perprof-py
A Python Package for Performance Profile

Raniere Gaia Costa da Silva raniere@ime.unicamp.br
Abel Soares Siqueira abel@ime.unicamp.br
Luiz-Rafael Santos lrsantos11@gmail.com

UNICAMP
IMECC

Introduction

Benchmarking optimization packages is very important in the optimization field, not only because it is one of the ways to compare solvers, but also to uncover deficiencies that could be overlooked. During benchmarking, a large amount of information is obtained, like CPU time, number of function evaluations, number of iterations and so on. This information, if presented as tables, can be difficult to be analyzed, due, for instance, to large amounts of data. Therefore, researchers started testing tools to better process and understand this data. One of the most widespread ways to do so is by using Performance Profile graphics as proposed by Dolan and Moré [1].

Performance Profile

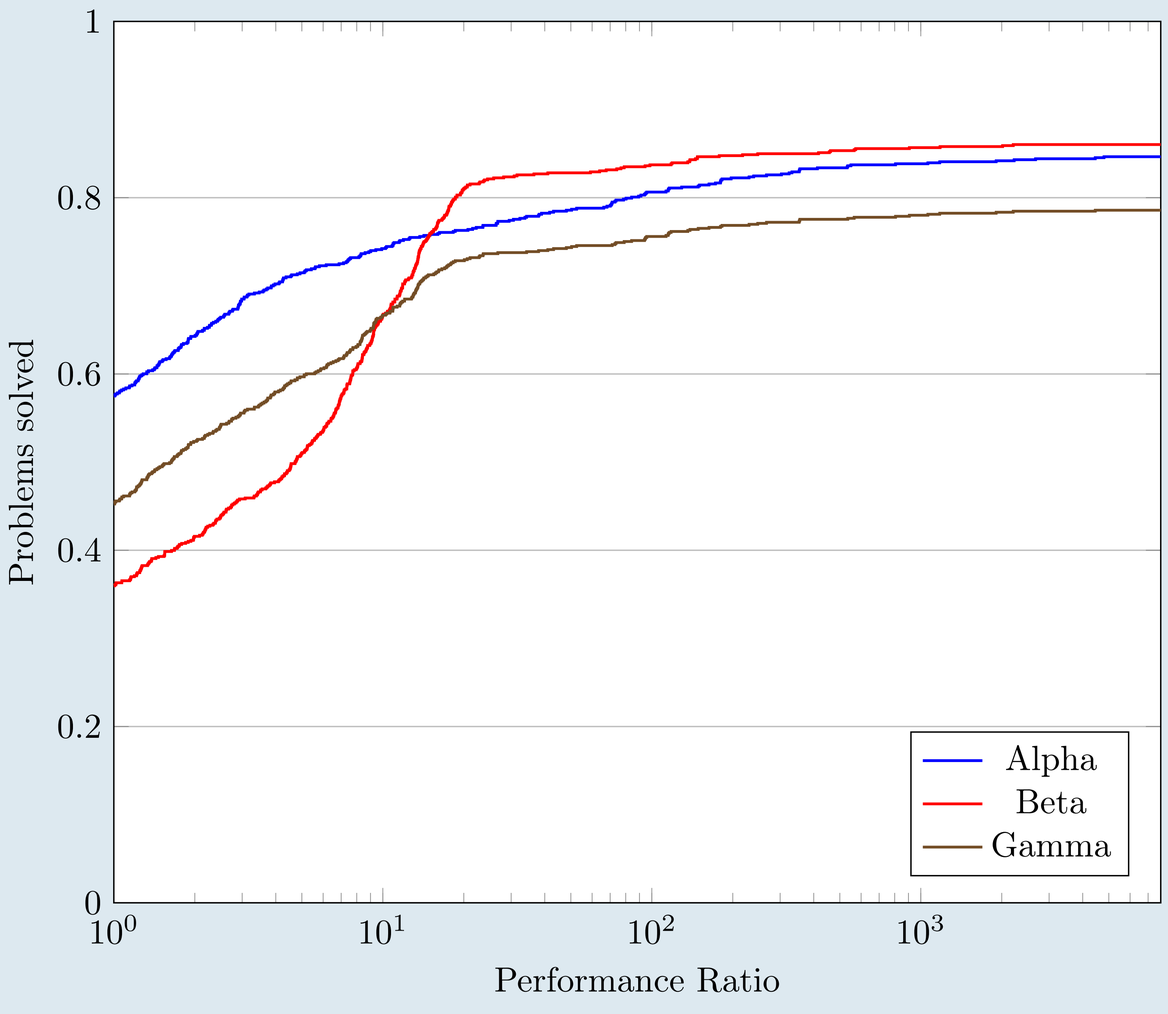
The performance profile of a solver is the cumulative distribution function of a performance metric, e.g., CPU time, number of functions evaluations, number of iterations, and so on, that we will call *cost*.
Given a set P of problems and a set S of solvers, for each problem $p \in P$ and solver $s \in S$, we define t_{ps} as the cost required to solve problem p by solver s and

$$r_{ps} = \frac{t_{ps}}{\min\{t_{ps} : s \in S\}}$$

as the performance ratio of solver s for the problem p when compared with the best performance by any solver on this problem.
Dolan and Moré [1] define

$$\rho_s(\tau) = \frac{|\{p \in P : r_{ps} \leq \tau\}|}{|P|}$$

where $\rho_s(\tau)$ is the probability for solver $s \in S$ to solve one problem within a factor $\tau \in \mathbb{R}$ of the best performance ratio. For a given τ, the best solver is the one with the highest value for $\rho_s(\tau)$.



