Comparison with State-of-the-Art: Traps and Pitfalls

Rafał Biedrzycki

Warsaw University of Technology
Institute of Computer Science
riedrzy@elka.pw.edu.pl

Popular road to publication

- Create some algorithm
- Compare to state-of-the-art
- If the algorithm is better for some problems, then write an article

Popular road to publication

- Create some algorithm
- Compare to state-of-the-art
- If the algorithm is better for some problems, then write an article

Popular road to publication

- Create some algorithm
- Compare to state-of-the-art
- If the algorithm is better for some problems, then write an article

Usual road to publication

- What is state-of-the-art? – known algorithms with source code available in researchers favorite programming language
- Cite the first paper that introduced algorithm used in comparison

Usual road to publication

- What is state-of-the-art? – known algorithms with source code available in researchers favorite programming language
- Cite the first paper that introduced algorithm used in comparison

Article-implementation gap

- Usually articles skip some details that are needed by the implementation
- These details can be filled in different ways by the developers
- Different implementations can give different results

Article-implementation gap

- Usually articles skip some details that are needed by the implementation
- These details can be filled in different ways by the developers
- Different implementations can give different results

Article-implementation gap

- Usually articles skip some details that are needed by the implementation
- These details can be filled in different ways by the developers
- Different implementations can give different results

Use trusted implementation

- What will happen if we download and use implementations created by one person, an author of a method
- CMA-ES will serve as an example of a good method, with high-quality implementations
- Implementations in Python, Matlab, C, Java were downloaded from the author's homepage

Use trusted implementation

- What will happen if we download and use implementations created by one person, an author of a method
- CMA-ES will serve as an example of a good method, with high-quality implementations
- Implementations in Python, Matlab, C, Java were downloaded from the author's homepage

Use trusted implementation

- What will happen if we download and use implementations created by one person, an author of a method
- CMA-ES will serve as an example of a good method, with high-quality implementations
- Implementations in Python, Matlab, C, Java were downloaded from the author's homepage

Experimental setup

- For all implementations, the same population size, initial σ, and maximal number of objective function evaluations were set
- The comparison and analysis of the results were performed by COCO using 24 noiseless single-objective functions formerly used in 2009 in Workshop on Real-Parameter Black-Box Optimization Benchmarking
- Bounds of the area of interest were used as bounds for constrained search, which better reflects a real-world application
- Python version comes with two constraint handling techniques – transformation (default) and weighted quadratic penalty
- A simple implementation in Python which was meant for reading, was also included in the experiments as it is used in direct translations into other languages

Experimental setup

- For all implementations, the same population size, initial σ, and maximal number of objective function evaluations were set
- The comparison and analysis of the results were performed by COCO using 24 noiseless single-objective functions formerly used in 2009 in Workshop on Real-Parameter Black-Box Optimization Benchmarking
- Bounds of the area of interest were used as bounds for constrained search, which better reflects a real-world application
- Python version comes with two constraint handling techniques – transformation (default) and weighted quadratic penalty
- A simple implementation in Python which was meant for reading, was also included in the experiments as it is used in direct translations into other languages

Experimental setup

- For all implementations, the same population size, initial σ, and maximal number of objective function evaluations were set
- The comparison and analysis of the results were performed by COCO using 24 noiseless single-objective functions formerly used in 2009 in Workshop on Real-Parameter Black-Box Optimization Benchmarking
- Bounds of the area of interest were used as bounds for constrained search, which better reflects a real-world application
- Python version comes with two constraint handling techniques – transformation (default) and weighted quadratic penalty
- A simple implementation in Python which was meant for reading, was also included in the experiments as it is used in direct translations into other languages

