CHAPTER 4 - Simplex Method Revisited

Section 4.14: Unrestricted-in-Sign Variables.

- Every LP we have considered up until this point considered non-negative decision variables. The simplex algorithm can only be applied for non-negative variables; otherwise the ratio test breaks down.
- There is a simple technique for converting the unrestricted-in-sign variable to a set of two variables that are both non-negative:

$$x_i = \text{urs} \xrightarrow{\text{define}} x_i = x_i' - x_i'' \quad \text{where } x_i' \geq 0,\ x_i'' \geq 0$$

Substitute the difference $x_i' - x_i''$ in for every instance of x_i.

- It turns out (we'll see why soon) that x_i' or x_i'' (or both) are zero i.e. both cannot be non-zero.

Case 1: $x_i' > 0$ and $x_i'' = 0$ $\longrightarrow$ $x_i = x_i'$

Case 2: $x_i' = 0$ and $x_i'' > 0$ $\longrightarrow$ $x_i = -x_i''$

Case 3: $x_i' = 0$ and $x_i'' = 0$ $\longrightarrow$ $x_i = 0$

WS #1 working with a basic urs example. Solve using simplex.

WS #2 Practical use of urs in an inventory model.

Section 4.1b: Goal Programming.

- Up until this point, most LPs that we have formulated have had well-defined (with some degree of certainty) objective functions. Further, the feasible space may lend an optimal solution to this objective.

- In some cases, the objective may not be clear or there may be multiple objectives with various degrees of priority. Further, the feasible space may not allow for every objective to be met. In this scenario, the decision maker may be faced with defining a satisfactory decision vs. can we maximize an alternative objective as a "satisficing" decision?

- In short, if there is no clear objective but we want to prioritize particular constraints, we implement deviational variables into the constraints to create "wiggle room" that allows the decision maker to satisfy relaxed conditions that will suffice at the end of the day.

- Deviational variables: Suppose we have a budget constraint of the form

$$10X_1 + 30X_2 \leq 10{,}000 \quad \text{(Budget)}$$

This constraint may be too restrictive on a particular objective so we "relax" the constraint by introducing s_i^- and s_i^+:

$$10X_1 + 30X_2 + s_1^- - s_1^+ = 10{,}000$$

Cannot both be nonzero.

- s_i^- → budget slack: if $s_i^- > 0$ → satisfied original constraint.
- s_i^+ → budget excess: if $s_i^+ > 0$ → violated over original constraint

Introduction of these variables allows for meeting or violating the constraint set will (which may help other goals).

Note: if meeting the budget is a priority, then we would want to minimize s_i^+ in our objective function!

- There are two methods for formulating a goal programming (GP) problem:

i) Weights Method: Suppose GP has n goals such that

$$\text{Minimize } G_i, \text{ for } i = 1, 2, 3, \ldots, n$$

Then we construct Z as

$$\text{Minimize } Z = w_1 G_1 + w_2 G_2 + \cdots + w_n G_n$$

where the weights reflect the preference regarding relative importance of each goal.

Ex: $w_1 = w_2 = \cdots = w_n = 1$ ← All goals are of equal importance.

ex: $w_1 > w_2 > w_3 > \cdots > w_n$ ← Goal 1 is most important.

G_{ri} are specific deviational variables that are implemented in the constraints. *this method is subjective and may lead to less than "optimal" solutions.

WS #1 Formulate the GP with weights method.

2.) Preemptive Method: Identify priority of goals:

$$G_1 \succ G_2 \succ \cdots \succ G_n$$

Formulate a sequence of LPs that solve each goal sequentially:

$$\begin{aligned} &\text{Minimize } G_{r1} = P_1 && (LP_1)\\ &\text{Minimize } G_{r2} = P_2 && (LP_2)\\ &\vdots \qquad\qquad \vdots && \vdots\\ &\text{Minimize } G_{rn} = P_n && (LP_n) \end{aligned}$$

Subject to appropriate constraints. Each P_n is some deviational variable (s_i^- or s_i^+). After an LP is solved, the value of P_i is placed as an equation constraint in the next LP.

WS #2 Formulate the GP with preemptive method.

WS #1 Formulate a GP using both methods.