

COMPARING THE SOLVERS FOR THE MIXED INTEGER LINEAR PROGRAMMING PROBLEMS AND THE SOFTWARE ENVIRONMENTS THAT CALL THEM*A.N. Ignatov¹, S.V. Ivanov¹*

¹Moscow Aviation Institute (National Research University), Moscow, Russian Federation
E-mail: alexei.ignatov1@gmail.com, sergeyivanov89@mail.ru

The paper presents a concept for comparing the solvers for the mixed integer linear programming problems and the software environments that call them. This concept involves multiple repetition of solving mathematical programming problems with the same initial data to take into account the fact that the computer operations time can be considered as random. It is also assumed to solve the mathematical programming problem with the same structure by varying the initial data to compare the solvers. The comparison is carried out for a number of practical mathematical programming problems. For example we consider the portfolio optimization problem with the probability criterion. Solvers CPLEX, Gurobi, MATLAB, SCIP are used in testing. The features of calling solvers in various software environments are described. In particular, a modification of the source codes for calling the CPLEX solver through the Opti Toolbox add-on in Matlab environment is provided. The components of the time required to obtain a solution for various solvers and software environments are described and studied in detail. It is shown that the operating time of the solver itself can be comparable to the time of reading data from files and the time of forming constraints in a mathematical programming problem.

Keywords: mixed integer linear programming; solver; comparison; software environment.

Introduction

Mixed integer linear programming problems, in which the criterion function and constraints are linear in optimized parameters, and the optimization variables themselves can be either continuous or integer, have many different practical applications. Such problems are used for modeling and traffic control in transport systems [1–7], portfolio optimization problems [8,9], facility location problems [10,11], resource allocation problems [12–15], etc.

A number of commercial and free solvers have been developed to solve mixed integer linear programming problems. Some of the most popular solvers are CPLEX, Gurobi, SCIP. The CPLEX solver was used, particularly, in [1–3, 5–8, 11, 13]. The Gurobi solver was used among other works in [10, 14]. Sometimes [15] articles cover both Gurobi and CPLEX. The good thing about the SCIP solver is that it allows you to solve even problems with nonlinear constraints. This solver was used, for example, in [12].

A comparison of different solvers can be found in [16–18]. CPLEX, Gurobi, XPRESS solvers are analyzed from the point of view of possible mathematical programming problems that they can solve, as well as the features of the functioning of these solvers in [16]. Solvers for mixed integer nonlinear programming problems are compared in [17]. The number of problems solved by one or another solver in a given period of time was used as a quality characteristic. There were 335 such problems in total. At the same time, the

question of how long it takes for one or another solver to solve this or that problem was not discussed. Also the question of how much input data influences the speed of obtaining a solution has not been studied. In addition, the issue of the influence of the software environment on the speed and optimality of the resulting solution was not discussed. The reducing of the computation time for solvers available in the early 2000s and late 2010s was investigated in [18]. It was noted in [18] that, depending on the order of filling the constraint matrix, the time to find a solution, generally speaking, is not the same. The influence of initial data on counting time has not been studied. It is not clear which solver is better.

Note that for testing solvers, you can select some of the problems from [19], however, the choice of one or another problem encounters the need to search for problems from various application areas, the importance of which is beyond doubt. In the future, we will study the portfolio optimization problem from [9], the trajectory correction of the satellite problem [20], the production planning problem [21], as well as the theoretical problem with a polyhedral loss function [22].

We develop a procedure for testing the solvers, which includes multiple repetition of solving the same mathematical programming problem, as well as solving a mathematical programming problem with the same variables, but different input data in this paper. Components for testing solvers are proposed. These components include both various time characteristics and characteristics associated with the “optimal” value of the criterion function produced by the solvers. The study of the work of solvers is carried out taking into account the fact that they can be launched from various environments. Various features of launching solvers from different software environments are discussed.

1. General Form of a Mixed Integer Linear Programming Problem

Let the vector of optimized variables have the form $u = \text{col}(u_1, u_2, \dots, u_n)$, and the variables with indices from the set $\mathcal{I} = \{i_1, i_2, \dots, i_L\}$ are integer, and the rest are continuous. Then, in general form, the mixed integer linear programming problem can be written as

$$c^T u \rightarrow \min_{u_{j_1} \in \mathbb{Z}, u_{j_2} \in \mathbb{R}, j_1 \in \mathcal{I}, j_2 \in \{1, \dots, n\} / \mathcal{I}} \quad (1)$$

under constraints

$$A_1 u \leq b_1, \quad (2)$$

$$A_2 u = b_2, \quad (3)$$

$$l \leq u \leq r, \quad (4)$$

where $A_1 \in \mathbb{R}^{m_1 \times n}$, $b_1 \in \mathbb{R}^{m_1}$, $A_2 \in \mathbb{R}^{m_2 \times n}$, $b_2 \in \mathbb{R}^{m_2}$ are some given matrices (vectors) of the corresponding dimension, where, in turn, m_1 is the number of inequality type constraints, m_2 is the number of equality type constraints. The components of columns l and r are numbers from the extended real axis. Note that, generally speaking, constraints (3) are redundant, since they can be rewritten in the form of constraints (2) with modification of the original matrix A_1 and vector b_1 . Constraints (4) are useful for specifying that some variables are binary, that is, taking the value 0 or 1. Note that if one or another of the constraints (2), (3) are not required, then the matrices and columns of these constraints should be set to zero. If there is no constraints from above (or below) on a component of the vector, then $+\infty$ ($-\infty$) is formally set in the corresponding coordinate of the vector r (or l).

2. Solver Comparison Procedure

To compare the work of solvers and the software environments that call them, we will select a number of practical mathematical programming problems. In each such problem we will vary the set of input data. Let us describe the procedure for comparing solvers for some practical problem.

