Bio-inspired Continuous Optimization: The Coming of Age

Anne Auger Nikolaus Hansen Nikolas Mauny
Raymond Ros Marc Schoenauer

TAO Team, INRIA Futurs, FRANCE
http://tao.lri.fr
First.Last@inria.fr

CEC 2007, Singapore, September 27., 2007

Problem Statement
Continuous Domain Search/Optimization

The problem

- Minimize a fitness function (*objective* function, *loss* function) in continuous domain

$$f : \mathcal{S} \subseteq \mathbb{R}^n \to \mathbb{R},$$

- in the Black Box scenario (direct search)

Hypotheses

- domain specific knowledge only used within the black box
- gradients are not available

Problem Statement
Continuous Domain Search/Optimization

The problem

- Minimize a fitness function (*objective* function, *loss* function) in continuous domain

$$f : \mathcal{S} \subseteq \mathbb{R}^n \rightarrow \mathbb{R},$$

- in the Black Box scenario (direct search)

Typical Examples

- shape optimization (e.g. using CFD) — curve fitting, airfoils
- model calibration — biological, physical
- parameter identification — controller, plants, images

Optimization techniques

Numerical methods

- Applied Mathematicians — Long history
- Heavily rely on theoretical convergence proofs

but

- Requires regularity — numerical gradient?
- Numerical pitfalls

Bio-inspired algorithms

- Computer Scientists — mostly from AI field
- Recent trendy methods — but divided!

but

- Computationally heavy
- No convergence proof — well, almost – see later

This talk

Goal

- Empirical comparison
- on some artificial testbed
- illustrating typical difficulties of continuous optimization
- between
  - some bio-inspired algorithms
  - and some (one!) deterministic optimization method(s)
- in the back-box scenario

without specific intensive parameter tuning

Bio-inspired Optimization Algorithms

Darwinian Artificial Evolution

Repeat (Parent selection – Variation – Survival selection)

Preselection: Results of CEC'05 Challenge

- Particle Swarm Optimization — Eberhart & Kennedy, 95
  - Perturb particle velocity $\rightarrow$ *best* and *local best*
  - Update *best* and *local best*
- Differential Evolution — Rainer and Storn, 95
  - Add difference vector(s)
  - Uniform crossover — with proba. $1 - CR$
  - Keep best of parent and offspring
- Covariance Matrix Adaptation-ES — Hansen & Ostermeier, 96
  - Gaussian mutation — + Update mutation parameters
  - Keep $\frac{\lambda}{2}$ best of λ offspring

BFGS

Gradient-based methods

$$\begin{cases} x_{t+1} = x_t - \rho_t d_t \\ \rho_t = Argmin_\rho\{\mathcal{F}(x_t - \rho d_t)\} \end{cases}$$ Line search

Choice of d_t, the descent direction?

BFGS: a Quasi-Newton method

- Maintain an approximation $\hat{H}_t$ of the Hessian of f
- Solve for d_t

$$\hat{H}_t d_t = \nabla f(x_t)$$

- Compute x_{t+1} and update $\hat{H}_t \rightarrow \hat{H}_{t+1}$
- Converges if quadratic approximation of $\mathcal{F}$ holds

around the optimum

- Reliable and robust

on quadratic functions!

1. Continuous optimization and stochastic search

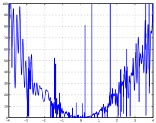
2. Problem difficulties
   - Ruggedness
   - Ill-Conditionning
   - Non-separability

3. Implementations and parameter settings

4. Experiments and results

5. Conclusion

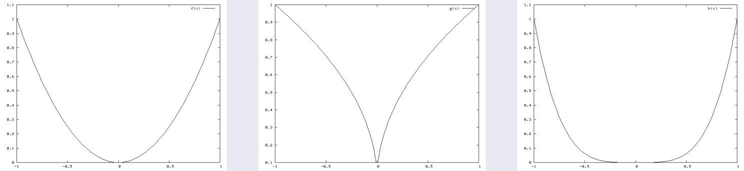
What makes a problem hard?



- Non-convexity
  invalidates most of deterministic theory
  - Ruggedness
    non-smooth, discontinuous, noisy
  - Multimodality presence of local optima
- Dimensionality line search is 'trivial'
  The magnifiscence of high dimensionality
- Ill-conditioning
- Non-separability

Monotonous transformation invariance

Monotonous transformations



$\sum x_i^2$ | $\sqrt{\sqrt{\sum x_i^2}}$ | $\left(\sum x_i^2\right)^2$

Invariance

- Comparison-based algorithms — PSO, DE, CMA-ES, ...
  are invariant w.r.t. monotonous transformations
- Gradient-based methods — BFGS, ...
  are not

Multimodality

Bio-inspired algorithms

- are global search algorithms
- but performance on multi-modal problems depends on population size

BFGS

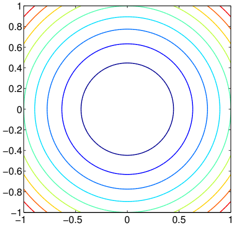
- has no population :-)
- but starting point is crucial — on multimodal functions
- Replace population by multiple restarts
  - from uniformly distributed points
  - or from the perturbed final point of previous trial

the Hessian is anyway reset to I_n

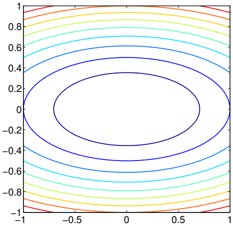
Ill-Conditionning

- The Condition Number (CN) of a positive-definite matrix $\boldsymbol{H}$ is the ratio of its largest and smallest eigenvalues
- If f is quadratic, $f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x}$), the CN of f is that of its Hessian $\boldsymbol{H}$
- More generally, the CN of f is that of its Hessian wherever it is defined.

Graphically, ill-conditioned means “squeezed” lines of equal function value



Increased $\longrightarrow$ condition number



Issue: The gradient does not point toward the minimum . . .

A priori discussion

Bio-inspired algorithms

- PSO and DE: population can point toward the minimum
- CMA-ES: covariance matrix can take longer to learn

BFGS

- Numerical gradient can raise numerical problems
- Hessian matrix can take longer to learn

Separability

Definition (Separable Problem)

A function f is separable if

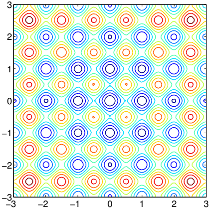
$$\arg\min_{(x_1,\ldots,x_n)} f(x_1,\ldots,x_n) = \left(\arg\min_{x_1} f(x_1,\ldots),\ldots,\arg\min_{x_n} f(\ldots,x_n)\right)$$

solve n independent 1D optimization problems

Example: Additively decomposable functions

$$f(x_1,\ldots,x_n) = \sum_{i=1}^{n} f_i(x_i)$$

e.g. Rastrigin function

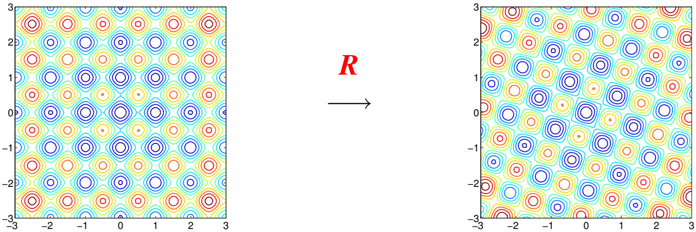


Designing Non-Separable Problems

Rotating the coordinate system

- $f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$ separable
- $f : \boldsymbol{x} \mapsto f(\boldsymbol{R}\boldsymbol{x})$ non-separable — $\boldsymbol{R}$ rotation matrix

