Journal of Advances and Scholarly Researches in Allied Education
Vol. 15, Issue No. 9, October-2018, ISSN 2230-7540

Linear Programming and Its Related Methods

Priyanka Sharma[1]* Dr. Renu Chaudhary[2]

[1] Research Scholar, Shri Venkateshwara University, Gajraula

Abstract – The subject of Linear Programming enlarges past the Simplex Method calculation, much as Linear Algebra enlarges past Gaussian Elimination, and the hypothesis behind it has enough substance to make study beneficial. This hypothesis serves to demonstrate why the Simplex Method moves ahead as it does, infers substitute methodologies to explaining Lp's, and might be utilized to formally demonstrate that a certain result is an ideal The presentation of simplex subordinates in example seek methods can prompt a noteworthy decrease in the amount of capacity assessments, for the same nature of the last emphasizes.

Keywords: Linear Programming, Function, Vector

---------------------------------X---------------------------------

INTRODUCTION

Linear Programming in a general form is the problem of maximizing a linear function in d variables subject to n linear inequalities. If, in addition, we require all variables to be nonnegative, we have an LP in standard form which can be written as follows.

$$\text{(LP)} \quad \text{maximize} \quad \sum_{j=1}^{d} c_j x_j$$
$$\text{subject to} \quad \sum_{j=1}^{d} a_{ij} x_j \le b_i \; (i=1,\dots,n), \qquad (1.1)$$
$$x_j \ge 0 \qquad (j=1,\dots,d),$$

where the cj, bi and aij are real numbers. By defining

$$x := (x_1,\dots,x_d)^T,$$
$$c := (c_1,\dots,c_d)^T,$$
$$b := (b_1,\dots,b_n)^T,$$
$$A := \begin{pmatrix} a_{11} & \cdots & a_{1d} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nd} \end{pmatrix}$$

this can be written in more compact form as

$$\text{(LP)} \quad \text{maximize} \quad c^T x$$
$$\text{subject to} \quad Ax \le b,$$
$$x \ge 0, \qquad (1.2)$$

where the relations <= and >= hold for vectors of the same length if and only if they hold component wise.

The vector c is called the cost vector of the LP, and the linear function z : x -> c^T x is called the objective function. The vector b is referred to as the the right-hand side of the LP. The inequalities

$$\sum_{j=1}^{d} a_{ij} x_j \le b_i ,$$

for i = 1; : : : : ; n and x_j >= 0, for j = 1; : : : ; d are the constraints of the linear program.

The LP is called feasible if there exists a non-negative vector x' satisfying Ax' <= b such an x' is called a feasible solution; otherwise the program is called infeasible. If there are feasible solutions with arbitrarily large objective function value, the LP is called unbounded; otherwise it is bounded. A linear program which is both feasible and bounded has a unique maximum value cT x' attained at a (not necessarily unique) optimal feasible solution x'. Solving the LP means finding such an optimal solution x' (if it exists). To avoid trivialities we assume that the cost vector and all rows of A are nonzero.

SIMPLEX ALGORITHM

The simplex algorithm starts of by introducing slack variables $x_{d+1,\dots\dots}$, x_{d+n} to transform the inequality system Ax <= b into an equivalent system of equalities and additional nonnegativity constraints on the slack variables. The slack variable x_{d+i} closes the gap between the left-hand side and right-hand side of the i-th constraint,

$$x_{d+i} := b_i - \sum_{j=1}^{d} a_{ij} \, x_j ,$$

for all i = 1...... n. The i-th constraint is then equivalent to

$$x_{d+i} >= 0;$$

and the linear program can be written as

$$\text{(LP)} \quad \text{maximize} \quad \sum_{j=1}^{d} c_j x_j$$

$$\text{subject to} \quad x_{d+i} = -\sum_{j=1}^{d} a_{ij} x_j \ (i = 1, ..., n), \quad (1.3)$$

$$x_j \geq 0 \quad (j = 1, ..., d+n),$$

or in a more compact form as

$$\begin{aligned} \text{(LP)} \quad & \text{maximize} \quad \underline{C}^T x \\ & \text{subject to} \quad \underline{A}x = b, \\ & \qquad\qquad x \geq 0, \quad (1.4) \end{aligned}$$

where A is the n X (d + n) - matrix

$$\underline{A} : (A/E), \qquad (1.5)$$

c is the (d + n) - vector

$$\underline{c} := \begin{pmatrix} c \\ 0 \\ \vdots \\ 0 \end{pmatrix} \qquad (1.6)$$

and x is the (d + n)-vector

$$x = \begin{pmatrix} x_0 \\ x_s \end{pmatrix},$$

where x_O is the vector of original variables, x_S the vector of slack variables.

