Simplex Optimization

The simplex Nelder-Mead algorithm (Amoeba search) is widely used to solve unconstrained function minimization problems even the discontinuous ones or subject to noise. The algorithm is an iterative process, meaning, it tries to replace the worst solution with a better one and reorder the simplex nodes. Each iteration, the search area is moving and deformed based on some rules. To understand the structure of the algorithm more closely, we begin with introducing some helpful stuff. Our main goal is :

$$\min_{\vec{X} \in \mathbb{R}^n} f(\vec{X})$$

$$f : \mathbb{R}^n \to \mathbb{R}$$

Where, f is the objective function we have discussed earlier and n is the dimension. And the simplex is a geometric figure that lives in that n dimension with $n{+}1$ vertices. In 2-D, it is a triangle, so we are going to denote it by Δ for simplicity and convenience.

$$\Delta = (\vec{x}_{best}\ ,\ \vec{x}_{other}\ ,\ \vec{x}_{worst})$$

where,

- $\vec{x}_{best}$ is the best solution within the vertices
- $\vec{x}_{worst}$ is the worst solution within the vertices
- $\vec{x}_{other}$ is the solution in between
  That is based on the fact that :

$$f(\vec{x}_{best}) \leq \quad f(\vec{x}_{other}) \ \leq \ f(\vec{x}_{worst})$$

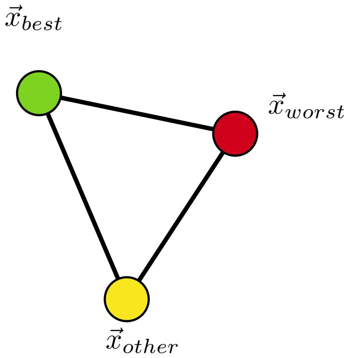


Figure 1: Initial state of particle at point of time t

The goal now is to adjust the parameter values of the worst point so that the simplex moves towards the minimum of the function with the direction of moving determined by the centroid.

Compute Centroid

The centroid represents the geometric mean between two points. It is calculated for every pair of vertices in the simplex except for the worst point.

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

In our case $n = 2$, the centroid is calculated for $\vec{x}_{best}$ and $\vec{x}_{other}$ only through the following simple equation:

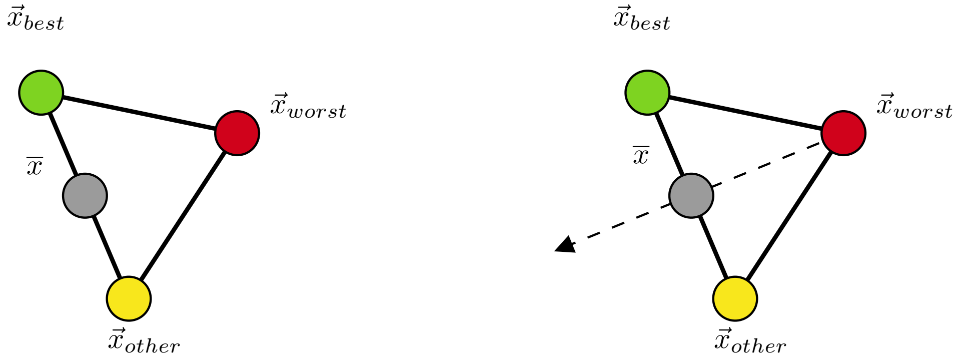
$$\overline{x} = (\vec{x}_{best} + \vec{x}_{other})/2$$

Now, all the operation we are going to execute on the simplex will take place with respect to the direction of vector $(\overline{x} - \vec{x}_{worst})$, second sub-figure.

Reflection

Reflection is the first operation we make on the simplex data structure. It is an attempt to see whether the value of the function at the reflected point is less than the worst solution. It can be computed through the upcoming equation.

$$\vec{x}_r = \overline{x} + \alpha(\overline{x} - \vec{x}_{worst})$$



(a) Intial state of particles

(b) Final state of particles

Figure 2: PSO Animation

where α is the reflection coefficient, positive real-valued parameter , typically $\alpha = 1$.

