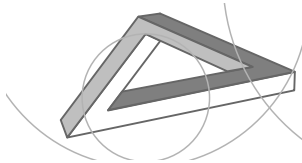
Bounding the population size of IPOP-CMA-ES on the Noiseless BBOB Testbed + Testing impact of tuning + Expensive optimization scenarios

Tianjun Liao and Thomas Stützle

IRIDIA, CoDE, Université Libre de Bruxelles (ULB)

fnrs LA LIBERTÉ DE CHERCHER



IRIDIA
Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle

ULB UNIVERSITÉ LIBRE DE BRUXELLES

Motivation

bounding population size

- IPOP-CMA-ES increases population size exponentially
- Question: are very large population useful?
- particular motivation:
  - Chen et al., 2012: large pop. size can be unhelpful in EAs
  - Wessing et al., 2011: when parameter tuning actually is parameter control

$\mapsto$ examine bounds on maximum population size

Motivation

bounding population size

- IPOP-CMA-ES increases population size exponentially
- Question: are very large population useful?
- particular motivation:
  - Chen et al., 2012: large pop. size can be unhelpful in EAs
  - Wessing et al., 2011: when parameter tuning actually is parameter control

$\mapsto$ examine bounds on maximum population size

tuning, expensive

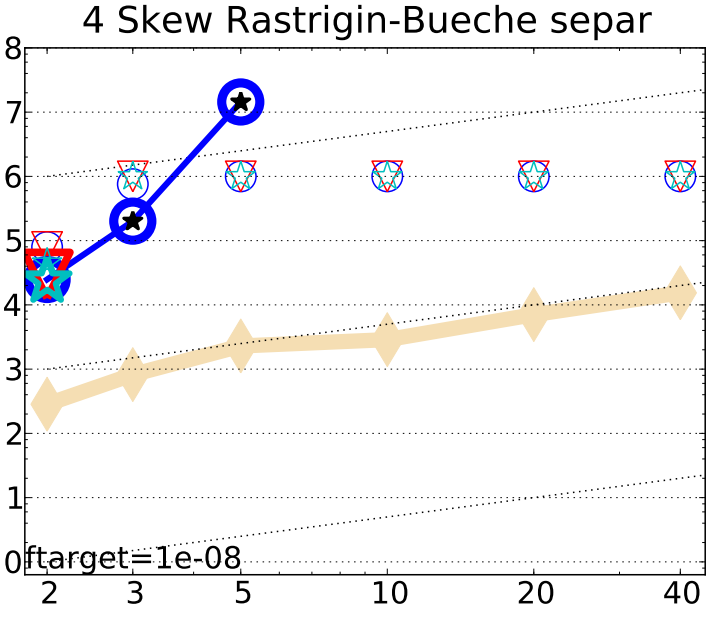
- tuning incurs only very light effort (in our case here)
- expensive functions: different look at data