It should be said that the speed of solvers, as well as the speed of reading data from files, is, generally speaking, not constant. In practice it can be set as a random variable [23]. In this regard, the same experiments are repeated several times in the future. Therefore, on the same set of input data, i.e. matrices A_1 , A_2 , vectors b_1 , b_2 and others, the mathematical programming problem (1) under constraints (2) – (4) will be solved R times by each solver. Next, for each solver and for each problem with the same input data, the median time to obtain a solution is selected (taking into account loading data from files). Thus, for each solver, a vector is formed from the median time values, from which the average, minimum, maximum value, as well as the standard deviation are calculated. The dimension of this vector coincides with the number of input data sets.

It must be said that the solver of a mixed integer linear programming problem may not find a feasible solution in some predetermined time. Or the solution provided by the solver is suboptimal due, for example, to limitations on computation time and computer memory used in the calculation. Therefore, in addition to the time characteristics for each solver, we present the number of times with the best value of the criterion function.

Let us explain the procedure for calculating this value. Using the same initial data, the mathematical programming problem is solved $Q \times R$ times (Q used solvers $\times R$ repetitions). For each set of initial inputs, the best value of the criterion function obtained over all solvers and over all repetitions is calculated. Next, for each solver, the number of cases is calculated when, on the studied set of initial data, the solver generated the best value of the criterion mentioned above. After that, this number of cases is summed over all sets of initial data for each solver, which is the number of times with the best value of the criterion function. Note that each solver runs $N \times R$ times.

Since mathematical programming problems are solved numerically, then the solver is considered to have found the best solution if the value of the criterion on the solution he generated differs from the best value of the criterion by amount not greater than ε .

3. Solvers Under Consideration

To test the operating time of various solvers (together with the time of reading supporting files for solving mathematical programming problems), we will use

- CPLEX solver version 12.5.1
 - launched from the CPLEX Studio IDE (hereinafter referred to as CPLEX IDE);
 - launched from the MATLAB 2014b environment (hereinafter referred to as CPLEX MATLAB);
 - launched using the Opti Toolbox add-on version 2.27 from the MATLAB 2014b environment (hereinafter referred to as CPLEX OPTI).
- SCIP solver version 4.0.0, launched using the Opti Toolbox add-on version 2.27 from the MATLAB 2014b environment (hereinafter referred to as SCIP OPTI);

- MATLAB 2014b's own solver, called by the `intlinprog` command (hereinafter referred to as MATLAB);
- Gurobi solver version 9.5.2, launched from the MATLAB 2014b environment (hereinafter referred to as Gurobi MATLAB);
- Gurobi solver version 9.5.2, launched from the R Studio 1.3.959 environment with version R 4.0.2 (hereinafter referred to as Gurobi R).

Of course, the time required to find a solution to a particular mathematical programming problem by one or another solver may vary from the setting of certain solver parameters. This parameter could be, for example, the method of searching for a solution. However, the sets of parameters for different solvers may differ; in this regard, we will compare “as is”, i.e. using the default values of the solver parameters.

It should also be noted that according to the comments to the source code of the Opti Toolbox add-on version 2.27, there is a conflict between it and the versions 12.5.0, 12.5.1, 12.6.0 of the CPLEX package and the MATLAB environment version not earlier than 2013a. Probably, in this regard, without additional actions with the source code, it is not possible to call the CPLEX solver from the Opti Toolbox add-on - the add-on simply “does not see” it – despite the presence of the installed CPLEX package, MATLAB has access to the folders where the file is located, establishing an interface between MATLAB and CPLEX, as well as copying the specified file to folders, containing the source codes of the Opti Toolbox add-on. In this regard, the authors of the article made “a modification” of the OptiSolver method¹ so that running the CPLEX solver using the add-on becomes possible. Potentially, it would be possible to do without launching CPLEX through the Opti Toolbox add-on, but such launch allows us to make the results of the SCIP and CPLEX solvers comparable.

For comparability of results delivered by different solvers, it is logical to use solvers of the same release time. However, Gurobi version 6.5.2, a contemporary one for the CPLEX environment and solver version 12.5.1, is unfortunately not available for download. Therefore, the maximum “old” version 9.5.2 is used. Due to the use of “old” version 12.5.1 for CPLEX, the most modern version of Gurobi 11.0.0 is not used. It should be noted that the developers state that version Gurobi 9.5.2 is supported in MATLAB versions R2019a-R2022a, however, this solver also works in MATLAB 2014b.

It should also be noted that the Opti Toolbox add-on version 2.27 checks the validity of the selected solver for solving a particular problem. However, such a check is carried out not by the type of function being optimized, but by the method of calling the solver. Namely, if the first parameter of the `opti` method is specified equal to `'fun'` and although a linear function is used for optimization, but declared outside the script calling the solver, then such a superstructure assumes that the problem of mixed integer nonlinear programming is being solved. Therefore, when calling the `opti` method, the first parameter was set to `'f'`, and the function to be optimized was set inside the script calling the solver.

Note that the solvers used are somewhat outdated. In particular, the CPLEX IDE software environment, and the corresponding solver alongside it, has received several updates since version 12.5.1. The use of relatively old versions of the solvers above is due to problems of obtaining licenses for new software.

¹In order to avoid confusion between the mathematical concept of a function and a function as part of a program/script, hereinafter the word *method* is used for a function as a part of a program/script.

For testing we will use a personal computer (Intel Core i5 4690, 3.5 GHz, 16 GB DDR3 RAM) with the Windows 7 operating system installed on it. Note that, generally speaking, the speed of solvers is influenced not only by the characteristics of the hardware, but also by the choice of operating system.

