

Simplex Optimization

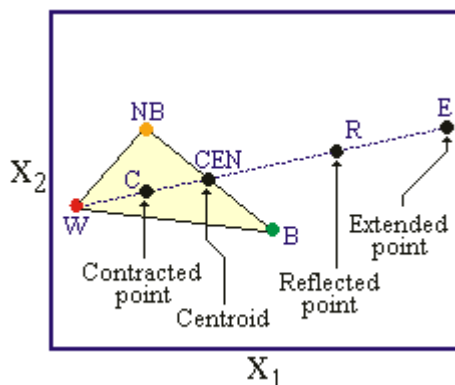
Theory

Simplex is a simple optimization algorithm seeking the vector of parameters corresponding to the global extreme (maximum or minimum) of any n -dimensional function $F(x_1, x_2, \dots, x_n)$, searching through the parameter space ("search area").

In chemistry, the goal may be the search for optimal conditions for obtaining the maximum yield of a compound, e.g. % yield as a function of reflux time and of excess of a particular reagent, or (in HPLC) the resolution of two or more chromatographic peaks as a function of the flow rate and the composition of the eluant.

The 2-dimensional simplex starts with 3 observations of the system response (e.g. yield % or resolution) obtained with 3 different (x_1, x_2) parameter settings ("guesses"). These 3 observations correspond to the vertices of a triangular constituting the 1st simplex. In the 3-dimensional space 4 initial observations are required defining a tetrahedral body, and so on for the impossible to be visualized spaces of higher than 3 dimensions.

From the evaluation of the response of each observation (RB: response of best point B, RNB: response of next to best point NB, RW: response of worst point W) the position of next point to be evaluated (i.e. the new set of experimental conditions) is indicated by either reflection, expansion, or contraction operations depicted in the following animated picture.



Briefly, the simplex algorithm follows these steps:

1. The position of centroid (point CEN in the midway between points B and NB) is calculated.
2. A reflection (vs CEN) of the worst-response point W is performed and the response RR of the reflected point R is evaluated.
3. If R is within the "search area" and its response RR is better than RW but no better than RB, then a new simplex is formed by replacing W with R. The process is repeated from step <1> with the new simplex.
4. If the response RR is even better, i.e. better than RB this is an indication that simplex is moving in the correct direction, therefore an extension to point E is tried (E is twice as far from CEN as R in the same direction).
5. If E is within the "search area" and its response RE is better than RR then W is replaced with E, otherwise W is replaced with R. The process is repeated from step <1> with the new simplex.
6. If the initial reflection fails, i.e. RR is worse than RW or R is not within the "search area", then a contraction is performed. The contracted point C, i.e. the point in the midway between W and CEN, replaces W. The process is repeated from step <1> with the new simplex.

The iterations are terminated when no more significant improvement of the response is observed on moving from one simplex to the other and/or the displacements are insignificant.

It should be stressed that when there are local extremes, it is highly probable the algorithm to fail and be trapped there, instead of the global extreme.

Applet

With this easy to use applet you can easily observe how simplex is trying to locate the global minimum within the "search area".

You have to click 3 points inside the "search area" to define the 1st simplex. Then click the Step button to move simplex toward the next position. Keep clicking Step until the termination of the search. The optimal point is marked with a white cross.

The blue-shaded contours allow the user to know where the global and local extremes are located, and how (and if) the simplex moves toward the correct direction. It is recommended the user to try to predict which will be the next action of the algorithm (i.e. reflection, extension, contraction) before clicking Step.

There are 6 different functions (F1, F2, .., F6) each one yielding a different response surface. F5 and F6 functions yield response surfaces with local minima, where, quite often (depending on the position of the 1st simplex, i.e. of the initial "guesses") the simplex algorithm is trapped, failing to locate the sought after global minimum.

ATTENTION:

For a full list of all applets click [here](#).

Page maintained by [Prof. C. E. Efstathiou](#)