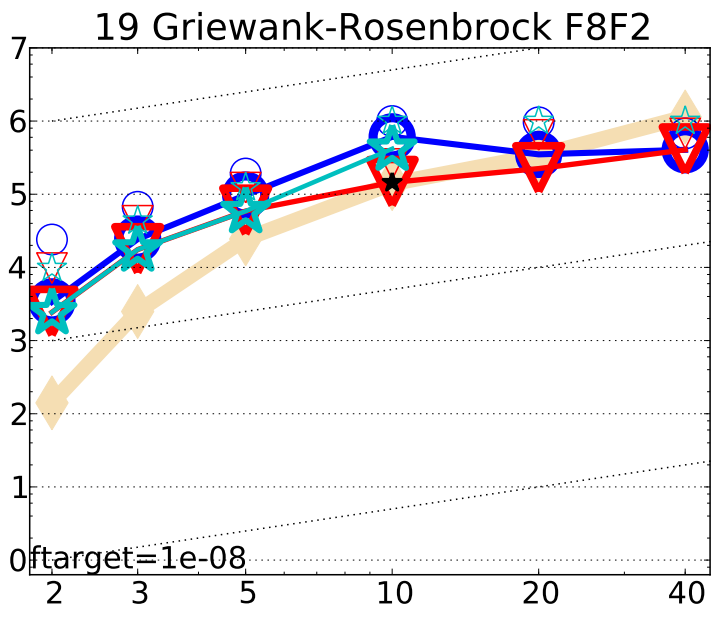
Bounded population size

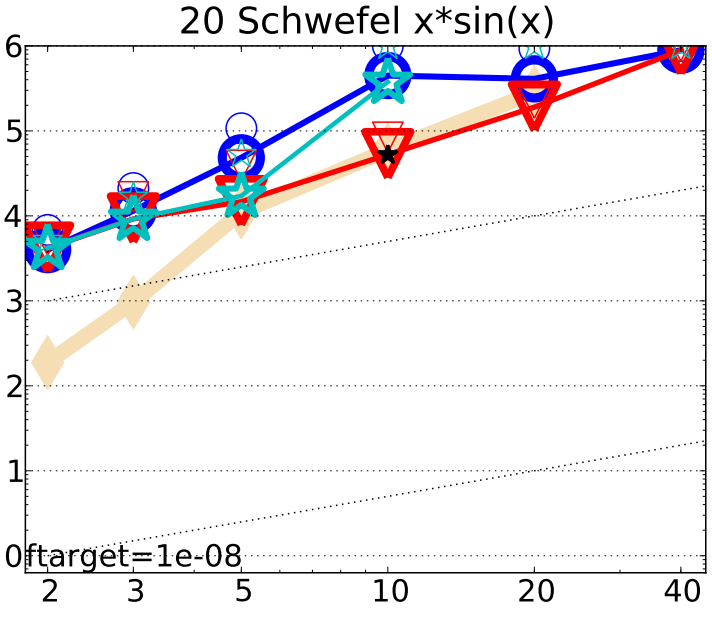
Experimental setup

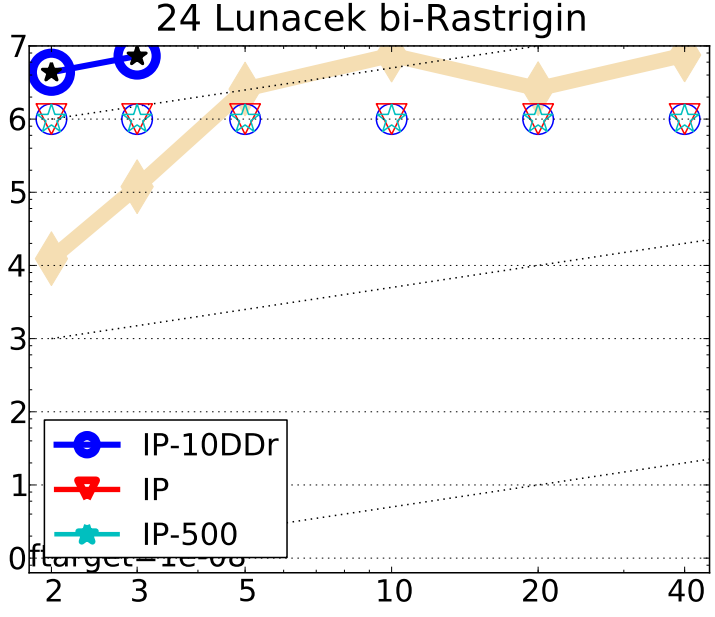
- IPOP-CMA-ES default version (C version, 10/16/10), modified to clamp bounds and limit maximum pop. size
- Recall pop-size settings of IPOP-CMA-ES
  - initial: $\lambda = 4 + \lfloor 3 \ln(D) \rfloor$
  - restart: $\lambda = 2 \cdot \lambda \mapsto \lambda_i = 2^i \cdot \lambda_0$
- poplation size bounds (limited experience, ad hoc)
  - fixed maximum to 500
  - dimension-dependent maximum of $10D^2$
- once maximum is reached, restart scheme at initial value