3.1. The Portfolio Optimization Problem with a Probability Criterion

First, let us consider the problem of forming a portfolio of securities using a probability criterion from two risky assets with returns X_1 and X_2 are components of a random vector $X = \text{col}(X_1, X_2)$. Suppose that this vector is discrete and has $K = 1900$ realizations $x_k = \text{col}(x_1^k, x_2^k)$ with equal probabilities $p_k = 1/K$, $k = \overline{1, K}^2$. We will assume that the desired capital φ is equal to 1,1, and the starting capital varies in such a way that $C_i = 2i/N$, $i = \overline{1, N}$. Let the capital be fully invested, and “short-sales” operations are prohibited, then³ for the i case, the matrix A_1 and the vector b_1 have the form

$$A_1 = \underbrace{\begin{pmatrix} -x_1^1 & -x_2^1 & \frac{\varphi}{C_i} & 0 & 0 & \dots & 0 & 0 \\ -x_1^2 & -x_2^2 & 0 & \frac{\varphi}{C_i} & 0 & \dots & 0 & 0 \\ -x_1^3 & -x_2^3 & 0 & 0 & \frac{\varphi}{C_i} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & & \vdots & \\ -x_1^{K-1} & -x_2^{K-1} & 0 & 0 & 0 & \dots & \frac{\varphi}{C_i} & 0 \\ -x_1^K & -x_2^K & 0 & 0 & 0 & \dots & 0 & \frac{\varphi}{C_i} \end{pmatrix}}_{K \text{ rows and } K+2 \text{ columns}}, b_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix}.$$

The matrix A_2 is a row, and the vector b_2 is a scalar, namely:

$$A_2 = (1 \ 1 \ 0 \ 0 \ \dots \ 0), b_2 = 1.$$

The set \mathcal{I} consists of the numbers 3, 4, ..., $K + 2$. In addition, $l = \text{col}(0, 0, 0, \dots, 0)$, $r = \text{col}(+\infty, +\infty, 1, \dots, 1)$. The vector of coefficients c of the criterion function has the form $c = -\text{col}(0, 0, 1, \dots, 1)/K$.

Note that the formulation of the portfolio analysis problem above is a discretized version of it. At the same time, to model returns in portfolio analysis, continuous distributions of returns are often used [24–26].

Having chosen $N = 1000$, we will test the operating time of various solvers for a certain set of realizations of the random vector X . These realizations can be found in [28]. Of course, to check other solvers or the solvers used in the article, but on other hardware it is more convenient to use ready-made A_1 , matrices, however, for only one capital value, the A_1 matrix takes up approximately 10 megabytes of disk space if stored in a matrix in .xlsx format. In this case, just reading the matrix in MATLAB in this format takes about 8 seconds. If you save the matrix in the internal MATLAB .mat format, then reading

²Note that the choice of exactly 1900 realizations is due to restrictions on the authors’ license to use the Gurobi package

³The criterion function and the corresponding constraints from [9] in the [9] notation are not given in this article in order to avoid confusion with the existing notation.

this matrix takes less than half a second, however, in this case, the comparability of the results with the CPLEX IDE software environment is lost, since this environment does not read .mat files. Due to this, only the Excel file containing implementations of the random vector X is supplied as input to both the CPLEX and MATLAB environments. The rest of the matrix A_1 , as well as other matrices and vectors necessary for the calculation, are generated in the program.

Let us choose $R = 3$. Undoubtedly, more than 3 repetitions allows you to better smooth out the results. However, as will be shown later, one of the solvers is quite slow. The components of the resulting time are the duration of loading data from a file, generating a vector of criterion function coefficients, matrices and constraint vectors, as well as the work of the solver itself. Let us set the permissible deviation ε equal to $5 \cdot 10^{-4}$.

Table 1

Comparison of solver performance for the first pool of examples in the portfolio optimization problem

Solver	CPLEX IDE	CPLEX MATLAB	CPLEX OPTI	SCIP OPTI	MATLAB	Gurobi MATLAB	Gurobi R
Minimum median operating time, [s]	0,766	0,493	0,567	0,514	0,514	0,594	0,058
Average median operating time, [s]	1,561	1,07	0,987	3,218	24,998	0,693	0,156
Maximum median operating time, [s]	11,468	50,067	4,385	9,153	1241,763	0,913	0,457
Standard deviation of median operating time, [s]	1,07	1,95	0,653	2,797	95,28	0,066	0,072
Number of times with the best value of criterion function, [units]	3000	3000	3000	3000	3000	3000	3000

As follows from Table 1, the worst results in terms of average operating time and standard deviation are produced by the built-in MATLAB solver. Noteworthy are the differences in the timing characteristics of the CPLEX solver invoked in different ways. This is, in particular, due to the fact that depending on the method of calling the solver, the parameters (solution method, permissible number of iterations) of its operation may differ. The best results in terms of minimum, average, maximum time, standard deviation are obtained by the Gurobi solver. On the same data set, the best values of the criterion function for all solvers for any of the three repetitions are the same. In this case, depending on the set of inputs, the time to obtain the optimal solution changes by several times, and sometimes by several orders.

It is obvious that the work time of the solvers depends not only on the amount of the investor's starting capital C_i , but also on the realization of the vector of returns. Therefore, let us analyze the performance characteristics of solvers for other values of vector of returns. We will assume such returns to be subject to a truncated normal distribution with parameters $0,1; 0,1^2$ and $0,1; 0,15^2$ respectively. The truncation rule can be found in [27]. Realizations of the vector of returns can again be found in [28]. We will again vary the starting value of capital. The results of such testing are presented in Table 2.

As can be seen from Table 2, on average, the median time when calling CPLEX using the Opti Toolbox add-on is almost three times less than when calling it directly

Table 2

Comparison of solver performance for the second pool of examples in the portfolio optimization problem

Solver \ Characteristics	CPLEX IDE	CPLEX MATLAB	CPLEX OPTI	SCIP OPTI	MATLAB	Gurobi MATLAB	Gurobi R
Minimum median operating time, [s]	0,755	0,47	0,533	0,491	0,508	0,576	0,055
Average median operating time, [s]	2,703	1,52	0,665	1,09	29,426	0,625	0,11
Maximum median operating time, [s]	1117,81	481,93	4,041	15,681	3556,663	4,516	4,167
Standard deviation of median operating time, [s]	36,586	16,395	0,446	1,916	204,408	0,206	0,216
Number of times with the best value of criterion function, [units]	3000	3000	2991	3000	3000	3000	3000

from Matlab. This is due, in particular, to the fact that in a number of cases a limit is reached on the number of iterations, branching vertices of the algorithm, and memory, the boundaries of which are set automatically. However, when calling CPLEX from the Opti Toolbox add-on, the solver in some cases does not find the optimal solution.

