UPPER BOUNDED VARIABLES IN LINEAR PROGRAMMING

G. B. Dantzig and S. M. Johnson

1. INTRODUCTION

With the growing awareness of the potentialities of the linear programming approach to both dynamic and static problems of industry, of the economy, and of the military, the main obstacle toward full application is the inability of current computational methods to cope with the magnitude of the technological matrices for many common situations. However, in certain cases, such as the now classical Hitchcock–Koopmans transportation model, it has been possible to solve the linear inequality system in spite of size because of simple properties of the system. This suggests that considerable research be undertaken to exploit certain special matrix structures in order to facilitate ready solution of larger systems.

Indeed, recent computational experience has made it clear that standard techniques such as the simplex algorithm, which have been used to solve successfully general systems involving two hundred equations on electronic computers (in any reasonable number of non-negative unknowns), are too tedious and lengthy to be practical for extensions much beyond this figure. Our purpose here will be to develop short-cut computational methods for solving an important class of systems involving upper bound restraints on the variables $f_{ij}$. 
Upper bounds on variables in linear programming are used quite commonly. Their use tends to increase as models become more realistic. Typically, in practice, this would mean that more activities enter the solution since the favored activities are not allowed to carry the whole load.

The following two classes of problems illustrate the use of upper bounds.

A. Capacity Restrictions

1. On factory production rates.
2. On traffic links in a transportation problem.

B. Convex Programming Problems

1. Many examples where the cost of an activity is a convex function of the level engaged in.
2. Linear programming under uncertainty.

2. The Statement of the Problem.

Consider a system of equations

\[
\sum_{j=1}^{n} a_{ij} x_j = b_i \quad x_j \geq 0 \quad (i = 1, 2, \ldots, m).
\]

\[
\sum_{j=1}^{n} c_j x_j = z,
\]

where it is desired to obtain values of \( x_j \) such that the form \( \sum c_j x_j \) is to be minimized.

The size of the array of coefficients associated with such a linear programming problem may become uncomfortably large
when, in addition to (1), many (or all) variables of the initial set have upper bounds. Thus, if each variable satisfied \(0 \leq x_j \leq a_j\), it is customary to add an additional variable, say \(x'_j\), and a new equation

\[(2) \quad x_j + x'_j = a_j \quad (x_j \geq 0, x'_j \geq 0)\]

to take care of each such restriction.

We shall refer to the original system plus these upper bound restrictions as the enlarged system.

To illustrate, a linear programming problem of the transportation type involving \(m\) destinations and \(n\) origins has a matrix involving \(m + n\) rows and \(m \cdot n\) variables \(x_{ij}\) associated with \(m \cdot n\) possible routes joining origins with destinations. Suppose there is a capacity limitation \(r_{ij}\) on a route so that, in addition to the original system of equations and linear inequalities, \(0 \leq x_{ij}\), one must impose \(m \cdot n\) additional restraints

\[x_{ij} + x'_{ij} = r_{ij} \quad (x_{ij} \geq 0, x'_{ij} \geq 0).\]

It is clear now that the original system has been expanded to \((mn + m + n)\) rows and \(2m \cdot n\) variables. A transportation problem with these added restraints is also called a "Capacitated Hitchcock Problem" \([4]\).

3. The Technique Illustrated:

The main idea which permits solutions to enlarged systems without very much more effort than smaller systems is based on a slight generalization of the simplex procedure.
The simplex algorithm is usually applied with the non-basic or independent variables at the fixed value zero. However, a little reflection makes it clear that the independent variables could be at any fixed value. The simplex criterion indicates that it pays to increase the value of a variable $x_s$ (and adjust the values of basic variables) if $\bar{c}_s < 0$. However, if $x_s$ is at its upper bound value this will not be possible. Similarly if $\bar{c}_s > 0$ it pays to decrease the value of $x_s$ unless $x_s$ is at its lower bound value (usually zero). If $x_s$ is neither at its upper nor lower bound value it pays to increase or decrease its value according as $\bar{c}_s < 0$ or $\bar{c}_s > 0$.

The purpose of this section is to show that the upper bound restraints may be provided for by applying the simplex algorithm to the original system with due care that the range of values of a variable appearing in a basic solution stays within its upper and lower bounds. A simple example will illustrate the technique. (The proof in general is easy and is left to the reader.)

Find numbers

\begin{align*}
0 \leq x_1 & \leq 4, \quad 0 \leq x_2 \leq 5, \quad 0 \leq x_3 \leq 1, \\
0 \leq x_4 & \leq 2, \quad 0 \leq x_5 \leq 3
\end{align*}

and minimum $z$ satisfying

\begin{align*}
\text{The relative cost coefficients, } \bar{c}_j, \text{ are the coefficients in the objective form after elimination of the basic variables.} 
\end{align*}
\[
\begin{align*}
\begin{bmatrix} x_1 & + & x_3 & - & 2x_4 \\ x_2 & - & x_3 & + & x_4 & + & 2x_5 \\ & - & 2x_3 & - & x_4 & + & x_5 & - & z \\ \end{bmatrix} & = \begin{bmatrix} 3 \\ 4 \\ 0 \end{bmatrix}
\end{align*}
\]

