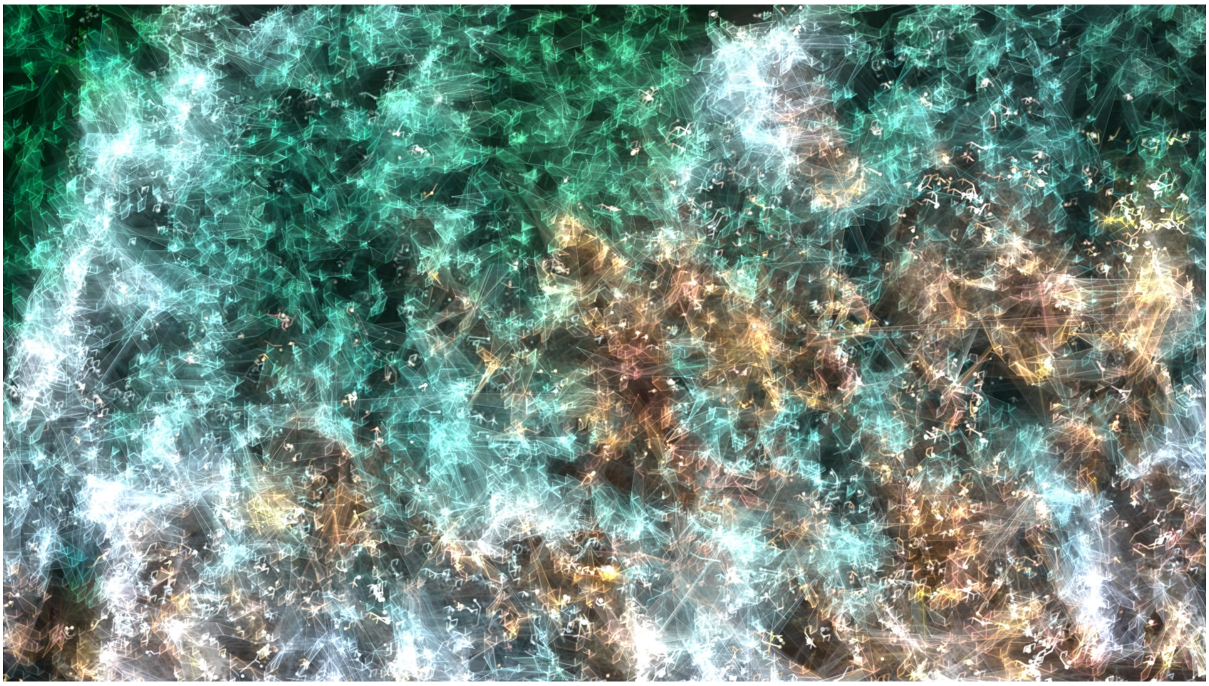
A (Somewhat) Simplified Explanation of the Nelder-Mead Search Method

Bret Battey • BatHatMedia.com

Revised 27 September 2019



A still from *Estuaries 3* (2018), an audiovisual composition I created with the assistance of my OptiNelder video filter. The Nelder-Mead triangles are quite clear in this image

My OptiNelder video filter entails the visualization of Nelder-Mead search agents seeking brightest or darkest points in a source image. The filter will be described in detail in an upcoming book chapter (TBA). The intent with this paper is to describe the Nelder-Mead algorithm itself in relatively simplified form to facilitate understanding of the process for artists and other non-specialists (including myself).

Let us conceive of all of the potential outputs of a mathematical function as forming a terrain – a contoured landscape. Optimization entails finding the lowest point in that landscape. The Nelder-Mead algorithm (Nelder and Mead 1965) is a classic "direct-search" method for such optimization. That is to say, it does not optimize by solving equations, but instead uses a heuristic (a rule-of-thumb process) to hunt for the solution.

Nelder-Mead can be used to solve high-dimensional problems (ones with many variables). But to aid understanding, below I present a two-dimensional (i.e. two-variable) application specifically rather than using the more abstract/generalized type of description that mathematicians love. The explanation here is based on the generalized n-dimensional formalization provided by M. H. Wright (1996).

One result of this is that, instead of referring to a "simplex" (a tetrahedron of some arbitrary number of dimensions), the below refers to a triangle – the type of simplex the Nelder-Mead algorithm would use to solve a two-dimensional problem.

Thus, we can say: To optimize our two-dimensional problem, the Nelder-Mead algorithm mutates a triangle iteratively so it moves over the terrain step-by-step. At each step, the algorithm uses what is discovers at the corners of the triangle to decide how to transform the triangle for the next step. Hopefully, the wandering triangle will ultimately discover the lowest point in the landscape. (There are various ways in which it might fail, but we will not address those here.)

The diagrams below are designed based on the standard coefficient settings for the Nelder-Mead algorithm. The coefficients are referred to in the original paper as ρ (rho) for reflection, χ (chi) for expansion, γ (gamma) for contraction and σ (sigma) for shrinkage. These should satisfy $\rho > 0, \chi > 1, 0 < \gamma < 1$, and $0 < \sigma < 1$. Standard choices for these values are $\rho = 1, \chi = 2, \gamma = 0.5$, and $\sigma = 0.5$ (Wright 1996).