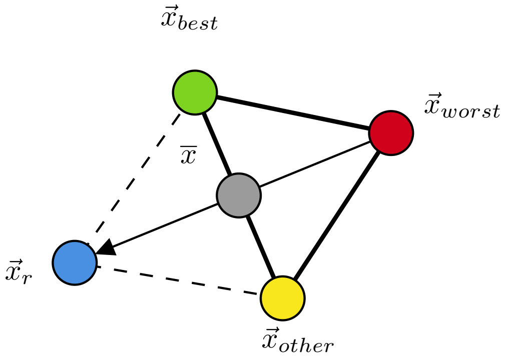


Figure 3: Initial state of particle at point of time t

If the objective function at the reflected point is less than the other point, but not the best, we set the reflected point to the worst.

$$f(\vec{x}_{best}) \leq \quad f(\vec{x}_r) \ < \ f(\vec{x}_{other}) \quad \implies \vec{x}_{worst} = \vec{x}_r$$

And we reorder the vertices as we mentioned above; in fact, we have to reorder the vertices of the simplex after each operation, which is the whole point of the algorithm.

Expansion

If it happened and the reflected point is better than the best point (in terms of fitness) $f(\vec{x}_r) < \quad f(\vec{x}_{best})$, maybe we can find better point a bit further from the reflected point, that is the idea of expansion.

$$\vec{x}_e = \vec{x}_r + \beta(\vec{x}_r - \overline{x})$$

where β is the expansion coefficient, with $\beta > 1$, typically $\beta = 2$.

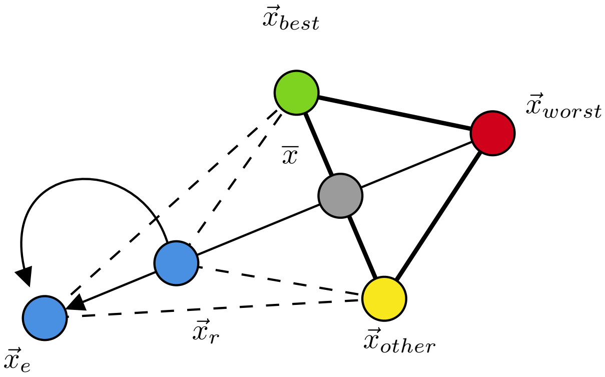


Figure 4: Initial state of particle at point of time t

Again, we check the quality of the new point $\vec{x}_e$, if it is better than the reflected point, then we are in the right direction, if not we assign the reflected point to the worst point normally which means, we have done a bad attempt .

$$\begin{aligned} f(\vec{x}_e) < \quad & f(\vec{x}_r) \quad \Longrightarrow \vec{x}_{worst} = \vec{x}_e \\ f(\vec{x}_e) \geq \quad & f(\vec{x}_r) \quad \Longrightarrow \vec{x}_{worst} = \vec{x}_r \end{aligned}$$

Contraction

If the reflected is worse from the beginning, then we can minimize the damage by contracting the simplex whether inside or outside, whichever is better.

Inside contraction when the reflected point is worse than all including the worst point, we try to contract the simplex towards the worst point, perhaps we get a better solution.

$$f(\vec{x}_r) \geq \quad f(\vec{x}_{worst})$$

$$\vec{x}_c = \overline{x} + \gamma(\vec{x}_{worst} - \overline{x})$$

where γ is the contraction coefficient, with $0 < \gamma < 1$, typically $\gamma = \frac{1}{2}$.
if the fitness value is better at the contracted point than the worst point, we replace the worst point with the contracted one, if not we moved in the wrong direction, maybe we have to go backwards.

$$f(\overrightarrow{x}_c) < \quad f(\overrightarrow{x}_{worst}) \quad \implies \overrightarrow{x}_{worst} = \overrightarrow{x}_c$$