Together with the objective function, the n equations for the x_{d+i} in (1.3) contain all the information about the LP. Following tradition, we will represent this information in tableau form where the objective function denoted by z is written last and separated from the other equations by a solid line. In this way we obtain the initial tableau for the LP.

$$\begin{aligned} x_{d+1} &= b_1 - a_{11}x_1 - \cdots - a_{1d}x_d \\ x_{d+n} &= b_n - a_{n1}x_1 - \cdots - a_{nd}x_d \\ \hline z &= c_1x_1 \cdots + c_dx_d \end{aligned} \qquad (1.7)$$

The compact form here is

$$\begin{aligned} x_s &= b - Ax_0 \\ \hline z &= c^T x_0 \end{aligned} \qquad (1.8)$$

An example illustrates the process of getting the initial tableau from an LP in standard.

Example 1.1 Consider the problem

$$\begin{aligned} \text{maximize} \quad & x_1 + x_2 \\ \text{subject to} \quad & -x_1 + x_2 \leq 1, \\ & x_1 \leq 3, \\ & x_2 \leq 2, \\ & x_1\ x_2 \geq 0. \qquad (1.9) \end{aligned}$$

After introducing slack variables x_3; x_4; x_5, the LP in equality form is

$$\begin{aligned} \text{maximize} \quad & x_1 + x_2 \\ \text{subject to} \quad & x_3 = 1 + x_1 - x_2, \\ & x_4 = 3 - x_1, \\ & x_5 = 2 - x_2, \\ & x_1, ...x_5 \geq 0. \qquad (1.10) \end{aligned}$$

From this we obtain the initial tableau

$$\begin{aligned} x_3 &= 1 + x_1 - x_2 \\ x_4 &= 3 - x_1 \\ x_5 &= 2 \qquad - x_2 \\ \hline z &= x_1 + x_2 \end{aligned} \qquad (1.11)$$

Abstracting from the initial tableau (1.7), a general tableau for the LP is any system T of n + 1 linear equations in the variables $x_{1}, .. , x_{d+n}$ and z, with the properties that

(i) T expresses n left-hand side variables x_B and z in terms of the remaining d righthand side variables x_N, i.e. there is an n-vector β, a d-vector γ, an n X d-matrix Λ and a real number z_0 such that T is the system

$$\begin{aligned} x_B &= \beta - \wedge x_N \\ \hline z &= z_0 + \gamma^T x_N \end{aligned} \qquad (1.12)$$

(ii) Any solution of (1.12) is a solution of (1.8) and vice versa.

By property (ii), any tableau contains the same information about the LP but represented in a different way. All that the simplex algorithm is about is constructing a sequence of tableaus by gradually rewriting them, finally leading to a tableau in which the information is represented in such a way that the desired optimal solution can be read off directly. We will immediately show how this works in our example.

Here is the initial tableau (1.11) to Example 1.1 again.

$$\begin{aligned} x_3 &= 1 + x_1 - x_2 \\ x_4 &= 3 - x_1 \\ x_5 &= 2 \quad\;\; - x_2 \\ \hline z &= \quad\;\; x_1 + x_2 \end{aligned}$$

By setting the right-hand side variables x_1, x_2 to zero, we find that the left-hand side variables x_3, x_4, x_5 assume nonnegative values $x_3 = 1$, $x_4 = 3$, $x_5 = 2$. This means, the vector x = (0, 0, 1, 3, 2) is a feasible solution of (1.10) and the vector $x_0 = (0, 0)$ is a feasible solution of (1.9). The objective function value z = 0 associated with this feasible solution is computed from the last row of the tableau. In general, any feasible solution that can be obtained by setting the right-hand side variables of a tableau to zero is called a basic feasible solution (BFS). In this case we also refer to the tableau as a feasible tableau. The left-hand side variables of a feasible tableau are called basic and are said to constitute a basis, the right-hand side ones are nonbasic. The goal of the simplex algorithm is now either to construct a new feasible tableau with a corresponding BFS of higher z-value, or to prove that there exists no feasible solution at all with higher z-value. In the latter case the BFS obtained from the tableau is reported as an optimal solution to the LP; in the former case, the process is repeated, starting from the new tableau.

In the above tableau we observe that increasing the value of x_1 i.e. making x_1 positive will increase the z-value. The same is true for x_2, and this is due to the fact that both variables have positive coefficients in the z-row of the tableau. Let us arbitrarily choose x_2. By how much can we increase x_2 ? If we want to maintain feasibility, we have to be careful not to let any of the basic variables go below zero. This means, the equations determining the values of the basic variables may limit x_2's increment.

