Applied Mathematical Sciences, Vol. 2, 2008, no. 5, 241 - 248

A Note on the Linear Programming Sensitivity Analysis of Specification Constraints in Blending Problems

Umit Akinc

Calloway School of Business and Accountancy
Wake Forest University, Winston-Salem NC, 27109, USA
Akinc@wfu.edu

Abstract

In blending problems there are typically specification constraints that limit the content of various properties of the blend that it acquires from the ingredients to certain maximum or minimum percentages of the total blend. For sake of linearity, these constraints are commonly included in the problem in a way that precludes direct sensitivity analysis with respect to changes in these percentages. This note shows that the sensitivity analysis with respect to changes in the target specification percentages can be derived from the ordinary sensitivity analysis with minimum additional effort; and that common LP codes can be slightly modified to facilitate sensitivity analysis of blending percentages.

Mathematics Subject Classification: 90C31

Keywords: Linear Programming, Blending Problems, Sensitivity Analysis.

Through its history, one of the most frequently cited application examples of linear programming has been the so-called "blending," problem in which various ingredients (inputs) are mixed into one or more blends (outputs) to satisfy some objective [1]. Blending problems, among other types of constraints, may include the "specification" constraints. These constraints limit certain properties (such as moisture) of the blend, which it inherits from the ingredients, to certain minimum or maximum percentages of the total. Most common LP software output does not allow direct sensitivity analysis of the target percentages. The purpose of this note is to show how simplex-based LP software may be slightly modified to enable direct and full sensitivity analysis of these target percentages.

A typical such constraint, say the k^{th} constraint of the problem, may be written generally as:

$$\sum_{j \in S_i} s_j x_j \Big/ \sum_{j \in S_i} x_j \le p \tag{1}$$

where s_j is the percentage of the property in question contained in ingredient j, and S_i is the subset of all ingredients which may be used in blend i. By selecting appropriate zeros and ones for s_j, Eq (1) can also model a common type of specification constraint—one which limits the percentage of a certain subset of ingredients in the blend. Since this form is not linear and thus unsuitable to linear programming it is equivalently included in the LP as:

$$\sum_{j \in S_i} s_j x_j - p \sum_{j \in S_i} x_j \leq 0 \qquad (2)$$

However this transformation recasts the sensitivity with respect to the RHS of Eq (1) into the sensitivity with respect to several technological constraints in Eq (2). This more difficult form of sensitivity has been studied extensively as part of LP theory. See for instance Simonnard [2, p. 145]. However, the theoretical results concerning the technological constraints have not been applied to the sensitivity analysis of target percentages; nor they have been implemented in the most commonly used LP software. Yet in many blending and mixing problems this parameter, p may be set as a matter of management policy and thus effect of adopting different policies on the optimal LP value may be quite a valuable guide.

The optimal dual price for the modified constraint, say d_k, gives the rate of change in the objective function as the right hand side (RHS) of (2) is perturbed within the range in which the optimal basis and thus d_k does not change (allowable range). Let $\Delta z / \Delta r$ denote this rate, where z is the objective function value and r is the RHS of (2) (currently zero); and Δ_r^+ and Δ_r^- denote the allowable increase and decrease in the RHS respectively. Although d_k gives some useful sensitivity information pertaining to the current optimal solution, it does not directly answer the more legitimate question of the behavior of the optimal solution for changes in the parameter *p,* the RHS of (1). The linear form (2) with a RHS of Δr is

$\sum_{j \in S_i} s_j x_j - p \sum_{j \in S_i} x_j \le \Delta r$. It can be re-written as $\sum_{j \in S_i} s_j x_j / \sum_{j \in S_i} x_j \le p + (\Delta r / \sum_{j \in S_i} x_j)$

which, in turn, means that changing the RHS of Eq (2) by Δr is equivalent to changing the RHS of (1) by $\Delta r / \sum_{j \in S_i} x_j$. Therefore we have:

$$\Delta p = \Delta r / \sum_{j \in S_i} x_j \qquad (3)$$

$$\Delta_p^+ = \Delta_r^+ / \sum_{j \in S_i} x_j \text{ and } \Delta_p^- = \Delta_r^- / \sum_{j \in S_i} x_j \qquad (4)$$

$$\Delta z / \Delta p = (\Delta z / \Delta r) \sum_{j \in S_i} x_j . \qquad (5)$$

Suppose that the optimal solution to the blending problem, with Eq. (2) and its sensitivity analysis information is available from a standard LP program. Although $\Delta z / \Delta r = d_k$ is constant in the allowable interval*;* $\Delta z / \Delta p$ may not be constant, that is *z(p)* may be non-linear in *p*. The behavior of the optimal solution vector and the objective function value, as *p* changes can be estimated by simply evaluating the quantity $d_k \sum_{j \in S_i} x_j$ at the current optimal solution x_j^o. However, this would only be approximate, because as the RHS of Eq (2) changes within the allowable range, while the optimal basis and d_k stay constant, the optimal x_j and thus $\Delta z / \Delta p$ may not. The quality of this approximation depends on the magnitude of the change in the quantity $\sum_{j \in S_i} x_j$. While in some cases this change might be small and can be ignored, in others it might be considerable enough to significantly distort the sensitivity results with respect to parameter, *p*.

