Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Bounds on the Opportunity Cost of Neglecting Reoptimization in Mathematical Programming

Osman Oğuz,

Bounds on the Opportunity Cost of Neglecting Reoptimization in Mathematical Programming

Osman Oğuz
Department of Industrial Engineering, Bilkent University, 06533 Ankara, Turkey
ooguz@bilkent.edu.tr

Postoptimality or sensitivity analysis are well-developed subjects in almost all branches of mathematical programming. In this note, we propose a simple formula which can be used to get preliminary bounds on the value of this type of analysis for a specific class of mathematical programming problems. We also show that our bounds are tight.
(*Sensitivity Analysis*; *Tolerance Limits*; *Worst-Case Analysis*)

1. Introduction

Data imprecision or variation is a source of great concern in linear and nonlinear programming. It has led to many forms of sophisticated mathematical analysis, especially in the case of linear programming. The most prominent among these is sensitivity analysis. Strictly speaking, it is the analysis of the effect of input data changes on an optimal solution of a linear program. All well-known books on the subject have chapters on sensitivity and/or post optimality analysis, e.g., Dantzig (1965), Bazaraa and Jarvis (1977), Chvatal (1983).

In the case of nonlinear programming, similar topics are discussed under names like perturbation analysis, stability analysis, or parametric analysis. See the book *Sensitivity and Stability Analysis in Nonlinear Programming* by A. V. Fiacco (1983) for a unified treatment. A collection of articles in a special issue of *Journal of Optimization Theory and Applications* (JOTA) also edited by A. V. Fiacco (1986) may be of interest in this area. Interval analysis may also be considered in this realm. Ratscheck and Voller (1991) provide a recent review of this subject.

More recent extensions of sensitivity analysis are multicriteria analysis, linear programming with interval objective function coefficients, and tolerance analysis as reflected in the works of Zeleny (1974), Yu and Zeleny (1976), Steuer (1981), and Wendell (1985).

The work of Wendell (1985) on tolerance limits for linear programming is similar in some ways to our approach in this study. More recently, he gave extensions and generalizations of his approach (1990, 1997). His tolerance limits are intervals for the objective function and the right-hand-side coefficients for which an existing optimal basis will remain optimal. More specifically the linear program:

$$\text{Maximize} \quad \sum_{j=1}^{n} (c_j + \alpha_j c_j')x_j,$$

$$\text{St:} \sum_{j=1}^{n} a_{ij}x_j = b_i \quad \text{for } i = 1, \ldots, m,$$

$$x_j \geq 0,\ j = 1, \ldots, n, \tag{1}$$

or

$$\text{Maximize} \quad \sum_{j=1}^{n} c_j x_j$$

$$\text{St:} \sum_{j=1}^{n} a_{ij}x_j = b_i + \beta_i b_i' \quad \text{for } i = 1, \ldots, m,$$

0025-1909/00/4607/1009$05.00
1526-5501 electronic ISSN

$$x_j \geq 0,\ j = 1, \ldots, n, \tag{2}$$

are the problems dealt with. Given an optimal basis for $\alpha_j = 0$ for $j = 1, \ldots, n$, or $\beta_i = 0$ for $i = 1, \ldots, m$, the maximum value of $\alpha \geq 0$ or $\beta \geq 0$ is sought such that whenever $-\alpha \leq \alpha_j \leq \alpha$ or $-\beta \leq \beta_i \leq \beta$ for each $j = 1, \ldots, n$, or $i = 1, \ldots, m$, the optimal basis remains unchanged. α and β are called "the maximum allowable tolerances" on variations in the values of c_j's and b_i's. Also, simple formulas are provided for computing the values of α and β using the information from the optimal simplex tableau. Bradley et al.'s (1977) "100% rule" which gives similar, easier to compute bounds, is another approach in this line.

Here, in this study, we look at the problem created by data changes from another point of view. Rather than determining ranges of data which allows a known optimal solution to remain optimal, we try to find a bound on the resulting loss when we stick to the known solution regardless of the changes in the data. We explain the logic behind the bound and its implications in the next section.

