



## Sensible Rules for Remembering Duals--The S-O-B Method

Arthur T. Benjamin

*SIAM Review*, Volume 37, Issue 1 (Mar., 1995), 85-87.

Stable URL:

<http://links.jstor.org/sici?sici=0036-1445%28199503%2937%3A1%3C85%3ASRFRDS%3E2.0.CO%3B2-J>

---

Your use of the JSTOR archive indicates your acceptance of JSTOR's Terms and Conditions of Use, available at <http://www.jstor.org/about/terms.html>. JSTOR's Terms and Conditions of Use provides, in part, that unless you have obtained prior permission, you may not download an entire issue of a journal or multiple copies of articles, and you may use content in the JSTOR archive only for your personal, non-commercial use.

Each copy of any part of a JSTOR transmission must contain the same copyright notice that appears on the screen or printed page of such transmission.

*SIAM Review* is published by Society for Industrial and Applied Mathematics. Please contact the publisher for further permissions regarding the use of this work. Publisher contact information may be obtained at <http://www.jstor.org/journals/siam.html>.

---

*SIAM Review*

©1995 Society for Industrial and Applied Mathematics

JSTOR and the JSTOR logo are trademarks of JSTOR, and are Registered in the U.S. Patent and Trademark Office. For more information on JSTOR contact [jstor-info@umich.edu](mailto:jstor-info@umich.edu).

©2003 JSTOR

## CLASSROOM NOTES

EDITED BY JACK MACKI

*This section contains brief notes of up to five typed pages. Articles should illustrate novel ideas and insights related to either the teaching of mathematics for applications or the teaching of the applications.*

*Submissions should be sent in triplicate to Jack Macki, Department of Mathematics, University of Alberta, Edmonton, Alberta, Canada T6G 2G1; e-mail: jmacki@vega.math.ualberta.ca; fax: (403) 492-6826. Authors are requested to provide e-mail and fax addresses, when possible, to facilitate efficient communication.*

### SENSIBLE RULES FOR REMEMBERING DUALS—THE S-O-B METHOD\*

ARTHUR T. BENJAMIN†

**Abstract.** We present a natural motivation and simple mnemonic for creating the dual LP of any linear programming problem.

**Key words.** linear program, duality

**AMS subject classification.** 90C05

A linear program (LP) is a problem of maximizing or minimizing a linear function subject to linear equality or inequality constraints. Associated with any LP is a *dual* LP with many important properties. We demonstrate (as in [1]) how the dual LP arises naturally from the original LP, and provide a method for creating the dual that is extremely easy to remember.

Consider the following LP:

$$\begin{array}{ll}
 \text{Maximize} & z = 30x_1 + 20x_2, \\
 \text{(A)} & \text{subject to} \quad x_1 + 2x_2 \leq 6, \\
 \text{(B)} & \quad \quad \quad 2x_1 + x_2 \leq 8, \\
 \text{(C)} & \quad \quad \quad x_2 \leq 2, \\
 \text{(D)} & -x_1 + x_2 \leq 1, \\
 & \quad \quad \quad x_1, \quad x_2 \geq 0.
 \end{array}$$

To find a lower bound for  $z^*$ , the maximum value of the objective function  $z$ , it suffices to find a *feasible* solution. For instance, since  $x_1 = 2$ ,  $x_2 = 1$  satisfies all constraints, we must have  $z^* \geq 80$ . Using  $x_1 = 10/3$  and  $x_2 = 4/3$ , this bound can be improved to  $z^* \geq 380/3$ .

How about an upper bound for  $z^*$ ?

By multiplying (B) by 20, we see that any feasible solution must satisfy  $40x_1 + 20x_2 \leq 160$ . Thus, since  $x_1 \geq 0$ ,  $30x_1 + 20x_2 \leq 40x_1 + 20x_2 \leq 160$ . Hence  $z^* \leq 160$ . For a better bound, we can multiply (A) by  $y_A = 10$  and (B) by  $y_B = 10$ , then add to get  $30x_1 + 30x_2 \leq 140$ . Since  $x_2 \geq 0$ ,  $30x_1 + 20x_2 \leq 30x_1 + 30x_2 \leq 140$ . Thus  $z^* \leq 140$ .

The problem of finding the “best” multipliers that take full advantage of this trick is itself a linear programming problem. Namely, we seek multipliers  $y_A, y_B, y_C, y_D$  to

$$\begin{array}{ll}
 \text{Minimize} & w = 6y_A + 8y_B + 2y_C + 1y_D, \\
 \text{(1)} & \text{subject to} \quad y_A + 2y_B + 0y_C - 1y_D \geq 30, \\
 \text{(2)} & \quad \quad \quad 2y_A + 1y_B + 1y_C + 1y_D \geq 20, \\
 & \quad \quad \quad y_A, y_B, y_C, y_D \geq 0.
 \end{array}$$

This is called the *dual* LP associated with the original *primal* LP. If its constraints are satisfied by choosing  $y_A = 10/3$ ,  $y_B = 40/3$ ,  $y_C = y_D = 0$ , we obtain  $z^* \leq 380/3$ . From

\*Received by the editors June 30, 1993; accepted for publication December 20, 1993.

†Harvey Mudd College, Claremont, California 91711 (benjamin@hmcvax.ac.hmc.edu).

our lower bounding we had  $z^* \geq 380/3$ ; therefore  $z^* = 380/3$ , and  $x_1 = 10/3$ ,  $x_2 = 4/3$  is an optimal solution to the primal LP. (What led to these particular  $y$ 's is not our concern here.)

