WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Català
Deutsch
Español
فارسی
Français
한국어
Bahasa Indonesia
Italiano
Nederlands
日本語
Polski
Русский
Edit links

Nelder–Mead method

From Wikipedia, the free encyclopedia

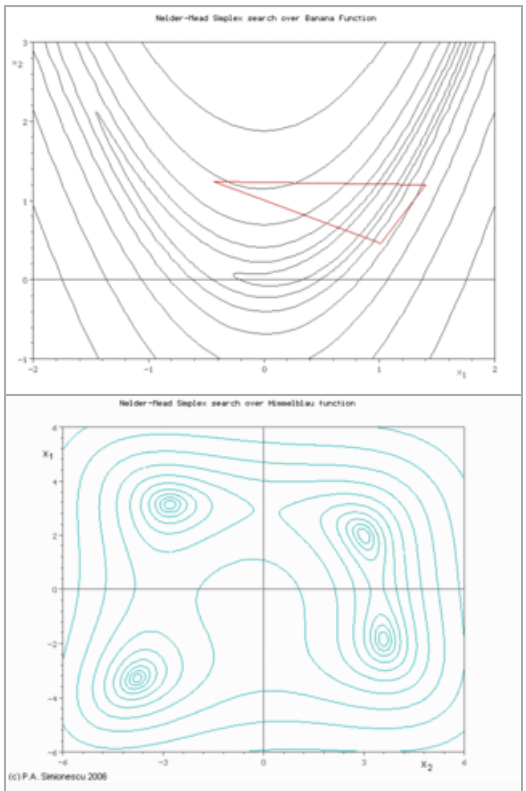
See simplex algorithm for Dantzig's algorithm for the problem of linear optimization.

The **Nelder–Mead method** or **downhill simplex method** or **amoeba method** is a commonly applied numerical method used to find the minimum or maximum of an objective function in a many-dimensional space. It is applied to nonlinear optimization problems for which derivatives may not be known. However, the Nelder–Mead technique is a heuristic search method that can converge to non-stationary points[1] on problems that can be solved by alternative methods.[2]

The Nelder–Mead technique was proposed by John Nelder & Roger Mead (1965).[3]

Contents [hide]

1 Overview
2 One possible variation of the NM algorithm
3 See also
4 References
4.1 Further reading
5 External links



Nelder–Mead simplex search over the Rosenbrock banana function (above) **and Himmelblau's function** (below)

Overview [edit]

The method uses the concept of a simplex, which is a special polytope of $n + 1$ vertices in n dimensions. Examples of simplices include a line segment on a line, a triangle on a plane, a tetrahedron in three-dimensional space and so forth.

The method approximates a local optimum of a problem with n variables when the objective function varies smoothly and is unimodal.

For example, a suspension bridge engineer has to choose how thick each strut, cable, and pier must be. These elements are interdependent, but it is not easy to visualize the impact of changing any specific element. Simulation of such complicated structures is often extremely computationally expensive to run, possibly taking upwards of hours per execution. An engineer may therefore prefer the Nelder-Mead method as it requires fewer evaluations per iteration than other optimization methods.

Nelder–Mead in n dimensions maintains a set of $n+1$ test points arranged as a simplex. It then extrapolates the behavior of the objective function measured at each test point, in order to find a new test point and to replace one of the old test points with the new one, and so the technique progresses. The simplest approach is to replace the worst point with a point reflected through the centroid of the remaining n points. If this point is better than the best current point, then we can try stretching exponentially out along this line. On the other hand, if this new point isn't much better than the previous value, then we are stepping across a valley, so we shrink the simplex towards a better point. An intuitive explanation of the algorithm is presented in [4]

> The downhill simplex method now takes a series of steps, most steps just moving the point of the simplex where the function is largest ("highest point") through the opposite face of the simplex to a lower point. These steps are called reflections, and they are constructed to conserve the volume of the simplex (and hence maintain its nondegeneracy). When it can do so, the method expands the simplex in one or another direction to take larger steps. When it reaches a "valley floor," the method contracts itself in the transverse direction and tries to ooze down the valley. If there is a situation where the simplex is trying to "pass through the eye of a needle," it contracts itself in all

directions, pulling itself in around its lowest (best) point.

