Exercise: Benchmarking Derivative-Free Optimization Algorithms

July 5, 2017

CEA/EDF/Inria summer school "Numerical Analysis" Université Pierre-et-Marie-Curie, Paris, France

Anne Auger, Asma Atamna, and Dimo Brockhoff Inria Saclay – Ile-de-France CMAP, Ecole Polytechnique









Requirements:

python 2.6 or 2.7

use Anaconda if not already installed: https://www.continuum.io/downloads)

note: python 3.x not yet supported

matplotlib >2.0

in principle via `pip install matplotlib`
otherwise, see http://matplotlib.org/users/installing.html

Objectives of the exercise:

- Get acquainted with the COCO platform at least with its postprocessing and visualization
- Or a construction of the 150+ algorithm data sets of COCO

	https://github.o	com/numbbo/c	oco	
numbbo/coco: Numerical >	+			
🛈 🔒 GitHub, Inc. (US) https://	github.com/numbbo/coco	C Search	☆ 自 ♥ ↓	
ost Visited Getting Started	COCO-Algorithms 🎧 numbbo/numbbo · Gi	💪 RandOpt 🛞 CMAP 🛞 Inria GitLab 🕤	RER B from lab	
This repository Sea	rch Pull requests	lssues Marketplace Gist	≰ +-	
] numbbo / coco		O Unwatch	✓ 15 ★ Unstar 38 ¥ Fork	k 24
<> Code ① Issues 13	4 🕅 Pull requests 1	Stop 1		
Jumerical Black-Box Opt	imization Benchmarking			Edit
dd topics				
dd topics To 16,007 commits	s ¥ 11 branches	© 31 releases	15 contributors	
add topics To 16,007 commits	s [§] 11 branches	© 31 releases	15 contributors	
dd topics ⑦ 16,007 commit: Branch: master ▼ New pull	request	© 31 releases	Upload files Find file Clone or down	nload -
Add topics T 16,007 commit: Branch: master New pull Drockho committed on C	s ¥ 11 branches request	© 31 releases Create new file /development	Upload files Find file Clone or down	nload - 20 Apr
dd topics ⑦ 16,007 commit: Branch: master → New pull Ŵ brockho committed on C ■ code-experiments	s P 11 branches request SitHub Merge pull request #1352 from numbbo/ A little more verbose error message wi	© 31 releases Create new file /development ···	Upload files Finc file Clone or down Latest commit 4b1497a on 2 a mont	nload - 20 Apr th ago
dd topics 16,007 commit: Branch: master • New pull W brockho committed on 0 code-experiments code-postprocessing	s P 11 branches request SitHub Merge pull request #1352 from numbbo/ A little more verbose error message will Hashes are back on the plots.	© 31 releases Create new file /development ···	Upload files Finc file Clone or down Latest commit 4b1497a on a a mont	nload - 20 Apr th ago th ago
The second secon	s P 11 branches request SitHub Merge pull request #1352 from numbbo/ A little more verbose error message will Hashes are back on the plots. Fixed preprocessing to work correctly with the preprocessing to work correctly with the plots.	© 31 releases Create new file /development hen suite regression test fails with the extended biobjective s	Upload files Find file Clone or down Latest commit 4b1497a on 2 a mont 3 month	nload - 20 Apr th ago th ago hs ago
The second secon	s P 11 branches request SitHub Merge pull request #1352 from numbbo/ A little more verbose error message will Hashes are back on the plots. Fixed preprocessing to work correctly will Update create-a-suite-howto.md	S 31 releases Create new file Vdevelopment	Upload files Find file Clone or down Latest commit 4b1497a on 2 a mont 3 month 4 month	nload - 20 Apr th ago th ago hs ago hs ago
Add topics The topics The topics The topics The topics The topic of topic of the topic of the topic of to	s II branches request SitHub Merge pull request #1352 from numbbo/ A little more verbose error message will Hashes are back on the plots. Fixed preprocessing to work correctly v Update create-a-suite-howto.md raising an error in bbob2009_logger.c	Solution Solution Solution Solution Create new file Create new file //development hen suite regression test fails with the extended biobjective s with the extended biobjective s when best_value is NULL. Plus s	Latest commit 4b1497a on 2 a mont 3 mont 2 yea	nload - 20 Apr th ago th ago hs ago hs ago rs ago
Idd topics Idd topics Idd topics Idd topics Idd topics Branch: master ▼ New pull Idd topics	s P 11 branches request SitHub Merge pull request #1352 from numbbo/ A little more verbose error message will Hashes are back on the plots. Fixed preprocessing to work correctly will Update create-a-suite-howto.md raising an error in bbob2009_logger.c raising an error in bbob2009_logger.c	Solution Solution Solution Solution Create new file Create new file //development hen suite regression test fails with the extended biobjective s with the extended biobjective s when best_value is NULL. Plus s when best_value is NULL. Plus s Solution	Latest commit 4b1497a on 2 a mont 3 mont 2 yea 2 yea	nload - 20 Apr th ago th ago hs ago rs ago rs ago
Add topics The format of the	s P 11 branches request SitHub Merge pull request #1352 from numbbo/ A little more verbose error message will Hashes are back on the plots. Fixed preprocessing to work correctly will Update create-a-suite-howto.md raising an error in bbob2009_logger.c raising an error in bbob2009_logger.c	Solution Solution Solution Solution Create new file Create new file Vdevelopment hen suite regression test fails with the extended biobjective s when best_value is NULL. Plus s when best_value is NULL. Plus s	Upload files Find file Clone or down Latest commit 4b1497a on 2 a mont 3 mont 4 mont 2 yea a yea	nload - 20 Apr th ago th ago hs ago rs ago ar ago

