Some Comments on 'Linear' Programming with Absolute-Value Functionals

M. R. Rao


Stable URL:
http://links.jstor.org/sici?sici=0030-364X%28197301%2F02%2921%3A1%3C373%3ASCO%27PW%3E2.0.CO%3B2-M

*Operations Research* is currently published by INFORMS.

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.

Please contact the publisher regarding any further use of this work. Publisher contact information may be obtained at http://www.jstor.org/journals/informs.html.

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.

The JSTOR Archive is a trusted digital repository providing for long-term preservation and access to leading academic journals and scholarly literature from around the world. The Archive is supported by libraries, scholarly societies, publishers, and foundations. It is an initiative of JSTOR, a not-for-profit organization with a mission to help the scholarly community take advantage of advances in technology. For more information regarding JSTOR, please contact support@jstor.org.
Some Comments on 'Linear' Programming with Absolute-Value Functionals

M. R. Rao

University of Rochester, Rochester, New York

(Received October 20, 1971)

This note gives simple and straightforward reasoning for obtaining the results of Shanno and Weil [Opns. Res. 19, 120–124 (1971)] for linear programming with absolute-value functionals.

The 'LINEAR' programming problem with absolute-value functionals, as stated by Shanno and Weil,\[5\] is as follows: max \( z = \sum c_j|x_j| \), subject to \( Ax=b \), with some or all \( x_j \) unrestricted in sign.

As they pointed out, the transformation frequently used for each unrestricted variable (e.g., \( x_j \)) in the constraints is to replace it by two nonnegative variables (e.g., \( x^+_j, x^-_j \)) and to give each of the new variables the appropriate \( c_j \) in the objective function. For instance, this approach is utilized in absolute-value regression\[1\] and in linear programming under uncertainty.\[2\] A mathematical analysis is presented in reference 5 to show that the simplex method (with unrestricted basis entry) will converge to an absolute maximum only when \( -c^+ \geq c^- \). The same result was obtained earlier by El Ag icy[3] in connection with two-stage programming under uncertainty.

In order to see this result, one merely has to note that, if the above condition is not satisfied for some variable \( x_k \), the transformed problem (i.e., replace \( x_k \) by \( x^+_k, x^-_k \) and give the corresponding objective function coefficients as \( c^+_k \) and \( c^-_k \), but do not include the restriction \( x^+_k \cdot x^-_k = 0 \)) would have an unbounded solution whenever a feasible solution exists. This is because, once a feasible solution is obtained, we can then increase \( x^+_k \) and \( x^-_k \) by an arbitrarily large but equal amount without violating the constraints. This, however, would make the objective function arbitrarily large, and consequently the solution is unbounded. The fact that the simplex method does not give a finite optimum when \( c^+_k > -c^-_k \) should come as no surprise, since the transformed problem has an unbounded solution. In such a case, as is to be expected, the simplex method identifies an unbounded solution, as shown in reference 5.

An example is given in reference 5 to show that, if some \( c_j \) (corresponding to an unrestricted variable \( x_j \)) is positive, the linear programming problem with absolute-value functionals may have a local optimum that is not necessarily a global optimum. In order to see why this may happen, one merely has to note that, if any \( c_j \) is positive, the objective function (to be maximized) is not concave. If all \( c_j \) are nonnegative, the objective function becomes convex and, if the feasible region is strictly bounded, there exists an optimal solution that is an extreme point.\[4\] However, adjacent-extreme-point methods would fail in general because of local optima.
A Note on Reinverting the Dantzig-Wolfe Type Decomposed LP Basis

William J. Reich, Jr.

Falstaff Brewing Corporation, St. Louis, Missouri

(Received April 5, 1971)

This note reports an efficient routine that has been developed and successfully applied to the reinversion of decomposed LP bases of the Dantzig-Wolfe type. In sum, if \( r \) is the total number of subproblems in the original LP problem and \( k \) is the number of subproblem vectors in the decomposed basis, then only \( k - r \) Gauss-Jordan iterations are needed to invert the decomposed basis. The larger \( r \), the more efficient and accurate the routine. This reduction in the number of Gauss-Jordan iterations is accomplished by finding the inverse of the wrong basis rapidly, and then making the necessary corrections to get the inverse of the right basis.

This NOTE describes an efficient routine for reinverting decomposed LP bases of the Dantzig-Wolfe type. The routine is 'efficient' in the sense that it improves accuracy by eliminating much of the computation necessary for inversion. Given the problem,

\[
\begin{align*}
\sum_{j=1}^{r} A_j x_j + I_m x_s &= b_0, \\
B_j x_j &= b_j, \\
\sum_{j=1}^{r} c_j x_j &= \min z,
\end{align*}
\]

where \( r \) is the number of subproblems; \( A_j \) is \( m \) by \( n_j \) for \( j = 1, 2, \ldots, r \); \( B_j \) is \( m \) by \( n_j \); \( b_0 \) has \( m \) components; \( b_j \) has \( m \) components; \( I_m \) is an \( m \times m \) identity matrix; \( x_s \) is a vector of \( m \) slack variables.

This 'decomposes' to,

\[
\begin{align*}
\sum_{j=1}^{r} \sum_{k=1}^{k_{ji}} P_{k_j} A_{j_k} x_{j_k} + I_m x_s &= b_0, \\
\sum_{k=1}^{k_{ji}} P_{k_j} &= 1, \quad (j = 1, 2, \ldots, r)
\end{align*}
\]