Significant differences ERT

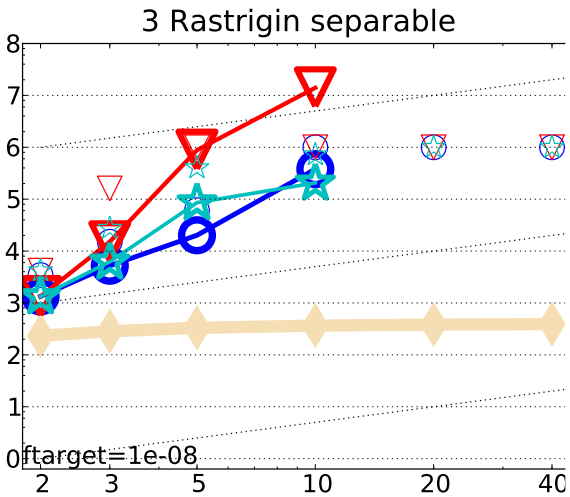


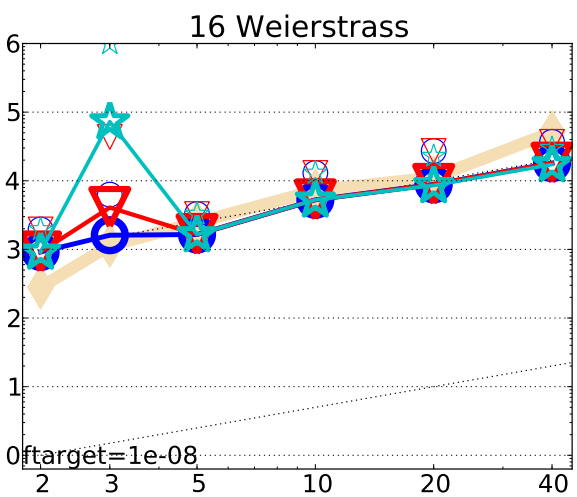


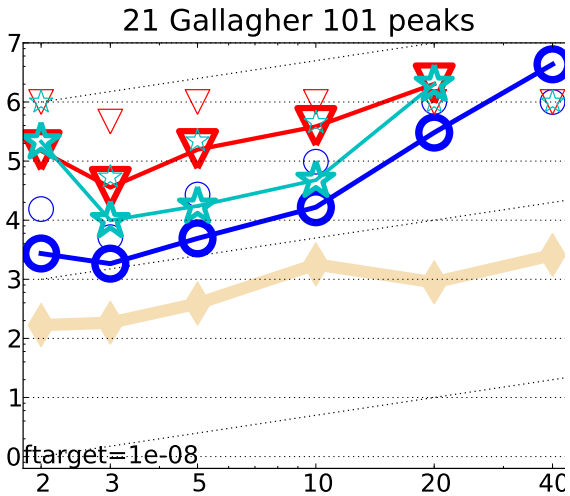


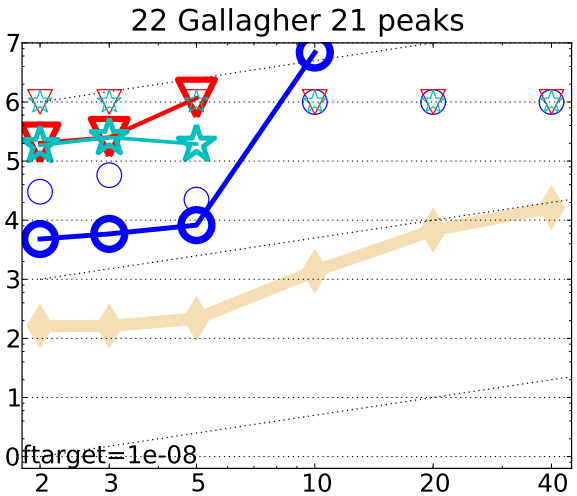


“Visible” differences

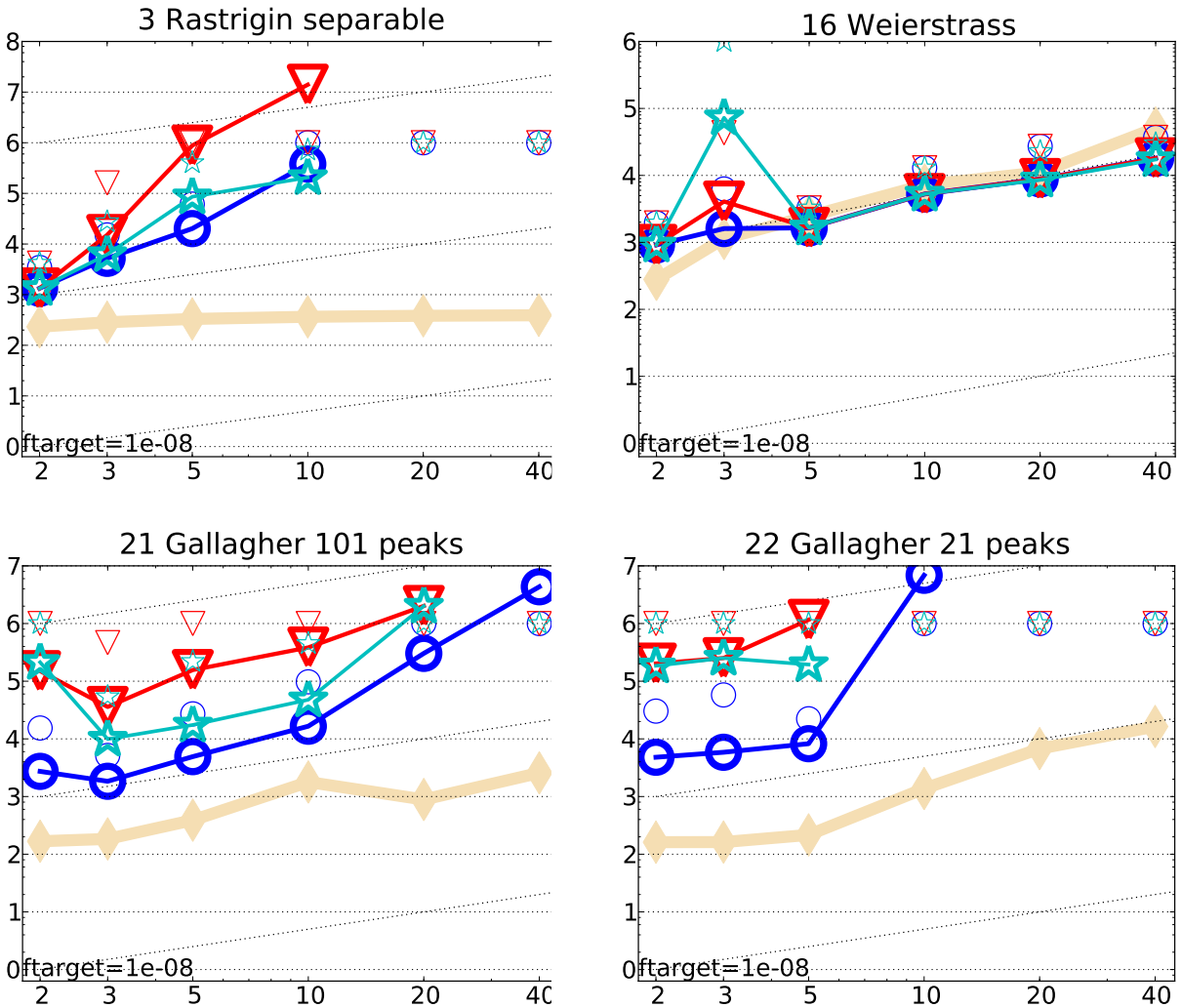








“Visible” differences



On many functions population-size bounds too loose to make an effect (apparently $f1, f2, f4 - f15, f17, f18$)

Tuning IPOP-CMA-ES

Tuning setup

- use $D = 10$ functions from SOCO special issue
- removed Sphere, Rosenbrock, Rastrigin, Schwefel 1.2 from SOCO set *error :-(Schaffer, f17*
- default version of `irace`
- tuning budget: 5000 runs of IPOP-CMA-ES each of $100D = 1000$ function evaluations
- short runs because same tuned version is used for expensive scenario
- measured cost function: evaluation function value at end of trial

Tuning IPOP-CMA-ES

Tuning setup

- use $D = 10$ functions from SOCO special issue
- removed Sphere, Rosenbrock, Rastrigin, Schwefel 1.2 from SOCO set *error :-(Schaffer, f17*
- default version of `irace`
- tuning budget: 5000 runs of IPOP-CMA-ES each of $100D = 1000$ function evaluations
- short runs because same tuned version is used for expensive scenario
- measured cost function: evaluation function value at end of trial

we are aware: wrong target for BBOB, not many parameters tuned, maybe wrong algorithm overall, etc.

Tuning IPOP-CMA-ES

Parameter	Internal parameter		default	tuned
a	Init pop size: λ_0	$= 4 + \lfloor a \ln(D) \rfloor$	3	2.675
b	Parent size: μ	$= \lfloor \lambda / b \rfloor$	2	1.351
c	Init step size: σ_0	$= c \cdot (B - A)$	0.5	0.102
d	IPOP factor: *ipop*	$= d$	2	2.88
e	*stopTolFun*	$= 10^e$	-12	-8.607
f	*stopTolFunHist*	$= 10^f$	-20	-14.77
g	*stopTolX*	$= 10^g$	-12	-9.529

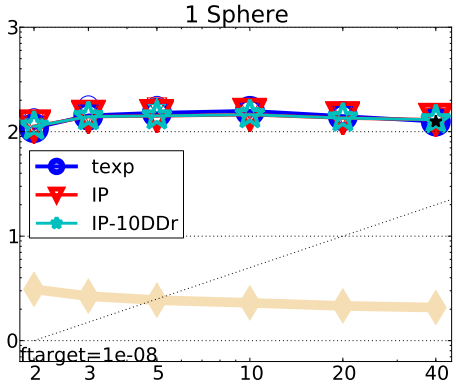
Tuning IPOP-CMA-ES

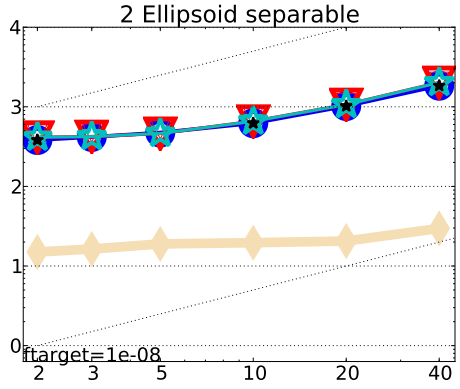
Parameter	Internal parameter		default	tuned
a	Init pop size: λ_0	$= 4 + \lfloor a \ln(D) \rfloor$	3	2.675
b	Parent size: μ	$= \lfloor \lambda / b \rfloor$	2	1.351
c	Init step size: σ_0	$= c \cdot (B - A)$	0.5	0.102
d	IPOP factor: *ipop*	$= d$	2	2.88
e	*stopTolFun*	$= 10^e$	-12	-8.607
f	*stopTolFunHist*	$= 10^f$	-20	-14.77
g	*stopTolX*	$= 10^g$	-12	-9.529

