Noisy Optimization

Sandra Astete-Morales, Marie-Liesse Cauwet, Adrien Couetoux, Jérémie Decock, Jialin Liu, Olivier Teytaud

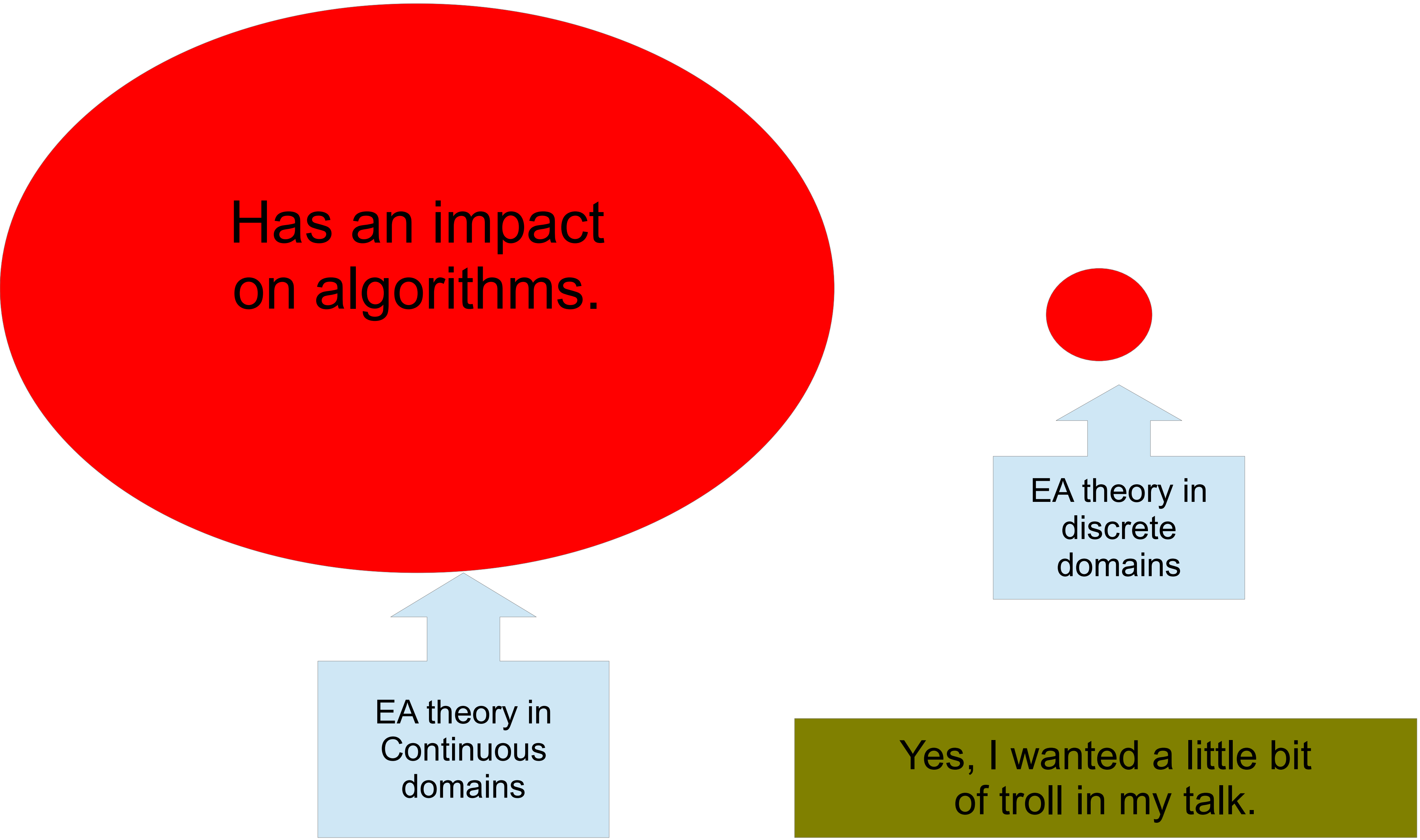
Noisy optimization

Astete-Morales, Cauwet,
Decock, Liu, Rolet, Teytaud

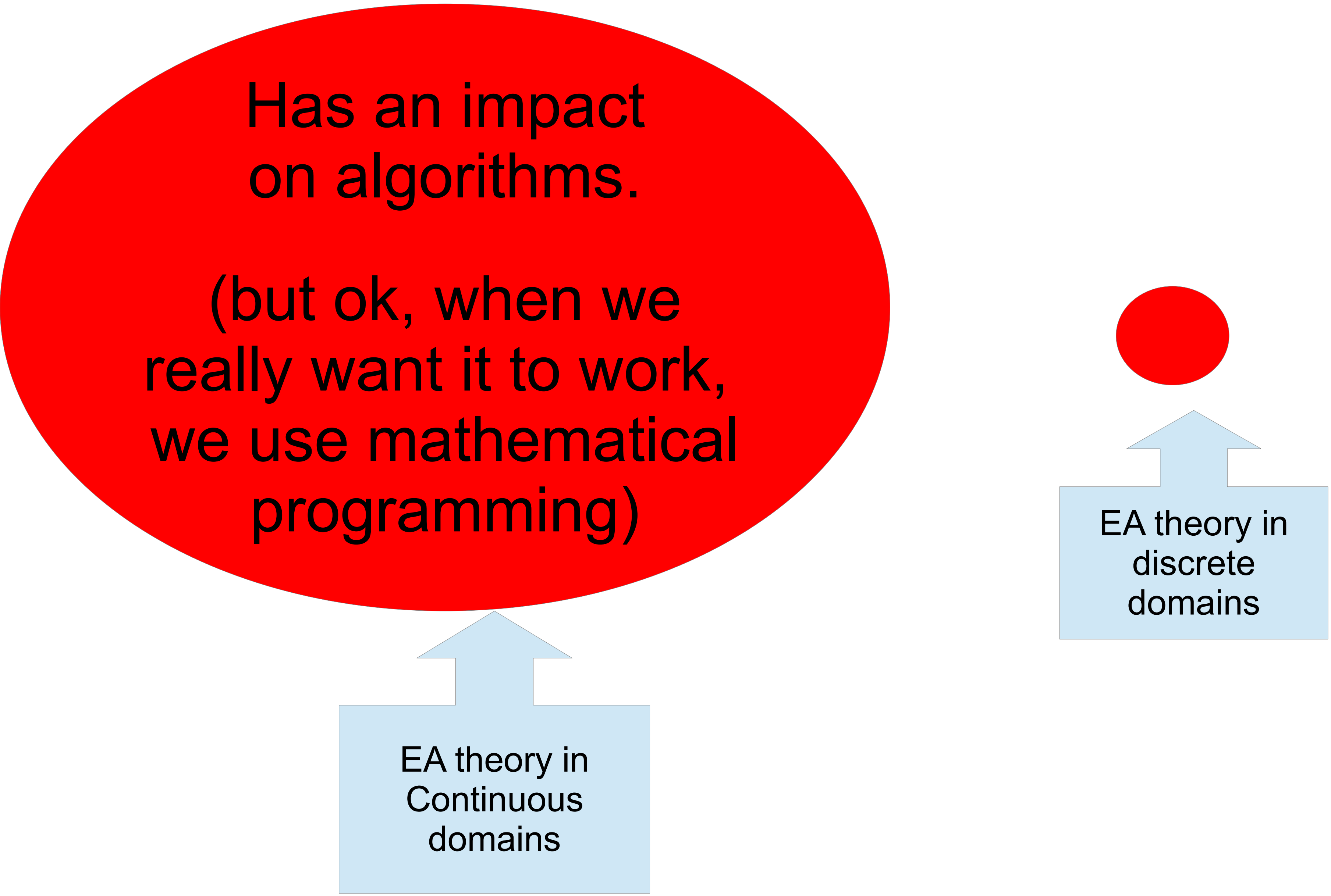
Thanks all !

- Runtime analysis
- Black-box complexity
- Noisy objective function

This talk about noisy optimization in continuous domains



In case I still have friends in the room, a second troll.



Noisy optimization: preliminaries (1)

$x_1, \ldots, x_m, \ldots$ Points at which the fitness function is evaluated (might be bad)

$y_1, \ldots, y_m, \ldots : y_m$ Obtained fitness values

$\tilde{x}_1, \ldots, \tilde{x}_m, \ldots$ Points which are recommended as Approximation of the optimum (should be good)

Noisy optimization: preliminaries (2)

À la Anne Auger:

In noise-free cases, you can get

$$\text{Log } ||x_n|| \sim - C\, n$$

with an ES

(log-linear convergence)

Noisy optimization: preliminaries (3)

A noise model (among others):

$$f(x) = ||x||^p + ||x||^z \times \text{Noise}$$

with “Noise” some i.i.d noise.

- Z=0 ==> additive constant variance noise
- Z=2 ==> noise quickly converging to 0

Part 1: what about ES ?

Let us start with:

- ES with nothing special
- Just reevaluate and same business as usual

Noisy optimization: log-log or log-linear convergence for ES ?

Set z=0 (**constant additive noise**).

Define r(n) = **number of revaluations at iteration n**.

- r(n)=poly(n)
- r(n)=expo(n)
- r(n)=poly (1 / step-size(n))

Results:

- Log-linear w.r.t. iterations
- Log-log w.r.t. evaluations

Noisy optimization: log-log or log-linear convergence for ES ?