Hansen, Ostermeier, & Gawelczyk, 95; Salomon, 96



Rotational invariance

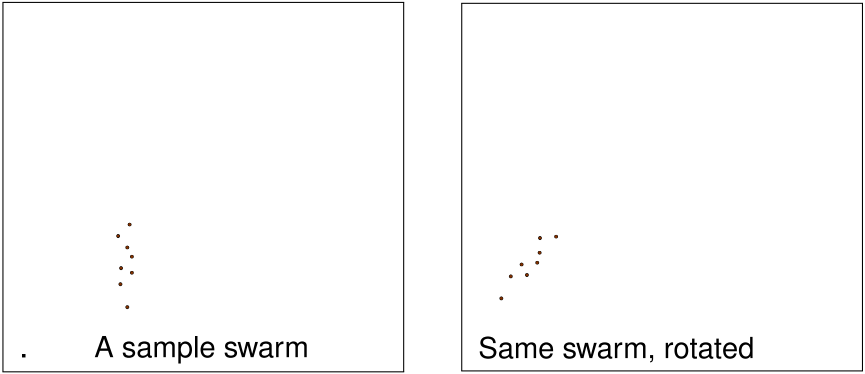
Bio-inspired algorithms

- PSO: is not rotational invariant — see next slide
- DE: Crossover is not rotational invariant
  Rotational invariance iff $CR = 1$
- CMA-ES: is rotational invariant

BFGS

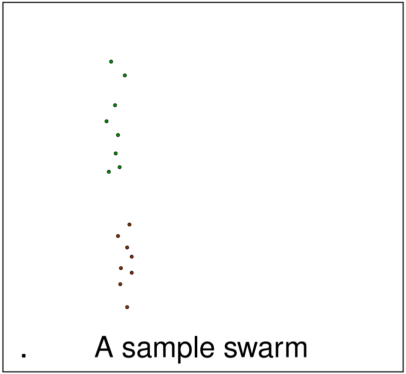
- Numerical gradient can raise numerical problems
  Added to ill-conditionning effects

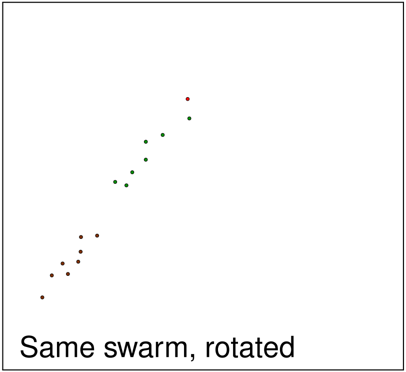
PSO and rotational invariance



$$V_i^j(t+1) = V_i^j(t) + \underbrace{c_1\,\mathcal{U}_i^j(0,1)(p_i^j - x_i^j(t))}_{\text{approach the "previous" best}} + \underbrace{c_2\,\tilde{\mathcal{U}}_i^j(0,1)(g_i^j - x_i^j(t))}_{\text{approach the "global" best}}$$

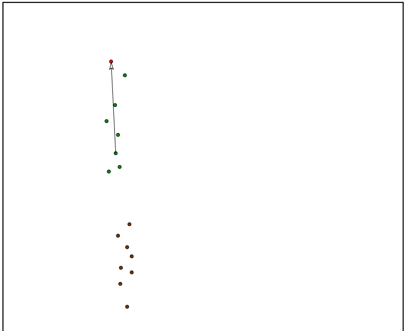
PSO and rotational invariance


. A sample swarm

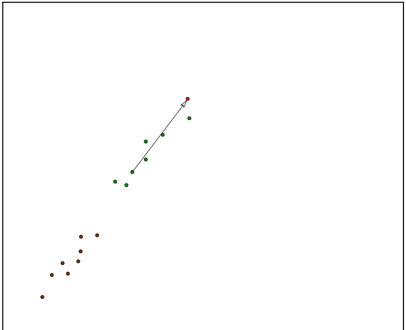

Same swarm, rotated

$$V_i^j(t+1) = V_i^j(t) + \underbrace{c_1\, \mathcal{U}_i^j(0,1)(p_i^j - x_i^j(t))}_{\text{approach the "previous" best}} + \underbrace{c_2\, \tilde{\mathcal{U}}_i^j(0,1)(g_i^j - x_i^j(t))}_{\text{approach the "global" best}}$$

PSO and rotational invariance



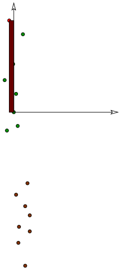
. A sample swarm



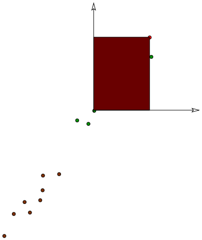
Same swarm, rotated

$$V_i^j(t+1) = V_i^j(t) + \underbrace{c_1 \mathcal{U}_i^j(0,1)(p_i^j - x_i^j(t))}_{\text{approach the "previous" best}} + \underbrace{c_2 \tilde{\mathcal{U}}_i^j(0,1)(g_i^j - x_i^j(t))}_{\text{approach the "global" best}}$$

PSO and rotational invariance



. A sample swarm



Same swarm, rotated

$$V_i^j(t+1) = V_i^j(t) + \underbrace{c_1\, \mathcal{U}_i^j(0,1)(p_i^j - x_i^j(t))}_{\text{approach the "previous" best}} + \underbrace{c_2\, \tilde{\mathcal{U}}_i^j(0,1)(g_i^j - x_i^j(t))}_{\text{approach the "global" best}}$$

The algorithms

'Default' implementations

- DE: Matlab code from
  http://www.icsi.berkeley.edu/~storn/code.html
  Not really giving default parameters
- PSO: Std PSO 2006, C code from
  http://www.particleswarm.info/Standard_PSO_2006.c
  Remark: C code $\neq$ Matlab code here
- CMA-ES: Matlab code from
  http://www.bionik.tu-berlin.de/user/niko/
- BFGS: Matlab built-in implementation — widely blindely used
  using numerical gradient
  + multiple restarts — local or global

The problem with DE

Control parameters

- NP, F, CR, Stopping Criterion . . .
- and strategy to generate difference vector
  - Perturb random or best
  - Number of difference vectors — 1 or 2
  - Slightly mutate perturbation
  - All of the above :-)

from public Matlab code

1. DE/rand/1 — the basic algorithm
2. DE/local-to-best/1
3. DE/best/1 with jitter
4. DE/rand/1 with per-vector-dither — $F = rand(F, 1)$
5. DE/rand/1 with global-dither *
6. either-or-algorithm

DE tuning

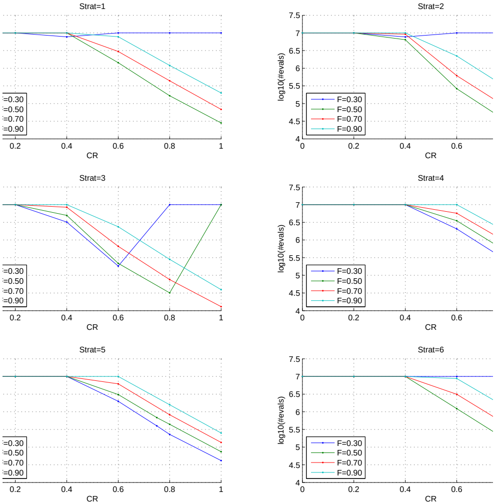
Experimental conditions

- Rotated ellipsoid, cond. number = 10^4
- Dimension $= 10$
- Stop when $f_{elli} < 10^{-6}$