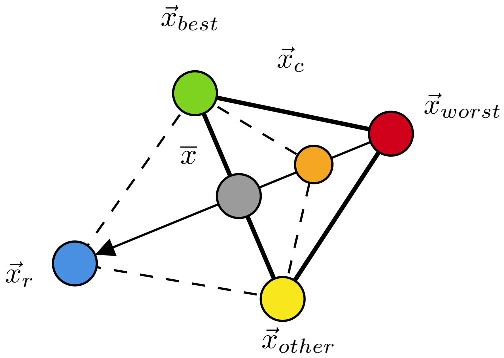


Figure 5: Initial state of particle at point of time t

Outside contraction When the reflected point is, however, worse than the “other” point , but better than the worst, we can contract in the outside direction (towards the reflected point).

$$f(\overrightarrow{x}_{other}) \leq \quad f(\overrightarrow{x}_r) \quad < \quad f(\overrightarrow{x}_{worst})$$
$$\overrightarrow{x}_c = \overline{x} + \gamma(\overrightarrow{x}_r - \overline{x})$$

And we also check weather we are going to accept the new point or not.

$$f(\overrightarrow{x}_c) < \quad f(\overrightarrow{x}_r) \quad \implies \overrightarrow{x}_{worst} = \overrightarrow{x}_c$$

Shrinking The idea of shrinking is to minimize the damage from a failed contraction, that is when $f(x_c) > \min[f(x_{worst}), f(x_r)]$, This can be done by moving the worst point and the other point towards the best point by :

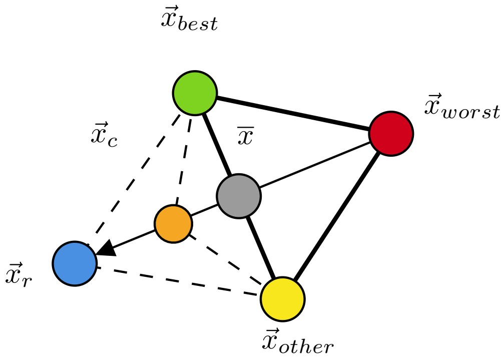


Figure 6: Initial state of particle at point of time t

$$\overrightarrow{x^*}_{worst} = \vec{x}_{best} + \delta(\vec{x}_{worst} - \vec{x}_{best})$$
$$\overrightarrow{x^*}_{other} = \vec{x}_{best} + \delta(\vec{x}_{other} - \vec{x}_{best})$$

where δ is the contraction coefficient, with $0 < \delta < 1$, traditionally $\delta = \frac{1}{2}$.

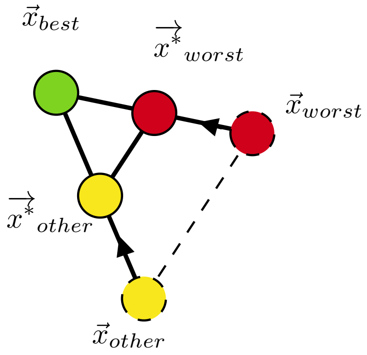

$\vec{x}_{best}$
$\overrightarrow{x^*}_{worst}$
$\vec{x}_{worst}$
$\overrightarrow{x^*}_{other}$
$\vec{x}_{other}$

Figure 7: Initial state of particle at point of time t

We do all of the previous until we reach the predefined maximum iterations or we cannot do better by reaching certain threshold, which can be done by checking the convergence each iteration

Convergence We say that , the simplex has converged to the best solution if it is sufficiently compact meaning that the distance between the each pair of vertices is adequately small.

$$\sigma = 2\ .\left[\ \frac{f(\overrightarrow{x}_{worst}) - f(\overrightarrow{x}_{best})}{f(\overrightarrow{x}_{worst}) + f(\overrightarrow{x}_{best}) + \epsilon}\right]$$

where ϵ is a tolerance defined based on the nature of the problem. Each iteration we check if :

$$\sigma < \epsilon$$

if not we proceed until the condition prevails.