LINEAR PROGRAMMING

Raja Grover, Naresh Kumar, Vishal Saini
Dronachaya college of engineering,Gurgaon

Abstract- In recent years the introduction and development of Interior-Point Methods has had a profound impact on optimization theory as well as practice, influencing the field of Operations Research and related areas. Development of these methods has quickly led to the design of new and efficient optimization codes particularly for Linear Programming. Consequently, there has been an increasing need to introduce theory and methods of this new area in optimization into the appropriate undergraduate and first year graduate courses such as introductory Operations Research and/or Linear Programming courses, Industrial Engineering courses and Math Modeling courses. The objective of this paper is to discuss the ways of simplifying the introduction of Interior-Point Methods for students who have various backgrounds or who are not necessarily mathematics majors.

Index Terms- **Interior-point methods, simplex method, Newton's method, linear programming, optimization, operations research, teaching issues**

I. INTRODUCTION

Linear programming (**LP**; also called **linear optimization**) is a method to achieve the best outcome (such as maximum profit or lowest cost) in a mathematicamodel whose requirements are represented by linear relationships. Linear programming is a special case of mathematical programming (mathematical optimization).

More formally, linear programming is a technique for the optimization of a linear objective function, subject to linear equality and linear inequality constraints. Its feasible region is a convex polytope, which is a set defined as the intersection of finitely many half spaces, each of which is defined by a linear inequality. Its objective function is a real-valued affine function defined on this polyhedron. A linear programming algorithm finds a point in the polyhedron where this function has the smallest (or largest) value if such a point exists.

Linear programs are problems that can be expressed in canonical form:

where **x** represents the vector of variables (to be determined), **c** and **b** are vectors of (known) coefficients, *A* is (known) matrix of coefficients, and $(\cdot)^{\mathrm{T}}$ is the matrix transpose. The expression to be maximized or minimized is called the *objective function* ($\mathbf{c}^{\mathrm{T}}\mathbf{x}$ in this case). The inequalities $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$ are the constraints which specify a convex polytope over which the objective function is to be optimized. In this context, two vectors are comparable when they have the same dimensions. If every entry in the first is less-than or equal-to the corresponding entry in the second then we can say the first vector is less-than or equal-to the second vector.

Linear programming can be applied to various fields of study. It is used in business and economics, but can also be utilized for some engineering problems. Industries that use linear programming models include transportation, energy, telecommunications, and manufacturing. It has proved useful in modeling diverse types of problems in planning, routing, scheduling, assignment, and design.

The problem of solving a system of linear equalities dates back at least as far as Fourier, after whom the method of Fourier–Motzkin elimination is named. The linear programming method was first developed by Leonid Kantorovich in 1937.[1] He developed it during World War II as a way to plan expenditures and returns so as to reduce costs to the army and increase losses incurred by the enemy. The method was kept secret until 1947 when George B. Dantzig published the simplex method and John von Neumann developed the theory ofduality as a linear optimization solution, and applied it in the field of game theory. Postwar, many industries found its use in their daily planning.

Dantzig's original example was to find the best assignment of 70 people to 70 jobs. The computing power required to test all the permutations to select the best assignment is vast; the number of possible configurations exceeds the number of particles in the observable universe. However, it takes only a moment to find the optimum solution by posing the problem as a linear program and applying the simplex algorithm. The theory behind linear programming drastically reduces the number of possible optimal solutions that must be checked.

The linear-programming problem was first shown to be solvable in polynomial time by Leonid Khachiyan in 1979, but a larger theoretical and practical breakthrough in the field came in 1984 when Narendra Karmarkar introduced a new interior-point method for solving linear-programming problems.