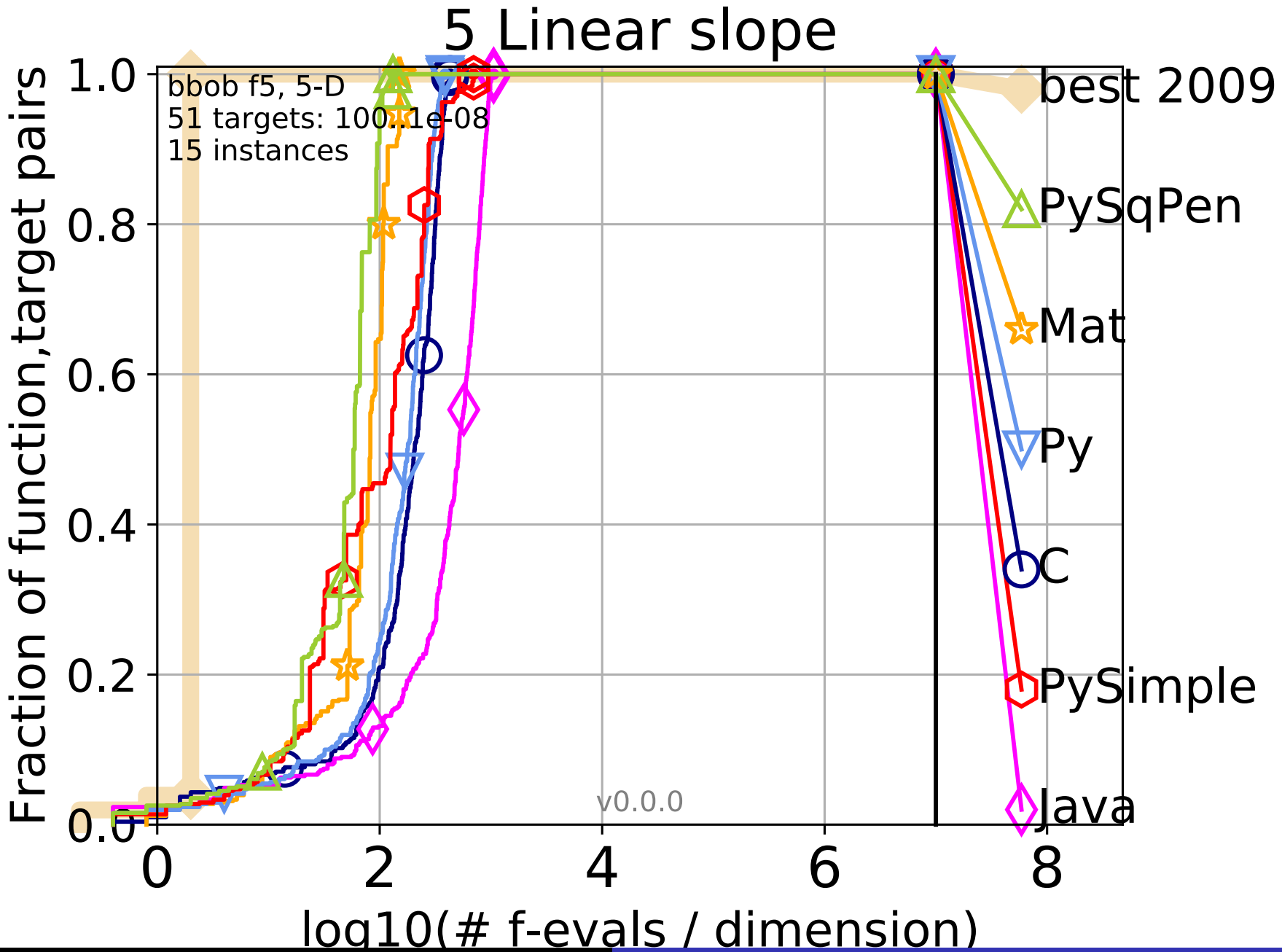
Experimental setup

- For all implementations, the same population size, initial σ, and maximal number of objective function evaluations were set
- The comparison and analysis of the results were performed by COCO using 24 noiseless single-objective functions formerly used in 2009 in Workshop on Real-Parameter Black-Box Optimization Benchmarking
- Bounds of the area of interest were used as bounds for constrained search, which better reflects a real-world application
- Python version comes with two constraint handling techniques – transformation (default) and weighted quadratic penalty
- A simple implementation in Python which was meant for reading, was also included in the experiments as it is used in direct translations into other languages

