266963	0.120076	0.569141
Q2	0.000637	0.293371	0.150800	0.645544
Q3	0.000661	0.329833	0.227895	0.789527

σ=0.05, n=30, randn, B=10	Miles	Cplex	Gurobi	Mosek
MAX	0.001271	0.297613	0.284027	0.652692
MIN	0.000574	0.097564	0.068833	0.127865
AVG	0.000660	0.158609	0.107071	0.250373
Num. Outliers	9	1	11	32
Q1	0.000598	0.122415	0.084305	0.145610
Q2	0.000619	0.147323	0.094033	0.157586
Q3	0.000666	0.191680	0.105250	0.412409

σ=0.4, n=30, rand, B=10	Miles	Cplex	Gurobi	Mosek
MAX	2.662336	18.644138	77.207288	1978.809075
MIN	0.021084	3.604093	12.066648	273.597023
AVG	0.133567	8.786895	38.547147	1080.113244
Num. Outliers	2	2	0	0
Q1	0.057762	6.118268	28.947506	653.054608
Q2	0.078205	8.484645	36.950025	929.496845
Q3	0.111072	10.545104	48.986172	1551.102561

σ=0.4, n=30, randn, B=10	Miles	Cplex	Gurobi	Mosek
MAX	0.043837	0.714923	0.712891	6.897621
MIN	0.000633	0.114020	0.071414	0.659534
AVG	0.006364	0.282885	0.221971	2.624047
Num. Outliers	36	8	10	16
Q1	0.000697	0.194893	0.112548	1.325309
Q2	0.000837	0.246250	0.176170	1.893824
Q3	0.006308	0.349635	0.268969	3.597116

σ=0.05, n=40, rand, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.003270	0.362480	0.334886	0.256686
MIN	0.001121	0.075647	0.098395	0.117141
AVG	0.001493	0.163384	0.157794	0.159920
Num. Outliers	25	3	9	13
Q1	0.001167	0.121074	0.138424	0.142668
Q2	0.001204	0.155268	0.151929	0.150181
Q3	0.001509	0.198396	0.162929	0.161276

σ=0.05, n=40, randn, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.003270	0.288779	0.194371	0.221679
MIN	0.001121	0.076501	0.090709	0.122054
AVG	0.001493	0.116599	0.138415	0.150697
Num. Outliers	25	2	0	1
Q1	0.001167	0.087908	0.123336	0.139885
Q2	0.001204	0.107387	0.137254	0.148232
Q3	0.001509	0.133453	0.155244	0.158325

σ=0.4, n=40, rand, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.006636	0.614239	1.122713	17.222088
MIN	0.001170	0.176613	0.152000	0.192544
AVG	0.002321	0.314322	0.482129	2.814194
Num. Outliers	5	3	1	4
Q1	0.001619	0.247780	0.328809	0.937050
Q2	0.002113	0.306628	0.469429	2.056821
Q3	0.002647	0.363973	0.603301	3.634671

σ=0.4, n=40, randn, B=1	Miles	Cplex	Gurobi	Mosek
MAX	0.002234	0.321833	0.235050	0.452954
MIN	0.001078	0.081131	0.098617	0.160211
AVG	0.001285	0.159769	0.129474	0.191218
Num. Outliers	14	2	5	4
Q1	0.001138	0.125658	0.117408	0.177629
Q2	0.001172	0.162988	0.124762	0.185620
Q3	0.001221	0.182300	0.134785	0.195118

σ=0.05, n=40, rand, B=10	Miles	Cplex	Gurobi	Mosek
MAX	0.005143	0.849333	0.257596	1.294442
MIN	0.001093	0.131773	0.079877	0.233545
AVG	0.001460	0.271819	0.110333	0.521373
Num. Outliers	24	11	13	5
Q1	0.001148	0.238450	0.090202	0.299867
Q2	0.001197	0.254626	0.097114	0.382043
Q3	0.001470	0.294626	0.108928	0.693786

σ=0.05, n=40, randn, B=10	Miles	Cplex	Gurobi	Mosek
MAX	0.002474	0.159256	0.105334	0.300861
MIN	0.001113	0.106297	0.071959	0.207965
AVG	0.001339	0.117189	0.086849	0.232465
Num. Outliers	18	6	0	6
Q1	0.001154	0.111202	0.082021	0.222490
Q2	0.001189	0.113974	0.085902	0.229295
Q3	0.001323	0.120378	0.091503	0.238494