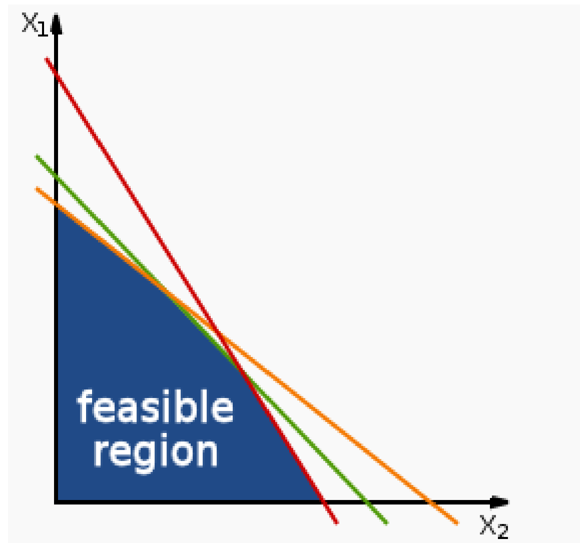
The problem of solving a system of linear equalities dates back at least as far as Fourier, after whom the method of Fourier–Motzkin elimination is named. The linear programming method was first developed by Leonid Kantorovich in 1937.[1] He developed it during World War II as a way to plan expenditures and returns so as to reduce costs to the army and increase losses incurred by the enemy. The method was kept secret until 1947 when George B. Dantzig published the simplex method and John von Neumann developed the theory ofduality as a linear optimization solution, and applied it in the field of game theory. Postwar, many industries found its use in their daily planning.

Dantzig's original example was to find the best assignment of 70 people to 70 jobs. The computing power required to test all the permutations to select the best assignment is vast; the number of possible configurations exceeds the number of particles in the observable universe. However, it takes only a moment to find the optimum solution by posing the problem as a linear program and applying the simplex algorithm. The theory behind linear programming drastically reduces the number of possible optimal solutions that must be checked.

The linear-programming problem was first shown to be solvable in polynomial time by Leonid Khachiyan in 1979, but a larger theoretical and practical breakthrough in the field came in 1984 when Narendra Karmarkar introduced a new interior-point method for solving linear-programming problems.

II. ALGORITHMS

See also: List of numerical analysis topics § Linear programming



In a linear programming problem, a series of linear constraints produces a convexfeasible region of possible values for those variables. In the two-variable case this region is in the shape of a convex simple polygon.

Basis exchange algorithms

Simplex algorithm of Dantzig

The simplex algorithm, developed by George Dantzig in 1947, solves LP problems by constructing a feasible solution at a vertex of the polytope and then walking along a path on the edges of the polytope to vertices with non-decreasing values of the objective function until an optimum is reached for sure. In many practical problems, "stalling" occurs: Many pivots are made with no increase in the objective function.[3][4] In rare practical problems, the usual versions of the simplex algorithm may actually "cycle".[4] To avoid cycles, researchers developed new pivoting rules.[5][6][3][4][7][8]

In practice, the simplex algorithm is quite efficient and can be guaranteed to find the global optimum if certain precautions against*cycling* are taken. The simplex algorithm has been proved to solve "random" problems efficiently, i.e. in a cubic number of steps,[9]which is similar to its behavior on practical problems.[3][10]

However, the simplex algorithm has poor worst-case behavior: Klee and Minty constructed a family of linear programming problems for which the simplex method takes a number of steps exponential in the problem size.[3][6][7] In fact, for some time it was not known whether the linear programming problem was solvable in polynomial time, i.e. of complexity class P.

Criss-cross algorithm

Like the simplex algorithm of Dantzig, the criss-cross algorithm is a basis-exchange algorithm that pivots between bases. However, the criss-cross algorithm need not maintain feasibility, but can pivot rather from a feasible basis to an infeasible basis. The criss-cross algorithm does not have polynomial time-complexity for linear programming. Both algorithms visit all 2^D corners of a (perturbed) cube in dimension *D*, the Klee–Minty cube, in the worst case.[8][11]

Interior point

Ellipsoid algorithm, following Khachiyan

This is the first worst-case polynomial-time algorithm for linear programming. To solve a problem which has *n* variables and can be encoded in *L* input bits, this algorithm uses $O(n^4L)$ pseudo-arithmetic operations on numbers with $O(L)$ digits. Khachiyan's algorithm and his long standing issue was resolved by Leonid Khachiyan in 1979 with the introduction of the ellipsoid method. The convergence analysis have (real-number) predecessors, notably the iterative methods developed by Naum Z. Shor and the approximation algorithms by Arkadi Nemirovski and D. Yudin.

