

Effective Programming Practices for Economists

Numerical Optimization

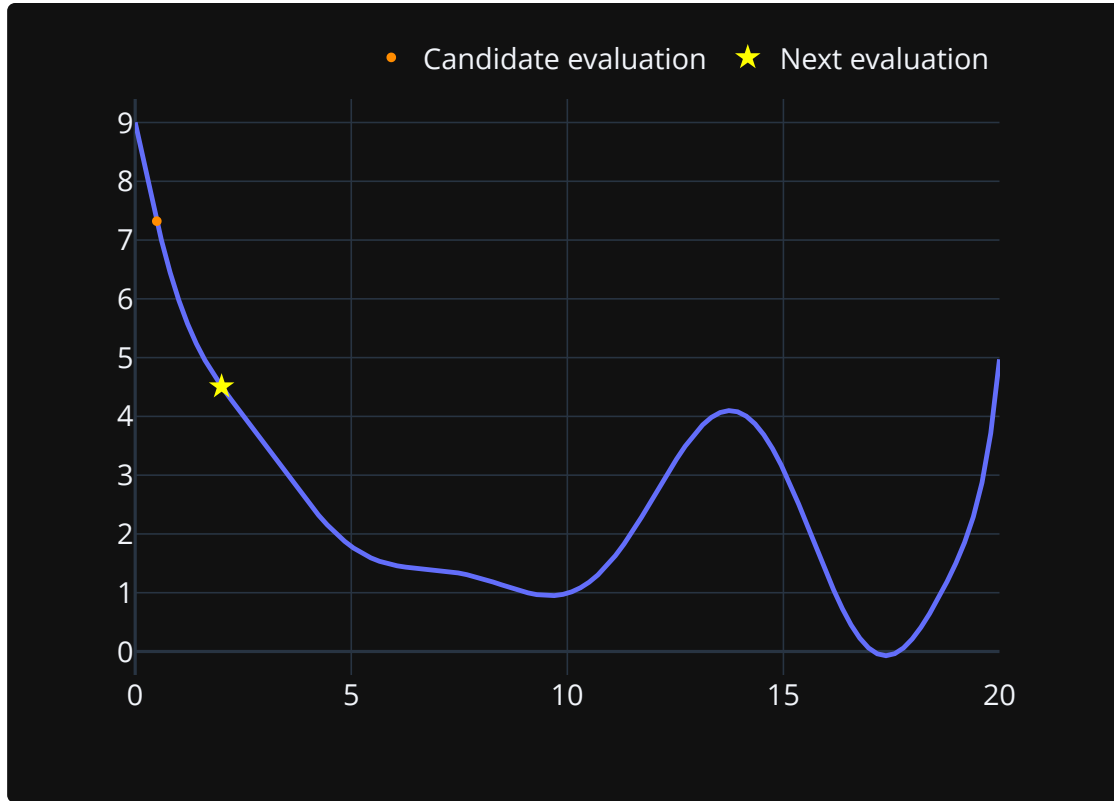
Derivative-Free Direct Search Algorithms

Janoś Gabler, Hans-Martin von Gaudecker, and Tim Mensinger

Basic Idea (optimagic docs)

