Journal of Science and Technology, Vol. 10 No. 4 (2018) p. 28-32

Creating a GUI Solver for Linear Programming Models in MATLAB

Lee Siaw Chong*, Chin Jia Xin

Faculty of Applied Sciences and Technology, Universiti Tun Hussein Onn Malaysia,
Pagoh Edu Hub, 84600, Muar, Johor, Malaysia

Received 15 September 2018; accepted 1 December 2018; available online 30 December 2018
DOI: https://10.30880/jst.2018.10.04.005

Abstract: The concept of linear programming (LP) was developed to find out the best solution among all feasible solutions in an optimization problem. This technique becomes much popular and attains great attention from researchers due to its wide application in engineering, computer science, marketing, military and industries. Nowadays, there are many commercial software that apply this technique to solve optimization problems, e.g. Excel Solver, TORA, AMPL, LINGO and MATLAB. In this paper, we aim to use MATLAB to develop a Graphical User Interface (GUI) solver for LPs, namely LpSolver. The LP methods that will be included in our solver are the simplex method, the Big-M method, the Two-Phase method and the Dual-Simplex method. We try to make our solver perform calculations in symbolic form so that the result will be free from rounding errors. Besides that, we added a few features such as creating animated 2-D graphs and generating a detailed tableau showing all intermediate iterative results; in which the user can use it to trace the convergence path that leads to the optimal solution. In the later part of this paper we test our solver with a simple classroom sized problem.

Keywords: Linear Programming; Simplex method; MATLAB.

1.0 Introduction

Linear programming (LP) is a one of the tool that solves optimization problems. It is a specific class of mathematical method, in which a linear function is maximized (or minimized) subject to some linear constraints. LP and its many forms have come into wide use since it was first proposed in 1947 by Dantzig [1]. It has been used to solve optimization problem in various industries such as banking, education, forestry, petroleum, and trucking since the development of the simplex algorithm.

LP become popular in academic circle, mainly for decision scientists (operations researchers and management scientists), as well as numerical analysts, mathematicians, and economists. Besides that, many industries use LP as a standard tool to allocate a finite set of resources in an optimal way. LP is also broad enough to encompass many interesting and important applications such as airline crew scheduling, shipping or telecommunication networks, oil refining and blending, and stock and bond portfolio selection [2].

Nowadays LP is used to solve large scale complex problem that requires a lot of computational effort. In fact, the rapid development of computer technology becomes the catalyst of the blooming of commercial LP solvers [3]. There are many commercial software packages to solve LP models such as Excel Solver, AMPL, LINGO, TORA, and the Optimization Toolbox in MATLAB [4]. However, most of the commercial packages are designed to solve large scale problems and lack of some features for classroom use.

This paper is aimed to develop a Graphical User Interface (GUI) solver for LPs using MATLAB namely LpSolver. This solver is specific for solving classroom sized problems and able to perform computation in symbolic form (the other applications can only perform calculations in numeric form). Symbolic computation includes computing expressions with symbolic variables and fractions. Our solver is also able to generate 2-D graph that allows the user to trace the optimizing process that leads to the optimal solution. This is particularly useful since certain method such as the dual simplex method has different converging path from the original simplex method. Graphical output helps user to understand the concepts behind each LP method base on its convergence path.

LP models itself exists in different classes such as LP with different types of constraints (equality and inequality) and right-hand side values (feasible and infeasible). Each class may require a specific solving method and the

*Corresponding author: sclee@uthm.edu.my

e-ISSN: 2600-7924/penerbit.uthm.edu.my/ojs/index.php/jst

methods that will be included in this paper are the simplex method, the Big-M method, the Two-Phase method and the Dual-Simplex method.

In this paper we consider the lack of features of the commercial solvers mentioned and included those features in our GUI solver. Table 1 shows the comparison of the features among the solvers and the our LpSolver. It can be seen that only the TORA solver and LpSolver has graphical output option and able to show the result by iteration whereas the other solvers do not have this option.

Table 1 Features for each solver.

Features	**Excel**	**TORA**	**AMPL**	**LINGO**	**MATLAB**	**LpSolver**
(1)	-	√	-	-	-	√
(2)	-	-	-	-	-	√
(3)	-	√	-	-	-	√
(4)	-	-	√	√	√	-
(5)	√	√	-	-	-	√

Note: (1) Graphical output, (2) Symbolic computation, (3) Result by iteration, (4) Command Line interface, (5) Mouse interface

2.0 Linear Programming Techniques

Here we show a typical LP that will be later used to test our solver. Model 1 is a simple LP that commonly used as an example in the first course of linear programming where there are only two independent variables involved, x_1 and x_2. All constraints are of '$\leq$' type but the LP techniques that we are going to discuss in this section can also cope with constraints of '$\geq$' or '$=$' type.