Design of Experiments

- 6 variants
- $F = \{0.3, 0.5, 0.7, 0.9\}$
- $CR = \{0.3, 0.5, 0.7, 0.9\}$
- $NP = \{3, 5, 10\} \times dim$
- 3 runs per setting :-(

864 runs



Poor DOE, yet totally unrealistic for RWA

DE Experimental Setting and Discussion

Couldn't decide among 2 variants:

DE2

- Strategy 2
- $F = 0.8$
- $CR = 1.0$
- $NP = 10 \times dim$

DE5

- Strategy 5
- $F = 0.4$
- $CR = 1.0$
- $NP = 10 \times dim$

Discussion

- DE2 very close to 'recommended' parameters
- Except $CR = 0.9$, but
- Rotational invariance iff $CR = 1$ $\equiv$ no crossover

The parameters

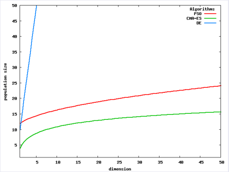
Common Parameters

- Default parameters — except for DE
- MaxEval = 10^7
- Fitness tolerance = 10^{-9}
- Domain: $[-20, 80]^d$ — Optimum not at the center
- 21 runs in each case — except BFGS when little success

Population size

Standard values: for $n = 10, 20, 40$

- PSO: $10 + floor(2\sqrt{n})$ — $16, 18, 22$
- CMA-ES: $4 + floor(3 \ln n)$ — $10, 12, 15$
- DE: $10 * n$ — $100, 200, 400$



To be increased for multi-modal functions — e.g. Rastrigin

The testbed

Test functions

- Ellipsoid function — quadratic
  both separable and rotated
  for different condition number
- Rosenbrock function — almost unimodal
  both original (non-separable) and rotated
  for different condition number
- Rastrigin function — highly multi-modal
  both separable and rotated
- DiffPow — convex, but 'flat'
- $\sqrt{\sqrt{\text{Ellipsoid}}}$ and $\sqrt{\sqrt{\text{DiffPow}}}$ — unimodal, but non-convex

Comparisons
from CEC 2005 challenge on continuous optimization

Goals
- Find the best possible fitness value
- At minimal cost — Number of function evaluations

Statistical Measures
- Must take into account both precision and cost
- Usual averages and standard deviations irrelevant to assess precision — of fitness values
- Issue: need to impose — for obvious practical reasons
  - precision threshold on fitness
  - maximum number of iterations

SP1 measure

SP1 – Success Performance “one”

Average required effort for success

$$SP1 = \frac{\text{avg \# evaluations}}{\text{proportion successful runs}}$$

- Effort to reach a given precision on the fitness (success)
- Same number of total fitness evaluations to allow comparisons
- Estimated after a fixed number of runs

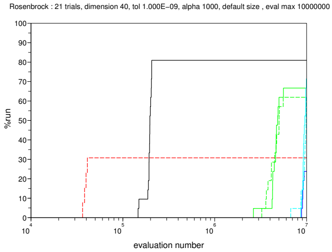
  High (unknown) variance in the case of few successes
- Could also be estimated after a fixed number of successes

  Would allow to control the variance

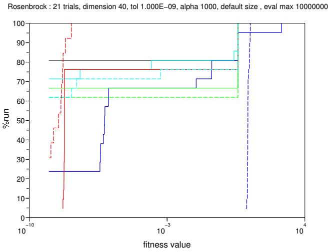
A single value, insufficient alone anyway

Cumulative distributions

Cope with both fitness threshold and bound on # evaluations



% runs
vs
eval to reach success threshold ($= 10^{-9}$)

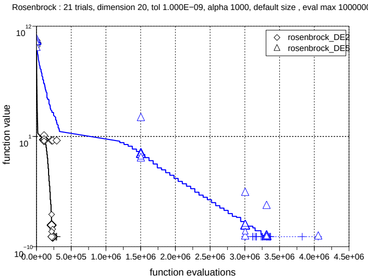


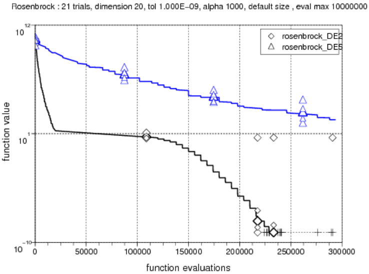
% runs
vs
fitness value reached before max eval ($= 10^7$)

Rosenbrock function, dim 40, cond. 1000

Dynamic Behaviors

Median, best, worse, 25-, 75-percentiles





All runs of 2 settings

Comparing DE2 and DE5 on Rosenbrock(1000), dim=20

Ellipsoid

- $f_{\text{elli}}(x) = \sum_{i=1}^{n} 10^{\alpha \frac{i-1}{n-1}} x_i^2 = x^T H_{\text{elli}} x$

$$H_{\text{elli}} = \begin{pmatrix} 1 & 0 & \cdots \\ & \ddots & \\ \cdots & 0 & 10^{\alpha} \end{pmatrix}$$

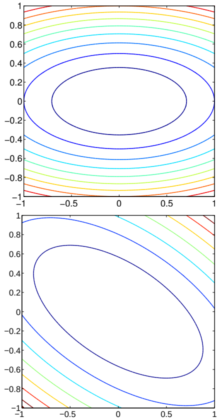
convex, separable

- $f_{\text{elli}}^{\text{rot}}(x) = f_{\text{elli}}(\boldsymbol{R}x) = x^T H_{\text{elli}}^{\text{rot}} x$

$\boldsymbol{R}$ random rotation

$H_{\text{elli}}^{\text{rot}} = \boldsymbol{R}^T H_{\text{elli}} \boldsymbol{R}$

convex, non-separable



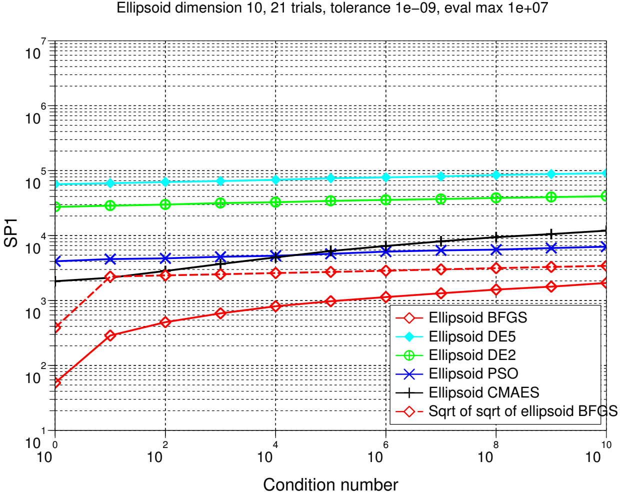
- $\text{cond}(H_{\text{elli}}) = \text{cond}(H_{\text{elli}}^{\text{rot}}) = 10^{\alpha}$