https://github.com/numbbo/coco _ **D** X 🖸 numbbo/coco: Numerical ... 🗙 + C Search 0 GitHub, Inc. (US) https://github.com/numbbo/coco C 🖻 Most Visited 🛞 Getting Started 🥰 COCO-Algorithms 😱 numbbo/numbbo · Gi... 🔀 RandOpt 🛞 CMAP 🛞 Inria GitLab 🕥 RER B from lab - 🛄 This repository Search Pull requests Issues Marketplace Gist Inumbbo / coco O Unwatch ▼ 15 * Unstar 38 Y Fork 24 Pull requests 1 <> Code ① Issues 134 Step 1: download COCO Numerical Black-Box Optimization Benchmarking Edit Add topics 16,007 commits 31 releases 2 11 branches 15 contributors New pull request Create new file Upload files Find file Clone or download -Branch: master • Use SSH 🔛 brockho committed on GitHub Merge pull request #1352 from numbbo/development 🛛 …. Clone with HTTPS ⑦ Use Git or checkout with SVN using the web URL. code-experiments A little more verbose error message when suite regression test fai 良 https://github.com/numbbo/coco.git code-postprocessing Hashes are back on the plots. code-preprocessing Fixed preprocessing to work correctly with the extended biobjectiv **Open in Desktop** Download ZIP Update create-a-suite-howto.md 4 months ago howtos .clang-format raising an error in bbob2009_logger.c when best_value is NULL. Plus s... 2 years ago raising an error in bbob2009_logger.c when best_value is NULL. Plus s... .hgignore 2 years ago AUTHORS small correction in AUTHORS a year ago E) LICENCE Lindata LICENICE