Unlike modern optimization methods, the Nelder–Mead heuristic can converge to a non-stationary point unless the problem satisfies stronger conditions than are necessary for modern methods.[1] Modern improvements over the Nelder–Mead heuristic have been known since 1979.[2]

Many variations exist depending on the actual nature of the problem being solved. A common variant uses a constant-size, small simplex that roughly follows the gradient direction (which gives steepest descent). Visualize a small triangle on an elevation map flip-flopping its way down a valley to a local bottom. This method is also known as the Flexible Polyhedron Method. This, however, tends to perform poorly against the method described in this article because it makes small, unnecessary steps in areas of little interest.

One possible variation of the NM algorithm [edit]

We are trying to minimize the function $f(\mathbf{x})$ where $\mathbf{x} \in \mathbb{R}^n$. Our current test points are $\mathbf{x}_1, \ldots, \mathbf{x}_{n+1}$.

- **1. Order** according to the values at the vertices:
  $$f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \cdots \leq f(\mathbf{x}_{n+1})$$
- **2.** Calculate $\mathbf{x}_o$, the centroid of all points except $\mathbf{x}_{n+1}$.
- **3. Reflection**

  Compute reflected point $\mathbf{x}_r = \mathbf{x}_o + \alpha(\mathbf{x}_o - \mathbf{x}_{n+1})$

  If the reflected point is better than the second worst, but not better than the best, i.e.:

  $f(\mathbf{x}_1) \leq f(\mathbf{x}_r) < f(\mathbf{x}_n)$,

  then obtain a new simplex by replacing the worst point $\mathbf{x}_{n+1}$ with the reflected point $\mathbf{x}_r$, and go to step 1.
- **4. Expansion**

  If the reflected point is the best point so far, $f(\mathbf{x}_r) < f(\mathbf{x}_1)$,

  then compute the expanded point $\mathbf{x}_e = \mathbf{x}_o + \gamma(\mathbf{x}_o - \mathbf{x}_{n+1})$

  If the expanded point is better than the reflected point, $f(\mathbf{x}_e) < f(\mathbf{x}_r)$

  then obtain a new simplex by replacing the worst point $\mathbf{x}_{n+1}$ with the expanded point $\mathbf{x}_e$, and go to step 1.

  Else obtain a new simplex by replacing the worst point $\mathbf{x}_{n+1}$ with the reflected point $\mathbf{x}_r$, and go to step 1.

  Else (i.e. reflected point is not better than second worst) continue at step 5.
- **5. Contraction**

  Here, it is certain that $f(\mathbf{x}_r) \geq f(\mathbf{x}_n)$

  Compute contracted point $\mathbf{x}_c = \mathbf{x}_o + \rho(\mathbf{x}_o - \mathbf{x}_{n+1})$

  If the contracted point is better than the worst point, i.e. $f(\mathbf{x}_c) < f(\mathbf{x}_{n+1})$

  then obtain a new simplex by replacing the worst point $\mathbf{x}_{n+1}$ with the contracted point $\mathbf{x}_c$, and go to step 1.

  Else go to step 6.
- **6. Reduction**

  For all but the best point, replace the point with

  $\mathbf{x}_i = \mathbf{x}_1 + \sigma(\mathbf{x}_i - \mathbf{x}_1)$ for all i $\in \{2, \ldots, n+1\}$. go to step 1.

Note: α, γ, ρ and σ are respectively the reflection, the expansion, the contraction and the shrink coefficient. Standard values are $\alpha = 1$, $\gamma = 2$, $\rho = -1/2$ and $\sigma = 1/2$.

For the **reflection**, since $\mathbf{x}_{n+1}$ is the vertex with the higher associated value among the vertices, we can expect to find a lower value at the reflection of $\mathbf{x}_{n+1}$ in the opposite face formed by all vertices point $\mathbf{x}_i$ except $\mathbf{x}_{n+1}$.

For the **expansion**, if the reflection point $\mathbf{x}_r$ is the new minimum along the vertices we can expect to find interesting values along the direction from $\mathbf{x}_o$ to $\mathbf{x}_r$.