2. Derivation of the New Bound

The problem dealt with has the following general form:

$$\text{Max}\{z = CX | X \in S\}, \tag{3}$$

where C is a $(1 \times n)$ vector of nonnegative cost coefficients, X is an $(n \times 1)$ vector of decision variables, and S is an arbitrary closed and bounded, nonempty set in R^n_+, i.e., there are no assumptions of convexity for S. Also, the components of the vector C which may be equal to zero remain fixed at zero throughout the analysis to follow, i.e., no changes in value are allowed for these coefficients. These restrictions on the cost vector cause a significant loss of generality; however, there are still a wide range of problems like TSP, many job shop and project scheduling problems, knapsack problems, etc., where the results to be presented apply. We could convert negative cost coefficients into positive ones using the complements of the corresponding variables. We define the complement of a variable x_j as $u_j - x_j$, where u_j is the upper bound on x_j which may be computed by solving:

$$\text{Maximize} \quad u_j = \{x_j | X \in S\}, \tag{4}$$

if they are not given explicitly. However, the interpretation of the bounds to be derived becomes rather difficult because of the resulting negative constant in the objective function in that case.

Consider now the following two instances of the above problem:

$$\text{Maximize} \quad z_1 = \{C^1X | X \in S\}, \tag{5}$$

and

$$\text{Maximize} \quad z_2 = \{C^2X | X \in S\}. \tag{6}$$

Also let X_1^* and X_2^* be optimal solutions (not necessarily unique) of these two problems with values $z_1 = C^1X_1^*$ and $z_2 = C^2X_2^*$ correspondingly. We can assume $z_1 > 0$ without loss of generality in the analysis to follow. Recall that we made the assumption that $c_i^1 = 0$ implies $c_i^2 = 0$.

Proposition. *If*

$$\frac{|c_i^1 - c_i^2|}{c_i^1} \leq \delta \tag{7}$$

for all i such that $c_i^1 \neq 0$, *then*

$$\frac{z_2 - z_3}{z_2} \leq \frac{2\delta}{1 + \delta} \tag{8}$$

where $z_3 = C^2X_1^*$.

Proof. The following is true by the definition of δ:

$$(1 - \delta)C^1 \leq C^2 \leq (1 + \delta)C^1.$$

Postmultiplying the left inequality by X_1^*, and the right inequality by X_2^* we obtain:

$$(1 - \delta)C^1X_1^* \leq C^2X_1^* \quad \text{and} \quad C^2X_2^* \leq (1 + \delta)C^1X_2^*.$$

The relationships $C^2X_1^* \leq C^2X_2^*$ and $C^1X_2^* \leq C^1X_1^*$ follow directly from the optimality of X_1^* and X_2^*. The two scalar inequalities obtained above, together with these relationships are sufficient to give the following:

$$(1 - \delta)z_1 \leq z_3 \leq z_2 \leq (1 + \delta)z_1.$$

The sought result is then obtained as a direct consequence of this string of inequalities as follows:

$$\frac{z_3}{z_2} \geq \frac{(1-\delta)z_1}{(1+\delta)z_1}, \tag{9}$$

and hence

$$1 - \frac{z_3}{z_2} \leq 1 - \frac{(1-\delta)z_1}{(1+\delta)z_1}, \tag{10}$$

which simplifies to:

$$\frac{z_2 - z_3}{z_2} \leq \frac{2\delta}{1+\delta} \quad \square \tag{11}$$

If we were to minimize instead of maximize, all other conditions and assumptions remaining the same, we would follow a similar line of reasoning in the proof to obtain:

$$\frac{z_3 - z_2}{z_3} \leq \frac{2\delta}{1+\delta}, \tag{12}$$

or, if we wish to have the same denominator

$$\frac{z_3 - z_2}{z_2} \leq \frac{2\delta}{1-\delta}. \tag{13}$$

Note that there are no restrictions on the form of the set S. Thus, the bounds obtained above apply to any mathematical programming problem with a nonnegative linear objective function. We can also apply the proposition to the dual of the problem in the case of linear programming, to determine limits on the change of the value of the objective function as a result of right-hand-side coefficient changes. We must assume the dual to have the form Minimize CX, subject to $AX \geq b$ where $b \geq 0$ for this case.