Projective algorithm of Karmarkar

Khachiyan's algorithm was of landmark importance for establishing the polynomial-time solvability of linear programs. The algorithm was not a computational break-through, as the simplex method is more efficient for all but specially constructed families of linear programs.

However, Khachiyan's algorithm inspired new lines of research in linear programming. In 1984, N. Karmarkar proposed a projective method for linear programming. Karmarkar's algorithm improved on Khachiyan's worst-case polynomial bound (giving $O(n^{3.5}L)$). Karmarkar claimed that his algorithm was much faster in practical LP than the simplex method, a claim that created great interest in interior-point methods.[12]

Path-following algorithms[edit]

In contrast to the simplex algorithm, which finds an optimal solution by traversing the edges between vertices on a polyhedral set, interior-point methods move through the interior of the feasible region. Since then, many interior-point methods have been proposed and analyzed. Early successful implementations were based on *affine scaling* variants of the method. For both theoretical and practical purposes, barrier function or path-following methods have been the most popular since the 1990s.[13]

Comparison of interior-point methods versus simplex algorithms

The current opinion is that the efficiency of good implementations of simplex-based methods and interior point methods are similar for routine applications of linear programming.[13] However, for specific types of LP problems, it may be that one type of solver is better than another (sometimes much better), and that the structure of the solutions generated by interior point methods versus simplex-based methods are significantly different with the support set of active variables being typically smaller for the later one.[14]

LP solvers are in widespread use for optimization of various problems in industry, such as optimization of flow in transportation networks.[15]

Approximate Algorithms for Covering/Packing LPs

Covering and packing LPs can be solved approximately in nearly-linear time. That is, if matrix *A* is of dimension $n\times m$ and has *N* non-zero entries, then there exist algorithms that run in time $O(N\cdot(\log N)^{O(1)}/\varepsilon^{O(1)})$ and produce $O(1\pm\varepsilon)$ approximate solutions to given covering and packing LPs. The best known sequential algorithm of this kind runs in time $O(N + (\log N)\cdot(n+m)/\varepsilon^2)$,[16] and the best known parallel algorithm of this kind runs in $O((\log N)^2/\varepsilon^3)$ iterations, each requiring only a matrix-vector multiplication which is highly parallelizable.

Standard form

Standard form is the usual and most intuitive form of describing a linear programming problem. It consists of the following three parts:

- A **linear function to be maximized**
  e.g. $f(x_1, x_2) = c_1 x_1 + c_2 x_2$
- **Problem constraints** of the following form
  e.g.

$$a_{11} x_1 + a_{12} x_2 \leq b_1$$
$$a_{21} x_1 + a_{22} x_2 \leq b_2$$
$$a_{31} x_1 + a_{32} x_2 \leq b_3$$

- **Non-negative variables**
  e.g.

$$x_1 \geq 0$$
$$x_2 \geq 0$$

The problem is usually expressed in *matrix form*, and then becomes:

$$\max\{c^{\mathrm{T}} x \mid Ax \leq b \wedge x \geq 0\}$$

Other forms, such as minimization problems, problems with constraints on alternative forms, as well as problems involving negative variables can always be rewritten into an equivalent problem in standard form.

Example

Suppose that a farmer has a piece of farm land, say L km^2, to be planted with either wheat or barley or some combination of the two. The farmer has a limited amount of fertilizer, F kilograms, and insecticide, P kilograms. Every square kilometer of wheat requires F_1 kilograms of fertilizer and P_1 kilograms of insecticide, while every square kilometer of barley requires F_2 kilograms of fertilizer and P_2 kilograms of insecticide. Let S_1 be the selling price of wheat per square kilometer, and S_2 be the selling price of barley. If we denote the area of land planted with wheat and barley by x_1 and x_2 respectively, then profit can be maximized by choosing optimal values for x_1 and x_2. This problem can be expressed with the following linear programming problem in the standard form:

Maximize: $S_1 \cdot x_1 + S_2 \cdot x_2$ (maximize the revenue—revenue is the "objective function")

Subject to:
$x_1 + x_2 \leq L$ (limit on total area)
$F_1 \cdot x_1 + F_2 \cdot x_2 \leq F$ (limit on fertilizer)
$P_1 \cdot x_1 + P_2 \cdot x_2 \leq P$ (limit on insecticide)
$x_1 \geq 0, x_2 \geq 0$ (cannot plant a negative area).