$\alpha = 1, \ldots, 10$

$\alpha = 6 \equiv$ axis ratio of 10^3, typical for real-world problem

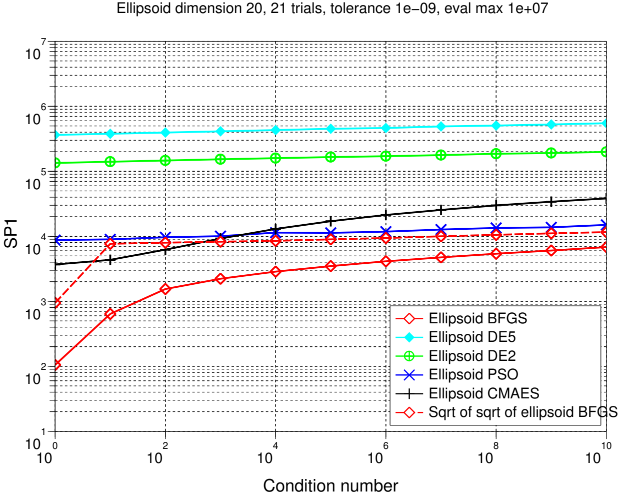
Separable Ellipsoid function

PSO, DE2, DE5, CMA-ES, and BFGS - Dimension 10



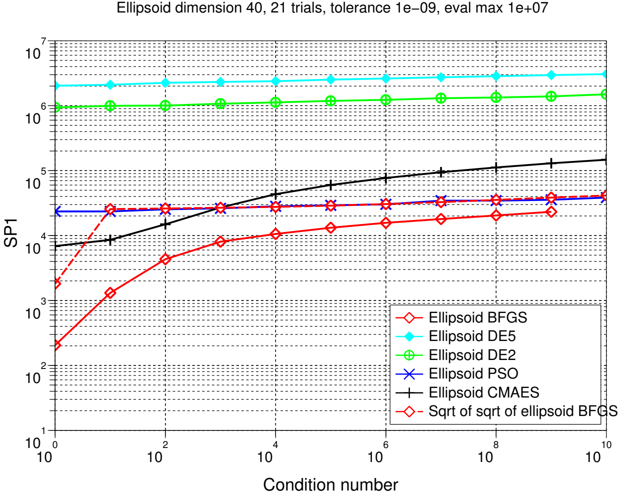
Separable Ellipsoid function

PSO, DE2, DE5, CMA-ES, and BFGS - Dimension 20



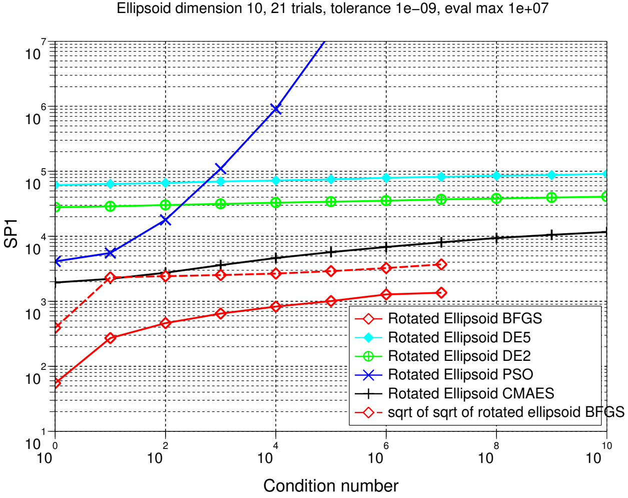
Separable Ellipsoid function

PSO, DE2, DE5, CMA-ES, and BFGS - Dimension 40



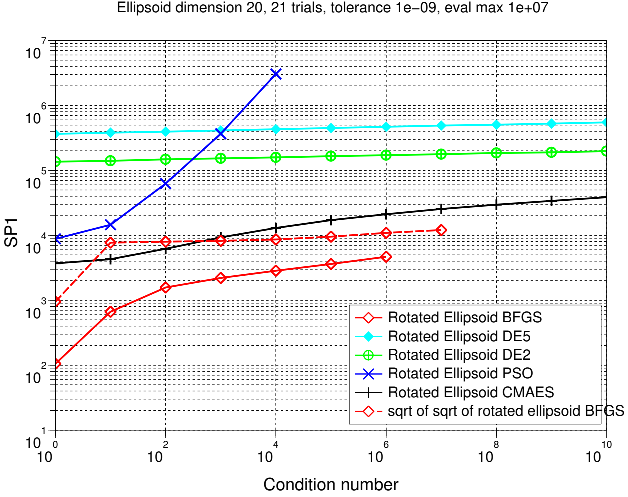
Rotated Ellipsoid function

PSO, DE2, DE5, CMA-ES, and BFGS - Dimension 10



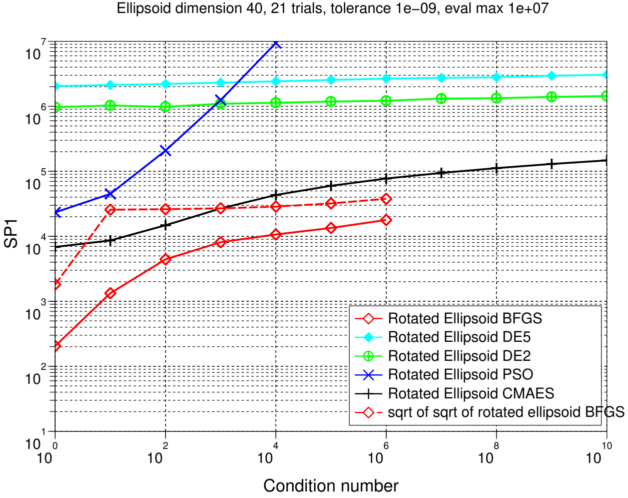
Rotated Ellipsoid function

PSO, DE2, DE5, CMA-ES, and BFGS - Dimension 20



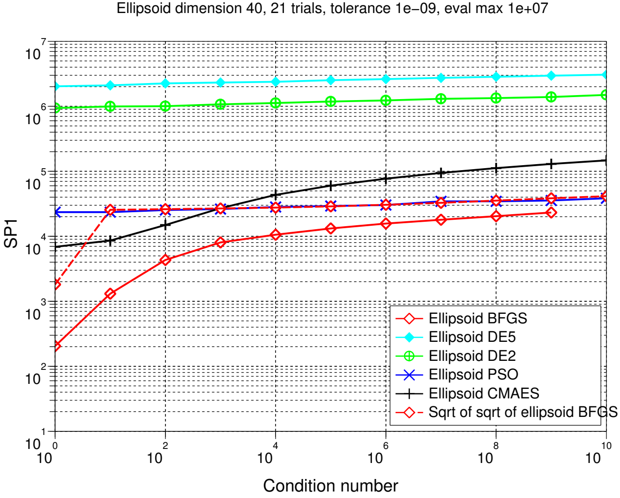
Rotated Ellipsoid function

PSO, DE2, DE5, CMA-ES, and BFGS - Dimension 40



Separable Ellipsoid function

PSO, DE2, DE5, CMA-ES, and BFGS - Dimension 40



Ellipsoid: Discussion

Bio-inspired algorithms

- Separable case: PSO and DE insensitive to conditionning
- ...but PSO rapidly fails to solve the rotated version
- ...while CMA-ES and DE ($CR = 1$) are rotation invariant
- DE scales poorly with dimension
  $d^{2.5}$ compared to $d^{1.5}$ for PSO and CMA-ES and BFGS

...vs BFGS

- BFGS fails to solve ill-conditionned cases
  Matlab message: *Roundoff error is stalling convergence*
  Line search couldn't find an acceptable point in the current search direction
- CMA-ES only 7 times slower
  on quadratic functions!