https://github.com/numbbo/coco

numbha/coco: Numerical									X
O GitHub, Inc. (US) https://github.com/numbbo/coco	C	Ce Search		☆	Ê		÷	Â	=
Most Visited 🛞 Getting Started 🎇 COCO-Algorithms 🕥 numbbo/numbbo · Gi	🔏 RandOpt 🛞 CMAP	🛞 Inria GitLab 🕥 RER I	B from lab						
2 In a system shell, cd into the coco, or coco-syers in	n> folder (framewo	rk root), where the	file do ny	can	be fc	ound			^
Type, i.e. execute, one of the following commands onc	ce	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,							
python do.py run-c									
python do.py run-java									
python do.py run-matlab									H
python do.py run-octave									
depending on which language shall be used to run the exp example experiment once. The build result and the example	periments. run-* wi e experiment code o	ill build the respecti can be found under	ve code and code-expe	d run rime	n the ents/	build	ł		
<pre>/<language> (<language>=matlab for Octave). python do</language></language></pre>	o.py lists all availab	le commands.							
3. On the computer where experiment data shall be post	t-processed, run								
python do.py install-postprocessing	nstallati	Step	2: st-p	ro	C	259	siı	าต	
to (user-locally) install the post-processing. From here on,	do.py has done its	job and is only nee	ded again f	or u	pdati	ing th	е		
builds to a new release.	000000000	a 8			31				
4. Copy the folder code-experiments/build/YOUR-FAVOR	RITE-LANGUAGE and	its content to anoth	ner location.	. In F	ytho	on it is			
sufficient to copy the file example_experiment.py . Ru vary, see the respective read-me's and/or example exp	un the example expe periment files:	riment (it already is	compiled).	As t	he de	etails			

- c read me and example experiment
- Java read me and example experiment

http://coco.gforge.inria.fr/doku.php?id=algorithms



[[algorithms]] **COMPARING CONTINUOUS OPTIMISERS: COCO** Show pagesource 🔜 Old revisions 📧 Recent changes 🔍 Sitemap 👰 Login Search The following table lists all algorithms related to the BBOB workshops and special sessions in the years 2009 till 2015 together with links to Navigation their data. In order to sort the table according to some columns, please click on the corresponding table header. If available, the source Home codes of the algorithms can be downloaded by clicking on the link with the corresponding algorithm name in the second column. Documentation Data Noiseless Data Noisy download latest old code related PDFs and Remarks No Algorithm Year Author(s) • Onew code homepage (Raw) (Raw) download new code directly noisvData O PDF 1 ALPS 2009 Hornby noiselessData BBOB 2016 noiselessData PDFnoiseless PDFnoisy noisyData 2 AMALGAM 2009 Bosman et al. BBOB 2015 @ GECCO PDFnoiseless PDFnoise Algorithms 3 BAYEDA Gallagher noiselessData noisyData 2009 BFGS noiselessData noisyData O PDFnoiseless O PDF 4 2009 Ros for the moment: PDFnoiseless PDF 5 BIPOP-CMA-ES 2009 Hansen noiselessData noisyData noiselessData PDF 6 Cauchy-EDA 2009 Pošík n/a **IPOP-CMA-ES** Auger and 😡 noiselessData O PDFnoiseless O PDF 7 CMA-ESPLUSSEL 2009 noisyData Hansen (algorithm 56) Korošec and PDFnoiseless PDF 2009 noiselessData noisyData 8 DASA Šilc García-Nieto noiselessData noisyData 😡 PDFnoiseless PDFn 9 DE-PSO 2009 et al. Downloads O PDF BBOB 2012 2009 Pošík noiselessData 10 DIRECT n/a algorithm is deterministic and thus, only run on each Algorithms instance once Results Downloads El-Abd and noiselessData noisyData O PDF 11 EDA-PSO 2009 Kamel BBOB 2010