Comparing the results in Tables 1 and 2, we notice that the average median time for some solvers has increased, and for others it has decreased. The common thing is that for both types of return distributions, the Gurobi solver works most consistently in terms of the range of the median time, as well as it has the smallest standard deviation of the median time in relation to other solvers.

3.2. The Problem of Correcting the Scalar Terminal State of a Satellite with a Geostationary Orbit

Next, we consider the problem of adjusting the scalar terminal state of a satellite with a geostationary orbit from [20]. Control in [20] formulation is understood as the magnitude of the corrective impulse communicated to the satellite depending on the range of deviations in which the value of its current deviation is. The control in [20] formulation is sought in the class of piecewise constant controls, therefore for this problem

$$A_1 = \underbrace{\begin{pmatrix} t_1(1 + y_1^1) & Z - (\varphi - \bar{z}^i) & 0 & 0 & \dots & 0 \\ t_1(1 + y_1^2) & 0 & Z - (\varphi - \bar{z}^i) & 0 & \dots & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ t_1(1 + y_1^{K-1}) & 0 & 0 & 0 & Z - (\varphi - \bar{z}^i) & 0 \\ t_1(1 + y_1^K) & 0 & 0 & 0 & 0 & Z - (\varphi - \bar{z}^i) \\ -t_1(1 + y_1^1) & Z - (\varphi + \bar{z}^i) & 0 & 0 & \dots & 0 \\ -t_1(1 + y_1^2) & 0 & Z - (\varphi + \bar{z}^i) & 0 & \dots & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ -t_1(1 + y_1^{K-1}) & 0 & 0 & 0 & Z - (\varphi + \bar{z}^i) & 0 \\ -t_1(1 + y_1^K) & 0 & 0 & 0 & 0 & Z - (\varphi + \bar{z}^i) \end{pmatrix}}_{2K \text{ rows and } K+1 \text{ columns}},$$

and, in addition, $b_1 = Z \cdot \text{col}(1, 1, \dots, 1)$, $A_2 = (0, 0, \dots, 0)$, $b_2 = 0$, $c = -\text{col}(0, 1, \dots, 1)/K$, $l = \text{col}(u_{\text{low}}, 0, \dots, 0)$, $r = \text{col}(u_{\text{up}}, 1, \dots, 1)$. In the notation above

- y_1^1, \dots, y_1^K are realizations of some random variable with a given distribution law, K is the number of such realizations;
- t_1 is the parameter which characterizes the influence of the corrective action on the deviation value, $t_1 > 0$;
- u_{low} and u_{up} are some acceptable limits for the value of the correction impulse;
- φ is permissible absolute value of the deviation.

Also $\bar{z}^i = 0.5(z^i + z^{i+1})$, $z^{i+1} = z^i(1 - 2i/N)$, $i = \overline{1, N}$. Finally, $Z = |t_1| \max\{|u_{\text{low}}|, |u_{\text{up}}|\}(1 + \max\{|y_1^1|, \dots, |y_1^K|\})$. The set \mathcal{I} consists of the numbers 2, 3, ..., $K + 1$.

We choose $z_1 = -3$, $t_1 = 1$, $u_{\text{up}} = -u_{\text{low}} = 10$, $\varphi = 1, 15$ as in [20]. The reader can find realizations, or rather pseudorandom numbers, from the normal distribution law with parameters 0 and $0,5^2$ in [28].

Let us put $K = 500$, $N = 50$. Let us compare the work of solvers for the above structure of matrices A_1 , A_2 , vectors b_1 , b_2 , as well as other vectors. Let us do 3 repetitions of solving the same mathematical programming problem with each solver ($R = 3$). Thus, each solver will solve 150 optimization problems. Let us choose the permissible deviation ε equal to 10^{-3} .

Table 3

Comparison of the performance of solvers for the correction of scalar terminal state of a satellite with a geostationary orbit problem

Solver \ Characteristics	CPLEX IDE	CPLEX MATLAB	CPLEX OPTI	SCIP OPTI	MATLAB	Gurobi MATLAB	Gurobi R
Minimum median operating time, [s]	0,76	0,39	0,428	0,403	0,427	0,447	0,016
Average median operating time, [s]	1,034	0,5	0,517	0,905	22,908	0,512	0,076
Maximum median operating time, [s]	1,533	0,871	0,866	1,837	430,21	0,593	0,159
Standard deviation of median operating time, [s]	0,231	0,125	0,116	0,489	66,38	0,05	0,054
Number of times with the best value of criterion function, [units]	150	150	150	150	150	150	150

Analyzing Table 3, we again come to the conclusion that even a small change in the set of initial data (in this case, the value \bar{z}^i) significantly affects the speed of obtaining the optimal solution. The fastest solver is Gurobi, launched from R Studio. At the same time, the average median time when calling the CPLEX and Gurobi solvers from MATLAB is less for CPLEX.

3.3. Quantile Optimization Problem for a Polyhedral Loss Function and Discrete Distribution of Random Parameters

Now consider the quantile optimization problem for a polyhedral loss function and a discrete distribution of random parameters from [22]. As for the previously discussed

problems, it is potentially possible to write the matrix A_1 and the vector b_1 in general form. However, this kind of notation will require a lot of additional explanations, so this type of matrix A_1 and vector b_1 are omitted from the article. The given vector and matrix for the data in example in [22] can be found in [28]. At the same time in contrast to the single level of confidence probability $\alpha = 0,7$, used in [22], we will examine confidence levels from 0,5 to 0,9 in increments of 0,1. Thus, in the future, 5 matrices A_1 are considered and $N = 5$ is obtained. Moreover, these matrices differ from each other in only one row.