Rosenbrock function (Banana)

$$f_{\text{rosen}}(x) = \sum_{i=1}^{n-1} \left[(1 - x_i)^2 + 10^{\alpha}(x_{i+1} - x_i^2)^2\right]$$

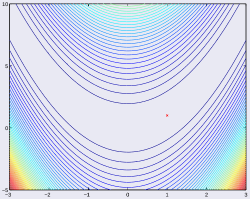
- Non-separable, but . . .

  also ran rotated version

- $\alpha = 2$, classical Rosenbrock function

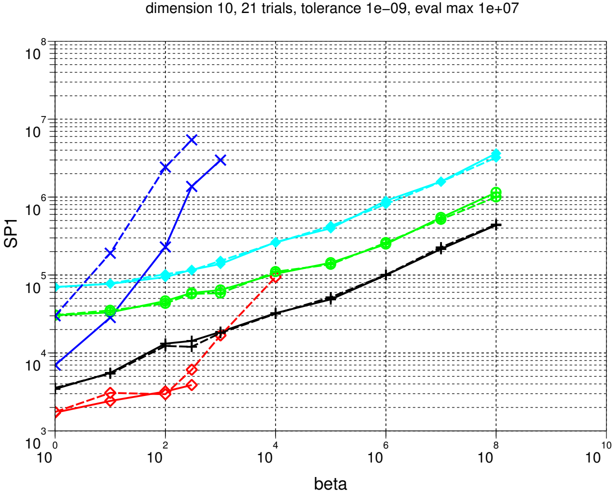
  $\alpha = 1, \ldots, 10^8$

- Multi-modal for dimension > 3



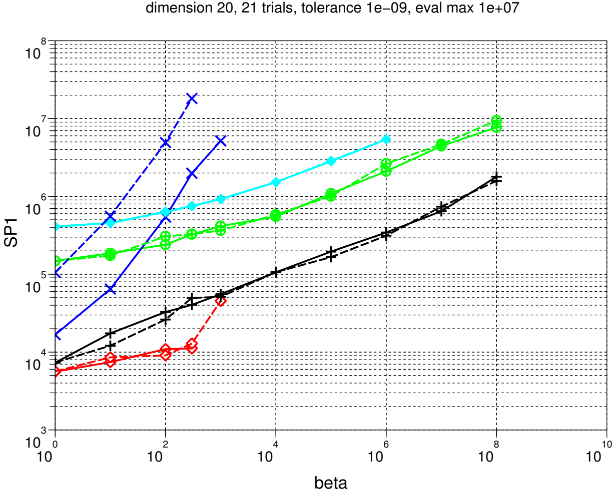
Rosenbrock functions

PSO, DE2, DE5, CMA-ES, and BFGS - Dimension 10



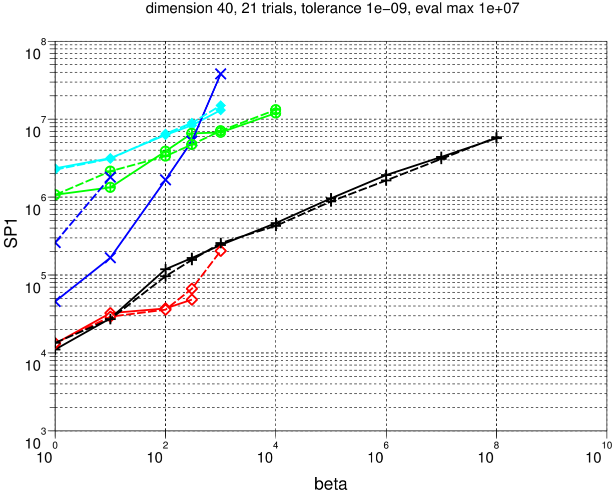
Rosenbrock functions

PSO, DE2, DE5, CMA-ES, and BFGS - Dimension 20



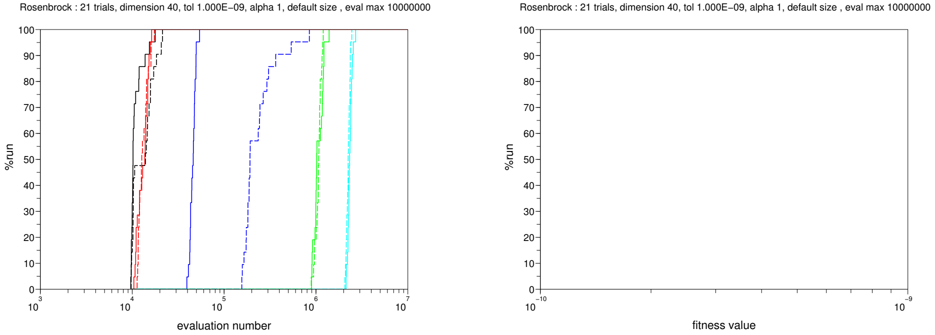
Rosenbrock functions

PSO, DE2, DE5, CMA-ES, and BFGS - Dimension 40



Rosenbrock function – Dim 40 – Cumulative distributions

PSO, DE2, DE5, CMA-ES, and BFGS – α 1

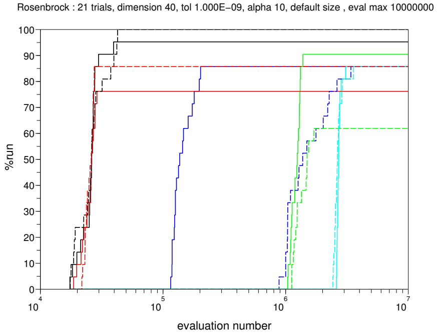


% success
vs
eval to reach success threshold ($= 10^{-9}$)

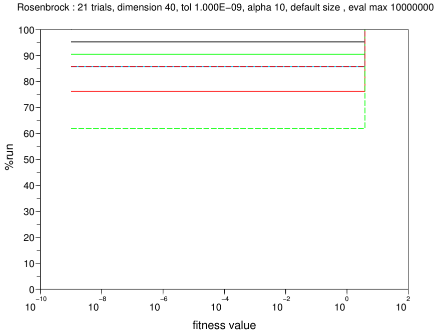
% success
vs
fitness value reached before max eval ($= 10^7$)

Rosenbrock function – Dim 40 – Cumulative distributions

PSO, DE2, DE5, CMA-ES, and BFGS – α 10



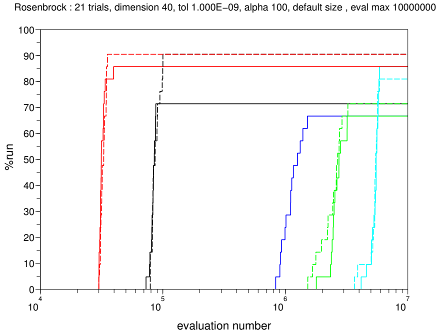
% success
vs
eval to reach success threshold ($= 10^{-9}$)

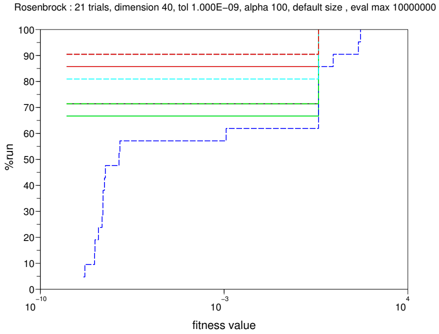


% success
vs
fitness value reached before max eval ($= 10^{7}$)

Rosenbrock function – Dim 40 – Cumulative distributions

PSO, **DE2**, **DE5**, **CMA-ES**, and **BFGS** – α 100



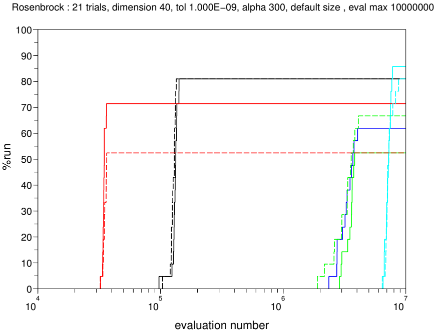


% success
vs
eval to reach success threshold ($= 10^{-9}$)