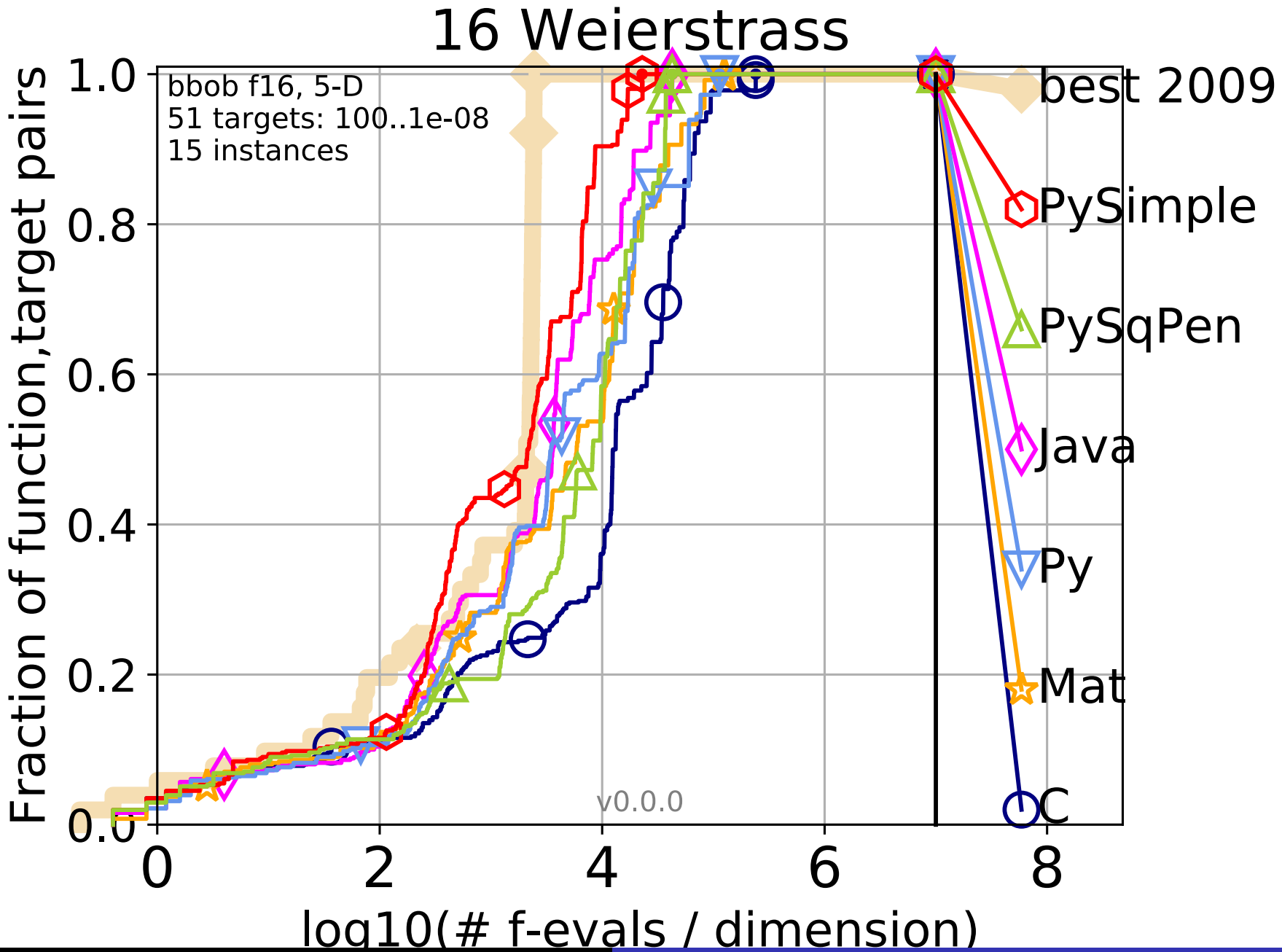
Experimental setup

- For all implementations, the same population size, initial σ, and maximal number of objective function evaluations were set
- The comparison and analysis of the results were performed by COCO using 24 noiseless single-objective functions formerly used in 2009 in Workshop on Real-Parameter Black-Box Optimization Benchmarking
- Bounds of the area of interest were used as bounds for constrained search, which better reflects a real-world application
- Python version comes with two constraint handling techniques – transformation (default) and weighted quadratic penalty
- A simple implementation in Python which was meant for reading, was also included in the experiments as it is used in direct translations into other languages