https://github.com/numbbo/coco

Coco/README.md at devel × +						
◯	C Search	☆	Ê	•	F 🕯	r =
lost Visited 🛞 Getting Started 🥰 COCO-Algorithms 🎧 numbbo/numbbo · Gi 🛽	🛃 RandOpt 🛞 CMAP 🛞 Inria GitLab 🕥 RER B fro	m lab				
Another entry point for your own experiments can be the c 5. Now you can run your favorite algorithm on the bbob bbob-biobj-ext suites (for multi-objective algorithms result_folder . By now, more suites might be availab	ode-experiments/examples folder. suite (for single-objective algorithms) or (). Output is automatically generated in the le, see below.	on the bbot e specified ()-biobj data	j and		
6. Postprocess the data from the results folder by typing						
python -m cocopp [-o OUTPUT_FOLDERNAME] YOURDATA	FOLDER [MORE_DATAFOLDERS]	ostpr	oc	es	S	
Any subfolder in the folder arguments will be searched for l different folders collected under a single "root" YOURDATAFC	ogged data. That is, experiments from diff DLDER folder. We can also compare more	rerent batch than one alç	ies can gorithn	be in n by		
ython -m cocopp IPOP-C	MA-ES ros nois	sele:	SS.	ta	ar.	g
A summary pdf can be produced via LaTeX. The correspond templates folder. Basic html output is also available in the templateBBOBarticle.html).	ling templates can be found in the code-p result folder of the postprocessing (file	oostprocess	ing/la	atex-		

- 7. Once your algorithm runs well, **increase the budget** in your experiment script, if necessary implement randomized independent restarts, and follow the above steps successively until you are happy.
- 8. The experiments can be **parallelized** with any re-distribution of single problem instances to batches (see example_experiment.py for an example). Each batch must write in a different target folder (this should happen

https://github.com/numbbo/coco

O coco/README.md at devel × +					×
(G GitHub, Inc. (US) https://github.com/numbbo/coco C	Cearch	☆ 自	♥ .	F 🕆	=
🔊 Most Visited 🛞 Getting Started COCO-Algorithms 🕥 numbbo/numbbo · Gi 💋 RandOpt 🛞 CMAP	🛞 Inria GitLab 🥎 RER B from lab				
Another entry point for your own experiments can be the code-experiments/o 5. Now you can run your favorite algorithm on the bbob suite (for single-ob bbob-biobj-ext suites (for multi-objective algorithms). Output is automa result_folder . By now, more suites might be available, see below.	examples folder. jective algorithms) or on the stically generated in the speci	bbob-biob fied data	j and		*
6. Postprocess the data from the results folder by typing python -m cocopp [-o OUTPUT_FOLDERNAME] YOURDATAFOLDER [MORE_DATA	FOLDERS]	proc	es	S	
Any subfolder in the folder arguments will be searched for logged data. That is different folders collected under a single "root" YOURDATAFOLDER folder. We ca	, experiments from different i an also compare more than or	oatches car ne algorithr	n be in m by		
python -m cocopp IPOP-CMA-ES_	ros_noisel	ess	.ta	ır.	gz
Alternative within (l)python:					

> cocopp.main("IPOP-CMA-ES_ros_noiseless.tar.gz")

Reminder: Measuring Performance Empirically

convergence graphs is all we have to start with...



number of function evaluations

ECDF:

Empirical Cumulative Distribution Function of the Runtime [aka data profile]

A Convergence Graph



First Hitting Time is Monotonous



15 Runs



15 Runs ≤ 15 Runtime Data Points



Empirical Cumulative Distribution



the ECDF of run lengths to reach the target

- has for each data point a vertical step of constant size
- displays for each x-value (budget) the count of observations to the left (first hitting times)

Empirical Cumulative Distribution



- 0.8. the higher the better (= more
 0.6 targets solved)
 - the more to the left the better (= targets solved quicker)

Aggregation



15 runs50 targets

Aggregation



15 runs50 targetsECDF with 750 steps

Fixed-target: Simulated Restarts



ECDFs with Simulated Restarts

What we typically plot are ECDFs of the simulated restarted algorithms:



Exercise

Objective:

investigate the performance of algorithms, available at

http://coco.gforge.inria.fr/doku.php?id=algorithms

- 56 IPOP-CMA-ES: CMA-ES with restarts and increasing popsize
 - 5 BIPOP-CMA-ES: two different popsize regimes
- 22 Nelder-Mead simplex (use better "NelderDoerr" version here)
 - 4 BFGS: quasi-Newton
- 14 Genetic Algorithm: discretization of cont. variables ("GA")
- 25 ONEFIFTH: (1+1)-ES with 1/5 rule

postprocess (now) and investigate the data (after a few more slides)

tip: use --omit-single option to save time

The single-objective BBOB functions

The bbob Testbed

• 24 functions in 5 groups:

1 Separable Functions			4 Multi-modal functions with adequate global structur					
f1	Sphere Function	f15	Rastrigin Function					
f2	Sellipsoidal Function	f16	Weierstrass Function					
f3	Rastrigin Function	f17	Schaffers F7 Function					
f4	Büche-Rastrigin Function	f18	Schaffers F7 Functions, moderately ill-conditioned					
f5	♥Linear Slope	f19	Composite Griewank-Rosenbrock Function F8F2					
2 Functions with low or moderate conditioning			5 Multi-modal functions with weak global structure					
f6	Attractive Sector Function	f20	Schwefel Function					
f7	Step Ellipsoidal Function	f21	Gallagher's Gaussian 101-me Peaks Function					
f8	Rosenbrock Function, original	f22	Gallagher's Gaussian 21-hi Peaks Function					
f9	Rosenbrock Function, rotated	f23	Katsuura Function					
3 F	unctions with high conditioning and unimodal	f24	Lunacek bi-Rastrigin Function					
f10	Sellipsoidal Function							
f11	ODiscus Function							
f12	Bent Cigar Function							
f13	Sharp Ridge Function							
f14	ODifferent Powers Function							

• 6 dimensions: 2, 3, 5, 10, 20, (40 optional)

Notion of Instances

- All COCO problems come in form of instances
 - e.g. as translated/rotated versions of the same function
- Prescribed instances typically change from year to year
 - avoid overfitting
 - 5 instances are always kept the same

Plus:

 the bbob functions are locally perturbed by nonlinear transformations

Notion of Instances



The single-objective noisy BBOB functions

bbob-noisy Testbed

- 30 functions with various kinds of noise types and strengths
 - 3 noise types: Gaussian, uniform, and seldom Cauchy
 - Functions with moderate noise
 - Functions with severe noise
 - Highly multi-modal functions with severe noise
 - ььоь functions included: Sphere, Rosenbrock, Step ellipsoid, Ellipsoid, Different Powers, Schaffers' F7, Composite Griewank-Rosenbrock
- 6 dimensions: 2, 3, 5, 10, 20, (40 optional)

COCO extended to multiobjective optimization

bbob-biobj Testbed

• 55 functions by combining 2 ььоь functions

1 S	eparable Functions	4 M	ulti-modal functions with adequate global structure
f1	Sphere Function √	f15	Rastrigin Function
f2	♥Ellipsoidal Function ✓	f16	Weierstrass Function
f3	Rastrigin Function	f17	Schaffers F7 Function ✓
f4	Büche-Rastrigin Function	f18	Schaffers F7 Functions, moderately ill-conditioned
f5	Linear Slope	f19	Composite Griewank-Rosenbrock Function F8F2
2 F	unctions with low or moderate conditioning	5 M	ulti-modal functions with weak global structure
f6		f20	Schwefel Function 🗸
f7	Step Ellipsoidal Function	f21	Gallagher's Gaussian 101-me Peaks Function ✓
f8	Rosenbrock Function, original	f22	Gallagher's Gaussian 21-hi Peaks Function
f9	Rosenbrock Function, rotated	f23	Katsuura Function
3 F	unctions with high conditioning and unimodal	f24	Lunacek bi-Rastrigin Function
f10	Ellipsoidal Function		
f11	ODiscus Function		
f12	Bent Cigar Function		
f13	Sharp Ridge Function ✓		
f14	♥Different Powers Function ✓		
-			