Example. Suppose that X^* is an optimal solution to an optimization problem with CX as the objective function. Also suppose that, for some reason, the coefficient vector has changed to C', but the maximum deviation of the components of C' from those of C is less than 5%. Then, by the proposition stated above, $C'X^*$ will differ from the optimum $C'X$ by at most $[(2 \times .05)/(1 + .05)] \times 100 = 9.52\%$.

Consider the linear program:

$$\text{Max}(1+\epsilon)x_1 + (1-\epsilon)x_2$$

$$\text{st: } x_1 + x_2 \leq 1 \quad \text{and} \quad x_1, x_2 \geq 0. \tag{14}$$

$(x_1 = 0, x_2 = 1)$ is an optimal solution of this linear program for $\epsilon = 0$. The bound indicated by the proposition above is realized exactly if we keep using $(x_1 = 0, x_2 = 1)$ as the optimal solution for any value of $\epsilon > 0$. This demonstrates that the bound is tight.

3. The New Bound and Wendell's Tolerance Limits

We have noted Wendell's (1985) tolerance limits approach among the most important sensitivity analysis techniques. Here we would like to point out the fact that our approach and his are complementary to each other. Suppose that Wendell's tolerance limit bound parameter α as explained in the first section of this paper is computed for $c_j = c'_j$ for $j = 1, \ldots, n$, and its value is greater than or equal to our parameter δ for a specific linear programming problem in the class discussed in this study. Then, obviously, our bound is redundant, because Wendell's bound tells us that $z_2 = C^2X_2^*$ will be equal to $z_3 = C^2X_1^*$, i.e., $X_1^* = X_2^*$ in our terminology.

Suppose on the other hand, we have $\delta > \alpha$. Then we can use this fact to tighten our bounds in the following manner. Wendell's limits allow us to change any c_j^1 to either $(1+\alpha)c_j^1$ or $(1-\alpha)c_j^1$, without affecting the optimality of the existing solution X_1^*. So, we can replace c_j^1 by $(1+\alpha)c_j^1$ whenever $((c_j^2 - c_j^1)/c_j^1) > \alpha$, and by $(1-\alpha)c_j^1$ when $((c_j^1 - c_j^2)/c_j^1) > \alpha$, and set $c_j^1 = c_j^2$ otherwise. After these replacements, the new vector C^1, comes closer to C^2, in other words, the deviations of the components of C^2 from those of C^1 become smaller, and that enables us to compute our bound with a smaller value of δ. The new δ denoted by δ' is not exactly equal to $\delta - \alpha$ as explained below.

Let us assume that $\delta = |c_{j^*}^1 - c_{j^*}^2|/c_{j^*}^1$, where j^* is the variable index with the maximum associated ratio. Although the possibility that the index j^* changes in the replacement procedure discussed above, let us assume it remains invariant to make the exposition short and simple. We know that $c_{j^*}^2 = (1+\delta)c_{j^*}^1$ or $c_{j^*}^2 = (1-\delta)c_{j^*}^1$ holds as a result. As a consequence of the replacement described above, $c_{j^*}^1$ is replaced by either $(1+\alpha)c_{j^*}^1$ or $(1-\alpha)c_{j^*}^1$. We can recompute the new value of δ, i.e., δ' as:

$$\delta' = ((1+\delta)c^1_{j^*} - (1+\alpha)c^1_{j^*})/((1+\alpha)c^1_{j^*})$$

or

$$\delta' = ((1-\alpha)c^1_{j^*} - (1-\delta)c^1_{j^*})/((1-\alpha)c^1_{j^*})$$

which gives:

$$\delta' = (\delta - \alpha)/(1+\alpha) \quad \text{or} \quad (\delta-\alpha)/(1-\alpha).$$

Relaxation of the assumption about the invariance of j^* would make obtaining a formula which gives the value of δ' in terms of δ more complicated. We omit deriving such a formula because our purpose was to show that δ' is not equal to $\delta - \alpha$, and we believe that the evidence provided is sufficient to show that fact.