According to [22] in the notation of this article $A_2 = (0, 0, \dots, 0)$, $b_2 = 0$. In addition, the set \mathcal{I} consists of the numbers 11, 12, ..., 18, $l = \text{col}(\underbrace{-\infty, \dots, -\infty}_{10 \text{ items}}, \underbrace{0, \dots, 0}_{8 \text{ pieces}}, -\infty)$, $r = \text{col}(\underbrace{+\infty, \dots, +\infty}_{10 \text{ items}}, \underbrace{1, \dots, 1}_{8 \text{ items}}, \underbrace{+\infty}_{18 \text{ items}})$. Vector $c = \text{col}(\underbrace{0, \dots, 0}_{18 \text{ items}}, 1)$.

Let us do 11 repetitions of solving the same mathematical programming problem with each solver, i.e. $R = 11$. Let us fix $\varepsilon = 10^{-5}$. We will write the results of the experiments in the Table 4.

Table 4

Comparison of the performance of solvers for the quantile optimization problem for a polyhedral loss function and a discrete distribution of random parameters

Solver	CPLEX IDE	CPLEX MATLAB	CPLEX OPTI	SCIP OPTI	MATLAB	Gurobi MATLAB	Gurobi R
Minimum median operating time, [s]	1,539	0,634	0,686	0,66	0,695	0,667	0,022
Average median operating time, [s]	1,798	0,65	0,691	0,703	0,698	0,671	0,026
Maximum median operating time, [s]	1,975	0,659	0,701	0,729	0,703	0,673	0,027
Standard deviation of median operating time, [s]	0,16	0,01	0,006	0,027	0,003	0,003	0,002
Number of times with the best value of criterion function, [units]	55	55	55	55	55	55	55

The best results for all studied characteristics, according to Table 4, were again demonstrated by the Gurobi solver, called from the R environment. The next in terms of average median running time is the CPLEX solver, called directly from MATLAB. Moreover, all solvers called from MATLAB for the problem under study give comparable results. At the same time, the time characteristics obtained when calculating using the CPLEX IDE environment are noticeably worse.

3.4. Bilevel Production Planning Problem

Next, consider the bilevel production planning problem from [21]. We will again vary the level α from 0,5 to 0,9 in increments of 0,1 (i.e. $N = 5$). The matrix A_1 and the vector b_1 can be found in [28]. Also $A_2 = (0, 0, \dots, 0)$, $b_2 = 0$. The set \mathcal{I} consists of the numbers 151, 152, ..., 310, $l = \text{col}(\underbrace{0, \dots, 0}_{312 \text{ items}}, -\infty)$, $r = \text{col}(\underbrace{+\infty, \dots, +\infty}_{150 \text{ items}}, \underbrace{1, \dots, 1}_{160 \text{ items}}, +\infty, +\infty, +\infty)$.

Vector $c = \text{col}(\underbrace{0, \dots, 0}_{312 \text{ items}}, 1)$.

Let us do 11 repetitions of solving the mathematical programming problem on the same input data with each solver, i.e. $R = 11$. We will write the results of the experiments in the Table 5. Let us set $\varepsilon = 10^{-5}$.

Table 5

Comparison of solver performance for the bilevel production planning problem

Solver \ Characteristics	CPLEX IDE	CPLEX MATLAB	CPLEX OPTI	SCIP OPTI	MATLAB	Gurobi MATLAB	Gurobi R
Minimum median operating time, [s]	2,45	1,014	1,045	1,607	60,48	1,056	0,148
Average median operating time, [s]	2,657	1,024	1,054	1,697	1426,1	1,066	0,158
Maximum median operating time, [s]	2,745	1,03	1,061	1,794	2491,4	1,077	0,173
Standard deviation of median operating time, [s]	0,127	0,009	0,008	0,077	1138,8	0,008	0,011
Number of times with the best value criterion function, [units]	55	55	55	55	33	55	55

If for the portfolio optimization problem the CPLEX solver, called through the Opti Toolbox add-on, did not always find the optimal solution, then for the production planning problem the built-in MATLAB solver has this property. When comparing the CPLEX and Gurobi solvers called directly from MATLAB, the CPLEX solver wins quite a bit in terms of average median running time.

Analyzing the results in Tables 1 – 5, we come to the conclusion that the built-in MATLAB solver and the SCIP solver are inferior to the CPLEX and Gurobi solvers. Additionally, using the CPLEX solver in conjunction with the Opti Toolbox add-on does not always result in an optimal solution.

4. Detailed Runtime Information for Some Solvers

As can be seen from Tables 1 – 5, solving the same problems with the same solvers called from different environments does not take the same time. In this regard, we will take a closer look at the time it takes to find a solution for some solvers and some environments. To do this, we select the problem of adjusting the orbit of the satellite for $K = 1900$, $i = 1$, $R = 21$, use the Gurobi, CPLEX solvers and calculate

- median time for loading data from files;
- median time of constraint formation;
- median time of searching for a solution by the solver himself.

The calculation results will be entered into the Table 6.

First, we note that in CPLEX IDE, when reading Excel files not in the main method, it is not possible to divide the time between actually reading the data and generating a model for subsequent solution. If the file with the source data is stored in .csv format and the file is read in the main method, then the preparation time for solving the problem is significantly reduced. These times are recorded in the CPLEX IDE* column. Moreover, regardless of the environment from which the CPLEX solver is called, the time required to solve the actual mathematical programming problem is approximately the same. The

Table 6

Comparison of the work of solvers in the correction of the scalar terminal state of the satellite problem

Solver	CPLEX IDE	CPLEX IDE*	CPLEX MATLAB	Gurobi MATLAB	Gurobi R
Median time to load data from a file	0,789	0,03	0,333	0,333	0,01
Median time of constraints formation		0,01	0,064	0,173	0,022
Median time to search for a solution by solver itself	1,31	1,4	1,139	0,238	0,244

same property is observed for the Gurobi solver, called from MATLAB and from R Studio. At the same time, the time for reading data and generating a model in R Studio is much faster than in MATLAB.