Using \( x_1 \) and \( x_2 \) as basic variables, the basic feasible solution is

\[(5) \quad \begin{bmatrix} 3 \\ 4 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \end{bmatrix},
\]

where the values of the basic variables are shown by brackets \([ \ ]\). It will be noted that the values of all variables are less than their upper bound restraints. Since \( c_3 = -2 < 0 \) it pays to increase the value of the variable \( x_3 \). Holding the values of the other non-basic variables fixed, the values of the variables become

\[(6) \quad \begin{bmatrix} 3-x_3 \\ 4+x_3 \\ x_3 \\ 0 \\ 0 \\ -2x_3 \end{bmatrix}.
\]

However, in this case it is not possible to increase \( x_3 \) to the value \( x_3 = 3 \) because \( 0 \leq x_3 \leq 1 \). Accordingly we increase \( x_3 \) to its upper bound value and hold it fixed there while keeping the same set of basic variables. The new "basic feasible solution" is

\[(7) \quad \begin{bmatrix} 2 \\ 5 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} -2 \end{bmatrix}.
\]

Since \( c_4 = -1 \), we now proceed to increase \( x_4 \), and the values of the variables become
The largest value of $x^4$ is not $5/2$ because $0 < x^4 < 2$, nor is it $x^4 = 2$, because in this case $x_1 = 2 + 2 \cdot 2 = 6$, which violates $0 < x_1 < 4$. Indeed the largest value of $x^4$ is $x^4 = 1$ at which value the basic variable $x_1$ hits its upper bound value 4. Accordingly we drop $x_1$ from the basic set, and replace it by $x^4$. The new canonical form relative to the variables $x_2$ and $x^4$ is obtained by using $-2x^4$ as pivot element in (4), thereby obtaining

$$
- \frac{1}{2} x_1 - \frac{1}{2} x_3 + x_4 = -\frac{3}{2}
$$

(9) $$
\frac{1}{2} x_1 + x_2 - \frac{1}{2} x_3 + 2x_5 = \frac{11}{2}
$$ $$
- \frac{1}{2} x_1 - \frac{5}{2} x_3 + x_5 - z = -\frac{3}{2}
$$

and the new basic feasible solution is

(10) $$
4, \ [3], 1, \ [1], 0, -3,
$$

since the variable $x_1$ enters the non-basic set at its upper bound value $x_1 = 4$. It will be noted that the basic variables are between their lower and upper bounds and the non-basic variables are either at their lower or upper bounds; the one at its lower bound, $x_5$, has its corresponding $\delta_5 = 1 \geq 0$, while the others at their upper bound values
$x_1$ and $x_3$, have $\bar{c}_1 = -1/2 < 0$, and $\bar{c}_3 = -5/2 < 0$. Since this, as we shall show in a moment, is the criterion for optimality in the bounded variable problem, no further iterations are required.

To prove optimality in this case let us substitute in (9) for the non-basic variables at upper bounds the expressions

$$(11) \quad x_1 = 4 - x_1^i, \quad x_3 = 1 - x_3^i;$$

we obtain immediately an equivalent linear programming problem

$$(12) \quad \frac{1}{2} x_1^i + \frac{1}{2} x_3^i + x_4 = 1$$
$$-\frac{1}{2} x_1^i + x_2 + \frac{1}{2} x_3^i + 2x_5 = 3$$
$$+\frac{1}{2} x_1^i + \frac{5}{2} x_3^i + x_5 - z = 3$$

where

$$(13) \quad 0 \leq x_1^i \leq 4, \quad 0 \leq x_2 \leq 5, \quad 0 \leq x_3^i \leq 1,$$
$$0 \leq x_4 \leq 2, \quad 0 \leq x_5 \leq 3.$$

It will be noted that $x_2 = 3$, $x_4 = 1$, and all other $x_j$ or $x_j^i = 0$ constitute an optimal basic feasible solution to (12) that does not violate (13), — hence must be an optimal solution to (12) and (13). This solution is obviously the same as (10).
4. **CONVEX-SEPARABLE OBJECTIVE FUNCTION** [2], [3]

Consider a situation in which the objective function, instead of being a linear form to be minimized, is of the form

\[
\sum_{j=1}^{n} \phi_j(x_j) = \min \quad (x_j \geq 0)
\]  

where \( \phi_j(x_j) \) is a convex function and the \( x_j \) are subject to

\[
\sum_{j=1}^{n} a_{1j}x_j = b_1 \quad (1 = 1, 2, \ldots, n).
\]

The trick here is to represent \( \phi_j(x) \) as an indefinite integral which is approximated by a sum over \( k \) fixed intervals.