In general, if our primal has standard form:

$$\begin{array}{ll} \text{Maximize} & \mathbf{c}^T \mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \end{array}$$

where  $A$  is an  $m$  by  $n$  real matrix,  $\mathbf{c} \in \mathbf{R}^n$ ,  $\mathbf{b} \in \mathbf{R}^m$ , then the associated dual LP is

$$\begin{array}{ll} \text{Minimize} & \mathbf{b}^T \mathbf{y} \\ \text{subject to} & \mathbf{A}^T \mathbf{y} \geq \mathbf{c}, \\ & \mathbf{y} \geq \mathbf{0}. \end{array}$$

The dual LP has many applications. The strong duality theorem of linear programming (see [1]) states that if the primal LP has an optimal solution, then its dual also has an optimal solution with the same objective function value. Computationally, it is sometimes easier to solve the primal LP by first finding a solution to its dual; the common “simplex algorithm” for LPs, aimed at either problem, will (in effect) deliver a solution to the other one as well.

If the primal LP is not in standard form, then it can be converted into standard form by various means. For instance, an equality constraint (say  $x_1 + x_2 = 1$ ) can be replaced by two inequality constraints ( $x_1 + x_2 \leq 1$ ,  $-x_1 - x_2 \leq -1$ ). But these modifications are unnatural, and the resulting dual LP needs to be converted to a more natural form. However, by using our earlier logic, we can directly create the natural dual LP of any primal LP.

For instance, if constraint (D) were changed to  $-x_1 + x_2 \geq 1$ , then to achieve a statement as is needed for the upper bounding, we must have  $y_D \leq 0$ . Hence the direction of an inequality constraint determines the sign of the corresponding multiplier. If, however, (B) is converted to  $2x_1 + x_2 = 8$ , then  $y_B$  becomes unrestricted in sign, since equality statements can be legitimately added to inequality statements. Likewise, the sign of a variable determines the direction of the corresponding dual inequality. For example, if  $x_2$  were required to be non-positive, then the direction of dual inequality (2) would be reversed. Similarly, if the variable  $x_1$  were unrestricted in sign, then  $30x_1 + 20x_2$  is no longer dominated by  $50x_1 + 70x_2$ , but would still be dominated by  $30x_1 + 70x_2$ . Hence dual equation (1) would become  $y_A + 2y_B - y_D = 30$ .

When the primal LP is a minimization problem, the dual LP is a maximization problem for finding multipliers that yield the greatest lower bound on the primal's objective function, and analogous rules apply. (In fact, the dual of the dual is the primal.)

Based on these results, we now present a simple method for creating the dual of any LP. We begin by labelling each variable of the LP as *sensible*, *odd*, or *bizarre*, depending on whether it is non-negative, unrestricted, or non-positive. (This terminology matches the usual interpretations of the variables in an LP model.) Next we label the constraints of the LP. A problem of maximizing some objective (like profit) is typically constrained by *upper bounds* on available resources. Thus we define  $\leq$  constraints as sensible,  $=$  constraints as odd, and  $\geq$  constraints as bizarre. Conversely, a typical minimization problem (say, minimizing costs) will be constrained by meeting or exceeding stipulated demands. Hence, for minimization problems,  $\geq$  constraints are defined to be sensible,  $=$  constraints are odd, and  $\leq$  constraints are bizarre.

Most students have no difficulty remembering that if the primal LP has variables  $x_1, \dots, x_n$ , objective function  $\mathbf{c}^T \mathbf{x}$ ,  $m$  by  $n$  constraint matrix  $A$ , and right-hand side vector  $\mathbf{b}$ , then the dual LP has variables  $y_1, \dots, y_m$  (one for each constraint) with objective function  $\mathbf{b}^T \mathbf{y}$ , constraint matrix  $A^T$  (one constraint per variable), and right-hand side vector  $\mathbf{c}$ . It is now a simple matter to remember that a constraint in the dual is sensible (S), odd (O), or bizarre (B), if the

corresponding variable in the primal is sensible, odd, or bizarre. Likewise, the “sensible-ness” of each dual variable is the same as that of its corresponding constraint in the primal.

Let’s apply these rules to determine the dual of the following LP:

$$\begin{array}{ll}
 \text{Minimize} & 3x_1 + 5x_2 + x_3 \\
 \text{subject to} & x_1 + x_2 + x_3 = 2, \quad (\text{O}) \\
 & 2x_1 - 3x_3 \leq 0, \quad (\text{B}) \\
 & x_1 \geq 0, x_2 \geq 0. \\
 & (\text{S}) \quad (\text{S}) \quad (\text{O})
 \end{array}$$

The primal constraints tell us that our dual variables  $y_1$  and  $y_2$  will be, respectively, odd and bizarre. Likewise, the first two dual constraints will be sensible and the third one will be odd, resulting in the following LP:

$$\begin{array}{ll}
 \text{Maximize} & 2y_1 \\
 \text{subject to} & y_1 + 2y_2 \leq 3, \quad (\text{S}) \\
 & y_1 \leq 5, \quad (\text{S}) \\
 & y_1 - 3y_2 = 1, \quad (\text{O}) \\
 & y_2 \leq 0. \\
 & (\text{O}) \quad (\text{B})
 \end{array}$$

Using the S-O-B method (some may prefer to use the word *unusual* instead of odd and call it the “SUB” method), all twenty of my students were able to correctly dualize the above primal, and virtually all were able to do the same problem (without warning or notes) a year later.

#### REFERENCES

- [1] V. CHVÁTAL, *Linear Programming*, W.H. Freeman and Company, New York, 1983, pp. 54–57.