Model 1:

Objective function:

Maximize $\quad z = 5x_1 + 4x_2$

Subject to $\quad 6x_1 + 4x_2 \leq 24$

$$x_1 + 2x_2 \leq 6$$

$$-x_1 + x_2 \leq 1$$

$$x_2 \leq 2$$

In Model 1, z is the objective function where we are interested to know its maximum value subject to the constraint stated. In Sections 2.1-2.4, we explain the solving procedure of LP with the Simplex method, the Big-M method, the Two-Phase method, and the Dual Simplex method.

2.1 Simplex Method

In mathematical optimization, the simplex method is a popular algorithm for LP which was developed by George B. Dantzig in 1947 [5]. Since many real-life problems are so large that makes hand calculation impractical, the need of an algorithm that systematically solves the problem becomes important. The simplex method, which is easily programmable suits this need [6]. The method itself is an iterative process that begins with an initial feasible solution, then repeatedly moves to a better solution. Finally, it terminates when an optimal solution has been found.

The simplex algorithm proceeds by performing successive Gauss-Jordan row operations which each give an improved basic feasible solution. A basic feasible solution is judged as an improvement if it increases (decreases) the value of the objective function in a maximizing (minimizing) problem compare to the previous objective function value. The choice of the pivot element at each step is determined by the optimality condition (entering variable selection) to improve the solution and feasibility condition (leaving variable selection) that used to ensure the feasible solution do not move out from the feasible region [4].

Optimality condition is used to obtain the entering variable. The entering variable in a maximization (minimization) problem is the non-basic variable with the most negative (positive) coefficient in the objective function. Ties are broken arbitrarily. The optimum is reached at the iteration where all the objective function coefficients are nonnegative (nonpositive).

Feasibility condition is required to find the leaving variable. For both the maximization and minimization problems, the leaving variable is the basic variable associated with the smallest nonnegative ratio which divide the non-basic variable with the corresponding right-hand-side value with strictly positive denominator. Similar to optimality condition, ties are broken arbitrarily.

Gauss-Jordan row operations identifies the entering variable column as the pivot column and the leaving variable row as the pivot row. The intersection of the pivot column and the pivot row is called the pivot element. In [4], the Gauss-Jordan computations uses two types of elementary row operation to compute a new basic solution for the;

1. pivot row:
    a. Replace the leaving variable in the basic column with the entering variable.
    b. New pivot row = Current pivot row ÷ Pivot element
2. all other rows, including the objective function

    New row = (Current row) – (Its pivot column coefficient) × (New pivot row).

The algorithm of the simplex method is as follows:

Step 0: Determine a starting basic feasible solution.

Step 1: Is there is an entering variable that can improve the current objective value (optimality condition)? Stop if there is no entering variable. Else, go to step 2.

Step 2: Select a leaving variable using the feasibility condition.

Step 3: Apply the Gauss-Jordan computations to determine the new basic solution. Go to step 1.

2.2 Big-M Method

The simplex algorithm discussed in the previous section is restricted to solve LP's with '$\leq$' constraints and nonnegative right-hand-side values only. To include the '$\geq$' and '$=$' constraints, the simplex method can be extended to the Big-M method by adding artificial variables which play the role of slacks at the first iteration and would not take part of any optimal solution [4]. The 'Big-M' refers to a large number associated with the artificial variables which represented by the letter M.

The steps in the algorithm are as follows:

Step 1: Modify the constraints so that the right-hand-side of each constraint is positive.

Step 2: Convert each inequality constraint to standard form. (Add slack variable S_i for constraint which contain '$\leq$' and for constraint that contain '$\geq$', add a surplus variable '$-S_i$').

Step 3: For any '$\geq$' and '$=$' constraints, introduce artificial variables 'R_i'.

Step 4: Let M be a very large positive number. If the LP problem is minimization (maximization), for each artificial variable add MR_i $(-MR_i)$ to the objective function.

Step 5: To ensure that we are begin with a canonical form, all artificial variables must be eliminated from the beginning objective function row by replacing the artificial variable R_i with corresponding R_i-row.

Step 6: Solve the problem using the usual simplex method [7].

The value of M must be chosen sufficiently large so that the artificial variable would not be part of any feasible solution. For a sufficiently large M, the optimal solution contains any artificial variables in the basis if and only if the problem is not feasible.

2.3 Two-Phase Method

The Two-Phase method is an alternative to the Big-M method for solving problems with '$\geq$' and '$=$' constraints. It introduces artificial variables to the same constraints as in the Big-M method. The Two-Phase method solve the LP problem within Two-Phases. Phase I is attempts to find a starting feasible solution and Phase II is responsible to solve the original problem when there has a starting basic feasible solution found in Phase I [7].