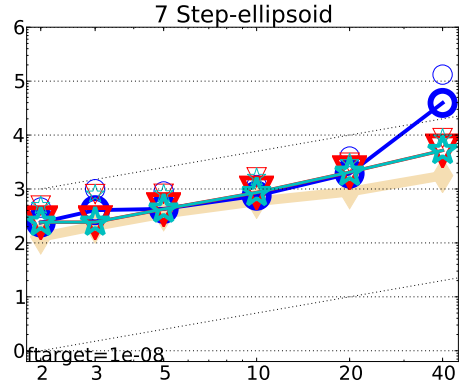
Following comparison

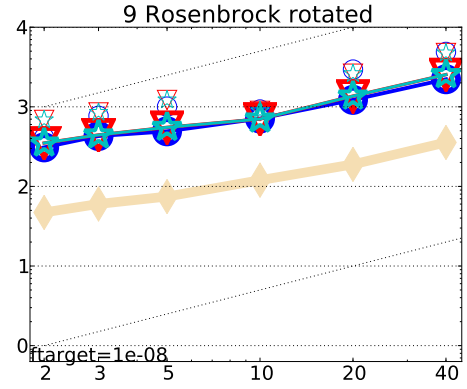
- default IPOP-CMA-ES
- IPOP-CMA-ES with pop. size bound $10D^2$
- IPOP-CMA-ES with pop. size bound $10D^2$, tuned

