12-1-1994

Review Of “Linear Programming And Related Problems” By E. Nering And A. W. Tucker

Stephen B. Maurer
Swarthmore College, smaurer1@swarthmore.edu

Let us know how access to these works benefits you

Follow this and additional works at: https://works.swarthmore.edu/fac-math-stat

Part of the Discrete Mathematics and Combinatorics Commons

Recommended Citation
https://works.swarthmore.edu/fac-math-stat/157

This Book Review is brought to you for free and open access by the Mathematics & Statistics at Works. It has been accepted for inclusion in Mathematics & Statistics Faculty Works by an authorized administrator of Works. For more information, please contact myworks@swarthmore.edu.
Review
Reviewed Work(s): Linear Programs and Related Problems by Evar D. Nering and Albert W. Tucker
Review by: Stephen B. Maurer
Published by: Mathematical Association of America
Stable URL: http://www.jstor.org/stable/2975180
Accessed: 19-10-2016 15:42 UTC

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at http://about.jstor.org/terms
Linear Programs and Related Problems, by Evar D. Nering and Albert W. Tucker, Academic Press, San Diego, 1993; 578 + pp

Reviewed by Stephen B. Maurer

This undergraduate text is a joint effort by the author of several fine texts (Nering) and one of the grand old men of mathematics and a former MAA President (Tucker). It should be on the shelf of anyone who teaches linear programming, and it invites careful consideration for classroom adoption. It provides an approach that deserves to be better known, for it combines theory and practice and makes key ideas (notably duality) particularly transparent. Most faculty will learn at least as much from the book as students will.

Linear programming (LP) is the study of optimizing linear functions subject to linear inequality constraints. The subject is blessed to have both beautiful theory and myriad applications. Consequently, there are many LP texts of several varieties. Most common are books with an Operations Research orientation, such as Hillier and Lieberman [6] and Bradley, Hax and Magnanti [2]. These books regard LP as a tool for modeling. The theory is usually there, but as a sideshow. They emphasize how to recognize and interpret a linear program in a real-world situation and how to apply algorithmic implementations that are efficient and numerically stable.

In contrast, books more mathematical in flavor tend to emphasize the theory and structure of the subject. They often dwell on the connection to n-dimensional polyhedra, and either downplay algorithms or highlight the more theoretical algorithmic issues from complexity theory. Books written in this vein are typically at the graduate level. A fine recent example is Schrijver [12]. Older and less advanced is Gass [5].

There are also LP books written by economists, which emphasize the economic applications and interpretations. Two classics are Dorfman, Samuelson and Solow [4] and Baumol [1].

This division of texts recapitulates history. When George Dantzig, at the Pentagon after World War II, was first able to secure funding to promote this new subject, three groups got started: one under Tucker at Princeton to develop the theory, another under Koopmans at Chicago to explore connections with economics, and the third under Dantzig to explore algorithms, specifically, the simplex algorithm. For more on the history, see [8, 9, 13].

Note the schism, deliberately introduced by the players themselves, between theory and practice. It doesn’t have to be that way. LP theory can be developed from the algorithms, and you can delight in both from the start. A key strength of this book is that it shows how. For instance, suppose an algorithm to compute the
optimum of a function $f$ has the properties that

— it has only a finite number of states
— it does not cycle among states
— if its current state neither provides an optimal value of $f$ nor shows that $f$ is unbounded, then the algorithm passes to another state.

Then it is a mere observation that $f$ must either attain an optimum or be unbounded. Yet it is just this observation that one may use with the simplex algorithm to show that any feasible linear program either attains an optimum or is unbounded. (Feasible means the domain of $f$ is nonempty.)

As the authors put it (p. 126)

Many people think of an algorithm as a method for finding a solution to a problem for which one already knows, by some other means, that a solution exists. We look on algorithms as much more than that and we use them for much more than that. In many cases an algorithm can supply the proof that the desired result exists.

In other words, Nering and Tucker are advocates of proof by algorithm, a refinement of the old idea of constructive proof.