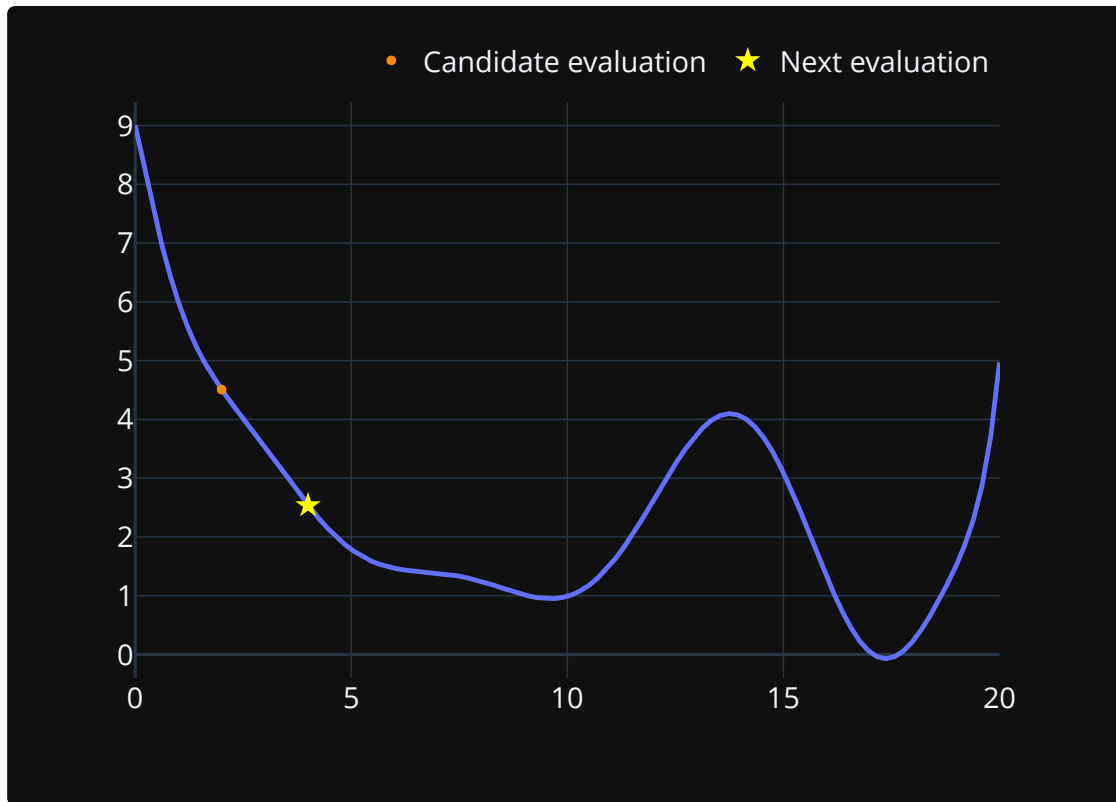
- Explore parameter space around current point systematically and accept the best value
- Also called pattern search because the points at which the function is evaluated form a pattern
- Easiest example for one dimensional problems:
 - Evaluate function at current point and one other point
 - Switch direction of other point if you got an increase in function value
 - Make steps larger after success
 - Make steps smaller after failure

Initial Evaluation



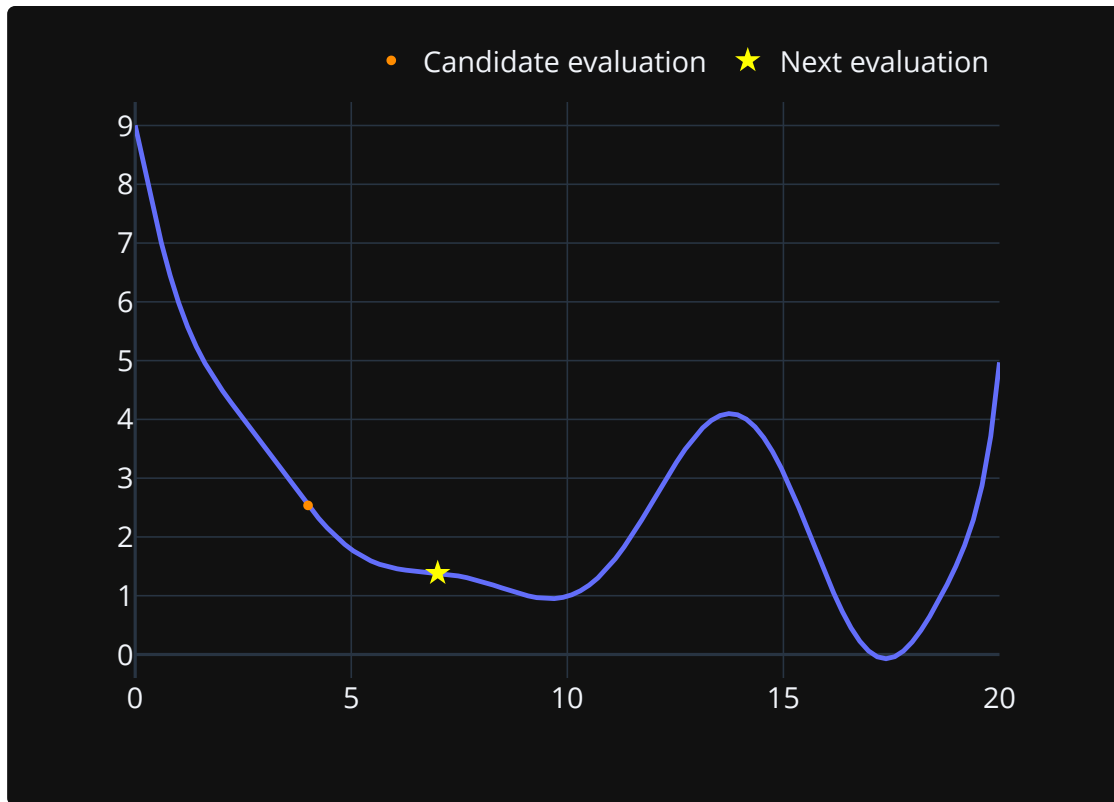
Candidate value $>$ initial value \Rightarrow reject, reverse direction

