A Practical Guide to Benchmarking and Experimentation

Nikolaus Hansen
Inria
Research Centre Saclay, CMAP, Ecole polytechnique, Université Paris-Saclay

GECCO '17 Companion, July 15-19, 2017, Berlin, Germany

ACM ISBN 978-1-4503-4939-0/17/07.
http://dx.doi.org/10.1145/3067695.3067711

Overview

- about experimentation (with demonstrations)
  making quick experiments, interpreting experiments,
  investigating scaling, parameter sweeps,
  invariance, repetitions, statistical significance…
- about benchmarking
  choosing test functions, performance measures,
  the problem of aggregation, invariance,
  a short introduction to the COCO platform…

Scientific Experimentation

- What is the aim? *Answer a question*, ideally quickly and comprehensively
- don't trust what you need to rely on without *good* reasons (code, claims, …)
  check/test yourself "everything", practice stress testing, helps also understanding
- run rather *many than few* experiments, as there are many questions to answer
  develops a feeling for the effect of setup changes
  to run many experiments they must be *quick to implement and run*
- run any experiment at least twice
  assuming that the outcome is stochastic
- display: *the more* the better, *the better* the better
  figures are *intuition pumps*
  it is rather impossible to underestimate the value of a good figure
  data is the only way experimentation can help to answer questions, therefore look at them!
- don't make minimising CPU-time a primary objective
  avoid spending time in implementation details to tweak performance
- there are many devils in the details, results may crucially depend on simple or intricate bugs or subtleties
  yet another reason to run many (slightly) different experiments
  check limit settings to give consistent results
- Testing Heuristics: We Have it All Wrong [Hooker 1995]
  "The emphasis on competition is fundamentally anti-intellectual and does not build the sort of insight that in the long run is conducive to more effective algorithms"

Jupyter IPython notebook

```
# download&install anaconda python
# shell cmd "conda create" in case a different Python version is needed
# shell cmd "pip install cma" to install a CMA-ES module (or see github)
# shell cmd jupyter-notebook and click on compact-ga.ipynb
%pylab nbagg
import time
import numpy as np  # already done in lab-mode
from scipy import stats
from experimentation import Results, down_sample, step_data
def mannwhitneyu(*args, **kwargs):  # defines function mannwithneyu
    """``scipy.stats.mannwhitneyu`` with ``alternative='two-sided'``
    as default (bug-fix)
    """
    kwargs.setdefault('alternative', 'two-sided')
    return stats.mannwithneyu(*args, **kwargs)
# shell command: jupyter nbextension enable codefolding/main
```

- Installing IPython is *not* a prerequisite to follow the tutorial
- for more and updated material, see
  slides: http://www.cmap.polytechnique.fr/~nikolaus.hansen/benchmarking-and-experimentation-gecco17-slides.pdf
  code: http://www.cmap.polytechnique.fr/~nikolaus.hansen/benchmarking-and-experimentation-gecco17-code.tar.gz
  at http://www.cmap.polytechnique.fr/~nikolaus.hansen/invitedtalks.html