9.4: Simplex Optimization

One strategy for improving the efficiency of a searching algorithm is to change more than one factor at a time. A convenient way to accomplish this when there are two factors is to begin with three sets of initial factor levels arranged as the vertices of a triangle (Figure $9.4.1$. After measuring the response for each set of factor levels, we identify the combination that gives the worst response and replace it with a new set of factor levels using a set of rules. This process continues until we reach the global optimum or until no further optimization is possible. The set of factor levels is called a **simplex**. In general, for k factors a simplex is a $k+1$ dimensional geometric figure [see Spendley, W.; Hext, G. R.; Himsworth, F. R. *Technometrics* **1962**, *4*, 441–461, and Deming, S. N.; Parker, L. R. *CRC Crit. Rev. Anal. Chem.* **1978** *7(3)*, 187–202]. Thus, for two factors the simplex is a triangle. For three factors the simplex is a tetrahedron.

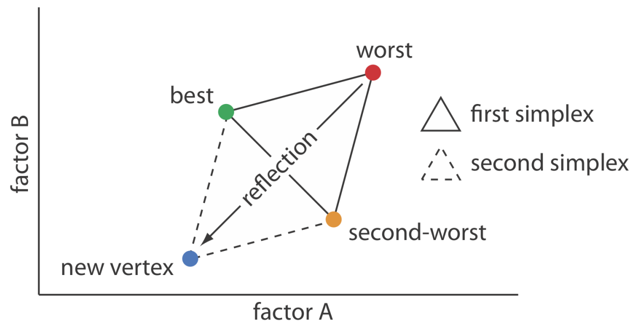

Figure 9.4.1. Example of a two-factor simplex. The original simplex is formed by the green, orange, and red vertices. Replacing the worst vertex with a new vertex moves the simplex to a new position on the response surface.

To place the initial two-factor simplex on the response surface, we choose a starting point (a, b) for the first vertex and place the remaining two vertices at (a + s_a, b) and (a + 0.5s_a, b + 0.87s_b) where s_a and s_b are step sizes for factor A and for factor B [see, for example, Long, D. E. *Anal. Chim. Acta* **1969**, *46*, 193–206]. The following set of rules moves the simplex across the response surface in search of the optimum response:

Rule 1. Rank the vertices from best (v_b) to worst (v_w).

Rule 2. Reject the worst vertex (v_w) and replace it with a new vertex (v_n) by reflecting the worst vertex through the midpoint of the remaining vertices. The new vertex's factor levels are twice the average factor levels for the retained vertices minus the factor levels for the worst vertex. For a two-factor optimization, the equations are shown here where v_s is the third vertex.

$$a_{v_n} = 2\left(\frac{a_{v_b} + a_{v_s}}{2}\right) - a_{v_w}$$

$$b_{v_n} = 2\left(\frac{b_{v_b} + b_{v_s}}{2}\right) - b_{v_w}$$

Rule 3. If the new vertex has the worst response, then return to the previous vertex and reject the vertex with the second worst response, (v_s) calculating the new vertex's factor levels using rule 2. This rule ensures that the simplex does not return to the previous simplex.

Rule 4. Boundary conditions are a useful way to limit the range of possible factor levels. For example, it may be necessary to limit a factor's concentration for solubility reasons, or to limit the temperature because a reagent is thermally unstable. If the new vertex exceeds a boundary condition, then assign it the worst response and follow rule 3.

Because the size of the simplex remains constant during the search, this algorithm is called a fixed-sized simplex optimization. The following example illustrates the application of these rules.

Example 9.4.1

Find the optimum for the response surface described by the equation

$$R = 5.5 + 1.5A + 0.6B - 0.15A^2 - 0.0254B^2 - 0.0857AB$$

using the fixed-sized simplex searching algorithm. Use (0, 0) for the initial factor levels and set each factor's step size to 1.00.

Solution

Letting a = 0, b =0, s_a = 1.00, and s_b = 1.00 gives the vertices for the initial simplex as

$$\text{vertex 1:}(a, b) = (0, 0)$$

$$\text{vertex 2:}(a + s_a, b) = (1.00, 0)$$

$$\text{vertex 3:}(a + 0.5 s_a, b + 0.87 s_b) = (0.50, 0.87)$$

The responses for the three vertices are shown in the following table

vertex	a	b	response
v_1	0	0	5.50
v_2	1.00	0	6.85
v_3	0.50	0.87	6.68

with v_1 giving the worst response and v_3 the best response. Following Rule 1, we reject v_1 and replace it with a new vertex; thus

$$a_{v_4} = 2\left(\frac{1.00 + 0.50}{2}\right) - 0 = 1.50$$

$$b_{v_4} = 2\left(\frac{0 + 0.87}{2}\right) - 0 = 0.87$$

The following table gives the vertices of the second simplex.

vertex	a	b	response
v_2	1.00	0	6.85
v_3	0.50	0.87	6.68
v_4	1.50	0.87	7.80

with v_3 giving the worst response and v_4 the best response. Following Rule 1, we reject v_3 and replace it with a new vertex; thus

$$a_{v_5} = 2\left(\frac{1.00 + 1.50}{2}\right) - 0.50 = 2.00$$

$$b_{v_5} = 2\left(\frac{0 + 0.87}{2}\right) - 0.87 = 0$$

The following table gives the vertices of the third simplex.

vertex	a	b	response
v_2	1.00	0	6.85
v_4	1.50	0.87	7.80
v_5	2.00	0	7.90

The calculation of the remaining vertices is left as an exercise. Figure $9.4.2$ shows the progress of the complete optimization. After 29 steps the simplex begins to repeat itself, circling around the optimum response of (3, 7).

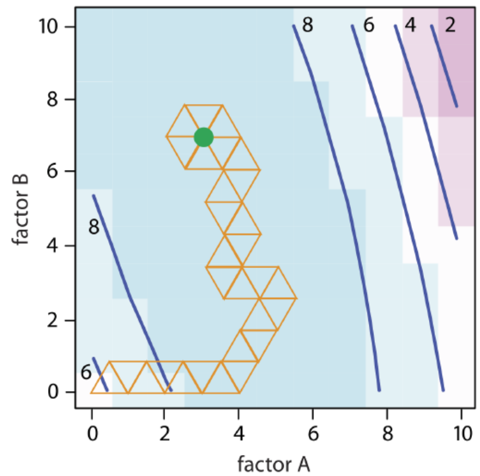

Figure 9.4.2. Progress of the fixed-size simplex optimization in Example 9.4.1. The green dot (•) marks the optimum response of (3,7). Optimization ends when the simplexes begin to circle around a single vertex.

The size of the initial simplex ultimately limits the effectiveness and the efficiency of a fixed-size simplex searching algorithm. We can increase its efficiency by allowing the size of the simplex to expand or to contract in response to the rate at which we approach the optimum. For example, if we find that a new vertex is better than any of the vertices in the preceding simplex, then we expand the simplex further in this direction on the assumption that we are moving directly toward the optimum. Other conditions might cause us to contract the simplex—to make it smaller—to encourage the optimization to move in a different direction. We call this a variable-sized simplex optimization. Consult this chapter's additional resources for further details of the variable-sized simplex optimization.