Comparing the operating time of the solvers themselves, we conclude that Gurobi is more efficient than CPLEX. However, we repeat, the comparison is not entirely “fair”, because the version of the Gurobi solver is more modern than the version of the CPLEX solver used.

Conclusion

In this work, a study of the operating time of various solvers was carried out for a number of practical problems of mathematical programming. It was found that even a small change in input data can lead to a significant decrease/increase in operating time. For all practical problems, it turned out that the Gurobi 9.5.2 solver is faster than the CPLEX 12.5.1 solver in terms of average median running time. A study was conducted of the running time components of solvers called from various software environments. As part of this study, it was revealed that a significant part of the work time can be caused by reading the initial data from the file, and not by the work of the solver itself.

Acknowledgments. *This work was supported by the Russian Science Foundation, project no. 23-21-00293.*

References

1. Ignatov A.N. On the Scheduling Problem of Cargo Transportation on a Railway Network Segment and Algorithms for its Solution. *Bulletin of the South Ural State University. Series: Mathematical Modelling, Programming and Computer Software*, 2021, vol. 14, no. 3, pp. 61–76. DOI: 10.14529/mmp210305
2. Ignatov A.N. On the Algorithm of Cargoes Transportation Scheduling in the Transport Network. *Automation and Remote Control*, 2023, vol. 84, no. 9, pp. 993–1004. DOI: 10.1134/S0005117923090096
3. Ignatov A.N., Naumov A.V. On the Problem of Increasing the Railway Station Capacity. *Automation and Remote Control*, 2021, vol. 82, no. 1, pp. 102–114. DOI: 10.1134/S0005117921010070

4. Ignatov A.N., Naumov A.V. On Time Selection for Track Possession Assignment at the Railway Station. *Bulletin of the South Ural State University. Series: Mathematical Modelling, Programming and Computer Software*, 2019, vol. 12, no. 3, pp. 5–16. DOI: 10.14529/mmp190301
5. Ye Yixin, Liang Shengming, Zhu Yushan. A Mixed-Integer Linear Programming-Based Scheduling Model for Refined-Oil Shipping. *Computers and Chemical Engineering*, 2017, vol. 99, pp. 106–116. DOI: 10.1016/j.compchemeng.2017.01.031
6. Flores-Luyo L., Agra A., Figueiredo R., Ocana E. Mixed Integer Formulations for a Routing Problem with Information Collection in Wireless Networks. *European Journal of Operational Research*, 2020, vol. 280, no. 2, pp. 621–638. DOI: doi.org/10.1016/j.ejor.2019.06.054
7. Yang Xiao, Bostel N., Dejax P. A MILP Model and Memetic Algorithm for the Hub Location and Routing Problem with Distinct Collection and Delivery Tours. *Computers and Industrial Engineering*, 2019, vol. 135, pp. 105–119. DOI: 10.1016/j.cie.2019.05.038
8. Benati S., Rizzi R. A Mixed Integer Linear Programming Formulation of the Optimal mean/Value-at-Risk Portfolio Problem. *European Journal of Operational Research*, 2007, vol. 176, no. 1, pp. 423–434. DOI: 10.1016/j.ejor.2005.07.020
9. Kibzun A.I., Ignatov A.N. Reduction of the Two-Step Problem of Stochastic Optimal Control with Bilinear Model to the Problem of Mixed Integer Linear Programming. *Automation and Remote Control*, 2016, vol. 77, no. 12, pp. 2175–2192. DOI: 10.1134/S0005117916120079
10. Ivanov S.V., Akmaeva V.N. Two-Stage Stochastic Facility Location Model with Quantile Criterion and Choosing Reliability Level. *Bulletin of the South Ural State University. Series: Mathematical Modelling, Programming and Computer Software*, 2021, vol. 14, no. 3, pp. 5–17. DOI: 10.14529/mmp210301
11. Albareda-Sambola M., Fernandez E., Hinojosa Y., Puerto J. The Multi-Period Incremental Service Facility Location Problem. *Computers and Operations Research*, 2009, vol. 36, no. 5, pp. 1356–1375. DOI: 10.1016/j.cor.2008.02.010
12. Ignatov A.N. On the Resource Allocation Problem to Increase Reliability of Transport Systems. *Lecture Notes in Computer Science*, 2023, vol. 13930, pp. 169–180. DOI: 10.1007/978-3-031-35305-5_11
13. Omu A., Choudhary R., Boies A. Distributed Energy Resource System Optimisation Using Mixed Integer Linear Programming. *Energy Policy*, 2013, vol. 61, pp. 249–266. DOI: 10.1016/j.enpol.2013.05.009
14. Knueven B., Ostrowski J., Watson J.-P. On Mixed-Integer Programming Formulations for the Unit Commitment Problem. *INFORMS Journal on Computing*, 2020, vol. 32, no. 4, pp. 857–876. DOI: 10.1287/ijoc.2019.0944
15. Veintimilla-Reyes J., Cattrysse D. Mixed Integer Linear Programming (MILP) Approach to Deal with Spatio-temporal Water Allocation. *Procedia Engineering*, 2016, vol. 162, pp. 221–229. DOI: 10.1016/j.proeng.2016.11.045
16. Anand R., Aggarwal D., Kumar V. A Comparative Analysis of Optimization Solvers. *Journal of Statistics and Management Systems*, 2017, vol. 20, no. 4, pp. 623–635. DOI: 10.1080/09720510.2017.1395182
17. Kronqvist J., Bernal Neira D.E., Lundell A., Grossmann I.E. A Review and Comparison of Solvers for Convex MINLP. *Optimization and Engineering*, 2019, vol. 20, no. 4, pp. 397–455. DOI: 10.1007/s11081-018-9411-8
18. Koch T., Berthold T., Pedersen J., Vanaret Ch. Progress in Mathematical Programming Solvers from 2001 to 2020. *EURO Journal on Computational Optimization*, 2022, vol. 10, no. 2, 32 p. DOI: 10.1016/j.ejco.2022.100031