Iteration 1



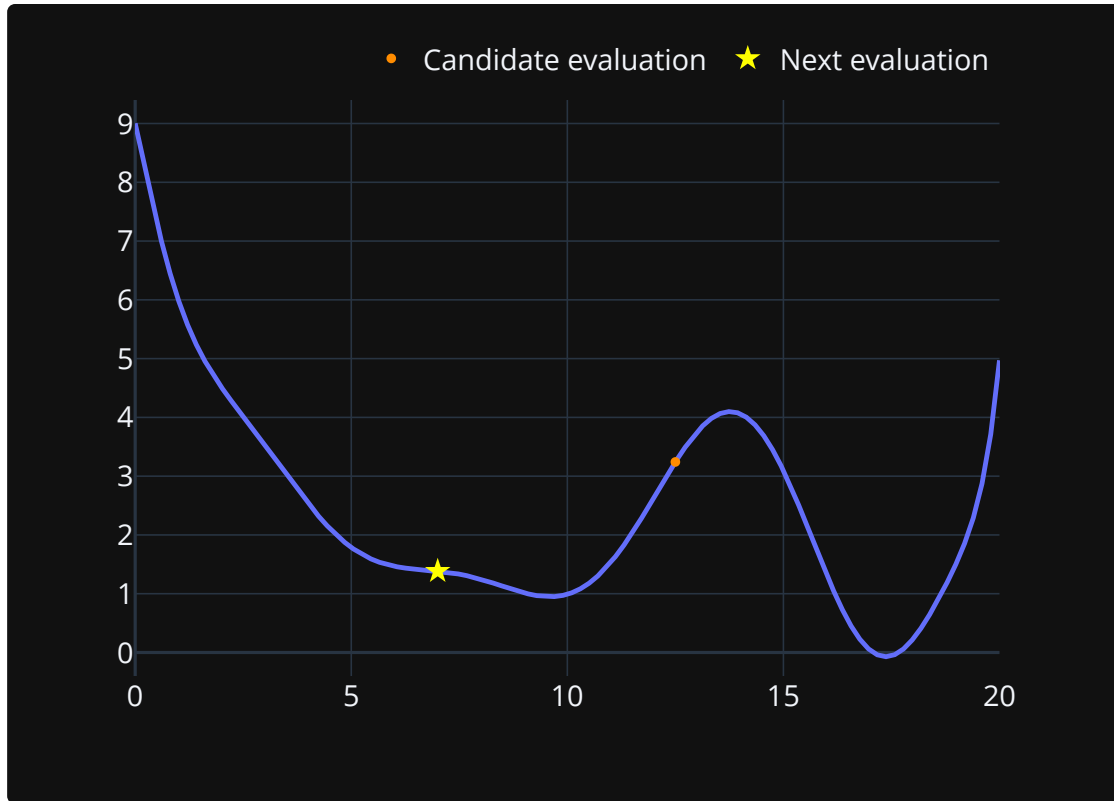
Candidate value $<$ initial value \Rightarrow accept, increase step length.