To see this clearly, let us note first that we may represent \( \phi(x) \) as

\[
\phi(x) = \int_0^x \phi'(u)du + k, \quad \phi(0) = k,
\]

where \( \phi(x) \) convex means \( \phi'(u) \) is nondecreasing and that \( \phi'(u) \) may be approximated by a histogram over some interval

\( 0 < u < c \) where \( c \) is some assumed very large upper bound for \( x \). This is illustrated below.

\[ h_1 \leq h_2 \leq h_3 \ldots \leq h_k \]
Here height of bars \( h_1 \leq h_{i+1} \) follows from the convexity of \( \phi \). We now replace \( x \) by

\[
x = \Delta_1 + \Delta_2 + \ldots + \Delta_n \quad \Delta_1 \geq 0
\]

where \( \Delta_i \) are non-negative variables and \( k \) is the fixed number of intervals into which the range of \( x \) was subdivided. It is easy to see that

\[
\phi(x) = \min \sum_{i=1}^{k} h_i \Delta_i \quad 0 \leq \Delta_i \leq d_i.
\]

Indeed, it is clear that the minimum is attained by choosing \( \Delta_1 = d_1, \Delta_2 = d_2, \ldots \) until the value of \( x \) is exceeded for some \( i \) in which case \( \Delta_i \) is set equal to the residual. Thus the effect of the minimization is to represent \( \phi(x) \) by the area under the histogram up to \( x \).

We now employ this approach to solve (14) and (15). The procedure is to replace \( x \) and \( \phi \) in the linear programming problem by

\[
(16) \quad x_j = \sum_{i=1}^{k} \Delta_{ij} \quad j = 1, 2, \ldots, n
\]

and

\[
(17) \quad \phi_j(x_j) = \sum_{i=1}^{k} h_{ij} \Delta_{ij} \quad 0 \leq \Delta_{ij} \leq d_{ij}.
\]

noting that since a minimum for \( \sum \phi_j(x_j) \) is sought in (14), this implies that the values of \( \Delta_{ij} \) satisfying (16) must, at
the minimum, satisfy \( \sum_{j} h_{1j} A_{1j} = \text{min} \).

This manner of treating convex separable objective functions greatly increases the number of variables without increasing the number of equations (instead of \( x \)'s we have many more \( A \)'s, each with an upper bound). However, it is the number of equations that, as a rule, determine the amount of work in the simplex method. Moreover, it should be noted that there are numerous short cuts possible due to the appearance of several columns with identical coefficients (except for the cost row) so that, in fact, it is quite simple to solve rapidly cases involving a convex separable objective form.

5. LINEAR PROGRAMMING UNDER UNCERTAINTY [2]

An example is a two-stage programming problem with the following structure. For the first stage

\[
\sum_{j=1}^{m} x_{1j} = a_i \quad (x_{1j} \geq 0)
\]

(18)

\[
\sum_{i=1}^{n} b_{1j} x_{1j} = u_j
\]

where \( x_{1j} \) represents the amount of the \( i \)th resource assigned to the \( j \)th destination and \( b_{1j} \) represents the number of units of demand at destination \( j \) that can be satisfied by one unit of resource \( i \). For the second stage
\[(19) \quad d_j = u_j + v_j - s_j \quad (j = 1, 2, \ldots, n)\]

where \(v_j\) is the shortage of supply and \(s_j\) is the excess of supply.

The total cost function is assumed to be of the form

\[c = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}x_{1j} + \sum_{j=1}^{n} \alpha_jv_j\]

i.e., depends linearly on the choice of \(x_{1j}\) and on the shortages \(v_j\) (which depend on assignments \(u_j\) and the demands \(d_j\)).

Our objective will be to minimize total expected costs. Let \(g_j(u_j | d_j)\) be the cost at a destination if the supply is \(u_j\) and the demand is \(d_j\). It is clear that

\[g_j(u_j | d_j) = \begin{cases} a_j(d_j - u_j) & \text{if } d_j \geq u_j \\ 0 & \text{if } d_j < u_j \end{cases}\]

we shall now give a result due to H. Scarf.

**Theorem:** The expected value of \(g_j(u_j | d_j)\), denoted by \(\mathcal{g}_j(u_j)\), is a convex function of \(u_j\).
Proof: Let \( p(d_j) \) be the probability density of \( d_j \); then

\[
\varphi_j(u_j) = a_j \int_{x=u_j}^{+\infty} (x-u_j)p(x)dx
\]

\[
= a_j \int_{x=u_j}^{+\infty} xp(x)dx - a_j u_j \int_{x=u_j}^{+\infty} p(x)dx
\]

whence differentiating \( \varphi(u) \)

\[
\varphi'_j(u_j) = -a_j \int_{x=u_j}^{+\infty} p(x)dx
\]

It is clear that \( \varphi'_j(u_j) \) is a nondecreasing function of \( u_j \) with \( \varphi''_j(u_j) \geq 0 \) and that \( \varphi_j(u_j) \) is convex.

Thus the expected value of the objective function is

\[
\text{Exp } c = \sum_i c_{ij} x_{ij} + \sum_{j=1}^n a_j \varphi_j(u_j)
\]

where \( \varphi_j(u_j) \) are convex functions. Thus the original problem has been reduced to minimizing (20) subject to (18). This permits application of the device already discussed for approximating such a problem by a standard linear programming problem in case the objective function can be represented by a sum of convex functions.
REFERENCES