Today, proof by algorithm is a well-known concept. Certainly in the field of combinatorial optimization it is the preferred method of proof—in this field existence proofs are not considered more esthetic. But since this preference is a rather drastic change from the mathematical esthetic of mid century, one can ask how the change came about. A reasonable assumption is that it came from Computer Science. However, it may be that Tucker had a lot to do with it. He started turning to the algorithmic viewpoint in the mid 1950s (see the interview with him [10]), and his group at Princeton included or interacted with almost all the early workers in combinatorial optimization. There is a fine math history research project waiting here—how did this new esthetic come about?

A second great strength of this book is its treatment of duality. Linear programs come in pairs. To take the canonical case, if the “primal” problem is to maximize $cx$ subject to $Ax \leq b$ and $x \geq 0$, then the dual is to minimize $vb$ subject to $vA \geq c$ and $v \geq 0$. (Lowercase letters are vectors, uppercase matrices.) Dual problems interact. For instance, if both problems are feasible, then it turns out both attain optima and the optimum values are equal. Duality has many important consequences, e.g., a lazy supervisor test for checking a claimed optimum, alternative algorithms, shadow prices in economics.

All LP texts cover duality, but usually as a somewhat mysterious add-on. This too mimics history, but there is a better way. Dual programs can be introduced simultaneously by the use of a special representation, the condensed or Tucker tableau:

$$
\begin{array}{cccccc}
 & x_1 & x_2 & \cdots & x_n & -1 \\
v_1 & a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\
v_2 & a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\
& \vdots & \vdots & \vdots & \vdots & \vdots \\
v_m & a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \\
-1 & c_1 & c_2 & \cdots & c_n & d \\
\end{array}
$$

= $u_1 = u_2 = \cdots = u_n = g$

= $-y_1 = -y_2$

= $v_1 = v_2 = \cdots = v_m$

= $f$

1994] REVIEWS 1023
The variables on the top multiply the columns to obtain the variables on the right, i.e., \( Ax + (-1)b = -y \), or equivalently, \( Ax + y = b \), or \( Ax \leq b \) if all variables are nonnegative. These row equations are the constraints, and the basement row equation, \( cx - d = f \), is the equation for the “objective” function \( f \) to be maximized. This is the primal problem. The variables on the left and bottom give the dual: minimize \( vb - d = g \) subject to \( vA - c = u \), or equivalently, \( vA \geq c \).

In brief, to develop the theory from (1), one shows by elementary linear algebra that the simplex algorithm just exchanges variables on the top and right, so that different variables become “basic” when the variables on the top are set to 0. Of course, the entries in the tableau must be updated too, so that the equal signs on the right are still correct after variables are exchanged. Marvelously, the same algebra maintains the equal signs on the bottom as the dual variables are exchanged. The conditions that make the basic solution of a tableau optimal are simple to understand and describe—the entries in \( b \) must be nonnegative and those in \( c \) nonpositive. By (skew) symmetry these same conditions describe optimality for the dual. Thus a condensed tableau can exhibit optimality for both problems simultaneously and, since \( x \) and \( v \) are 0 for a basic solution, the objective functions \( f \) and \( g \) have the same value, \( d \).

Almost all LP books use tableaus, but most use some version of extended form, where \( A = [a_{ij}] \) in (1) gets replaced by \( [A I] \) and the “slack” variables \( y \) in (1) move to the columns of \( I \). But it is much harder to even see a dual, let alone develop its theory, in these formats. A few books do use condensed tableaus, but without the variables on the left and bottom, which is the key to handling duality. To my knowledge, the only other texts with true condensed tableaus are Kemeny Snell and Thompson’s “Finite Mathematics”, third edition only [7], and Rothenberg [11].

Part I of Nering and Tucker develops linear programming itself, along the lines described above. Part II treats various related problems—related in that most can be stated as LPs but their special form allows for special algorithms. The emphasis on proof by algorithm continues, and duality is often used to clarify the algorithms (but not as often as it could be). For instance, the minimax theorem of matrix games is reduced to the existence-duality theorem of LP. Kuhn’s Hungarian method for the minimum weight assignment problem is shown to work by increasing the sum of the dual variables until the primal and dual objective functions are equal. The same approach is used for analyzing Dantzig’s algorithm for the transportation problem. Other topics include various network problems—transshipment, maximum flow, shortest path—and an introduction to nonlinear programming—the Karush-Kuhn-Tucker Theorem (still known as the Kuhn-Tucker theorem to many) and various quadratic programs.