***Caveat emptor*: I am an audiovisual composer and algorithm hacker, not a mathematician or an optimization specialist. Use the following at your own risk! Engineers and mathematicians will probably be best advised to refer instead to primary sources.**

1. Start

The process starts with an initial triangle. It *might* be placed an initial best guess about where the solution lies, or it might be placed randomly or on some other basis. Then the following steps are iterated.

2. Sort

Find the function values (the height of the landscape) at the three vertices of the triangle and sort the vertices from low to high to give $x_{1,} x_2$, and x_3.

3. Reflect

Compute the reflection point x_r by reflecting the worst (highest) point (x_3) around the centroid point x_m of the best points (x_1 and x_2). In our triangle case, the centroid point will be at the midpoint of a line between the two best points. The reflection distance is scaled by ρ. (See Fig. 1.)

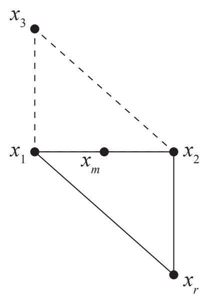


Fig. 1. Calculation of reflection point x_r by reflecting the worst point x_3 around the centroid x_m of the best points: x_1 and x_2. The dotted line represents the original triangle. The distance between x_m and x_r can be scaled by a coefficient ρ, shown here as if $\rho = 1$.

If the function value of x_1 is less than or equal to that of this reflection point $x_{r,}$ which in turn is less than the worst retained point (x_2), we accept this new point as an "improvement" over x_3. Therefore, the process retains x_r as a new point, discards x_3 , and jumps to step 7. Otherwise, we proceed to step 4.

4. Expand

If the reflection point is less than the function value of x_1, it is “downhill” compared to the given triangle, and we continue with the remainder of step 4. Otherwise we jump to Step 5.

To see if this trajectory continues downhill, the process calculates an expansion point x_e. One can think of the expansion as pushing out even further on a line formed from the centroid point to the reflection point, scaled by χ. (See Fig. 2.)

If the expansion point returns a function value even lower than x_r, we assume that this is on the right track: x_e is taken as a new point for next triangle and we jump to step 7. Otherwise, if the expansion point is greater than or equal to the reflection point, it does not represent an improvement; x_r is retained as the new point and we jump to step 7.

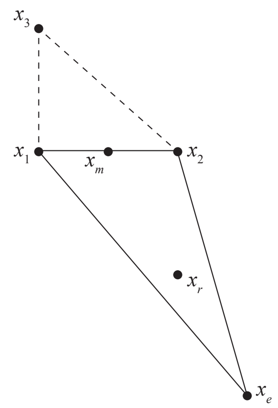


Fig. 2. Calculation of expansion point x_e by projecting further on the line from the centroid to the reflection point. The projection can be scaled by coefficient χ, shown here with $\chi = 2$.

5. Contract

If the reflection point is greater than or equal to x_2, then it lies “uphill”; it is time to try a contraction to find a potentially better point, either “outside” or “inside” the original triangle. (See Fig. 3.)

If the function value of $x_2 \leq x_r < x_3$, form an “outside contraction” to determine x_c. It will be a point on a line between the centroid x_m and the reflection point, with the distance scaled by γ. If the function value at $x_c \leq x_r$, then it is a better choice. We retain x_c as our new point and jump to step 7.

On the other hand, if the reflection point $x_r > x_3$ (that is, it is even higher than our worst point), perform an “inside contraction” to determine x'_c, which will lie on the line between the centroid x_m and the worst point, x_3. The distance is scaled by γ. If the function value at x'_c is less than x_3, then it is a better choice. We retain x'_c as our new point and jump to step 7. Otherwise, jump to step 6.

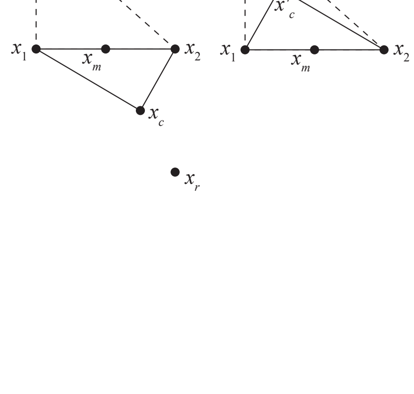


Fig. 3. Calculation of outside and inside contractions, with $\gamma = 0.5$.

6. Shrink

Retain x_1 while calculating two new vertices, v_2 and v_3, by "pulling in" x_2 and x_3 to create a triangle of the same shape but smaller size. The scaling is controlled by σ. See Fig 4.

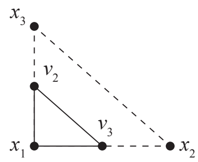


Fig. 4. Calculation of a shrink step, with the triangle scaled by $\sigma = 0.5$.

7. Evaluate

Determine if the process should terminate. This may be decided on the basis of the function values of the three points being sufficiently close to one another, or on the triangle becoming sufficiently small, or on the basis of a maximum number of allowed iterations. If the process does not meet a termination condition, return to step 2 above.

Bibliography

Nelder, J.A.; Mead, R. (1965) A Simplex Method for Function Minimization. *The Computer Journal*. 7(4). Oxford Academic. pp. 308-313.

Wright, M. H. (1996) Direct Search Methods: Once Scorned, Now Respectable. *Pittman Research Notes in Mathematics Series: Numerical Analysis 1995*. Harlow, UK: Addison Wesley Longman. pp. 191-208.