Linear Programming Notes: Simplex Algorithm Summary

The Simplex Algorithm.

- Step I:

  Write system in **basic form**.

- Step II:

  Check Row 0. If all entries are non-negative, STOP. You have a solution. If there is a negative entry in Row 0, pick a negative entry (it does not matter which one). If the entry in the array row i column j is called a_{ij}. Call the column you pick column j^* (so that $a_{ij^*} < 0$. j^* is the pivot column.

- Step III:

  Check entries in pivot column. If all entries are non-positive, STOP. Your problem is unbounded. If there is a positive entry in Column j^*, pick a pivot row (it **does** matter which one) using the minimum ratio rule: Denote the entry in Row i of the value column by v_i. Since the system is in basic form, $v_i \geq 0$. Pivot on Row i^* where $\frac{v_{i^*}}{a_{i^*j^*}} \leq \frac{v_i}{a_{ij^*}}$ for all i such that $a_{ij^*} > 0$.

  In other words: Look at all of the numbers in Column j^*. If none are positive, stop (unbounded problem). If at least one is positive, pick the Row that minimizes the ratio between the Row's value and the Row's entry in Column j^*.

  (If there is a tie in the minimum ratio, then pick any row that minimizes the ratio.)

- Step IV:

  Pivot on Row i^* and Column j^*. That is, rewrite the system so that Column j^* has a one as its entry in Row i^* and zeros elsewhere. Algebraically, the new Row i^* is equal to $a_{i^*j^*}{-1}$ times the old Row i^*, and all other Rows are updated by the formula:

  $$\text{New Row } i = \text{Old Row } i - \frac{a_{ij^*}}{a_{i^*j^*}} \text{ times Old Row } i^*.$$

- Step V:

  Return to Step II. Your system should be in basic form; the basis should differ from the previous basis in exactly one way (now variable j^* is in the basis); the value of the objective function should be larger than in the previous step.

The Algorithm cannot repeat the same basis twice (because then it would give the same value twice, but the value increases in each step). So it terminates in a finite number of steps (because there are only a finite number of possible bases), either with a solution or with proof that the problem is unbounded.

Remember, an array is in basic form if it represents the linear programming problem as a system of equations involving non-negative variables; the objective is to maximize x_0; the array has a basis; and the value column in non-negative.

Loose Ends

1. Can you always write an L.P. in basic form?

   No. You can write an L.P. in basic form if and only if the L.P. is feasible. If the problem is not feasible, then you cannot write the problem in basic form (because the basic form provides a feasible solution to the constraints). If the problem is feasible, then you can write the problem in basic form. This requires some work, but Excel does it automatically.

   If the problem is written in the form: $\max x_0$ subject to $Ax \le b$, $x \ge 0$ **and** $b_i > 0$ for all i (all right-hand sides positive), then it is easy to write the problem in basic form (by adding slack variables). This is the only kind of problem that I will ask you to solve by hand.

2. When there are many negative entries in Row 0, does it matter which one I pick?

   No. They will all lead to the solution eventually.

3. What happens when I want to pivot in column j^* but all of the entries are negative or zero?

   The problem is unbounded. You can increase x_{j^*} as much as you want, continuing to increase x_0, without violating feasibility.

4. Why is the minimum ratio rule important?

   If you do not pivot according to this rule, you will not be able to write the system in basic form (either you will not have a basis or an entry in the value column will become negative).

5. Why does the value column stay positive and the value of the objective function increase after every step?

   These things happen because of the clever way we chose to pivot. As long as the entries in the value column are positive, each pivot strictly increase x_0. As long as the ratio between value column and Column j^* has a unique minimum, pivot operations maintain strict positivity of the value column.

6. What happens if there is more than one way to satisfy the minimum ratio rule?

   This leads to a zero in the value column. In principle, this could ruin the algorithm: you might pivot in such a way as to leave the value of the objective function unchanged and therefore it is possible to cycle through the same set of bases over and over again. In theory, one solves this by making the pivot rule more complicated. In practice, it is not a problem. In 172A, you don't need to worry about it.