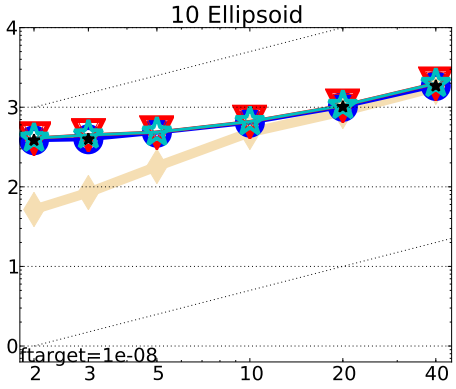
Significant differences ERT

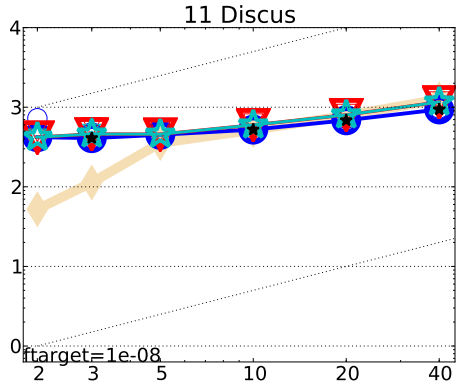


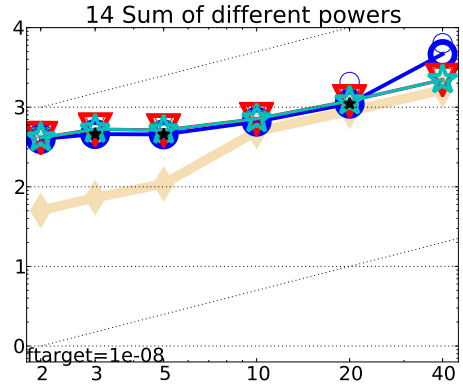


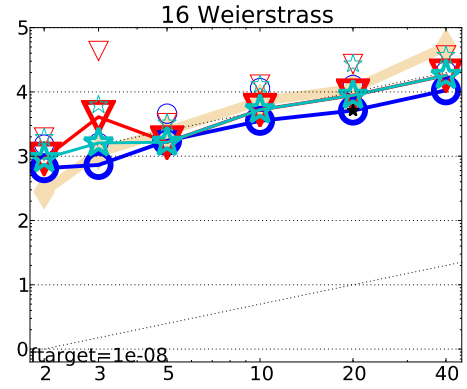


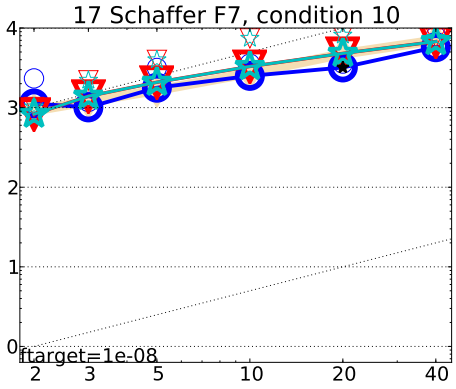


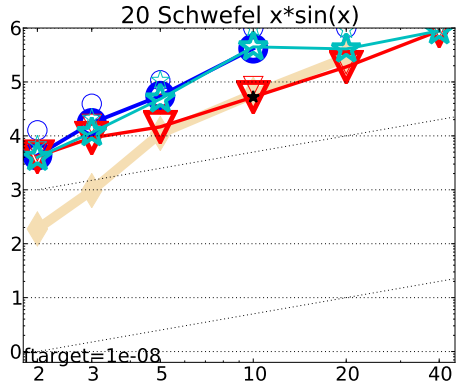


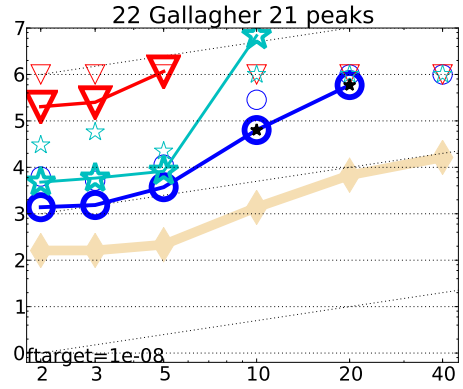


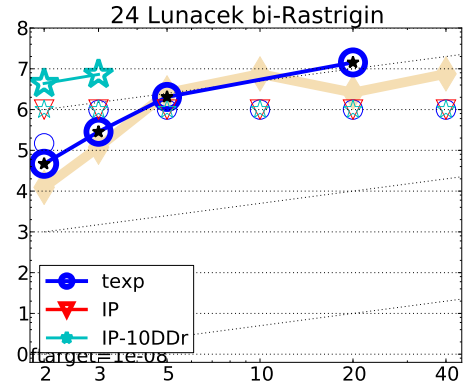




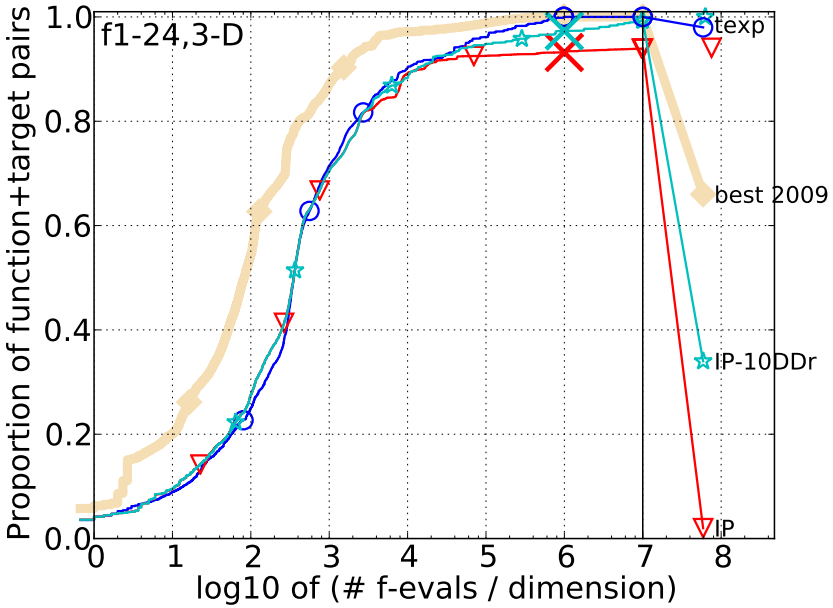


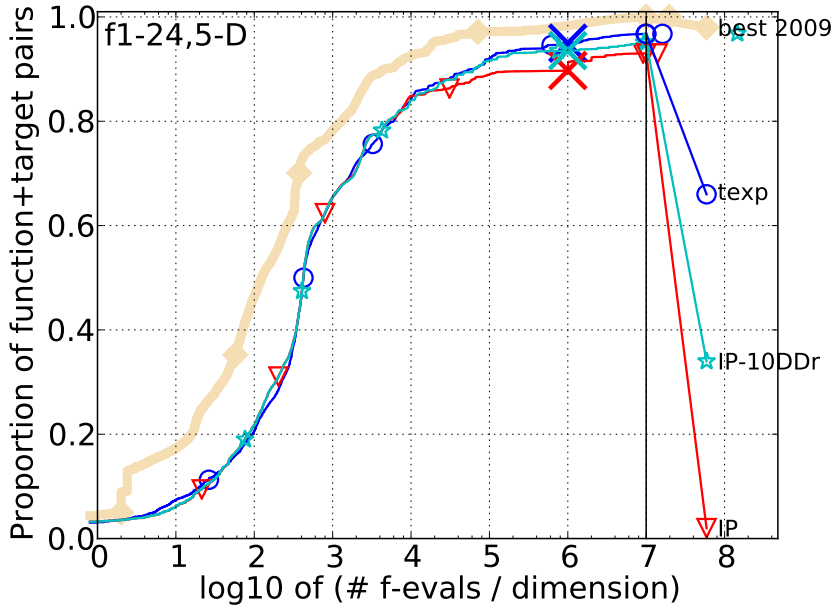


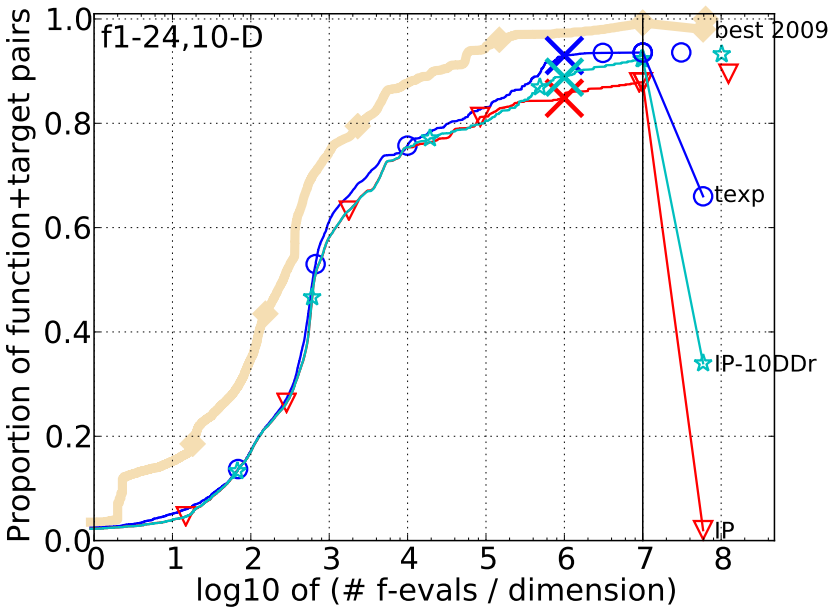


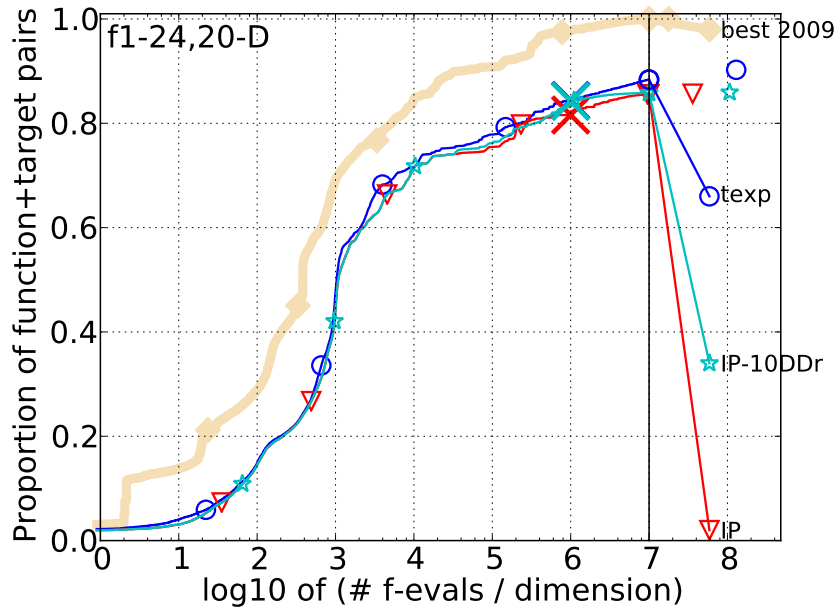


General scenario









Expensive scenario

no changes, same algorithm (tuned IPOP-CMA-ES with maximum population size bound (here not effective)