The standard LP solution with Eq (2) however, can be used to perform a full and exact sensitivity analysis of the solution for changes in the parameter, *p*. Let $\mathbf{x}(\Delta r)$ be the vector of solution values as a function of the change in the RHS of (2), vector **b** the original right hand sides of the LP with Eq (2), and **B** the current optimal basis. We have:

$$\mathbf{x}(\Delta r) = \mathbf{B}^{-1}(\mathbf{b} + \mathbf{u}_k \Delta r) = \mathbf{x^o} + \mathbf{B}^{-1}\mathbf{u}_k \Delta r, \tag{6}$$

where $\mathbf{u}_k$ is the k^{th} unit vector. In (6), the term, $\mathbf{B}^{-1}\mathbf{u}_k$ traces out the k^{th} column of the optimal basis inverse. Let us denote the elements of this column by β_{jk}. Therefore, we have: $\sum_{j \in S_i} x_j(\Delta r) = \sum_{j \in S_i} x_j^o + \Delta r \sum_{j \in \bar{S}_i} \beta_{jk}$, where $\bar{S}_i$ is the subset of S_i corresponding to the currently basic vectors. With this information the exact limits of p, Δ_p^- and Δ_p^+ are obtained as:

$$\Delta_p^- = \Delta_r^- / (\sum_{j \in S_i} x_j^o + \Delta_r^- \sum_{j \in \bar{S}_i} \beta_{jk}) \text{ and } \Delta_p^+ = \Delta_r^+ / (\sum_{j \in S_i} x_j^o + \Delta_r^+ \sum_{j \in \bar{S}_i} \beta_{jk}). \tag{7}$$

Also the exact rate of change in the objective function value, as the RHS of Eq. (1) changes, may be written as: $\Delta z / \Delta p = \Delta z / \Delta r \,(\sum_{j \in S_i} x_j^o + \Delta r \sum_{j \in \bar{S}_i} \beta_{jk})$. To express this rate in terms of Δp rather than Δr , we can solve $\Delta r = \Delta p\,(\sum x_j^o + \Delta r \sum \beta_{jk})$ for Δr and substitute in above to give:

$$\Delta z / \Delta p = d_k \left[\sum_{j \in S_i} x_j^o + \sum_{j \in S_i} \beta_{jk} \left(\frac{\Delta p \sum x_j^o}{1 - \Delta p \sum \beta_{jk}} \right) \right], \tag{8}$$

which is the average rate of change in z, as *p* changes by an amount Δ*p*. The optimal value of the LP as *p* changes by Δp within the allowable range, can be obtained by multiplying (8) through by Δp and simplifying

$$z(\Delta p) = z^o + d_k \left[\frac{\Delta p \sum x_j^{\,o}}{1 - \Delta p \sum \beta_{jk}} \right]. \tag{9}$$

In Eq (9), when $\Delta p \sum \beta_{jk} = 1$, $z(\Delta p)$ becomes undefined. However this troublesome situation does not occur, because $1 - \Delta p \sum \beta_{jk} > 0 \quad \forall \quad \Delta_p^- \le \Delta p \le \Delta_p^+$. To prove this, assume that $\sum \beta_{jk} < 0$. As Δp changes in the negative direction from 0, it reaches Δ_p^- (≤ 0) before it becomes $1/\sum \beta_{jk}$, that is to say $\Delta_p^- > 1/\sum \beta_{jk}$. This is easily shown by substituting $\Delta_r^- /(\sum x_j^o + \Delta_r^- \sum \beta_{jk})$ for Δ_p^- and rearranging terms to get $\frac{\Delta_r^- \sum \beta_{jk}}{\sum x_j^o + \Delta_r^- \sum \beta_{jk}} < 1$. This is true whenever $\sum x_j^o > 0$, since both $\sum \beta_{jk}$ and Δ_r^-

are non-positive. Furthermore, since $1/\sum\beta_{jk} < \Delta_p^- \le \Delta p$ we have $1-\Delta p\sum\beta_{jk} > 0$. A similar argument holds for the case $\sum\beta_{jk} > 0$.

It is also straightforward to examine the functional behavior of $z(\Delta p)$ as p changes in the allowable range. The first "derivative" of $z(\Delta p)$ is $d_k\frac{\sum x_j^o}{(1-\Delta p\sum\beta_{jk})^2}$ and thus has the same sign as d_k. Thus $z(\Delta p)$ is–non-decreasing if $d_k \ge 0$; non-increasing otherwise. The "second derivative" is $2d_k\frac{\sum\beta_{jk}\sum x_j^o}{(1-\Delta p\sum\beta_{jk})^3}$ which implies that if $\sum\beta_{jk} = 0$, then $z(\Delta p)$ is linear in the allowable interval; also since $\sum x_j^o$ and $1-\Delta p\sum\beta_{jk}$ are both non-negative, $z(\Delta p)$ is convex if d_k and $\sum\beta_{jk}$ have the same sign, concave otherwise.

The above suggests that standard LP software packages can be modified slightly to enable users to conduct exact sensitivity analyses of specification-type constraints. All that is required, possibly as a user selectable option, is to report the $\sum\beta_{jk}$ quantities corresponding to those constraints that the user has coded as "specification-type" during input.

REFERENCES

[1] George Dantzig, *Linear Programming and extensions.* Princeton University Press, Princeton, N.J., 1963

[2] Michel Simmonard, *Linear Programming.* Prentice Hall, Englewood Cliffs, N.J., 1966.

Received: June 5, 2007