Consider the first equation

$$x_3 = 1 + x_1 - x_2 \qquad (1.13)$$

Together with the implicit constraint $x_3 >= 0$, this equation lets us increase x_2 up to the value $x_2 = 1$ (the other nonbasic variable x_1 keeps its zero value). The second equation

$$x_4 = 3 - x_1$$

does not limit the increment of x_2 at all, and the third equation

$$x_5 = 2 - x_2$$

allows for an increase up to the value $x_2 = 2$ before x_5 gets negative. The most stringent restriction therefore is $x_3 >= 0$, imposed by (1.13), and we will increase x_2 just as much as we can, so we get $x_2 = 1$ and $x_3=0$. From the remaining tableau equations, the values of the other variables are obtained as

$$x_4 = 3 - x_1 = 3;$$

$$x_5 = 2 - x_2 = 1;$$

To establish this as a BFS, we would like to have a tableau with the new zero variable x_3 replacing x_2 as a nonbasic variable. This is easy, the equation (1.13) which determined the new value of x_2 relates both variables. This equation can be rewritten as:

$$x_2 = 1 + x_1 - x_3;$$

and substituting the right-hand side for x_2 into the remaining equations gives the new tableau

$$\begin{aligned} x_2 &= 1 + x_1 - x_3 \\ x_4 &= 3 - x_1 \\ x_5 &= 1 - x_1 + x_3 \\ \hline z &= 1 + 2x_1 - x_3 \end{aligned}$$

with corresponding BFS x = (0, 1, 0, 3, 1) and objective function value z = 1.

DISCUSSION

This process of rewriting a tableau into another one is called a pivot step, and it is clear by construction that both systems have the same set of solutions. The effect of a pivot step is that a nonbasic variable (in this case x_2) enters the basis, while a basic one (in this case x_3) leaves it. Let us call x_2 the entering variable and x_3 the leaving variable.

In the new tableau, we can still increase x_1 and obtain a larger z-value. x_3 cannot be increased since this would lead to smaller z-value. The first equation puts no restriction on the increment, from the second one we get $x_1 < = 3$ and from the third one x1 <= 1. So the third one is the most stringent, will be rewritten and substituted into the remaining equations as above. This means, x_1 enters the basis, x_5 leaves it, and the tableau we obtain is

$$\begin{array}{llllll} x_2 & = 2 & & & - & x_5 \\ x_4 & = 2 & - & x_3 & + & x_5 \\ x_1 & = 1 & + & x_3 & - & x_5 \\ \hline z & = 3 & + & x_3 & - & 2x_5 \end{array}$$

with BFS x = (1, 2, 0, 2, 0) and z = 3. Performing one more pivot step (this time with x_3 the entering and x_4 the leaving variable), we arrive at the tableau

$$\begin{array}{llllll} x_2 & = 2 & & & - & x_5 \\ x_4 & = 2 & - & x_4 & + & x_5 \\ x_1 & = 3 & - & x_4 & & \\ \hline z & = 5 & - & x_4 & - & x_5 \end{array} \qquad (1.14)$$

with BFS x = (3, 2, 2, 0, 0) and z = 5. In this tableau, no non-basic variable can increase without making the objective function value smaller, so we are stuck. Luckily, this means that we have already found an optimal solution. Why? Consider any feasible solution x' = ($x'_{1,\ldots\ldots\ldots}$, x'_5) for (1.10), with objective function value z_0. This is a solution to (1.11) and therefore a solution to (1.14). Thus,

$$z_0 = 5 - x'_4 - x_5'$$

must hold, and together with the implicit restrictions x_4; x_5 ¸ 0 this implies z_0 <= 5.

CONCLUSION

The tableau even delivers a proof that the BFS we have computed is the unique optimal solution to the problem: z = 5 implies $x_4 = x_5 = 0$, and this determines the values of the other variables. Ambiguities occur only if some of the non-basic variables have zero coefficients in the z-row of the final tableau. Unless a specific optimal solution is required, the simplex algorithm in this case just reports the optimal BFS it has at hand.

REFERENCES

K. L. Clarkson (2014). A Las Vegas Algorithm for Linear and Integer Programming when the Dimension is Small. In Proceedings of 29th Annual IEEE Symposium Foundations of Computer Science.

T. H. Cormen, C. E. Leiserson, and R. L. Rivest: Introduction to Algorithms. The MIT Press, Cambridge, MA.

G. B. Dantzig (2013). Linear Programming and Extensions. Princeton University Press, Princeton, NJ.

M. E. Dyer (1986). On a Multidimensional Search Technique and Its Application to the Euclidean One-Center Problem. SIAM J. Comput.

M. E. Dyer and A. M. Frieze: A Randomized Algorithm for Fixed-Dimensional Linear Programming. Math. Programming.

H. Edelsbrunner and R. Seidel: Voronoi diagrams and arrangements. Discrete Comput. Geom.

B. Gartner: A subexponential algorithm for abstract optimization problems. In Proceedings of 33rd Annual IEEE Symposium on Foundations of Computer Science.

B. Gartner and E. Welzl: Vapnik-Chervonenkis dimension and pseudo-hyperplane arrangements. Discrete Comput. Geom.

B. Gartner and E. Welzl: Linear programming randomization and abstract frameworks. In Proceedings of 13th Annual Symposium on Theoretical Aspects of Computer Science (STACS), volume 1046 of Lecture Notes in Computer Science.

Corresponding Author

Priyanka Sharma*

Research Scholar, Shri Venkateshwara University, Gajraula