Tucker once said that his goal in mathematics has always been to “unify and simplify”. The fruits of this attitude appear in this book. It’s a shame that his condensed tableau format is not better known.

These days most talk in LP circles is about completely different approaches: Khachian’s 1979 ellipsoid algorithm, Karmarkar’s 1984 interior point method, and more recent variations. Khachian’s algorithm was important because it is polynomial: there is a polynomial \( P \) so that his algorithm will solve every LP problem in \( P(n) \) steps, where \( n \) is the amount of input data. However, in practice the Ellipsoid algorithm is much slower than simplex. (The simplex algorithm takes linear time on average but special cases take exponential time.) Karmarkar’s algorithm is not only polynomial, but fast in practice on at least some sorts of problems. The jury is still out on what is the best commercial method.
Nering and Tucker are thin, and mostly nontechnical, on the new methods. For Khachian’s algorithm, no algebraic description or analysis is given, nor are there numerical examples or homework problems. For Karmarkar’s algorithm, some algebraic detail is given, one iteration of one problem is shown, and there is one homework problem. (This material appears in the final chapter of Part I, which also introduces numerical issues and provides ties to other simplex approaches.)

This thinness is understandable. There are no small, complete examples for the new methods, and their theory is much more complicated than for the simplex method. Also, no one has shown a nice way to develop the existence and duality theory of LP from these algorithms. Since building the theory from algorithms is a key theme of this book, the authors rightly emphasize simplex.

Nonetheless, since the new methods are at least part of the future, it would be good for students to get more than a passing nod at them even in a first course. Software seems called for. A program with an option to hide the numbers and show pictures might be best. To my knowledge, no text has appeared that takes this approach.

How will students like this book? The authors have made LP theory simple (but still rigorous!) by reducing it to displays and arguments that are carried forward by linear equations and numerical calculations. The writing is clear, straightforward and leisurely, but also somewhat bland. The book is written more in essay style than in textbook chunk style—example, theorem, proof, sidebar, vignette, etc. Much as this chunk style has been criticized, students are used to it and it does allow them clear touchstones and stopping points in what is usually difficult material for them. In Nering and Tucker, important points often appear in the middle of paragraphs in pages full of text.

Based on my experience teaching from a preliminary version some years ago, and more recent reports, I would say that students won’t find the book hard to understand—the usual math book complaint—but they won’t find it exciting either. It can’t be very neat mathematics if it is just about linear equations and a lot of number manipulation, can it? Of course it can, but the teacher’s biggest job in using this text will be to explain to students why. Alas, if you make something sufficiently simple for readers who don’t know that it used to be complicated, they won’t appreciate what they have!

A very nice set of real-world-like examples and problems are introduced in the first chapter: problems too simplified to be really applied, but they show the way. In later chapters, the problems (though not the examples) are mostly purely mathematical. Problems appear at the ends of chapters only. A rather complete set of solutions is supplied in the back, an unusual feature. Also, menu-driven software is provided with the book—for PCs, but a Macintosh version is due by the end of 1993. With this software one can do the arithmetic for all the problems, in the format used by the book, either all at once, or step by step. The data for all the problems is already read in. I found this software serviceable but not effortless. The instructions are terse.

Readers intrigued by this book might also look at Chvátal [3]. It is like-spirited in doing theory, algorithmics and applications simultaneously, for a similar audience. It has a somewhat livelier format and style, and a large variety of problems. It covers many more topics and is more advanced (however, much advanced material is identified with small print and can be skipped). On the other hand, Chvátal does not have Tucker tableaus (or any tableaus), and duality, though done early, is not present from the start.
To conclude, a founding father of mathematical programming, involved in it for 45 years, has put the full maturity of his perspective into this book. His perspective is quite personal, and the excitement and beauty seen by both authors may not come across fully to you or your students. But if you teach LP you impoverish yourself by not taking a look.

Note: Prof. Maurer was a Ph.D. student of Tucker, and learned LP from him.

REFERENCES


Department of Mathematics
Swarthmore College
Swarthmore, PA 19081-1397
smaurer1@cc.swarthmore.edu

The real danger is not that computers will begin to think like men, but that men will begin to think like computers.

—Sydney J. Harris