Effective Programming Practices for Economists

Scientific Computing

Visualizing optimizer histories

Janoś Gabler and Hans-Martin von Gaudecker

Motivation

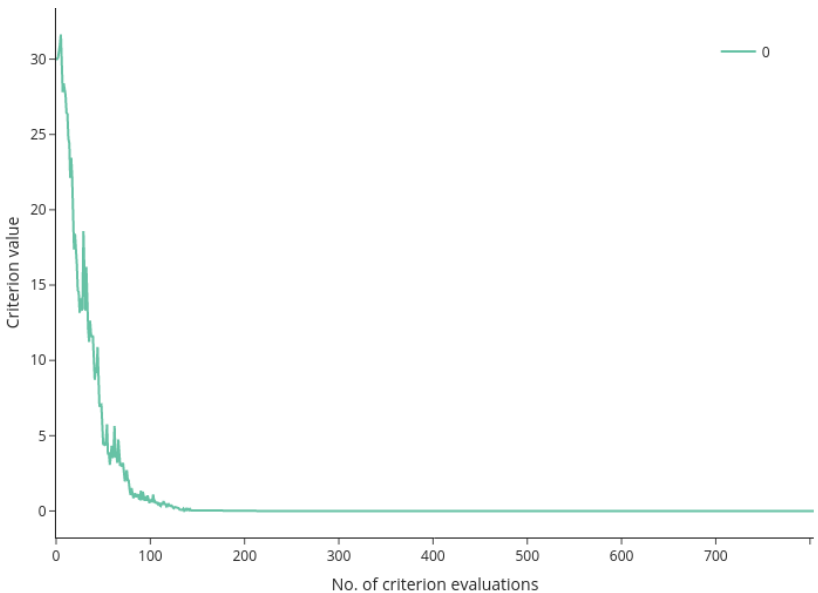
- You rarely have a guarantee that an optimizer will work
  - Assumptions of convergence proofs might not hold in practice
  - You might get stuck in local optima
  - Floating point calculations are never exact
- But you can compare the performance of optimizers
  - Which one finds the lower function value?
  - Which one decreases the function more quickly?
- The `criterion_plot` makes this very easy!

Criterion plot

We assume you have done an optimization and the result is called `res`

Criterion plot

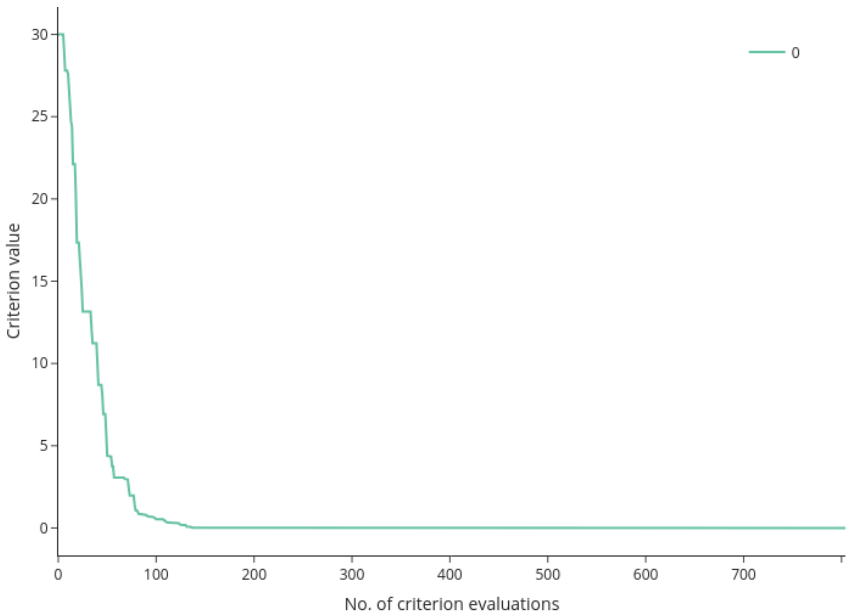
```
em.criterion_plot(res)
```



- First argument can be:
  - `` `OptimizeResult` ``
  - path to log file
  - list or dict thereof
- Dictionary keys are used for legend

Criterion plot

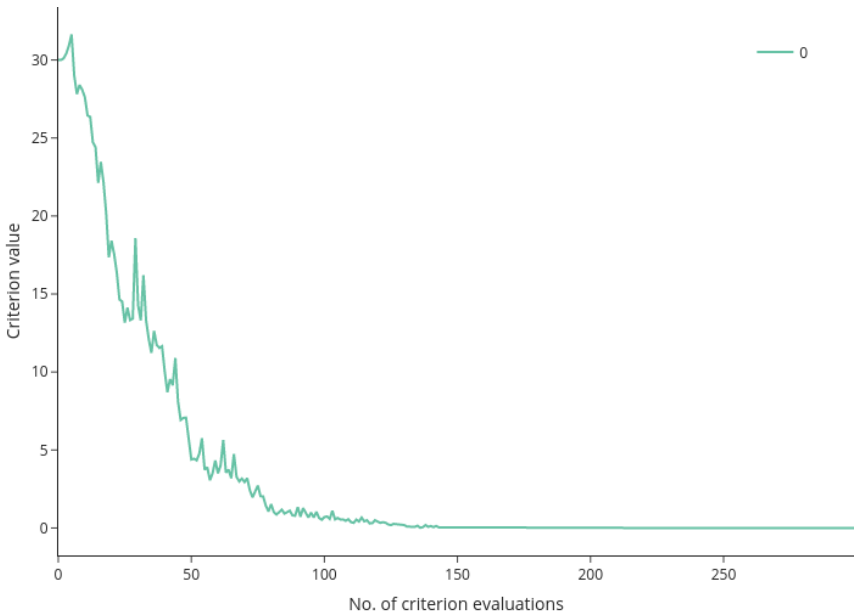
```
em.criterion_plot(res, monotone=True)
```



- **monotone=True** shows the current best value
- useful if there are extreme values in history

Criterion plot

```
em.criterion_plot(res, max_evaluations=300)
```



- **max_evaluations** limits the x-axis

Criterion plot for multiple optimizations

```python
def sphere(x):
    return x @ x

results = {}
for algo in ["scipy_neldermead", "nlopt_neldermead", "fides"]:
    results[algo] = em.minimize(
        sphere,
        np.arange(10),
        algorithm=algo,
    )

em.criterion_plot(results, max_evaluations=200)
```