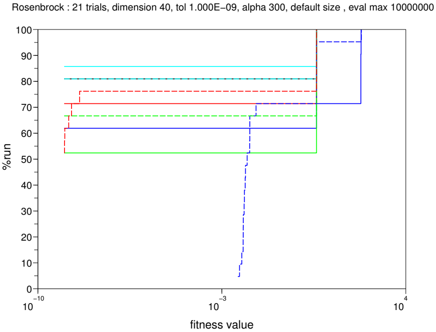
% success
vs
fitness value reached before max eval ($= 10^7$)

Rosenbrock function – Dim 40 – Cumulative distributions

PSO, **DE2**, **DE5**, **CMA-ES**, and **BFGS** – α 300



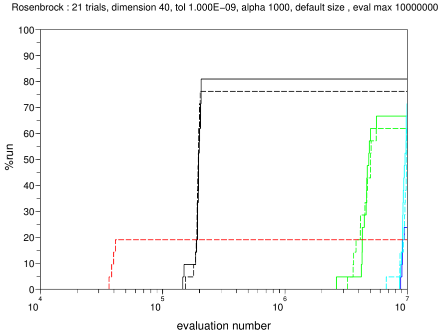
% success
vs
eval to reach success threshold ($= 10^{-9}$)

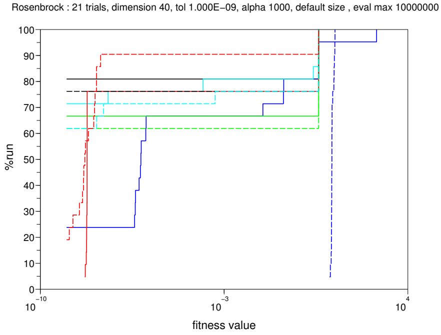


% success
vs
fitness value reached before max eval ($= 10^7$)

Rosenbrock function – Dim 40 – Cumulative distributions

PSO, **DE2**, **DE5**, **CMA-ES**, and **BFGS** – α 1000



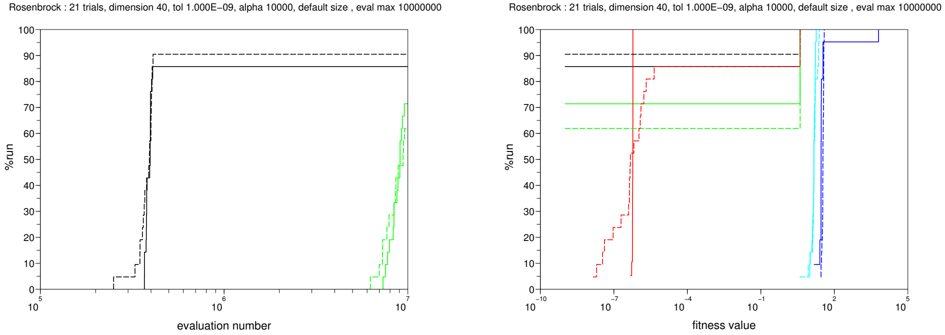


% success
vs
eval to reach success threshold ($= 10^{-9}$)

% success
vs
fitness value reached before max eval ($= 10^7$)

Rosenbrock function – Dim 40 – Cumulative distributions

PSO, **DE2**, **DE5**, **CMA-ES**, and **BFGS** – α 10000



% success
vs
eval to reach success threshold ($= 10^{-9}$)

% success
vs
fitness value reached before max eval ($= 10^7$)

Rosenbrock: Discussion

Bio-inspired algorithms

- PSO sensitive to non-separability
- DE still scales badly with dimension

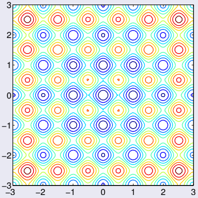
. . . vs BFGS

- Numerical premature convergence on ill-condition problems
- Both local and global restarts improve the results

Rastrigin function

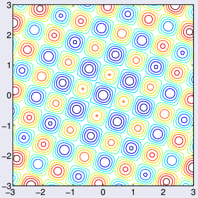
$f_{\text{rast}}(x) = 10n + \sum_{i=1}^{n} x_i^2 - 10\cos(2\pi x_i)$

- separable
- multi-modal



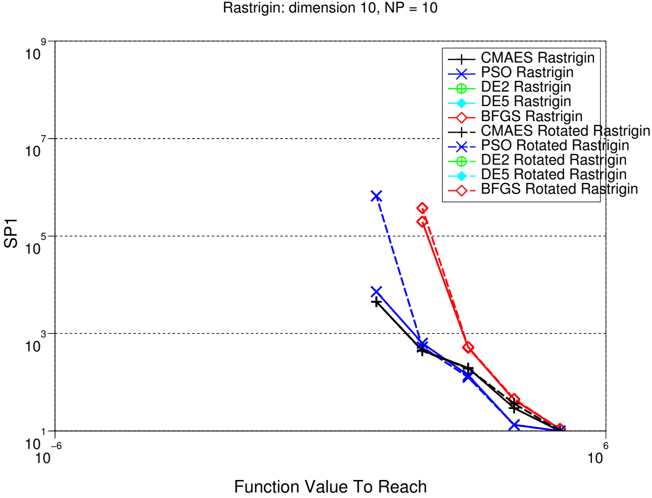
$f_{\text{rast}}^{\text{rot}}(x) = f_{\text{rast}}(\boldsymbol{R}x)$

- $\boldsymbol{R}$ random rotation
- non-separable
- multimodal



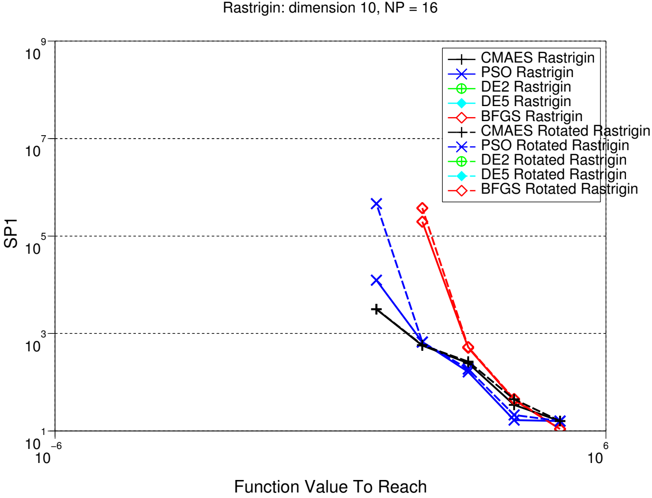
Rastrigin function - SP1 vs fitness value

PSO, DE2, DE5, CMA-ES, and BFGS - PopSize 10



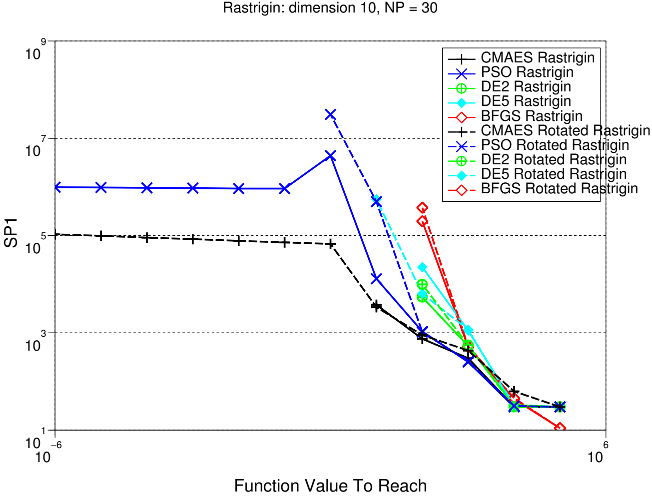
Rastrigin function - SP1 vs fitness value