- r(n)=expo(n) ==> log(||~x_n||) ~ -C log(n)
- r(n)=poly (1 / ss(n)) ==> log(||~x_n||) ~ -C log(n)

Noise' = Noise after averaging over r(n) revaluations

<u>Proof:</u> *enough revaluations for no misranking*

- d(n) = sequence decreasing to zero.
- **p(n) = P(|E f(xi)-E f(xj)| <= d(n))** (≈same fitness)
- **p'(n) = λ^2 P(| Noise'(xi)-Noise'(xj) | >d(n))**
- choose coeffs so that, if linear convergence in the noise-free case, **∑ p(n)+p(n') < δ**

So with simple revaluation schemes

- z=0 ==> log-log (for #evals)
- z=large (2?) ==> not finished checking, we believe it is log-linear

Now:

What about "races" ?

Races = revaluations until statistical difference.

Part 2

- Bernoulli noise model (sorry, another model...)
- Combining bandits and ES for runtime analysis (see also work by V. Heidrich-Meisner and C. Igel)
- Complexity bounds by information theory (#bits of information)

Bernoulli objective function

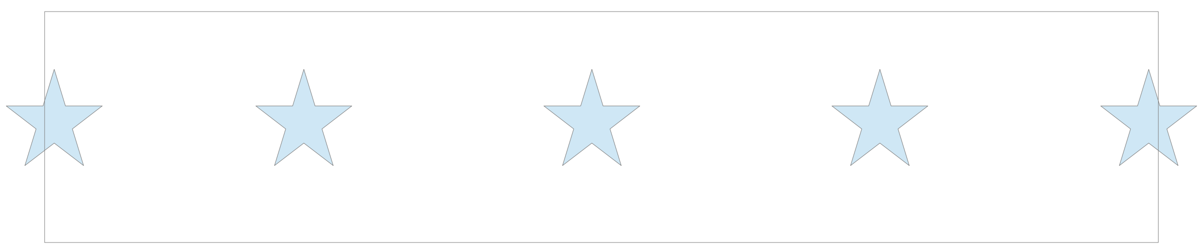
(convenient for applying information theory; what you get is bits of information.)

$D = [0,1]^d$ and

$$f_{\boldsymbol{x}^*,\beta,\gamma}(\boldsymbol{x}) = \mathcal{B}\left(\gamma\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}^*\|}{\sqrt{d}}\right)^{\beta} + (1-\gamma)\right).$$

ES + races

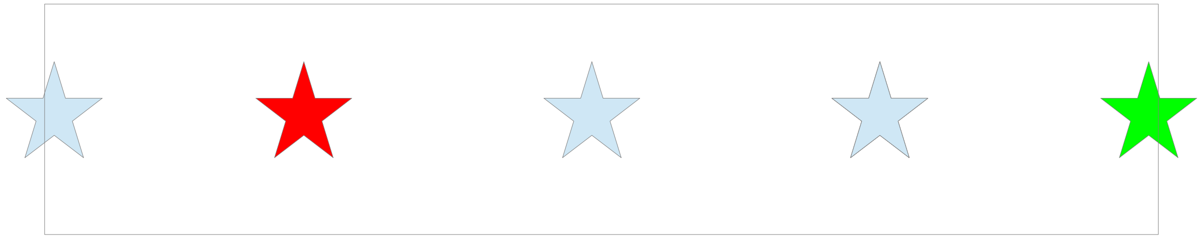
- The race stops when you know ***one of*** the good points
- Assume sphere (sorry)
- Sample five in a row, on one axis



- At some point two of them are statistically different (because of the sphere)
- Reduce the domain accordingly
- Now split another axis

ES + races

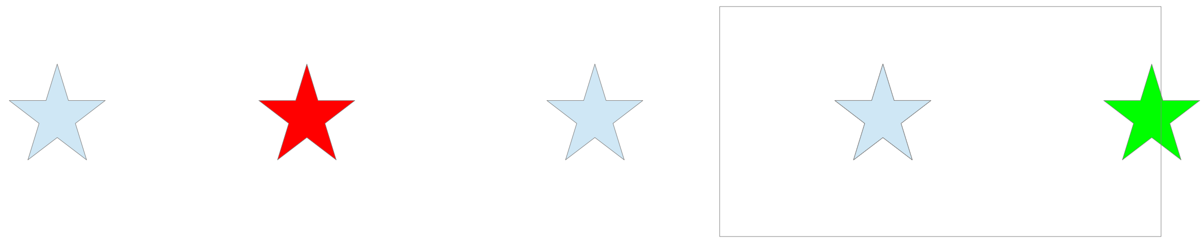
- The race stops when you know ***one of*** the good points
- Assume sphere (sorry)
- Sample five in a row, on one axis