σ=0.4, n=40, rand, B=10 (*)	Miles	Cplex	Gurobi	Mosek
MAX	1.126402	523.907142	4131.609260	5008.283234
MIN	0.075499	39.260990	162.742904	2107.376079
AVG	0.421278	172.632878	1252.882469	4914.902983
Num. Outliers	0	2	1	5
Q1	0.156501	71.966388	344.225769	4617.791916
Q2	0.390491	120.663294	1119.459183	5004.458286
Q3	0.594044	169.139358	1225.033402	5006.017255

σ=0.4, n=40, randn, B=10	Miles	Cplex	Gurobi	Mosek
MAX	0.002669	0.666001	0.358033	1.499021
MIN	0.001099	0.293156	0.147093	0.386709
AVG	0.001764	0.432544	0.249022	0.992804
Num. Outliers	0	0	0	0
Q1	0.001157	0.348903	0.197209	0.753406
Q2	0.001776	0.412275	0.233002	0.979136
Q3	0.002297	0.478789	0.306926	1.240660

(4) Error table for ILS

All are mosek

n=10	rand	randn
$\sigma = 0.05$	0%	0%
$\sigma = 0.4$	34%	12%

n=20	rand	randn
$\sigma = 0.05$	0%	0%
$\sigma = 0.4$	39%	0%

n=30	rand	randn
$\sigma = 0.05$	0%	0%
$\sigma = 0.4$	34%	0%

n=40	rand	randn
$\sigma = 0.05$	0%	0%
$\sigma = 0.4$	60%	0%

(5) Error table for MILS

All are mosek

n=10	rand	randn
$\sigma = 0.05$	24%	31%
$\sigma = 0.4$	21%	21%

n=20	rand	randn
$\sigma = 0.05$	25%	21%
$\sigma = 0.4$	18%	21%

n=30	rand	randn
$\sigma = 0.05$	21%	21%
$\sigma = 0.4$	13%	24%

n=40	rand	randn
$\sigma = 0.05$	20%	5%
$\sigma = 0.4$	90%	20%

(6) Error table for BILS

All are mosek

n=10 (100)	rand	randn
$\sigma = 0.05$, B=1	0%	0%
$\sigma = 0.4$, B=1	0%	0%
$\sigma = 0.05$, B=10	12%	0%
$\sigma = 0.4$, B=10	25%	5%

n=20 (100)	rand	randn
$\sigma = 0.05$, B=1	0%	0%
$\sigma = 0.4$, B=1	0%	0%
$\sigma = 0.05$, B=10	10%	0%
$\sigma = 0.4$, B=10	28%	3%

n=30 (100)	rand	randn
$\sigma = 0.05$, B=1	0%	0%
$\sigma = 0.4$, B=1	14%	0%
$\sigma = 0.05$, B=10	0%	0%
$\sigma = 0.4$, B=10	26%	0%

n=40 (20)	rand	randn
$\sigma = 0.05$, B=1	0%	0%
$\sigma = 0.4$, B=1	12%	0%
$\sigma = 0.05$, B=10	0%	0%
$\sigma = 0.4$, B=10	10%	0%

8. References

[1]. X.-W. Chang. MILES: MATLAB package for solving mixed integer least squares problems,

School of Computer Science, McGill University. Retrieved from:

http://www.cs.mcgill.ca/~chang/software/MILES.php

[2]. Cplex, I. I. (2009). V12. 1: User's Manual for CPLEX. International Business Machines Corporation, 46(53), 157.

https://www.gurobi.com/documentation/9.1/refman/MATLAB_the_model_argument.html

[3] LLC Gurobi Optimization. 2021. Gurobi Optimizer Reference Manual. http://www.gurobi.com

[4] MOSEK ApS. 2021.The MOSEK optimization toolbox for MATLAB manual. Version 9.2 https://docs.mosek.com/9.2/toolbox/index.html

[5] MATLAB. (2020). 9.8.0.1451342 (R2020a) Update 5. Natick, Massachusetts: The MathWorks Inc.