Expensive scenario

no changes, same algorithm (tuned IPOP-CMA-ES with maximum population size bound (here not effective)

just a different look at the same data

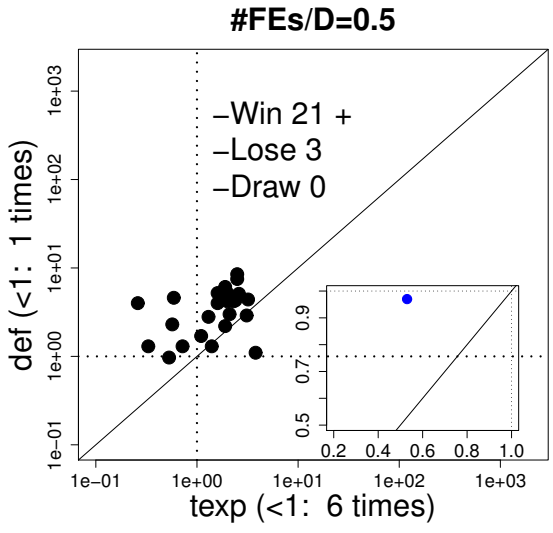
Expensive scenario

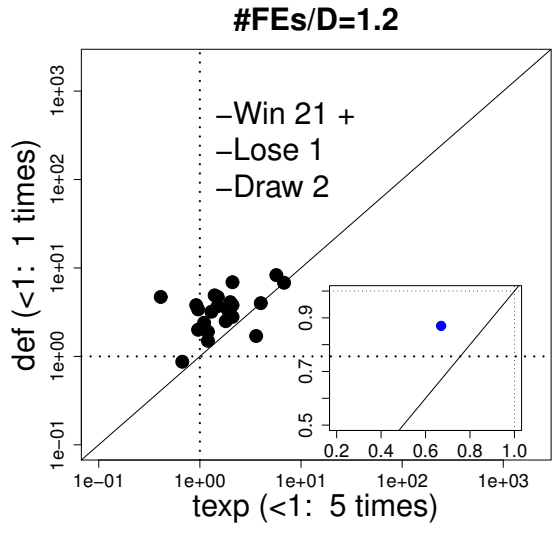
no changes, same algorithm (tuned IPOP-CMA-ES with maximum population size bound (here not effective)

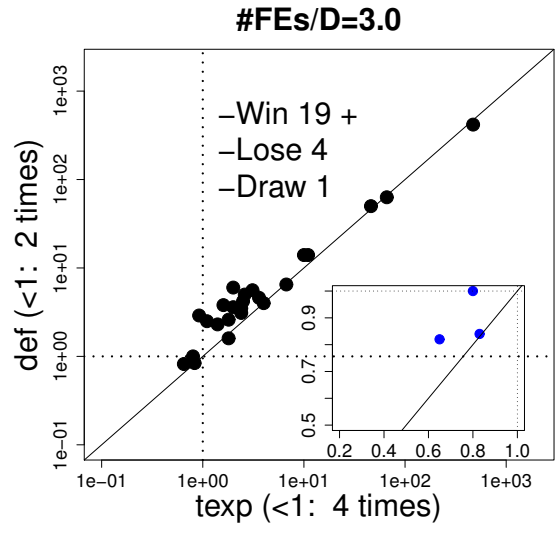
just a different look at the same data

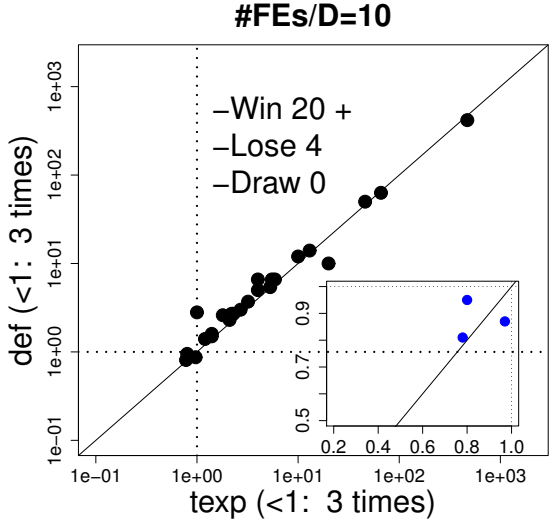
Motivation: direct search may not be so bad after all . . .

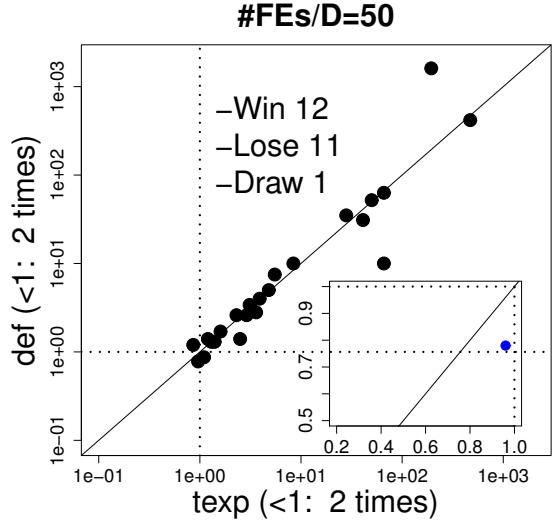
Expensive scenario, comparison to default IPOP-CMA-ES