Concerning the **contraction**: If $f(\mathbf{x}_r) > f(\mathbf{x}_n)$ we can expect that a better value will be inside the simplex formed by all the vertices $\mathbf{x}_i$.

Finally, the **reduction** handles the rare case that contracting away from the largest point increases f, something that cannot happen sufficiently close to a non-singular minimum. In that case we contract towards the

lowest point in the expectation of finding a simpler landscape.

The initial simplex is important, indeed, a too small initial simplex can lead to a local search, consequently the NM can get more easily stuck. So this simplex should depend on the nature of the problem.

See also [edit]

- Derivative-free optimization
- COBYLA
- NEWUOA
- LINCOA
- Nonlinear conjugate gradient method
- Levenberg–Marquardt algorithm
- Broyden–Fletcher–Goldfarb–Shanno or BFGS method
- Differential evolution
- Pattern search (optimization)
- CMA-ES

References [edit]

1. ^ *a* *b*
   - Powell, Michael J. D. (1973). "On Search Directions for Minimization Algorithms". *Mathematical Programming* **4**: 193–201. doi:10.1007/bf01584660.
   - McKinnon, K.I.M. (1999). "Convergence of the Nelder–Mead simplex method to a non-stationary point". *SIAM J Optimization* **9**: 148–158. doi:10.1137/S1052623496303482. (algorithm summary online).
2. ^ *a* *b*
   - Yu, Wen Ci. 1979. “Positive basis and a class of direct search techniques”. *Scientia Sinica* [*Zhongguo Kexue*]: 53—68.
   - Yu, Wen Ci. 1979. “The convergent property of the simplex evolutionary technique”. *Scientia Sinica* [*Zhongguo Kexue*]: 69–77.
   - Kolda, Tamara G.; Lewis, Robert Michael; Torczon, Virginia (2003). "Optimization by direct search: new perspectives on some classical and modern methods". *SIAM Rev.* **45**: 385–482. doi:10.1137/S003614450242889.
   - Lewis, Robert Michael; Shepherd, Anne; Torczon, Virginia (2007). "Implementing generating set search methods for linearly constrained minimization". *SIAM J. Sci. Comput.* **29**: 2507–2530. doi:10.1137/050635432.
3. ^ Nelder, John A.; R. Mead (1965). "A simplex method for function minimization". *Computer Journal* **7**: 308–313. doi:10.1093/comjnl/7.4.308.
4. ^
   - Press, WH; Teukolsky, SA; Vetterling, WT; Flannery, BP (2007). "Section 10.5. Downhill Simplex Method in Multidimensions". *Numerical Recipes: The Art of Scientific Computing* (3rd ed.). New York: Cambridge University Press. ISBN 978-0-521-88068-8.

Further reading [edit]

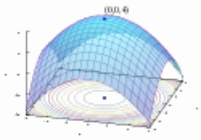
- Avriel, Mordecai (2003). *Nonlinear Programming: Analysis and Methods.* Dover Publishing. ISBN 0-486-43227-0.
- Coope, I. D.; C.J. Price, 2002. “Positive bases in numerical optimization”, *Computational Optimization & Applications*, Vol. 21, No. 2, pp. 169–176, 2002.

External links [edit]

- Nelder–Mead (Simplex) Method
- Nelder–Mead (Downhill Simplex) explanation and visualization with the Rosenbrock banana function
- Nelder–Mead Search for a Minimum
- John Burkardt: Nelder–Mead code in Matlab - note that a variation of the Nelder–Mead method is also implemented by the Matlab function fminsearch.
- Nelder–Mead online for the calibration of the SABR model - Application in Finance.
- SOVA 1.0 (freeware) - Simplex Optimization for Various Applications
- [1] - HillStormer, a practical tool for nonlinear, multivariate and constrained Simplex Optimization by Nelder Mead.

v · t · e **Optimization: Algorithms, methods, and heuristics** [hide]

Unconstrained nonlinear: Methods calling ...	[show]
Constrained nonlinear	[show]
Convex optimization	[show]
Combinatorial	[show]
Metaheuristics	[show]



Categories (Algorithms and methods · Heuristics) · **Software**

Categories: Optimization algorithms and methods | Operations research