PSO, **DE2**, **DE5**, **CMA-ES**, and **BFGS** - PopSize 16



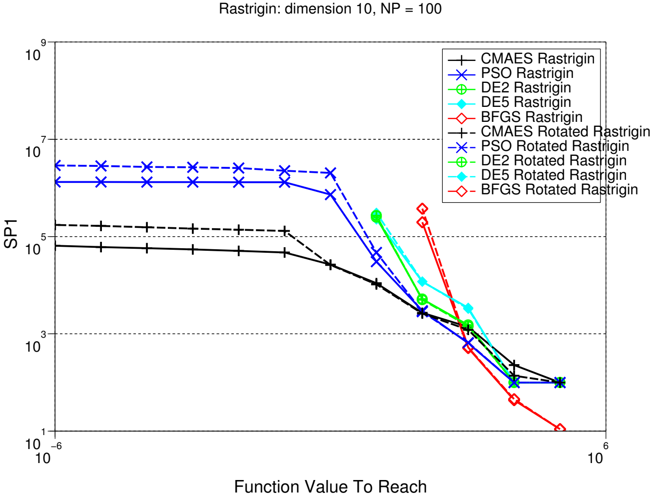
Rastrigin function - SP1 vs fitness value

PSO, DE2, DE5, CMA-ES, and BFGS - PopSize 30



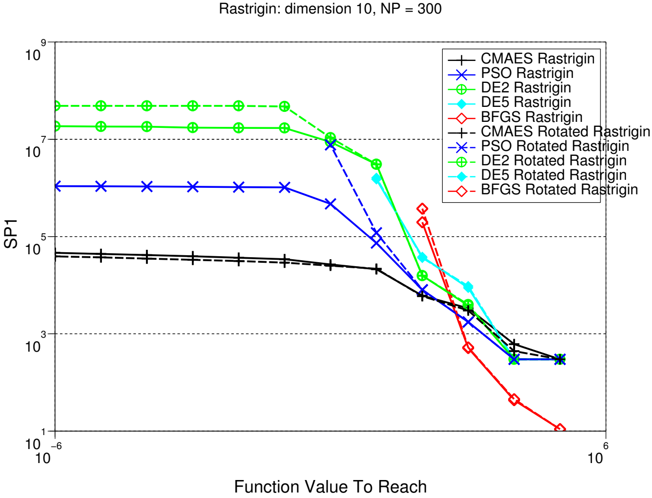
Rastrigin function - SP1 vs fitness value

PSO, **DE2**, **DE5**, **CMA-ES**, and **BFGS** - PopSize 100



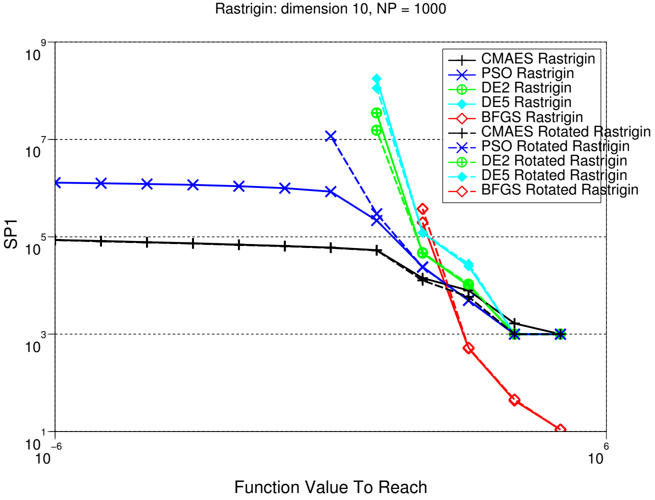
Rastrigin function - SP1 vs fitness value

PSO, DE2, DE5, CMA-ES, and BFGS - PopSize 300



Rastrigin function - SP1 vs fitness value

PSO, DE2, DE5, CMA-ES, and BFGS - PopSize 1000



Rastrigin function - Cumulative distributions

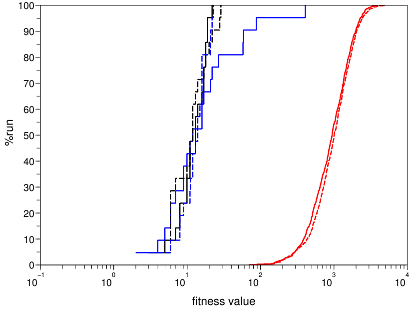
PSO, DE2, DE5, CMA-ES, and BFGS - PopSize 10

Rastrigin : 21 trials, dimension 10, tol 1.000E−09, alpha 10, default size , eval max 10000000

%run

evaluation number

Rastrigin : 21 trials, dimension 10, tol 1.000E−09, alpha 10, default size , eval max 10000000



% success
vs
eval to reach success threshold ($= 10^{-9}$)

% success
vs
fitness value reached before max eval ($= 10^{7}$)

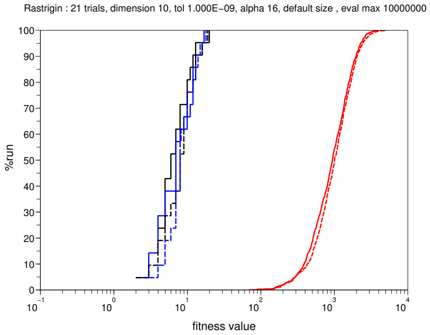
Rastrigin function - Cumulative distributions

PSO, DE2, DE5, CMA-ES, and BFGS - PopSize 16

Rastrigin : 21 trials, dimension 10, tol 1.000E−09, alpha 16, default size , eval max 10000000

%run

evaluation number

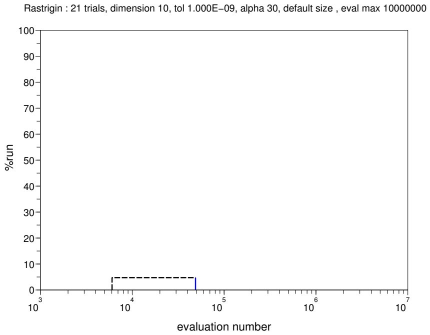


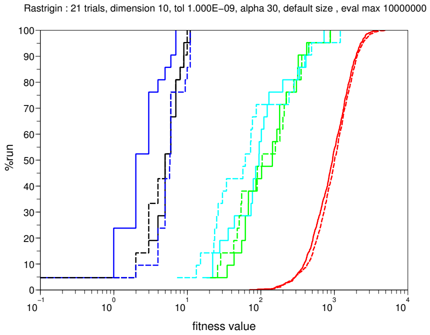
% success
vs
eval to reach success threshold ($= 10^{-9}$)

% success
vs
fitness value reached before max eval ($= 10^{7}$)

