Deterministic Optimization

Unconstrained Optimization: Derivative-Free Methods

Shabbir Ahmed
Anderson-Interface Chair and Professor
School of Industrial and Systems Engineering

Multivariate functions

Optimality Conditions

Learning objectives:

- Recognize the Nelder-Mead Method

Derivative-free Algorithms for Unconstrained Optimization

$$(P): \quad \min\{f(x) \mid x \in \mathbb{R}^n\}.$$

- Golden Section search ($n = 1$)
- Quadratic fit ($n = 1$)
- Nelder-Mead method ($n > 1$)

The Nelder-Mead Method

$$\min\{f(x) \mid x \in \mathbb{R}^n\}.$$

Each iteration maintains an ordered set of $n+1$ solution points, i.e. at iteration k, the solution points are labeled $x_1^k, \ldots, x_{n+1}^k$ such that

$$f(x_1^k) \leq f(x_2^k) \leq \ldots \leq f(x_{n+1}^k).$$

Each iteration requires function evaluations and sorting.

No formal convergence theory but works well in practice.

The Nelder-Mead Method

Step 0. Choose $n+1$ distinct solution points $x_1^0, \ldots, x_{n+1}^0$. Set the iteration counter $k = 0$.

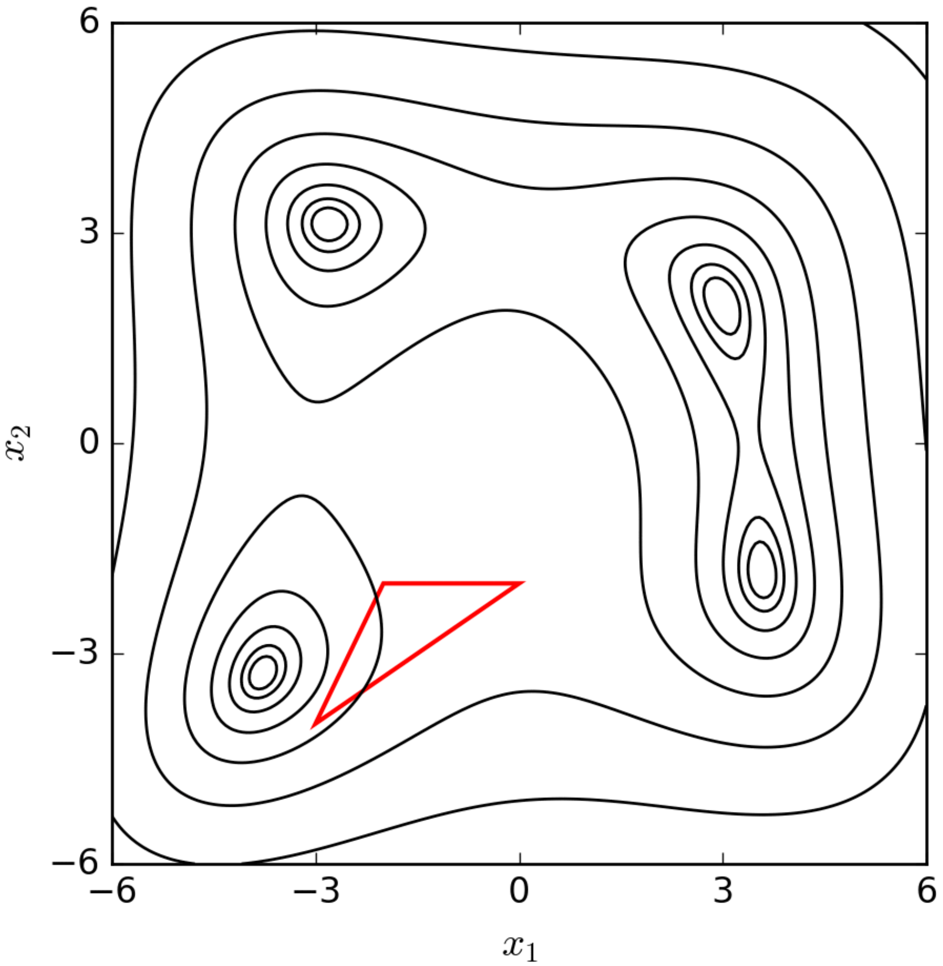
Step 1. Order the solution points. Compute the best-n centroid $\overline{x}^k = (1/n) \sum_{i=1}^n x_i^k$.

Step 2. If $\sum_{i=1}^n |f(x_i^k) - f(\overline{x}^k)| < \epsilon$, STOP and report the better of x_1^k and $\overline{x}^k$.

The Nelder-Mead Method

Step 3. Try to find a better solution point x_b^k along the direction $(\overline{x}^k - x_{n+1}^k)$ using various rules. If you find one, replace x_{n+1}^k by x_b^k, update $k \leftarrow k+1$ and go to step 1.

Step 4. Shrink the current solution set towards the best solution x_1^k by $x_i^{k+1} \leftarrow 0.5(x_1^k + x_i^k)$ for all $i = 1, \ldots, n+1$. Update k and go to step 1.



$$f(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2.$$

Source:
https://en.wikipedia.org/wiki/Nelder%E2%80%93
Mead_method

Implemented in the Matlab routine `fminsearch`.

See also `scipy.optimize.minimize(method='Nelder-Mead')`

```
>>>
>>>
>>> def f(x):
...     return (x[0]-2)**2 + (x[1]-1)**2
...
>>> res = minimize(f,[0,0],method='Nelder-Mead')
>>> res.x
array([ 2.00003382,  0.99997307])
>>>
>>>
```

Summary

- The Nelder-Mead method is a numerical algorithm for minimizing a multivariate function using only function evaluations
- It is not guaranteed to converge but often works well