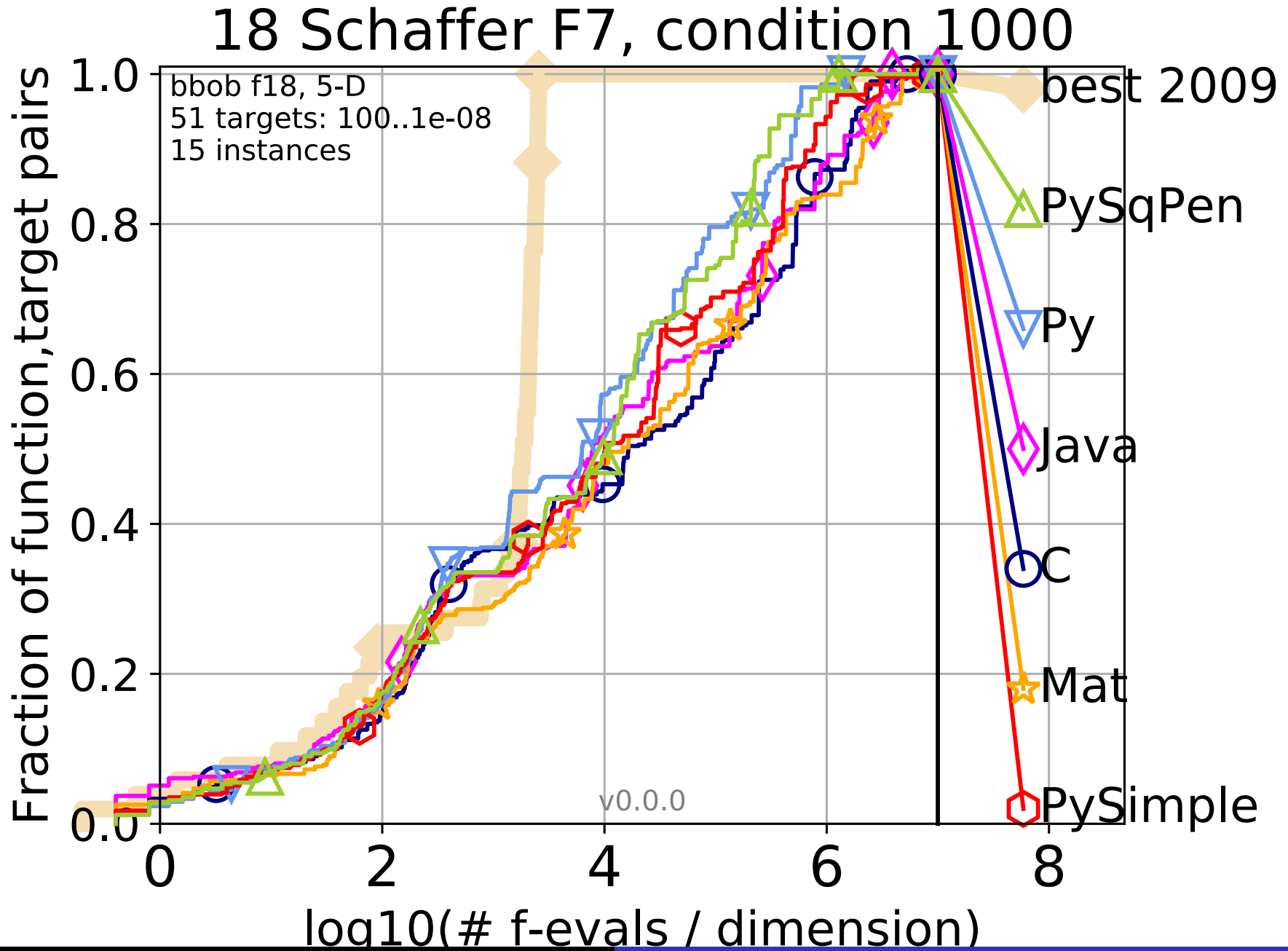
Results on selected functions in 5D



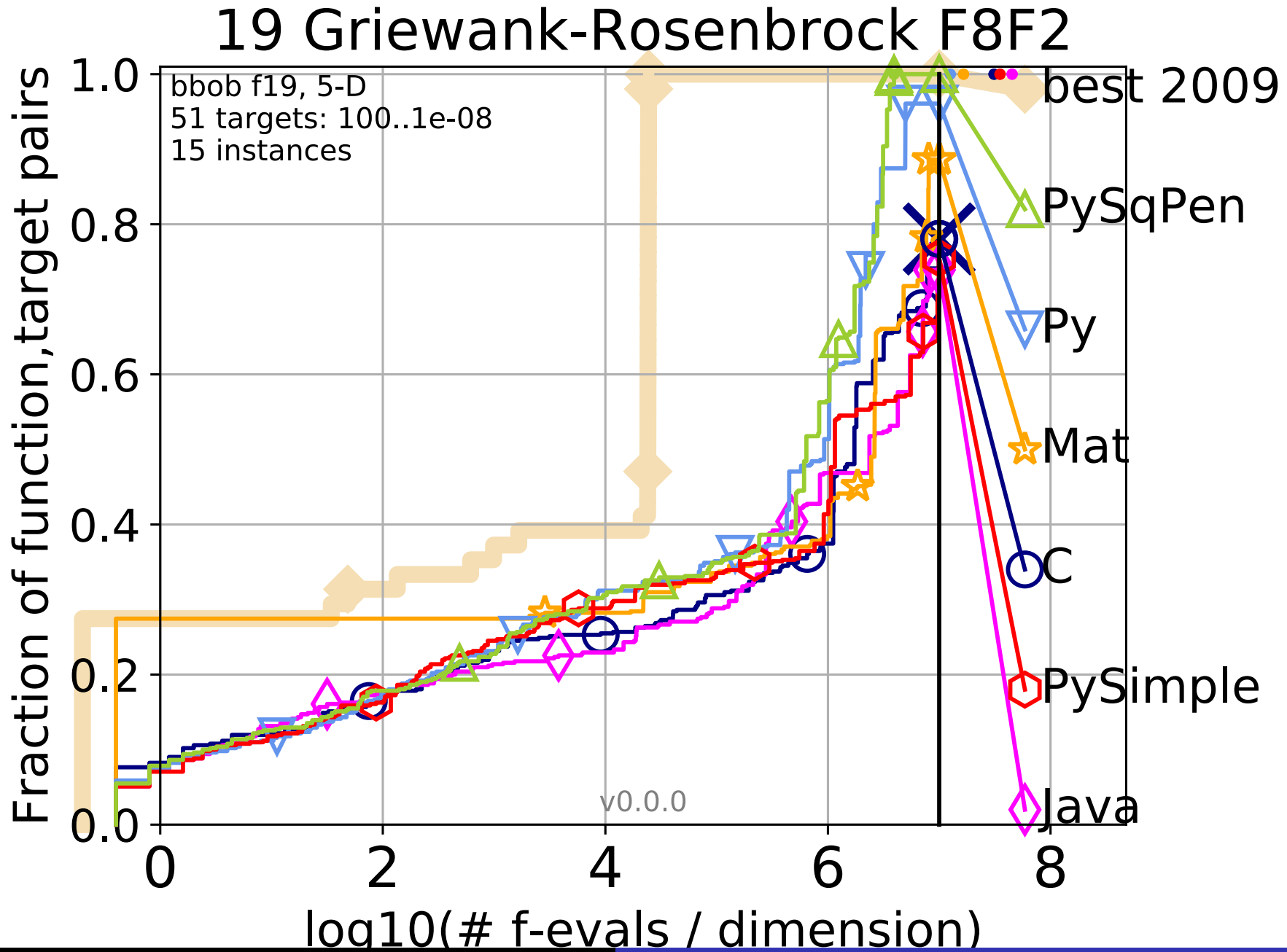
Results on selected functions in 5D



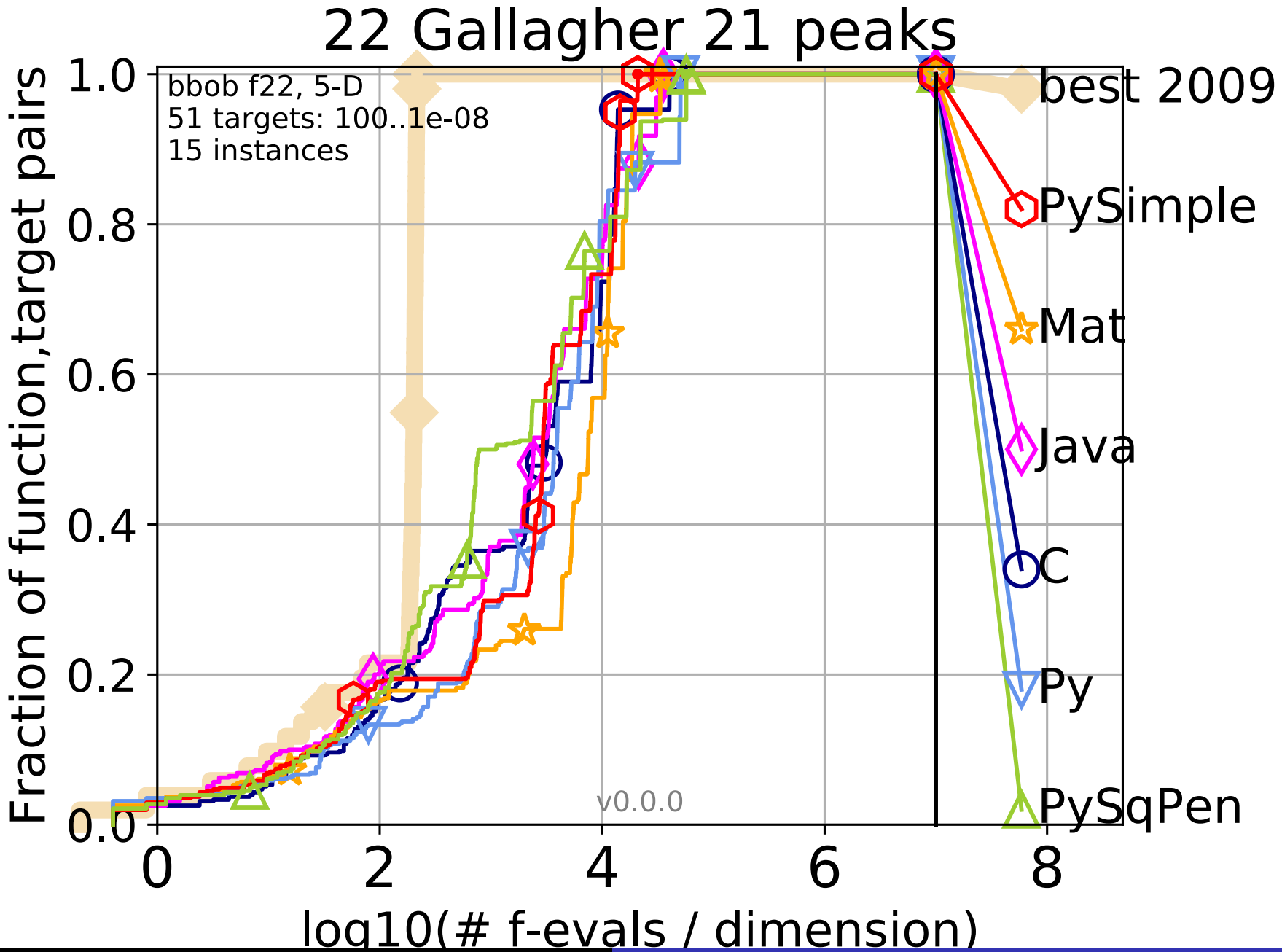
Results on selected functions in 5D



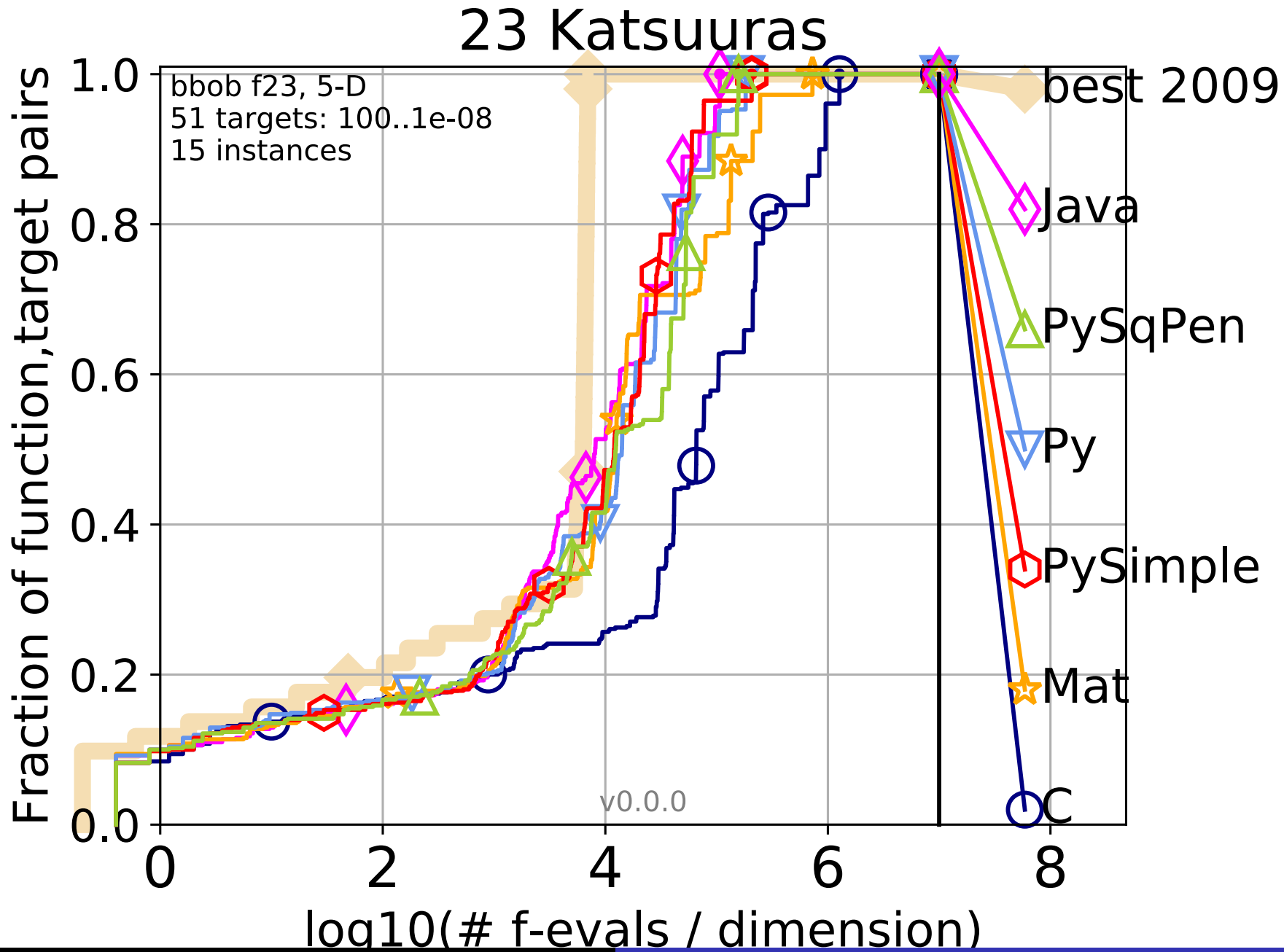
Results on selected functions in 5D



Results on selected functions in 5D



Results on selected functions in 5D



Meaning of values in table

- The numbers in tables show average runtime divided by the best value measured during BBOB-2009 competition
- The target error level was set to 10^{-5}
- The half difference between 10 and 90%- tile of bootstrapped run lengths was put in braces as dispersion measure

Meaning of values in table

- The numbers in tables show average runtime divided by the best value measured during BBOB-2009 competition
- The target error level was set to 10^{-5}