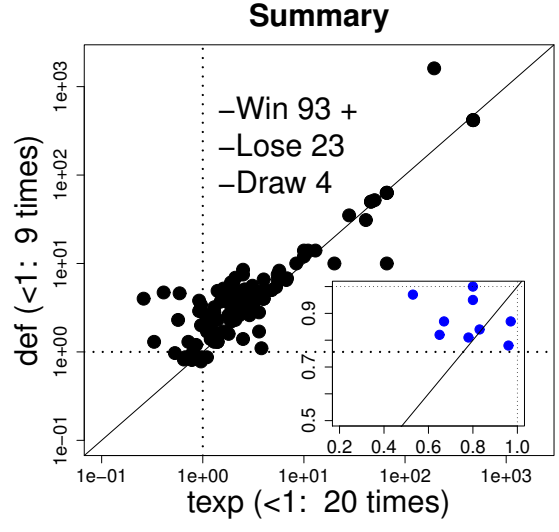




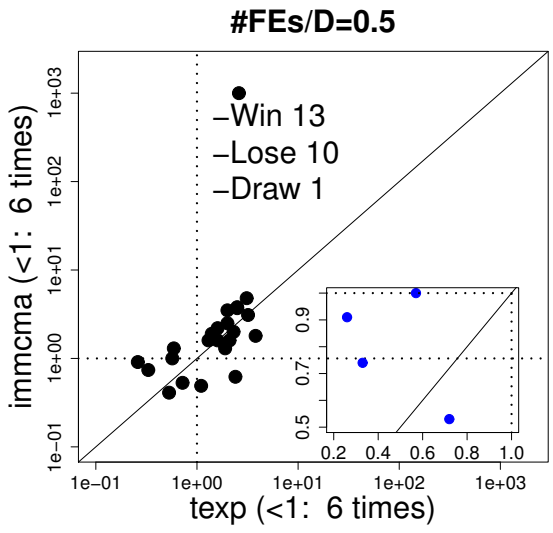


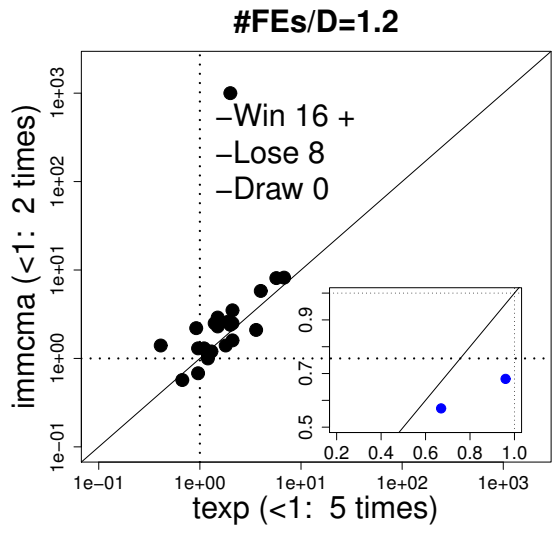


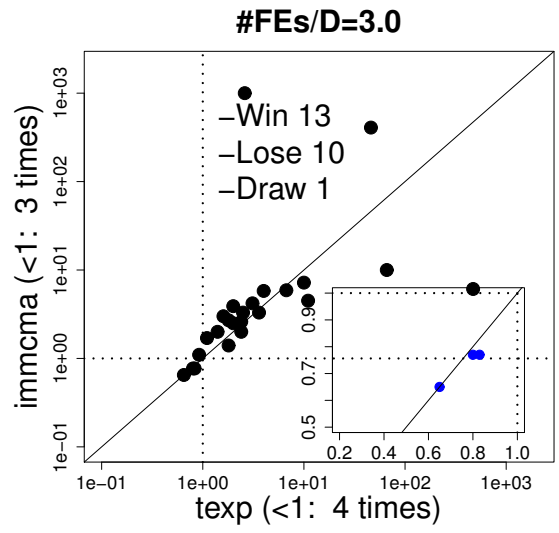


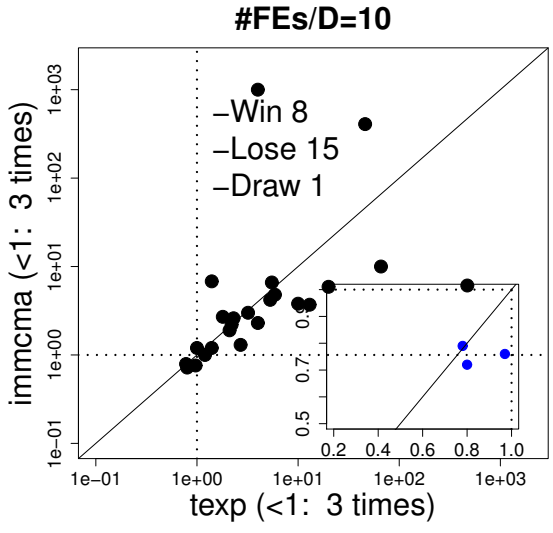


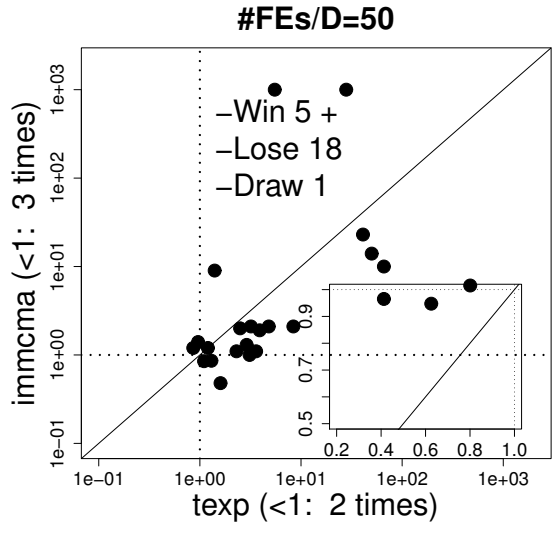
Expensive scenario, comparison to ImmCMA-ES