The Two-Phase simplex method proceeds as follows:

Step 1: Bring the constraints into equality form by adding the necessary slack variables and artificial variables to the constraints (same as the step 2 and 3 of the Big-M method).

Step 2: Phase I: find a basic feasible solution that always minimize the sum of the artificial variables, regardless of whether the LP is maximization or minimization.

Step 3: If some artificial variable has a positive value in the optimal solution, the original problem is infeasible; stop.

Step 4: Phase II: solve the original problem, starting from the basic feasible solution found in phase I. Apply the ordinary simplex method to obtain an optimal solution.

As with the Big-M method, the column for any artificial variable may be dropped from future tableaus as soon as the artificial variable leaves the basis. The Big-M method and Phase I of the Two-Phase method make the same sequence of pivots.

2.4 Dual-Simplex Method

For the primal simplex algorithm, we always concerned the starting basis are primal feasible (right-hand side are all nonnegative) from the beginning until the final iteration but dual infeasible (non-optimal of objective function), which means we aim to achieve dual feasible solution. In other words, we maintain primal feasibility and drive toward dual feasibility throughout the process. Meanwhile, dual simplex algorithm works in just opposite fashion. The starting basis are now dual feasible from the beginning until the final iteration but primal infeasible, and we aim to achieve primal feasible solution. Throughout the process we maintain dual feasibility and drive toward primal feasibility [8].

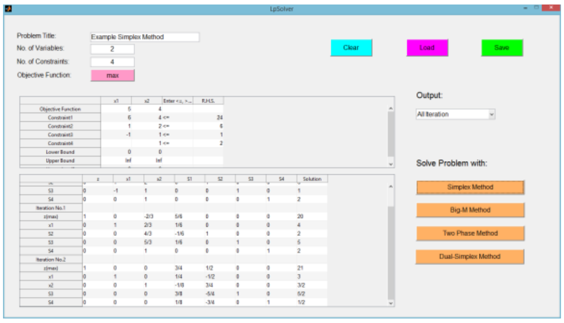
Dual feasibility condition is used to obtain the leaving variable. The leaving variable is the basic variable with the most negative right-hand-side value. Ties are broken arbitrarily. The optimum is reached at the iteration where all the basic variables are nonnegative.

Dual optimality condition is required to find the entering variable. The row with the leaving variable will be the pivot row. The entering variable is the smallest nonnegative ratio of coefficient of x_j in z-row divided with coefficient x_j in pivot row. Similar to feasibility condition, ties are broken arbitrarily. If all the ratios are negative, the problem has no feasible solution.

The two requirements need to considered before starting are the optimality and infeasibility of the LP. The objective function must satisfy the optimality condition of the regular simplex method and all the constraints only can be contain '$\leq$'. Inequalities of the type '$\geq$' are converted to '$\leq$' by multiplying both sides of the inequality by -1. If the equation contains '$=$' constraint, it can be replaced by two inequalities which are basic variables $\leq$ right-hand side and basic variables $\geq$ right-hand side. The starting solution must be infeasible if at least one of the right-hand-side value of inequalities is negative [4].

3.0 Results and Discussions

In this paper, a solver, namely LpSolver that solves the LP models in MATLAB has been created. The paper will only focus on the problem which the decision variables are nonnegative. Fig. 1 shows the sample output of the Model 1.



Problem Title: Example Simplex Method
No. of Variables: 2
No. of Constraints: 4
Objective Function: max

Output: All Iteration

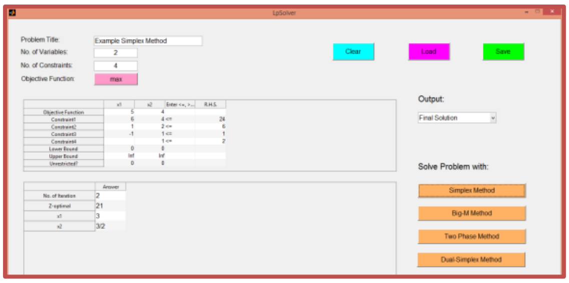
Solve Problem with:

	x1	x2	Enter <=, >...	R.H.S.
Objective Function	5	4		
Constraint1	6	4	<=	24
Constraint2	1	2	<=	6
Constraint3	-1	1	<=	1
Constraint4		1	<=	2
Lower Bound	0	0		
Upper Bound	Inf	Inf		

	z	x1	x2	S1	S2	S3	S4	Solution
S3	0	-1	1	0	0	1	0	1
S4	0	0	1	0	0	0	1	2
Iteration No.1								
z(max)	1	0	-2/3	5/6	0	0	0	20
x1	0	1	2/3	1/6	0	0	0	4
S2	0	0	4/3	-1/6	1	0	0	2
S3	0	0	5/3	1/6	0	1	0	5
S4	0	0	1	0	0	0	1	2
Iteration No.2								
z(max)	1	0	0	3/4	1/2	0	0	21
x1	0	1	0	1/4	-1/2	0	0	3
x2	0	0	1	-1/8	3/4	0	0	3/2
S3	0	0	0	3/8	-5/4	1	0	5/2
S4	0	0	0	1/8	-3/4	0	1	1/2

Fig. 1 All Iteration result for simplex method.