bbob-biobj Testbed

• 55 functions by combining 2 ььоь functions

1 Separable Functions			4 Multi-modal functions with adequate global structure									
f1	f1 Sphere Function			f15 🕢 Rastrigin Function 🗸								
f2	Sellipsoidal Function √			f16 Weierstrass Function								
f3	Rastrigin Function			f17 Schaffers F7 Function								
f4	Büche-Rastrigin Function		f_1	fa	f_6	f_8	f_{12}	f_{14}	f_{15}	f_{17}	f_{20}	f_{21}
f5	♥Linear Slope	f,	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10
2 F	2 Functions with low or moderate conditionin			<u>12</u>	10	<u>11</u>	1.2	0.5	<u></u>	61.7	12	510
f6	Sttractive Sector Function √	J_2		<u>TII</u>	<u>TI 2</u>	<u>TI 3</u>	<u>114</u>	<u>T15</u>	116	<u>TI /</u>	118	<u>119</u>
f7	Step Ellipsoidal Function	f_6			<u>f20</u>	<u>f21</u>	<u>f22</u>	<u>f23</u>	<u>f24</u>	<u>f25</u>	<u>f26</u>	<u>f27</u>
f8		f_8				<u>f28</u>	<u>f29</u>	<u>f30</u>	<u>f31</u>	<u>f32</u>	<u>f33</u>	<u>f34</u>
f9	Rosenbrock Function, rotated	f_{13}					<u>f35</u>	<u>f36</u>	<u>f37</u>	<u>f38</u>	<u>f39</u>	<u>f40</u>
3 F	unctions with high conditioning and unimo	f_{14}						<u>f41</u>	<u>f42</u>	<u>f43</u>	<u>f44</u>	<u>f45</u>
f10	Sellipsoidal Function	f_{15}							<u>f46</u>	<u>f47</u>	<u>f48</u>	<u>f49</u>
f11	ODiscus Function	f_{17}								<u>f50</u>	<u>f51</u>	<u>f52</u>
f12	Bent Cigar Function	f_{20}									f53	f54
f13	Sharp Ridge Function √	fai										f55
f14	Different Powers Function	J 21										

bbob-biobj Testbed

- 55 functions by combining 2 ььоь functions
- 15 function groups with 3-4 functions each
 - separable separable, separable moderate, separable ill-conditioned, ...
- 6 dimensions: 2, 3, 5, 10, 20, (40 optional)
- instances derived from ъъоъ instances:
 - more or less 2i+1 for 1st objective and 2i+2 for 2nd objective
 - exceptions: instances 1 and 2 and when optima are too close
- no normalization (algo has to cope with different orders of magnitude)
- for performance assessment: ideal/nadir points known

- Pareto set and Pareto front unknown
 - but we have a good idea of where they are by running quite some algorithms and keeping track of all nondominated points found so far
- Various types of shapes

Example: sphere with sphere



Example: sharp ridge with sharp ridge



Example: sphere with Gallagher 101 peaks





Exercise

Objective:

investigate the data you just postprocessed:

- a) which algorithms are the best ones?
- b) does this depend on the dimension?
- c) look at single graphs: can we say something about the algorithms' invariances, e.g. wrt. rotations of the search space?
- d) what's the impact of covariance-matrix-adaptation?
- e) what do you think: are the displayed algorithms well-suited for problems with larger dimension?

Paper Discussion: "Dynamic Search in Fireworks Algorithm"

http://eprints.cs.univie.ac.at/4082/1/PID3181839.pdf

Objectives of the exercise:

Learn how to read an optimization paper critically in order to be able to do high-quality reviews

but also to get the most information from it for practical purposes

Output: Understand the reasoning behind the COCO concepts by looking at how others benchmark optimization algorithms