The Journal of Open Source Software

PyCUTEst: an open source Python package of optimization test problems

Jaroslav Fowkes[1], **Lindon Roberts**[2,3], **and Árpád Bűrmen**[4]

1 Science and Technology Facilities Council, Rutherford Appleton Laboratory, Harwell Campus, Didcot, Oxfordshire, OX11 0QX, UK **2** Mathematical Sciences Institute, Building 145, Science Road, Australian National University, Canberra ACT 2601, Australia **3** School of Mathematics and Statistics, Carslaw Building, University of Sydney, Camperdown NSW 2006, Australia **4** Faculty of Electrical Engineering, University of Ljubljana, Tržaška cesta 25, SI-1000 Ljubljana, Slovenia

DOI: 10.21105/joss.04377

Software

- Review
- Repository
- Archive

Editor: Patrick Diehl

Reviewers:

- @stsievert
- @jonjoncardoso

Submitted: 24 March 2022
Published: 25 October 2022

Summary

Solving mathematical optimization problems is a critical task for many disciplines, ranging from cutting-edge scientific research to the management of financial portfolios. Due to the inherent complexity of such problems, a plethora of different algorithms and software have been developed for solving them, and this necessitated a standard collection of test problems on which optimization algorithms and software can be evaluated. PyCUTEst provides efficient access to the extensive CUTEst (Gould et al., 2015) library of nonlinear optimization test problems, long a standard test set for nonlinear optimization. In particular, PyCUTEst:

- assists numerical algorithm and software developers in testing new ideas against a state-of-the-art collection of test problems that span small- and large-scale, constrained and unconstrained, nonlinear optimization problems;
- allows scientists and other users of optimization software to compare candidate algorithms and software on standard test problems, helping them select the tools best suited to their needs; and
- is easy to install via pip and our detailed documentation provides instructions on how users can easily install the underlying CUTEst test collection.

In short, our aim is that PyCUTEst will make it easier for users to test new and existing optimization algorithms and software in Python. With over 15,000 downloads at the time of writing, we firmly believe that PyCUTEst is well on the way to achieving this aim.

State of the field

The CUTEst (Gould et al., 2015) library is a widely used collection of nonlinear optimization test problems, based on the original CUTE (Bongartz et al., 1995) and CUTEr (Gould et al., 2003) packages. It has a collection of over 1,500 problems, many of which are parametrized to allow for variable dimensions through user-selectable parameters. However, despite the popularity of CUTEst, it is currently only accessible through Fortran, C, or MATLAB interfaces provided with the main package, or through the Julia interface CUTEst.jl (Orban et al., 2020). In particular, it is not possible to use CUTEst in Python, even though Python is widely used in numerical computing and has a large ecosystem of open source software for nonlinear optimization.

The other widely used packages that encode optimization test problems are the modelling languages AMPL (Inc., n.d.) and GAMS (Corp., n.d.). Although both provide Python interfaces, and in fact many CUTEst problems have been translated into AMPL (Yurttan, n.d.), they are

proprietary packages. An open-source alternative to AMPL and GAMS is the Julia package JuMP (Dunning et al., 2017).

Statement of need

PyCUTEst gives Python users access to the full CUTEst (Gould et al., 2015) collection of optimization test problems via a simple interface for compiling problems (that automatically generates a C interface to the underlying Fortran package). To the best of our knowledge, this is the only available Python package for accessing the CUTEst library that is stable and maintained.

The main benefits of the PyCUTEst package are that it:

- enables the use of the CUTEst test collection by the sizeable community of Python optimization software developers and users; and
- allows simple benchmarking of optimization algorithms and software in Python against a widely used standard collection of test problems.

Our aim is for PyCUTEst to make it easier for both optimization users and software developers to develop and test new and existing algorithms and software in Python. Since its inception, just over four years ago at the time of writing, PyCUTEst has had over 15,000 downloads and we believe is well on the way to achieving this aim.

Acknowledgements

We would like to thank Nick Gould for his helpful advice when we were developing PyCUTEst and Coralia Cartis for suggesting that we make PyCUTEst more widely available.

References

Bongartz, I., Conn, A. R., Gould, N., & Toint, Ph. L. (1995). CUTE: Constrained and unconstrained testing environment. *ACM Transactions on Mathematical Software*, *21*(1), 123–160. https://doi.org/10.1145/200979.201043

Corp., G. D. (n.d.). *GAMS - cutting edge modelling*. https://www.gams.com/ (accessed 2022-03-16).

Dunning, I., Huchette, J., & Lubin, M. (2017). JuMP: A modeling language for mathematical optimization. *SIAM Review*, *59*(2), 295–320. https://doi.org/10.1137/15M1020575

Gould, N. I. M., Orban, D., & Toint, P. L. (2003). CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. *ACM Transactions on Mathematical Software*, *29*(4), 373–394. https://doi.org/10.1145/962437.962439

Gould, N. I. M., Orban, D., & Toint, P. L. (2015). CUTEst: A constrained and unconstrained testing environment with safe threads for mathematical optimization. *Computational Optimization and Applications*, *60*(3), 545–557. https://doi.org/10.1007/s10589-014-9687-3

Inc., A. O. (n.d.). *AMPL: Streamlined modeling for real optimization*. https://ampl.com/ (accessed 2022-03-16).

Orban, D., Siqueira, A. S., & contributors. (2020). *CUTEst.jl: Julia's CUTEst interface*. https://github.com/JuliaSmoothOptimizers/CUTEst.jl. https://doi.org/10.5281/zenodo.1188851

Yurttan, H. (n.d.). *Cute models*. https://vanderbei.princeton.edu/ampl/nlmodels/cute/index.html (accessed 2022-03-16).