Fig. 2 shows that when the user choose Final Solution as the output option, the table displays the final results only, i.e. the optimal solution (optimal z and decision variables $x_1$and x_2) along with the number of iterations. All intermediate iterative results are excluded.



Problem Title: Example Simplex Method
No. of Variables: 2
No. of Constraints: 4
Objective Function: max

Output: Final Solution

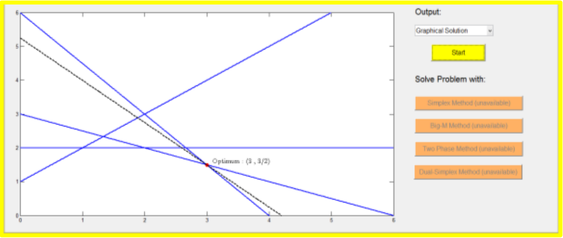
Solve Problem with:

	x1	x2	Enter <=, >...	R.H.S.
Objective Function	5	4		
Constraint1	6	4	<=	24
Constraint2	1	2	<=	6
Constraint3	-1	1	<=	1
Constraint4		1	<=	2
Lower Bound	0	0		
Upper Bound	Inf	Inf		
Unrestricted?	0	0		

	Answer
No. of Iteration	2
Z-optimal	21
x1	3
x2	3/2

Fig. 2 Final Solution output for simplex method.

If Graphical Solution is selected, a Start button will appear under the pop-up menu. Clicking the Start button activates the graphical output as in Fig. 3. The Graphical Solution shows how the objective function line gradually moves toward the optimal solution by seeking the intersection between the constraints that has the optimal objective value. However, this option is restricted to LP with two decision variables only.


0
1
2
3
4
5
6
0
1
2
3
4
5
6
Optimum : (3 , 3/2)
Start
Simplex Method (unavailable)
Big-M Method (unavailable)
Two Phase Method (unavailable)
Dual-Simplex Method (unavailable)

Output: Graphical Solution

Solve Problem with:

Fig. 3 Graphical output for Model 1.

4.0 Conclusions

Linear programming (LP) is a branch of optimization methods that finds the best outcome of a certain objective function based on linear constraints. The Simplex algorithm is one of the widely-used LP technique that iteratively improves the objective value by moving around the adjacent points on the boundary of the feasible region. We developed a MATLAB GUI solver for LPs, namely LpSolver that enables user to solve LP problems with a single click and avoids jumbling with convoluted programming codes. Four methods are considered in our solver: Simplex method, Big-M method, Two-Phase method and Dual-Simplex method.

To make our solver handier, we allow it to compute in symbolic form (fractions included) and generates graphical output for 2-D models. LpSolver smartly chooses the suitable solving method if the user picks an unsuitable one. Users are even allowed to create a data file of their own and use the load button to load it into LpSolver rather than input the data sequentially which can be a hassle if the data size is large. However, LpSolver is not recommended to solve large scale problems (> 50 constraints, > 100 variables) since computing in symbolic form can be quite time consuming and the Simplex method itself may converges slowly. To improve the convergence speed, advanced method such as interior point method can be considered to be coded in LpSolver for future work.

Those who are interested to try our solver can send an email to us. We do not charge for the solver but as a token of appreciation we hoped that users can give feedbacks to us to improve our work.

REFERENCES

[1] Lenstra, J., Rinnooy Kan, A., & Schrijver, A. (1991). History of Mathematical Programming: a collection of personal reminiscences. North-Holland, Amsterdam.

[2] Gupta, D. (2015). Strategic Allocation of Resources Using Linear Programming Model with Parametric Analysis. Anchor Academic Publishing, Hamburg.

[3] Hiller, F.S. and Lieberman, G.J. (2010) Introduction to Operations Research. McGraw-Hill, New York.

[4] Taha, H. A. (2011). Operations Research: An Introduction (9th ed.). Pearson, New Jersey.

[5] Murty, K. G. (1983). Linear programming. John Wiley & Sons Inc, New York.

[6] Carter, M. W., & Price, C. C. (2000). Operations Research: A Practical Introduction. CRC Press, Florida.

[7] Winston, W. L. (2004). Operations Research: Applications and Algorithms (4th ed.). Canada: Curt Hindrichs.

[8] Jensen, P. A., & Jonathan, B. F. (2003). Operations research models and methods. John Wiley & Sons Inc, New York.