19. Gleixner A., Hendel G., et al. MIPLIB 2017: Data-Driven Compilation of the 6th Mixed-Integer Programming Library. *Mathematical Programming Computation*, 2021, vol. 13, pp. 443–490. DOI: 10.1007/s12532-020-00194-3
20. Ignatov A.N. On Solution of the Problem of Correcting Scalar Terminal State of an Aircraft for an Arbitrary Distribution of a Multiplicative Perturbation. *Trudy MAI*, 2016, no. 87, 19 p. (in Russian)
21. Dempe S., Ivanov S., Naumov A. Reduction of the Bilevel Stochastic Optimization Problem with Quantile Objective Function to a Mixed-Integer Problem. *Applied Stochastic Models in Business and Industry*, 2017, vol. 33, no. 5, pp. 544–554. DOI: 10.1002/asmb.2254
22. Ivanov S.V., Naumov A.V. Algorithm to Optimize the Quantile Criterion for the Polyhedral Loss Function and Discrete Distribution of Random Parameters. *Automation and Remote Control*, 2012, vol. 73, pp. 105–117. DOI: 10.1134/S0005117912010080
23. Borisov A., Ivanov A. Stochastic Time Complexity Surfaces of Computing Node. *Mathematics*, 2023, vol. 20, no. 11, pp. 43–79. DOI: 10.3390/math11204379
24. MacLean L.C., Thorp E.O., Zhao Yonggan, Ziemba W. How Does the Fortune’s Formula Kelly Capital Growth Model Perform? *The Journal of Portfolio Management Summer*, 2011, vol. 37, no. 4, pp. 96–111. DOI: 10.3905/jpm.2011.37.4.096
25. Ziemba W.T., Wickson R.G. *Stochastic Optimization Models in Finance*. Singapore, World Scientific, 2006.
26. Alexander G.J., Baptista A.M. Portfolio Performance Evaluation Using Value at Risk. *The Journal of Portfolio Management Summer*, 2003, vol. 29, no. 4, pp. 93–102. DOI: 10.3905/jpm.2003.319898
27. Ignatov A.N. On the Construction of Positional Control in a Multistep Portfolio Optimization Problem with Probabilistic Criterion. *Automation and Remote Control*, 2020, vol. 81, pp. 2181–2193. DOI: 10.1134/S0005117920120036
28. SolversMILP. Available at: https://github.com/al-ignatov/SolversMILP_comparison (accessed on 02.05.2024)

Received May 17, 2024

УДК 004.4+51-3

DOI: 10.14529/mmp240305

СРАВНЕНИЕ РЕШАТЕЛЕЙ ЗАДАЧ СМЕШАННОГО ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ И ВЫЗЫВАЮЩИХ ИХ ПРОГРАММНЫХ СРЕД

А.Н. Игнатов¹, С.В. Иванов¹

¹Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация

В работе приводится концепция сравнения решателей задач смешанного целочисленного линейного программирования и вызывающих их программных сред. Эта концепция предполагает многократное повторение решения задач математического программирования с одними и теми же исходными данными для учета того, что время

выполнения операций компьютером можно рассматривать как случайное. Для сравнения решателей также предполагается варьировать исходные данные при решении задачи математического программирования той же структуры. Сравнение проводится для ряда практических задач математического программирования. Например, рассматривается задача оптимизации портфеля ценных бумаг с вероятностным критерием. В тестировании используются решатели CPLEX, Gurobi, MATLAB, SCIP. В работе разбираются особенности вызова решателей в различных программных средах. В частности, описывается модификация исходных кодов для вызова решателя CPLEX через надстройку Opti Toolbox в среде Matlab. Детально описываются и исследуются компоненты времени получения решения для различных решателей и программных сред. Показывается, что время работы самого решателя может быть сравнимо со временем чтения данных из файлов и временем формирования ограничений в задаче математического программирования.

Ключевые слова: смешанное целочисленное линейное программирование; решатель; сравнение; программная среда.

Литература

1. Игнатов, А.Н. О задаче составления расписания грузоперевозок на участке железнодорожной сети и алгоритмах ее решения / А.Н. Игнатов // Вестник ЮУрГУ. Серия: Математическое моделирование и программирование. – 2021. – Т. 14, № 3. – С. 61–76.
2. Ignatov, A.N. On the Algorithm of Cargoes Transportation Scheduling in the Transport Network / A.N. Ignatov // Automation and Remote Control. – 2023. – V. 84, № 9. – P. 993–1004.
3. Ignatov, A.N. On the Problem of Increasing the Railway Station Capacity / A.N. Ignatov, A.V. Naumov // Automation and Remote Control. – 2021. – V. 82, № 1. – P. 102–114.
4. Игнатов, А.Н. О выборе времени для установления «технологического окна» на железнодорожной станции / А.Н. Игнатов, А.В. Наумов // Вестник ЮУрГУ. Серия: Математическое моделирование и программирование. – 2019. – Т. 12, № 3. – С. 5–16.
5. Ye, Yixin. A Mixed-Integer Linear Programming-Based Scheduling Model for Refined-Oil Shipping / Yixin Ye, Shengming Liang, Yushan Zhu // Computers and Chemical Engineering. – 2017. – V. 99. – P. 106–116.
6. Flores-Luyo, L. Mixed Integer Formulations for a Routing Problem with Information Collection in Wireless Networks / L. Flores-Luyo, A. Agra, R. Figueiredo, E. Ocana // European Journal of Operational Research. – 2020. – V. 280, № 2. – P. 621–638.
7. Xiao, Yang. A MILP Model and Memetic Algorithm for the Hub Location and Routing Problem with Distinct Collection and Delivery Tours / Xiao Yang, N. Bostel, P. Dejax // Computers and Industrial Engineering. – 2019. – V. 135. – P. 105–119.
8. Benati, S. A Mixed Integer Linear Programming Formulation of the Optimal Mean/Value-at-Risk Portfolio Problem / S. Benati, R. Rizzi // European Journal of Operational Research. – 2007. – V. 176, № 1. – P. 423–434.
9. Kibzun, A.I. Reduction of the Two-Step Problem of Stochastic Optimal Control with Bilinear Model to the Problem of Mixed Integer Linear Programming / A.I. Kibzun, A.N. Ignatov // Automation and Remote Control. – 2016. – V. 77, № 12. – P. 2175–2192.
10. Иванов, С.В. Двухэтапная стохастическая модель размещения предприятий с квантильным критерием и выбором уровня надежности / С.В. Иванов, В.Н. Акмаева // Вестник ЮУрГУ. Серия: Математическое моделирование и программирование. – 2021. – V. 14, № 3. – P. 5–17.