- At some point two of them are statistically different
- Reduce the domain accordingly
- Now split another axis

ES + races

- The race stops when you know ***one of*** the good points
- Assume sphere (sorry)
- Sample five in a row, on one axis



- At some point two of them are statistically different
- Reduce the domain accordingly
- Now split another axis

Black-box complexity: the tools

- Ok, we got a rate for both evaluated and recommended points.
- What about lower bounds ?
- UR = uniform rate = bound for ***worse*** evaluated point
  (see also: cumulative regret)

Proof technique for precision e with Bernoulli noise:

b=#bits of information (yes, it's a rigorous proof)

b > log2(nb of possible solutions with precision e)

b < sum of q(n) with

q(n) = proba that fitness(x*,w) ≠ fitness(x**,w)

Rates on Bernoulli model

$\log||x(n)|| < C - \alpha \log(n)$

"flatness" β	Proved rate for R-EDA in [16] ("flatness" on an envelope of the fitness function; the fitness function does not have to be flat around $\boldsymbol{x}^*$)	R-EDA experimental rate in [14] (on functions with invariances)	This paper (lower bound under locality assumption)
Framework $\gamma = 1$ (small noise)			
1	$\alpha \geq 1$	$\alpha = 1$	$\alpha \leq 1$
2	$\alpha \geq 1/2$	$\alpha = 1/2$	$\alpha \leq \mathbf{1/2}$
4	$\alpha \geq 1/4$	$\alpha = 1/4$	$\alpha \leq \mathbf{1/4}$
Framework $\gamma < 1$ (large noise)			
1	$\alpha \geq 1/2$	$\alpha = 1/2$	$\alpha \leq 1$
2	$\alpha \geq 1/4$	$\alpha = 1/4$	$\alpha \leq \mathbf{1/2}$
4	$\alpha \geq 1/8$	$\alpha = 1/8$	$\alpha \leq \mathbf{1/4}$

Other algorithms reach 1/2. So, yes, sampling close to the optimum makes algorithms slower sometimes.

Conclusions of parts 1 and 2

- **Part 1:** with constant noise, log-log convergence; improved with noise decreasing to 0
- **Part 2:**
  - When using races, sampling should be careful
  - sampling not too close to the optimum can improve convergence rates

Part 3: Mathematical programming and surrogate models

- Fabian 1967: good convergence rate for recommended points (evaluated points: not good) with stochastic gradient
- z=0, constant noise:

Log ||~xn|| ~ C - log (n)/2 (rate=-1/2)

- Chen 1988: this is optimal
- Rediscovered recently in ML conferences

Newton-style information

- Estimate the gradient & Hessian by finite differences
- s(SR) = slope(simple regret)

= log | E f (~xn)-Ef(x*)| / log(n)

- s(CR) = slope(cumulative regret)

= log | sum E f(xi) – Ef(xi)| / log(n)

$$\sigma_n = A / n^{\alpha}$$

Step-size for finite differences

$$\lambda_n = B \lceil n^{\beta} \rceil$$

#revals per point

Newton-style information

- Estimate the gradient & Hessian by finite differences
- s(SR) = slope(simple regret) **= log | E f (~xn)-Ef(x*)| / log(n)**
- s(CR) = slope(cumulative) **= log | sum E f(xi) – Ef(xi)| / log(n)**

(same rate as Fabian for z=0; better for z>0)

ES-friendly Criterion; considers the worst points

z	optimized for CR		optimized for S	
	$s(SR)$	$s(CR)$	$s(SR)$	$s(CR)$
0 (constant var)	$\alpha \simeq \infty, \beta \simeq 4\alpha + 1^+$		$\beta = 5\alpha, \alpha = 0^+$	
	$-\frac{1}{2}$	$\frac{1}{2}$	-1	1
1 (linear var)	$\alpha \simeq \infty, \beta \simeq 2\alpha + 1^+$			
	-1	0	-1	0
2 (quadratic var)	$\alpha \simeq \infty, \beta > 1$			
	$-\infty$	0	$-\infty$	0

Conclusions

- **Compromise between SR and CR** (ES better for CR)
- **Information theory** can help for proving complexity bounds in the noisy case as well
- **Bandits + races = require careful sampling** (if two points very close, huge #revals)
- **Newton-style algorithms** are fast … when z>0
- **No difference between evaluated points & recommended points ==> slow rates** (similar to simple regret vs cumulative regret debates in bandits)

I have no xenophobia against discrete domains :-)

http://www.lri.fr/~teytaud/discretenoise.pdf

is a Dagstuhl-collaborative work on discrete optimization (thanks Youhei, Adam, Jonathan).