- The half difference between 10 and 90%- tile of bootstrapped run lengths was put in braces as dispersion measure

Meaning of values in table

- The numbers in tables show average runtime divided by the best value measured during BBOB-2009 competition
- The target error level was set to 10^{-5}
- The half difference between 10 and 90%- tile of bootstrapped run lengths was put in braces as dispersion measure

Results in 5D

Fun.	C	Java	Matlab	Python	Py. sim.	Py. sq. pen.	Py. sq
f1	88 (8)	**78** (13)	84 (6)	86 (10)	89 (7)	81 (8)	
f2	46 (5)	44 (3)	41 (2)	**27** (3)	43 (6)	28 (4)	
f3	379 (249)	**283** (246)	341 (395)	601 (254)	472 (702)	328 (315)	
f4	7045 (1e4)	6783 (7417)	9937 (2e4)	6026 (6540)	5732 (2212)	**3908** (2539)	
f5	140 (24)	351 (42)	47 (15)	115 (13)	97 (73)	**33** (9)	
f6	2.6 (0.3)	2.4 (0.5)	2.4 (0.2)	2.5 (0.2)	**2.3** (0.3)	2.4 (0.1)	
f7	13 (12)	7.2 (10)	9.4 (13)	**2.8** (2)	8.1 (4)	2.9 (5)	
f8	12 (5)	10 (2)	12 (4)	12 (15)	10 (2)	**9.1** (1)	