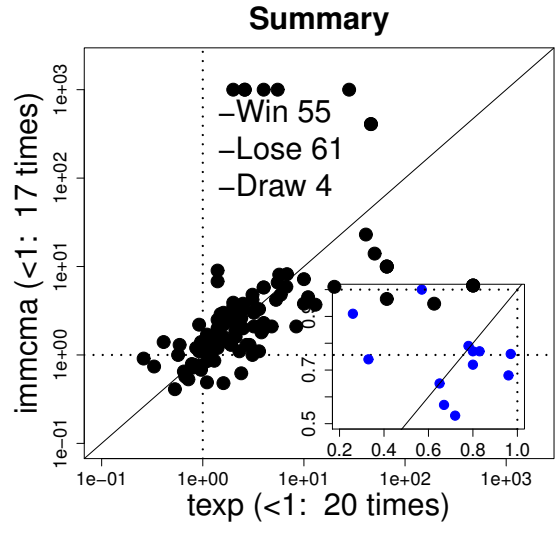




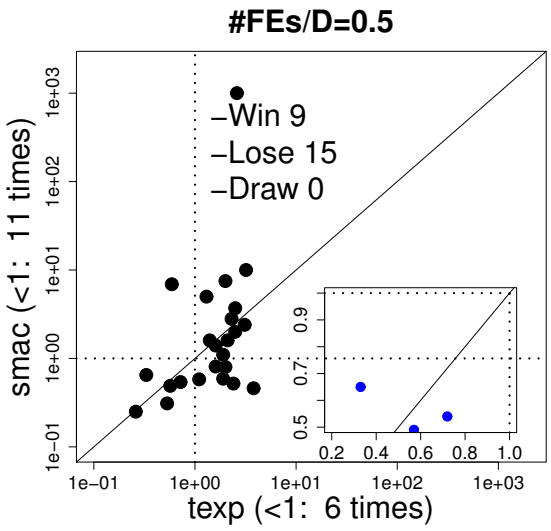


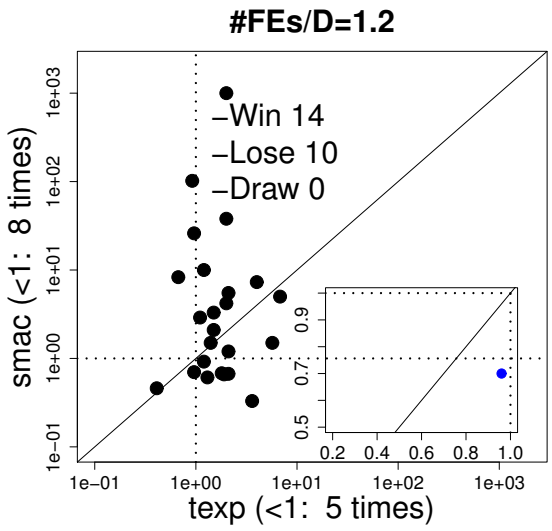


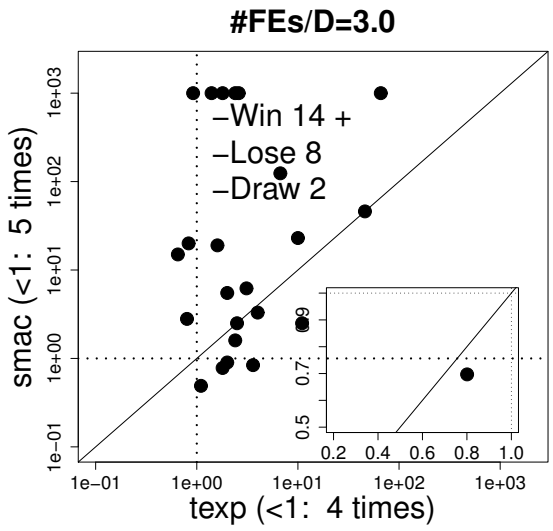


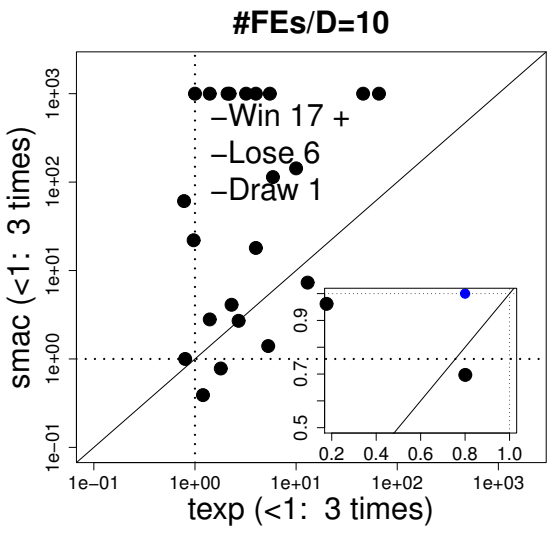


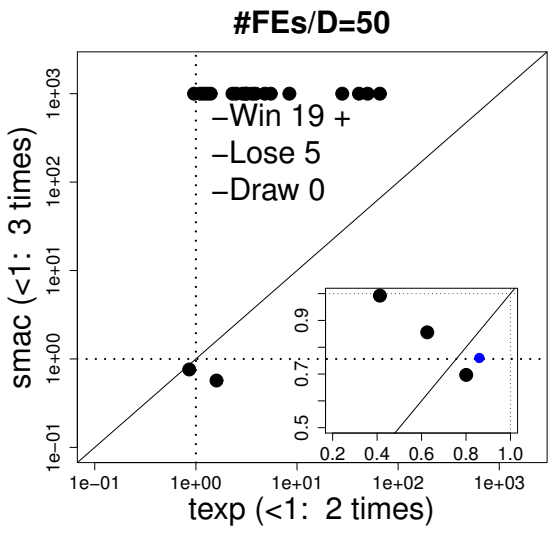
Expensive scenario, comparison to SMAC-BBOB

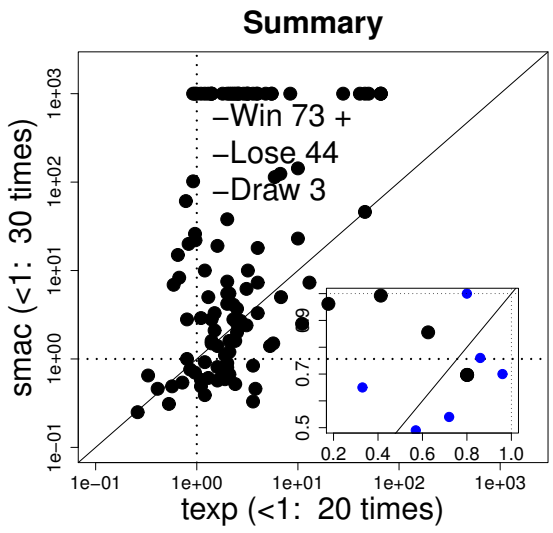












Conclusions

- limiting max. population size of IPOP-CMA-ES can help
- tuning can further improve performance
  - main differences on weakly structured multi-modal functions (also other benchmarks such as CEC'05 / '13)
- IPOP-CMA-ES anyway rather robust