Which in matrix form becomes:

maximize $\begin{bmatrix} S_1 & S_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

subject to $\begin{bmatrix} 1 & 1 \\ F_1 & F_2 \\ P_1 & P_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} L \\ F \\ P \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$

Standard form

Standard form is the usual and most intuitive form of describing a linear programming problem. It consists of the following three parts:

- A **linear function to be maximized**

  e.g. $f(x_1, x_2) = c_1 x_1 + c_2 x_2$

- **Problem constraints** of the following form

  e.g.

$$a_{11} x_1 + a_{12} x_2 \quad \leq b_1$$
$$a_{21} x_1 + a_{22} x_2 \quad \leq b_2$$
$$a_{31} x_1 + a_{32} x_2 \quad \leq b_3$$

- **Non-negative variables**

  e.g.

$$x_1 \geq 0$$
$$x_2 \geq 0$$

The problem is usually expressed in *matrix form*, and then becomes:

$$\max\{c^{\mathrm{T}} x \mid Ax \leq b \wedge x \geq 0\}$$

Other forms, such as minimization problems, problems with constraints on alternative forms, as well as problems involving negative variables can always be rewritten into an equivalent problem in standard form.

REFERENCES

[1] L.V. Kantorovich: *A new method of solving some classes of extremal problems*, Doklady Akad Sci USSR, 28, 1940, 211-214.

[2] G.B Dantzig: *Maximization of a linear function of variables subject to linear inequalities*, 1947. Published pp. 339–347 in T.C. Koopmans (ed.):*Activity Analysis of Production and Allocation*, New York-London 1951 (Wiley & Chapman-Hall)

[3] J. E. Beasley, editor. *Advances in Linear and Integer Programming*. Oxford Science, 1996. (Collection of surveys)

[4] R. G. Bland, New finite pivoting rules for the simplex method, *Math. Oper. Res.* 2 (1977) 103–107.

[5] Karl-Heinz Borgwardt, *The Simplex Algorithm: A*

[6] *Probabilistic Analysis*, Algorithms and Combinatorics, Volume 1, Springer-Verlag, 1987. (Average behavior on random problems)

[7] Richard W. Cottle, ed. *The Basic George B. Dantzig*. Stanford Business Books, Stanford University Press, Stanford, California, 2003. (Selected papers by George B. Dantzig)

[8] George B. Dantzig and Mukund N. Thapa. 1997. *Linear programming 1: Introduction*. Springer-Verlag.

[9] George B. Dantzig and Mukund N. Thapa. 2003. *Linear Programming 2: Theory and Extensions*. Springer-Verlag. (Comprehensive, covering e.g. pivoting and interior-point algorithms, large-scale problems, decomposition following Dantzig-Wolfe and Benders, and introducing stochastic programming.)

[10] Edmonds, J. and Giles, R., "A min-max relation for submodular functions on graphs," Ann. Discrete Math., v1, pp. 185–204, 1977

[11] Fukuda, Komei; Terlaky, Tamás (1997). Thomas M. Liebling and Dominique de Werra, ed. "Criss-cross methods: A fresh view on pivot algorithms". *Mathematical Programming: Series B* (Amsterdam: North-Holland Publishing Co.) **79** (1—3): 369–395. doi:10.1007/BF02614325. MR 1464775.

[12] Gondzio, Jacek; Terlaky, Tamás (1996). "3 A computational view of interior point methods". In J. E. Beasley. *Advances in linear and integer programming*. Oxford Lecture Series in Mathematics and its Applications **4**. New York: Oxford University Press. pp. 103–144. MR 1438311. Postscript file at website of

Gondzio and at McMaster University website of Terlaky.

[13] Murty, Katta G. (1983). *Linear programming*. New York: John Wiley & Sons, Inc. pp. xix+482. ISBN 0-471-09725-X. MR 720547. (comprehensive reference to classical approaches).