f9	12 (2)	12 (4)	12 (6)	14 (9)	13 (4)	**11** (4)	
f10	8.4 (4)	5.2 (0.4)	4.8 (0.7)	4.3 (2)	4.7 (0.3)	**3.1** (0.2)	
f11	3.2 (1)	2.9 (0.2)	2.7 (0.2)	2.2 (1)	2.8 (0.3)	**1.5** (0.2)	
f12	6.8 (5)	5.7 (4)	4.2 (3)	6.4 (5)	6.2 (2)	**3.4** (2)	
f13	3.4 (0.8)	3.6 (0.9)	3.3 (0.6)	**2.1** (0.7)	3.1 (0.4)	2.3 (1)	
f14	11 (0.8)	12 (2)	11 (2)	7 (0.8)	11 (1)	**6.4** (0.8)	
f15	**20** (29)	**20** (22)	44 (24)	23 (36)	25 (26)	35 (48)	
f16	24 (31)	5.7 (8)	11 (11)	12 (20)	**2.5** (3)	8.2 (7)	
f17	20 (13)	**7.2** (5)	13 (6)	12 (12)	16 (11)	12 (13)	
f18	140 (104)	133 (313)	245 (325)	53 (43)	109 (108)	**40** (66)	
f19	358 (448)	389 (306)	290 (144)	117 (34)	382 (272)	**86** (67)	
f20	106 (252)	61 (59)	102 (120)	49 (31)	49 (60)	**35** (13)	
f21	15 (20)	**7.1** (5)	12 (12)	14 (4)	13 (15)	18 (18)	
f22	39 (31)	47 (78)	56 (47)	59 (33)	**35** (28)	43 (136)	
f23	50 (91)	**5** (3)	17 (36)	6.6 (8)	6 (4)	7.7 (12)	
f24	∞	∞	**60** (60)	62 (44)	∞	∞	