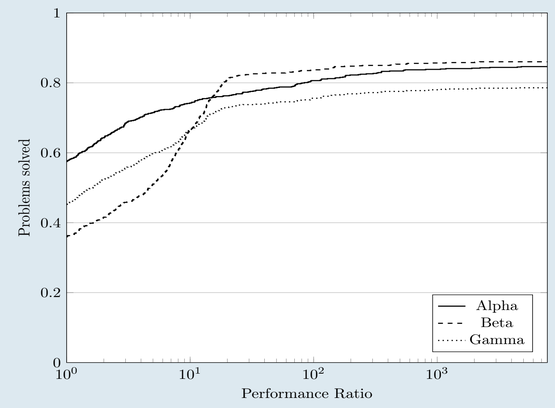
perprof-py

We implemented a free/open software that generates performance profiles using data provided by the user in a friendly manner. This software produces graphics in PDF using LaTeX with PGF/TikZ and pgfplots packages, in PNG using matplotlib and can also be easily extended to use other plot libraries. The software is implemented in Python3 with support for internationalization and is available at `https://github.com/lpoo/perprof-py` under GPLv3 license.

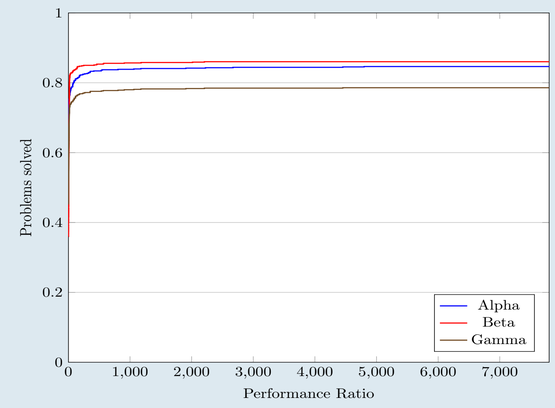
Features

Some of perprof-py features are:

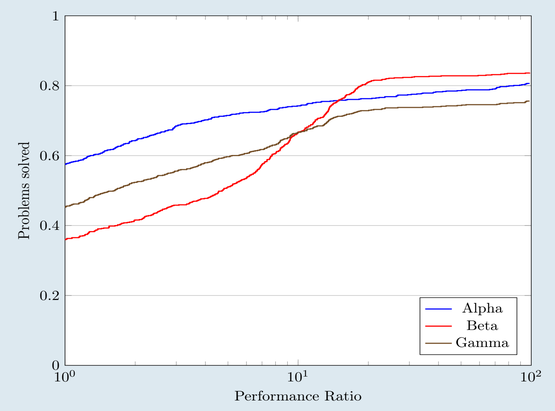
- high quality output (in more than one format);
- lots of native options/filters;
- input processing to limit solver minimum and maximum costs and set of problems;
- native internationalization support.



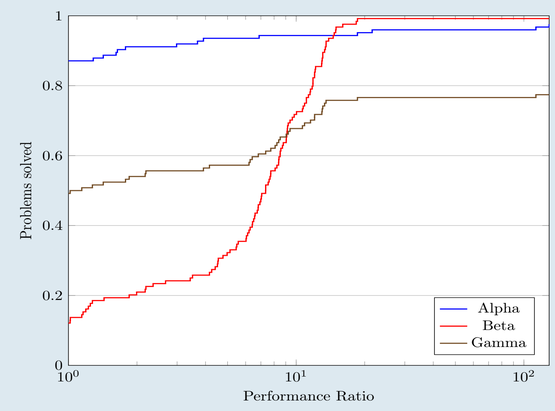
Black and white



Linear scale



Performance ratio limit



Subset (Hock-Schittkowski)

Besides the previous features, the package can be easily used in a range of external tools that support Python.

Input file

Each solver to be used in the benchmark must have a file like:

```
YAML information
Problem01 exit01 time01
Problem02 exit02 time02
```

In the YAML information you can set the name of the solver, and some flags for perprof-py. Each line beyond that has 3 columns that mean, in order:

- The name of the problem;
- Exit flag;
- Elapsed time.

Future works

Some of our next improvements consist of:

- add metadata support and also support data profiles as defined by Moré and Wild [2];
- add support for the analysis of the objective value, and the primal and dual infeasibilities;
- add more backends and more output formats.

References

[1] Elizabeth D. Dolan and Jorge J. Moré. “Benchmarking Optimization Software with Performance Profiles”. In: *Mathematical Programming* 91.2 (2002), pp. 201–213.

[2] J. Moré and S. Wild. “Benchmarking Derivative-Free Optimization Algorithms”. In: *SIAM J. Optimization* 20 (1) (2009), pp. 172–191.