To illustrate, consider the first numerical example discussed in the preceding section; if the value of α is computed as being equal to 3%, then we would use δ' = .02/1.03 or .02/.97 instead of .05 and g[illegible]n bound of 3.81% or 4.04% instead of 9.52%. For small values (like in this example) of α, one can set new δ equal to $\delta - \alpha$ without losing much accuracy in the calculation of the bound.

We have to note, however, that this sort of bound tightening is limited only to linear programming and cannot be used in integer programming, for example, since the tolerance limits are not readily available with the optimal solution in integer programming.

4. Conclusions and Remarks

We have tried to explain a new bound for use in sensitivity and worst-case analysis for optimization problems. It has informative value in its own right by stating that small perturbations in the objective function coefficients of some mathematical programming problems cannot put the current optimum solution relatively too far off the true optimal value to be computed after the perturbations, as long as the objective function is linear with nonnegative coefficients. In fact, if the perturbations are small, and if there are some transaction costs in implementing a new solution (which might be the case in changing portfolios for example), one may well be justified in sticking to the present solution, considering that the expected gain from reoptimization may be negligible.

This has some implications for the use of approximation or heuristic algorithms for obtaining near optimal solutions to some difficult problems which may require excessive computation times to find the optimum. Suppose one has a large traveling salesman problem which must be solved repetitively for some frequently changing cost coefficients. Solving the problem once to optimality, and using it for as long as the cost changes are within some prespecified limits, may well be a better alternative than using the heuristic or approximation algorithms frequently, knowing the worst-case bounds which come with most such algorithms.

The simplicity of the bound is another advantage. One may not need even a calculator to determine the proposed bounds. The ease of obtaining make them good candidates for being used as preliminary guidelines for more sophisticated sensitivity analysis techniques.[1]

[1] The author wishes to thank an associate editor for suggestions which improved the presentation of the main proof.

References

Bazaraa, M. S., J. J. Jarvis. 1977. *Linear Programming and Network Flows*. John Wiley & Sons, New York.

Bradley, S. P., A. C. Hax, T. L. Magnanti. 1977. *Applied Mathematical Programming*. Addison-Wesley, Reading, MA.

Chvatal, V. 1983. *Linear Programming*. Freeman and Co., New York.

Dantzig, G. B. 1963. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ.

Fiacco, A. V. 1983. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*. Academic Press, New York.

——, ed. 1986. Special issue on mathematical programming with data perturbations. *J. Optim. Theory and Appl.* **48**.

Ratschek, H., R. L. Voller. 1991. What can interval analysis do for global optimization? *J. Global Optim.* **1** 111–130.

Steuer, R. E. 1981. Algorithms for linear programming with interval objective function coefficients. *Math. Oper. Res.* **6** 333–348.

Ward, J. E., R. E. Wendell. 1990. Approaches to sensitivity analysis in linear programming. *Ann. Oper. Res.* **27** 3–38.

Wendell, R. E. 1985. Tolerance approach to sensitivity analysis in linear programming. *Management Sci.* **31** 564–578.

——. 1997. Linear programming: The tolerance approach. T. Gal, H. J. Greenberg, eds. *Advances in Sensitivity Analysis and Parametric Programming*. Kluwer, Boston, MA.

Yu, P. L., M. Zeleny. 1976. Linear multiparametric programming by multicriteria simplex method. *Management Sci.* **23** 159–170.

Zeleny, M. 1974. *Linear Multiobjective Programming*. Springer-Verlag, New York.

Accepted by Thomas Liebling; received November 1997. This paper was with the author 9 months for 3 revisions.