Meaning of values in table

- Algorithms that use the same bound constraint handling were compared in pairs using the Wilcoxon rank-sum test along with the Bonferroni correction by the number of functions
- The star means that there is a statistically significant difference with p-value 0.05. The number k after the star shows the p-value was 10^{-k}

Meaning of values in table

- Algorithms that use the same bound constraint handling were compared in pairs using the Wilcoxon rank-sum test along with the Bonferroni correction by the number of functions
- The star means that there is a statistically significant difference with p-value 0.05. The number k after the star shows the p-value was 10^{-k}

Results in 5D

Fun.	C	Matlab	Python	Py. sq. pen.	Py. sq. pen. vs Matlab	Py. vs C
f1	88 (8)	84 (6)	86 (10)	81 (8)		
f2	46 (5)	41 (2)	**27** (3)	28 (4)	★4	★4
f3	379 (249)	341 (395)	601 (254)	328 (315)		
f4	7045 (1e4)	9937 (2e4)	6026 (6540)	**3908** (2539)		
f5	140 (24)	47 (15)	115 (13)	**33** (9)		★
f6	2.6 (0.3)	2.4 (0.2)	2.5 (0.2)	2.4 (0.1)		
f7	13 (12)	9.4 (13)	**2.8** (2)	2.9 (5)		
f8	12 (5)	12 (4)	12 (15)	**9.1** (1)	★	
f9	12 (2)	12 (6)	14 (9)	**11** (4)		
f10	8.4 (4)	4.8 (0.7)	4.3 (2)	**3.1** (0.2)	★4	★
f11	3.2 (1)	2.7 (0.2)	2.2 (1)	2.8 (0.3)	★4	★
f12	6.8 (5)	4.2 (3)	6.4 (5)	**3.4** (2)		
f13	3.4 (0.8)	3.3 (0.6)	**2.1** (0.7)	2.3 (1)	★2	★3
f14	11 (0.8)	11 (2)	7 (0.8)	**6.4** (0.8)	★4	★3
f15	**20** (29)	44 (24)	23 (36)	35 (48)		
f16	24 (31)	11 (11)	12 (20)	8.2 (7)		
f17	20 (13)	13 (6)	12 (12)	12 (13)		
f18	140 (104)	245 (325)	53 (43)	**40** (66)	★	
f19	358 (448)	290 (144)	117 (34)	**86** (67)		
f20	106 (252)	102 (120)	49 (31)	**35** (13)		
f21	15 (20)	12 (12)	14 (4)	18 (18)		
f22	39 (31)	56 (47)	59 (33)	43 (136)		
f23	50 (91)	17 (36)	6.6 (8)	7.7 (12)		★
f24	∞	**60** (60)	62 (44)	∞		

Tracing the differences

- Bound constraint handling
- Internal stopping conditions (sanity checks) – 8 in C, 11 in Python; stopTolFun, stopTolFunHist, stopTolX are different
- After setting Python like C, Python was interrupted on function 19
- Implementing different versions of the method – Python implements ActiveCMA version
- Different heuristics used to detect and escape from flat areas of the fitness
- Different values used in initialization of internal recombination weights

Tracing the differences

- Bound constraint handling
- Internal stopping conditions (sanity checks) – 8 in C, 11 in Python; stopTolFun, stopTolFunHist, stopTolX are different
- After setting Python like C, Python was interrupted on function 19
- Implementing different versions of the method – Python implements ActiveCMA version
- Different heuristics used to detect and escape from flat areas of the fitness
- Different values used in initialization of internal recombination weights

Tracing the differences

- Bound constraint handling
- Internal stopping conditions (sanity checks) – 8 in C, 11 in Python; stopTolFun, stopTolFunHist, stopTolX are different
- After setting Python like C, Python was interrupted on function 19
- Implementing different versions of the method – Python implements ActiveCMA version
- Different heuristics used to detect and escape from flat areas of the fitness
- Different values used in initialization of internal recombination weights

Tracing the differences

- Bound constraint handling
- Internal stopping conditions (sanity checks) – 8 in C, 11 in Python; stopTolFun, stopTolFunHist, stopTolX are different
- After setting Python like C, Python was interrupted on function 19
- Implementing different versions of the method – Python implements ActiveCMA version
- Different heuristics used to detect and escape from flat areas of the fitness
- Different values used in initialization of internal recombination weights

Tracing the differences

- Bound constraint handling
- Internal stopping conditions (sanity checks) – 8 in C, 11 in Python; stopTolFun, stopTolFunHist, stopTolX are different
- After setting Python like C, Python was interrupted on function 19
- Implementing different versions of the method – Python implements ActiveCMA version
- Different heuristics used to detect and escape from flat areas of the fitness
- Different values used in initialization of internal recombination weights

Tracing the differences

- Bound constraint handling
- Internal stopping conditions (sanity checks) – 8 in C, 11 in Python; stopTolFun, stopTolFunHist, stopTolX are different
- After setting Python like C, Python was interrupted on function 19
- Implementing different versions of the method – Python implements ActiveCMA version
- Different heuristics used to detect and escape from flat areas of the fitness
- Different values used in initialization of internal recombination weights

Article-implementation relation

- Which article describes the CMA-ES?
- Authors of the CMA-ES cited sequence of four articles when referring to CMA-ES
- Article-implementation relation is not clear:
  - there are no references in the C code
  - there are two references connected with additional improvements of the method in Python
  - there are five references in Matlab and five in Java, four of them are common for both implementations

