



Noisy Optimization

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

► **To cite this version:**

| Sandra Astete-Morales, Marie-Liesse Cauwet, Adrien Couetoux, Jérémie Decock, Jialin Liu, et al..
| Noisy Optimization. Dagstuhl seminar 13271, 2013, Dagstuhl, Germany. 2013. <hal-00844305>

HAL Id: hal-00844305

<https://hal.inria.fr/hal-00844305>

Submitted on 14 Jul 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Noisy optimization



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




Thanks all !

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


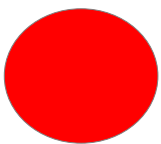
This talk about noisy optimization in continuous domains



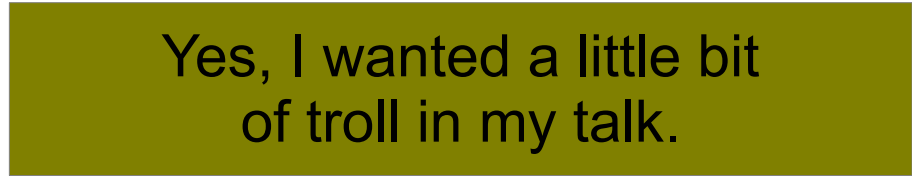
Has an impact on algorithms.



EA theory in Continuous domains



EA theory in discrete domains



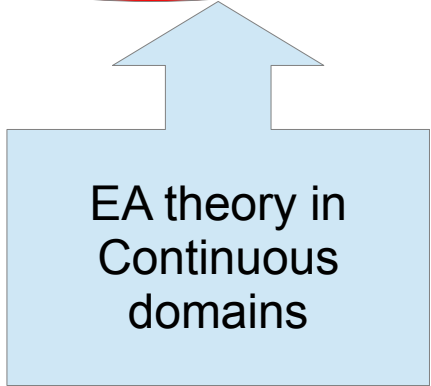
Yes, I wanted a little bit of troll in my talk.

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

Has an impact on algorithms.

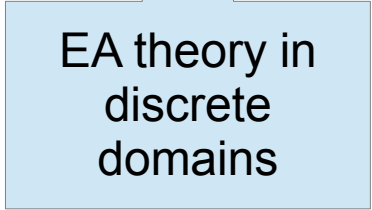
(but ok, when we really want it to work, we use mathematical programming)

EA theory in Continuous domains



```
graph BT; A[EA theory in Continuous domains] --> B(( )); B --- C[Has an impact on algorithms.]; C --- D["(but ok, when we really want it to work, we use mathematical programming)"];
```

EA theory in discrete domains



```
graph BT; A[EA theory in discrete domains] --> B(( )); B --- C[Has an impact on algorithms.]; C --- D["(but ok, when we really want it to work, we use mathematical programming)"];
```

Noisy optimization: preliminaries (1)

x_1, \dots, x_m, \dots

Points at which the fitness function
is evaluated (might be bad)

$y_1, \dots, y_m, \dots : y_m$

Obtained fitness values

$\tilde{x}_1, \dots, \tilde{x}_m, \dots$

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.

- $Z=0 \implies$ additive constant variance noise
- $Z=2 \implies$ 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 .**

- $r(n)=\text{poly}(n)$
- $r(n)=\text{expo}(n)$
- $r(n)=\text{poly} (1 / \text{step-size}(n))$

Results:

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

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

- $r(n)=\text{expo}(n) \implies \log(\|\tilde{x}_n\|) \sim -C \log(n)$
- $r(n)=\text{poly}(1 / \text{ss}(n)) \implies \log(\|\tilde{x}_n\|) \sim -C \log(n)$

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

Proof: *enough revaluations for no misranking*

- $d(n)$ = sequence decreasing to zero.
- $p(n) = P(|E f(x_i) - E f(x_j)| \leq d(n))$ (\approx same fitness)
- $p'(n) = \lambda^2 P(| \text{Noise}'(x_i) - \text{Noise}'(x_j) | > d(n))$
- choose coeffs so that, if linear convergence in the noise-free case, $\sum p(n) + p(n') < \delta$

So with simple revaluation schemes

- $z=0 \implies \log\text{-log}$ (for #evals)
- $z=\text{large (2?) } \implies$ 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