Iteration 2



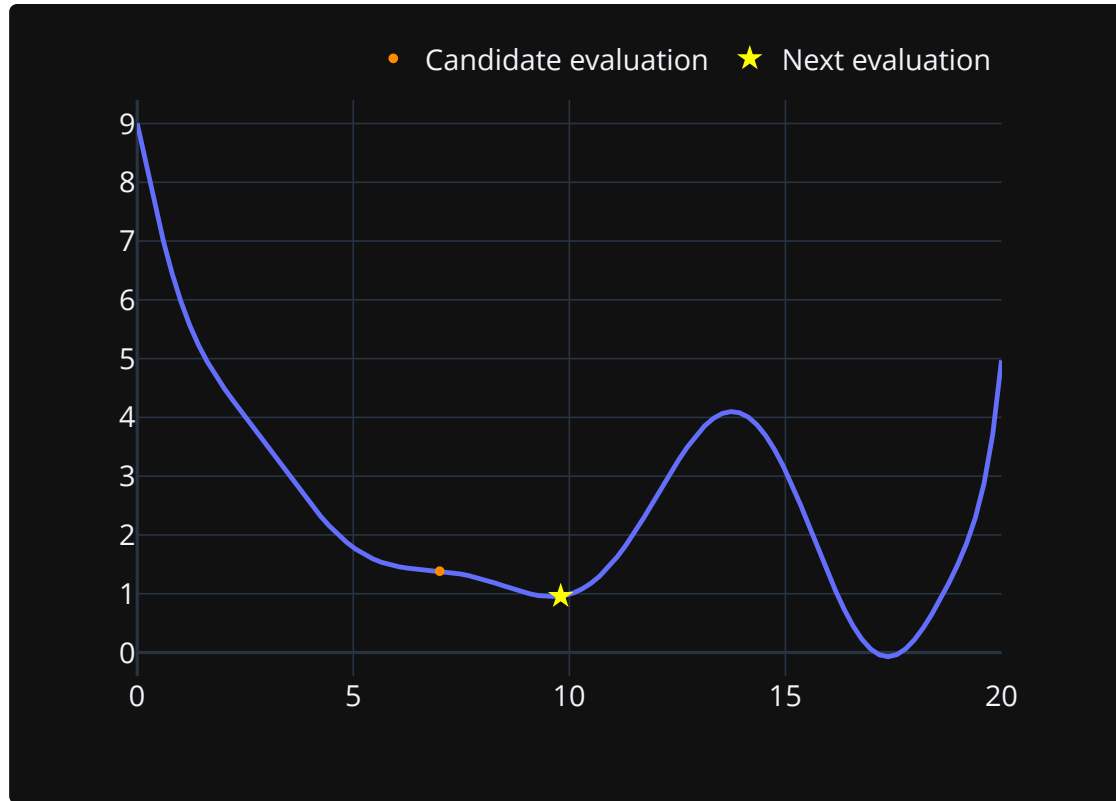
Candidate value $<$ initial value \Rightarrow accept, increase step length.

Iteration 3



Candidate value $>$ initial value \Rightarrow reject, decrease step length.

Iteration 4



Converge eventually (gets worse in both directions).

Some Remarks

- Adjusting the step size and switching to promising directions is complicated in real algorithms
- These algorithms only use the information which function value is smallest, not by how much
- Makes them slow but robust to small amounts of noise
- It does not help for large amounts of noise
- Most famous example is the Nelder-Mead algorithm
 - Very widely used
 - Very seldomly the best choice

A real algorithm: Nelder Mead