11. Albareda-Sambola, M. The Multi-Period Incremental Service Facility Location Problem / M. Albareda-Sambola, E. Fernandez, Y. Hinojosa, J. Puerto // *Computers and Operations Research*. – 2009. – V. 36, № 5. – P. 1356–1375.
12. Ignatov, A.N. On the Resource Allocation Problem to Increase Reliability of Transport Systems / A.N. Ignatov // *Lecture Notes in Computer Science*. – 2023. – V. 13930. – P. 169–180.
13. Omu, A. Distributed Energy Resource System Optimisation Using Mixed Integer Linear Programming / A. Omu, R. Choudhary, A. Boies // *Energy Policy*. – 2013. – V. 61. – P. 249–266.
14. Knueven, B. On Mixed-Integer Programming Formulations for the Unit Commitment Problem / B. Knueven, J. Ostrowski, J.-P. Watson // *INFORMS Journal on Computing*. – 2020. – V. 32, № 4. – P. 857–876.
15. Veintimilla-Reyes, J. Mixed Integer Linear Programming (MILP) Approach to Deal with Spatio-Temporal Water Allocation / J. Veintimilla-Reyes, D. Cattrysse // *Procedia Engineering*. – 2016. – V. 162. – P. 221–229.
16. Anand, R. A Comparative Analysis of Optimization Solvers / R. Anand, D. Aggarwal, V. Kumar // *Journal of Statistics and Management Systems*. – 2017. – V. 20, № 4. – P. 623–635.
17. Kronqvist, J. A Review and Comparison of Solvers for Convex MINLP / J. Kronqvist, D. Bernal Neira, A. Lundell, I.E. Grossmann // *Optimization and Engineering*. – 2019. – V. 20. – P. 397–455.
18. Koch, T. Progress in Mathematical Programming Solvers from 2001 to 2020 / T. Koch, T. Berthold, J. Pedersen, Ch. Vanaret // *EURO Journal on Computational Optimization*. – 2022. – V. 10, № 2. – 32 p.
19. Gleixner, A. MIPLIB 2017: Data-Driven Compilation of the 6th Mixed-Integer Programming Library / A. Gleixner, G. Hendel, et al. // *Mathematical Programming Computation*. – 2021. – V. 13. – P. 443–490.
20. Игнатов, А.Н. О решении задачи корректирования скалярного терминального состояния летательного аппарата при произвольном распределении мультипликативного возмущения / А.Н. Игнатов // *Труды МАИ*. – 2016. – № 87. – 19 с.
21. Dempe, S. Reduction of the Bilevel Stochastic Optimization Problem with Quantile Objective Function to a Mixed-Integer Problem / S. Dempe, S. Ivanov, A. Naumov // *Applied Stochastic Models in Business and Industry*. – 2017. – V. 33, № 5. – P. 544–554.
22. Ivanov S.V. Algorithm to Optimize the Quantile Criterion for the Polyhedral Loss Function and Discrete Distribution of Random Parameters / S.V. Ivanov, A.V. Naumov // *Automation and Remote Control*. – 2012. – V. 73. – P. 105–117.
23. Borisov, A. Stochastic Time Complexity Surfaces of Computing Node / A. Borisov, A. Ivanov // *Mathematics*. – 2023. – V. 20, № 11. – P. 43–79.
24. MacLean, L.C. How does the fortune’s formula Kelly capital growth model perform? / L.C. MacLean, E.O. Thorp, Yonggan Zhao, W. Ziemba // *The Journal of Portfolio Management Summer*. – 2011. – V. 37, № 4. – P. 96–111.
25. Ziemba, W.T. *Stochastic Optimization Models in Finance* / W.T. Ziemba, R.G. Wickson. – Singapore: World Scientific, 2006.
26. Alexander, G.J. Portfolio Performance Evaluation Using Value at Risk / G.J. Alexander, A.M. Baptista // *The Journal of Portfolio Management Summer*. – 2003. – V. 29, № 4. – P. 93–102.

27. Ignatov, A.N. On the Construction of Positional Control in a Multistep Portfolio Optimization Problem with Probabilistic Criterion / A.N. Ignatov // Automation and Remote Control. – 2020. – V. 81, № 12. – P. 2181–2193.
28. SolversMILP. – URL: https://github.com/al-ignatov/SolversMILP_comparison (дата обращения 02.05.2024)

Алексей Николаевич Игнатов, кандидат физико-математических наук, кафедра «Теория вероятностей и компьютерное моделирование», Московский авиационный институт (национальный исследовательский университет) (г. Москва, Российская Федерация), alexei.ignatov1@gmail.com.

Сергей Валерьевич Иванов, доктор физико-математических наук, кафедра «Теория вероятностей и компьютерное моделирование», Московский авиационный институт (национальный исследовательский университет) (г. Москва, Российская Федерация), sergeyivanov89@mail.ru.

Поступила в редакцию 17 мая 2024 г.