Rastrigin function - Cumulative distributions

PSO, **DE2**, **DE5**, **CMA-ES**, and **BFGS** - PopSize 30



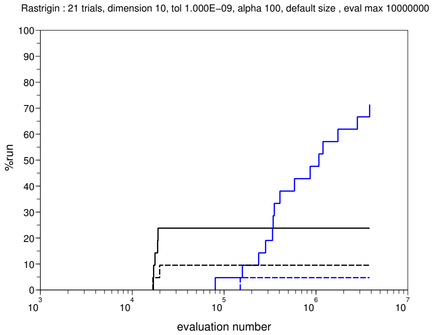


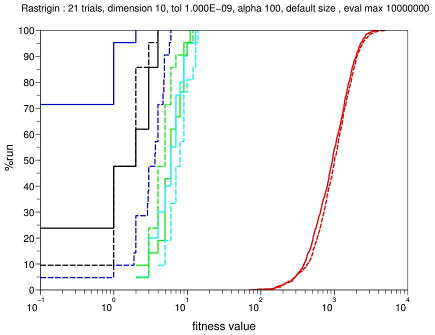
% success
vs
eval to reach success threshold ($= 10^{-9}$)

% success
vs
fitness value reached before max eval ($= 10^7$)

Rastrigin function - Cumulative distributions

PSO, DE2, DE5, CMA-ES, and BFGS - PopSize 100



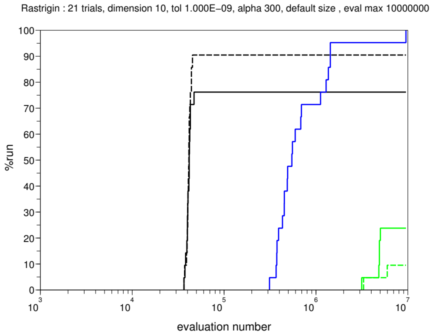


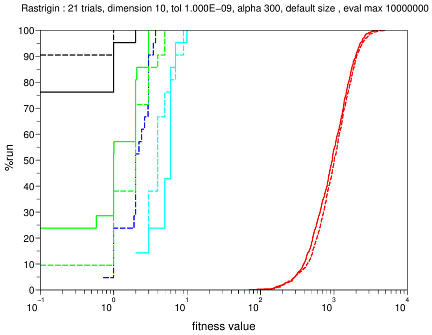
% success
vs
eval to reach success threshold ($= 10^{-9}$)

% success
vs
fitness value reached before max eval ($= 10^{7}$)

Rastrigin function - Cumulative distributions

PSO, DE2, DE5, CMA-ES, and BFGS - PopSize 300



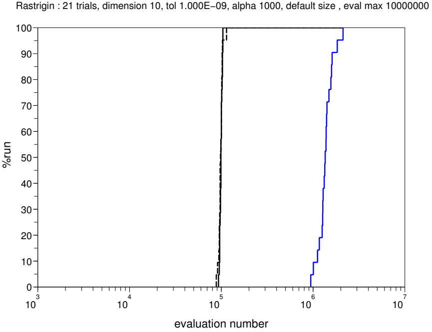


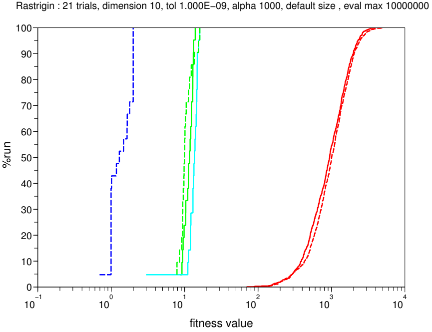
% success
vs
eval to reach success threshold ($= 10^{-9}$)

% success
vs
fitness value reached before max eval ($= 10^{7}$)

Rastrigin function - Cumulative distributions

PSO, **DE2**, **DE5**, **CMA-ES**, and **BFGS** - PopSize 1000





% success
vs
eval to reach success threshold ($= 10^{-9}$)

% success
vs
fitness value reached before max eval ($= 10^7$)

Rastrigin: Discussion

Bio-inspired algorithms

- Increasing population size improves the results

  Optimal size is algorithm-dependent
- CMA-ES and PSO solve separable case

  PSO about 100 times slower
- Only CMA-ES solves the rotated Rastrigin reliably

  requires popSize ≥ 300

...vs BFGS

- Gets stuck in local optima
- Whatever the restart strategies

  No numerical premature convergence

Away from “quadraticity“

Non-linear scaling invariance

- Comparison-based algorithms are insensitive to monotonous transformations   True for DE, PSO and all ESs
- BFGS is not   and convergence results do depend on convexity

Other test functions

- Simple transformation of ellispoid

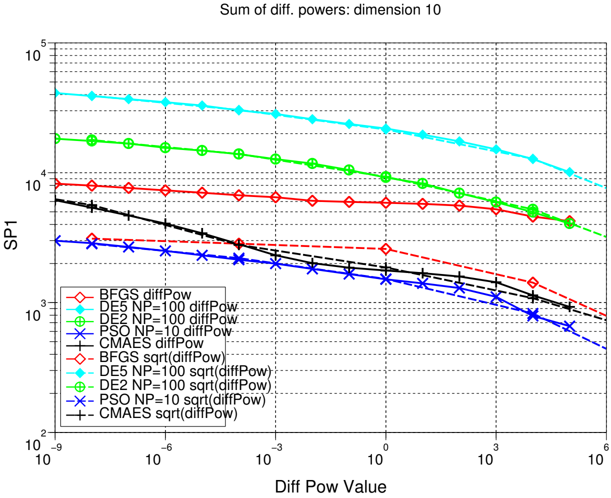
$$f_{\mathrm{SSE}}(x) = \sqrt{\sqrt{f_{\mathrm{elli}}(x)}}$$

- The DiffPow function   and $\sqrt{\sqrt{DiffPow}}$

$$f_{\mathrm{DiffPow}}(x) = \sum(|x_i|^{2+(10*i)})$$

DiffPow – SP1 vs Fitness Values

PSO, DE2, DE5, CMA-ES, and BFGS - Dimension 10



Sqrt and DiffPow: Discussion

Bio-inspired algorithms

- Invariant — as expected!
- PSO performs best — DiffPow is separable

. . . vs BFGS

- Worse on $\sqrt{\sqrt{\text{Ellipsoid}}}$ than on Ellipsoid
- Better on $\sqrt{\sqrt{\text{DiffPow}}}$ than on DiffPow

  'closer' to quadraticity?
- Premature numerical convergence for high CN . . .
  fixed by the "local restart" strategy

1. Continuous optimization and stochastic search
2. Problem difficulties
3. Implementations and parameter settings
4. Experiments and results
   - Experimental conditions
   - Outputs and Performance measures
   - Results
5. Conclusion

What is missing?

- The algorithms
  - Other deterministic methods — e.g. Scilab procedures
  - PCX crossover operator — Specific Evolution Engine
- The testbed
  - Noisy functions
  - Constrained functions
- The statistics
  - Confidence bounds for SP1 and other precision/cost measures
- Real-world functions
  - Which ones ???
  - Do complete experiments

Summary

Bio-inspired algorithms

- All are monotonous-transformation invariant
- PSO very good ...only on separable (easy) functions
- DE poorly scales up with the dimension

Sensitive to non-separability when $CR < 1$

- CMA-ES is a clear best choice

Redo experiments with parameter-less restart version

BFGS

- Optimal choice for quasi-quadratic functions
- but can suffer from numerical premature convergence for high condition number
- Even with restart procedures, fails on multimodal problem

dim 10 only here

The coming of age

The message to our Applied Maths colleagues

Bio-inspired vs BFGS

- CMA-ES only 7 times slower than BFGS

on (quasi-)quadratic functions.

but

- is less hindered by high conditionning,
- is monotonous-transformation invariant,
- is a global search method!

Moreover,

- Theoretical results are catching up
  - Linear convergence for SA-ES with bound on the CV speed — Auger, 05
  - On-going work for CMA-ES