Article-implementation relation

- Which article describes the CMA-ES?
- Authors of the CMA-ES cited sequence of four articles when referring to CMA-ES
- Article-implementation relation is not clear:
  - there are no references in the C code
  - there are two references connected with additional improvements of the method in Python
  - there are five references in Matlab and five in Java, four of them are common for both implementations

Article-implementation relation

- Which article describes the CMA-ES?
- Authors of the CMA-ES cited sequence of four articles when referring to CMA-ES
- Article-implementation relation is not clear:
  - there are no references in the C code
  - there are two references connected with additional improvements of the method in Python
  - there are five references in Matlab and five in Java, four of them are common for both implementations

Article-implementation relation

- Which article describes the CMA-ES?
- Authors of the CMA-ES cited sequence of four articles when referring to CMA-ES
- Article-implementation relation is not clear:
  - there are no references in the C code
  - there are two references connected with additional improvements of the method in Python
  - there are five references in Matlab and five in Java, four of them are common for both implementations

Article-implementation relation

- Which article describes the CMA-ES?
- Authors of the CMA-ES cited sequence of four articles when referring to CMA-ES
- Article-implementation relation is not clear:
  - there are no references in the C code
  - there are two references connected with additional improvements of the method in Python
  - there are five references in Matlab and five in Java, four of them are common for both implementations

Best practice recommendations

- Publishers should require the availability of the source code for all new optimization methods
- Authors should reveal how all parameters were set up, not only in the proposed method but also in methods used for comparison
- The code used for running experiments should be available
- Authors should use the most up-to-date trusted implementation of the state-of-the-art and reveal its origin, name, and version
- Authors of implementations should define article-implementation relation
- Authors of articles and authors of implementations should identify the method used for constraint handling

Best practice recommendations

- Publishers should require the availability of the source code for all new optimization methods
- Authors should reveal how all parameters were set up, not only in the proposed method but also in methods used for comparison
- The code used for running experiments should be available
- Authors should use the most up-to-date trusted implementation of the state-of-the-art and reveal its origin, name, and version
- Authors of implementations should define article-implementation relation
- Authors of articles and authors of implementations should identify the method used for constraint handling

Best practice recommendations

- Publishers should require the availability of the source code for all new optimization methods
- Authors should reveal how all parameters were set up, not only in the proposed method but also in methods used for comparison
- The code used for running experiments should be available
- Authors should use the most up-to-date trusted implementation of the state-of-the-art and reveal its origin, name, and version
- Authors of implementations should define article-implementation relation
- Authors of articles and authors of implementations should identify the method used for constraint handling

Best practice recommendations

- Publishers should require the availability of the source code for all new optimization methods
- Authors should reveal how all parameters were set up, not only in the proposed method but also in methods used for comparison
- The code used for running experiments should be available
- Authors should use the most up-to-date trusted implementation of the state-of-the-art and reveal its origin, name, and version
- Authors of implementations should define article-implementation relation
- Authors of articles and authors of implementations should identify the method used for constraint handling

Best practice recommendations

- Publishers should require the availability of the source code for all new optimization methods
- Authors should reveal how all parameters were set up, not only in the proposed method but also in methods used for comparison
- The code used for running experiments should be available
- Authors should use the most up-to-date trusted implementation of the state-of-the-art and reveal its origin, name, and version
- Authors of implementations should define article-implementation relation
- Authors of articles and authors of implementations should identify the method used for constraint handling

Best practice recommendations

- Publishers should require the availability of the source code for all new optimization methods
- Authors should reveal how all parameters were set up, not only in the proposed method but also in methods used for comparison
- The code used for running experiments should be available
- Authors should use the most up-to-date trusted implementation of the state-of-the-art and reveal its origin, name, and version
- Authors of implementations should define article-implementation relation
- Authors of articles and authors of implementations should identify the method used for constraint handling

Conclusions

- The choice of a particular implementation of even a popular and standard algorithm may have a substantial impact on the results obtained in research studies or applications
- Many articles do not provide information about implementations used in experiments, which puts in question the utility of their findings
- The sources of discrepancies are frequently hidden in the auxiliary code
- The difference in the outcome of implementations also stems from implementing different versions of the algorithm

Conclusions

- The choice of a particular implementation of even a popular and standard algorithm may have a substantial impact on the results obtained in research studies or applications
- Many articles do not provide information about implementations used in experiments, which puts in question the utility of their findings
- The sources of discrepancies are frequently hidden in the auxiliary code
- The difference in the outcome of implementations also stems from implementing different versions of the algorithm

Conclusions

- The choice of a particular implementation of even a popular and standard algorithm may have a substantial impact on the results obtained in research studies or applications
- Many articles do not provide information about implementations used in experiments, which puts in question the utility of their findings
- The sources of discrepancies are frequently hidden in the auxiliary code
- The difference in the outcome of implementations also stems from implementing different versions of the algorithm

Conclusions

- The choice of a particular implementation of even a popular and standard algorithm may have a substantial impact on the results obtained in research studies or applications
- Many articles do not provide information about implementations used in experiments, which puts in question the utility of their findings
- The sources of discrepancies are frequently hidden in the auxiliary code
- The difference in the outcome of implementations also stems from implementing different versions of the algorithm

The end

Thank you for watching!