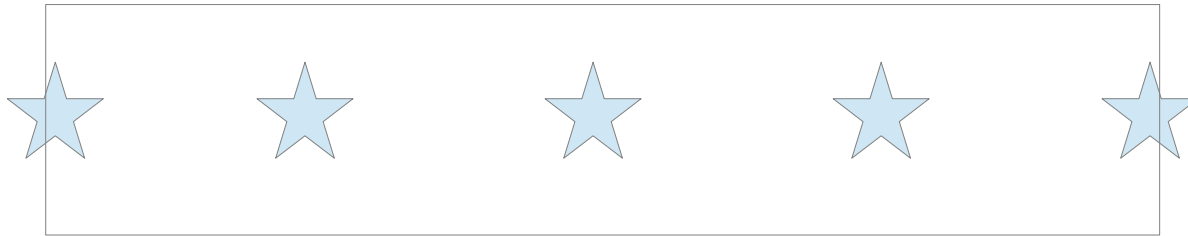
(convenient for applying information theory; what you get is bits of information.)

$$D = [0, 1]^d \text{ and}$$

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

ES + races

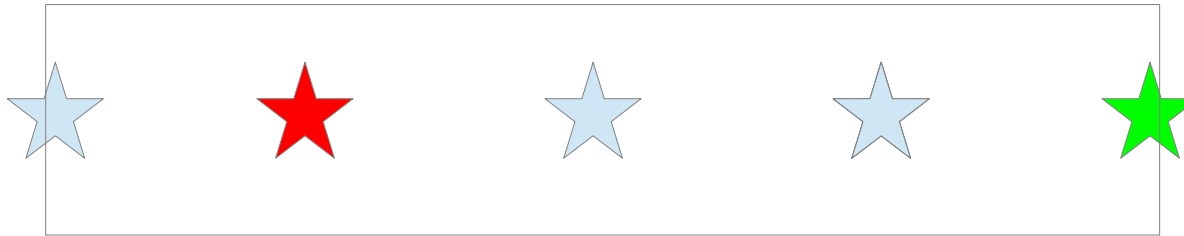
- The race stops when you know **one of** the good points
- Assume sphere (sorry)
- Sample **five** in a row, on one axis



- 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 ϵ with Bernoulli noise:

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

$b > \log_2(\text{nb of possible solutions with precision } \epsilon)$

$b < \text{sum of } q(n) \text{ with}$

$q(n) = \text{proba that } \text{fitness}(x^*, w) \neq \text{fitness}(x^{**}, w)$

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 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	$\alpha \geq 1/2$	$\alpha = 1/2$	$\alpha \leq 1/2$
4	$\alpha \geq 1/4$	$\alpha = 1/4$	$\alpha \leq 1/4$
Framework $\gamma < 1$ (large noise)			
1	$\alpha \geq 1/2$	$\alpha = 1/2$	$\alpha \leq 1$
2	$\alpha \geq 1/4$	$\alpha = 1/4$	$\alpha \leq 1/2$
4	$\alpha \geq 1/8$	$\alpha = 1/8$	$\alpha \leq 1/4$

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:

$$\text{Log } \|\tilde{x}_n\| \sim C - \log(n)/2 \quad (\text{rate}=-1/2)$$

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

Newton-style information

- Estimate the gradient & Hessian by finite differences

- $s(\text{SR}) = \text{slope}(\text{simple regret})$

$$= \log | \mathbb{E} f(\tilde{x}_n) - \mathbb{E} f(x^*) | / \log(n)$$

- $s(\text{CR}) = \text{slope}(\text{cumulative regret})$

$$= \log | \text{sum } \mathbb{E} f(x_i) - \mathbb{E} f(x_i) | / \log(n)$$

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

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

Step-size for
finite differences

#revals per point

Newton-style information

- Estimate the gradient & Hessian by finite differences
- $s(SR) = \text{slope}(\text{simple regret}) = \log | E f(\tilde{x}_n) - E f(x^*) | / \log(n)$
- $s(CR) = \text{slope}(\text{cumulative}) = \log | \text{sum } E f(x_i) - E f(x_i) | / \log(n)$

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

z	optimized for CR		optimized for SR	
	$s(SR)$	$s(CR)$	$s(SR)$	$s(CR)$
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

ES-friendly
Criterion;
considers
the worst
points

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 \implies 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).