Hybridization of dynamic optimization methodologies

Jérémie Decock

Advisor: Olivier Teytaud

I-Lab: LRI - Inria & Artelys

Ph.D. Defense

November 28, 2014



I-Lab: LRI - Inria & Artelvs

Decock

Subject

Comparison and hybridation of Sequential Decision Making (SDM) algorithms

- Long-term reward
- Interdependence among decisions
- Exponential number of solutions



I-Lab: LRI - Inria & Artelvs

Application field

Decision aids for Power systems management

Outline

Introduction

Introduction

State of the art

State of the art in Sequential Decision Making State of the art in Power Systems

Contributions

Direct Value Search Noisy Optimization: Lower Bounds on Runtimes Noisy Optimization: Variance Reduction

3

I-Lab: LRI - Inria & Artelvs

Conclusion

Conclusions and perspectives

Introduction •0000000000	State of the art	Contributions	Conclusion 000000	References
Introduction				

Introduction

▲ロト ▲唇 ▶ ▲ 臣 ▶ ▲ 臣 ▶ ― 臣 | 毛 つの(I-Lab: LRI - Inria & Artelys

4

Hybridization of dynamic optimization methodologies

Energy policies

Optimizing energy management: a crucial problem in many respects

5

I-Lab: LRI - Inria & Artelvs

- economics
- political and geopolitical
- ecological and health
- societal

Energy policies

New technical constraints:

- intermittent: solar power plants and wind turbines
- distribution network security: more and more interconnections increase blackout risks

To sum up

 As there are more and more constraints, decision are more and more complex;

6

I-Lab: LRI - Inria & Artelvs

Environmental, economical and political consequences of inappropriate choices are more and more severe.

Context and Motivations

Need for better decision aids tools

Prospective models are very valuable decision support. They are commonly used to forecast and optimise the short and mid term electricity production and long term investment strategies.

Our goals

- create new decision aid tools accounting for the latest political, economical, environmental, sociological and technical constraints
- help electricity managers to find better exploitation and investment strategies reflecting realistic large-scale models.

I-Lab: LRI - Inria & Artelvs

Unit Commitment Problems



- A multi-stage problem
- Energy demand (forecast)
- Energy production:
 - Hydroelectricity (N water stocks)

I-Lab: LRI - Inria & Artelvs

- Thermal plants
- Water flow through stock links

Goal

Decide, for each power plant, its power output at time t so as to satisfy the demand with the lowest possible cost.

Challenges

Issues

- Limited forecast (e.g. demand and weather)
- Renewable energies increase production variability
- Transportation introduces constraints

Specifics

- Can't assume Markovian process
 - E.g. weather (influences production and demand)

I-Lab: LRI - Inria & Artelvs

- Increase the state space dimension
- Avoid simplified models to avoid model errors

10

I-Lab: LRI - Inria & Artelys

Optimization and Model Errors

- Optimization errors
- Model errors
 - Simplification errors
 - Anticipativity errors
 - Statistical errors

Optimization and Model Errors

- NEWAVE versus ODIN: comparison of stochastic and deterministic models for the long term hydropower scheduling of the interconnected Brazilian system. M. Zambelli et al., 2011.
- Dynamic Programming and Suboptimal Control: A Survey from ADP to MPC. D. Bertsekas, 2005. (MPC = deterministic forecasts)

11

I-Lab: LRI - Inria & Artelvs

Renewable energy forecasts ought to be probabilistic!
 P. Pinson, 2013 (WIPFOR talk)

Example: POST (ADEME)



High scale investment studies

- (e.g. Europe + North Africa)
 - ► Long term (2030 2050)
 - Huge (non-stochastic) uncertainties
 - Future technologies
 - Future laws
 - Future demand
 - Investment problem
 - Interconnections
 - Storage
 - Smart grids
 - Power plants

Hybridization of dynamic optimization methodologies

Stochastic Control



13

I-Lab: LRI - Inria & Artelvs

Modelling Sequential Decision Making Problems

Notations

- $s_t = \text{state at time } t$
- $a_t =$ action (decision) at time t
- ► $s' = s_{t+1}$
- T(s, a, s') = probability of $s \rightarrow s'$ when action a
- r(s, a) = reward when action a in s
- $\pi(s) =$ action to execute in s (policy)

14

Introduction State of the art Contributions Conclusion Conclusion Conclusion

State of the art in Sequential Decision Making

State of the art in Sequential Decision Making

15

I-Lab: LRI - Inria & Artelys

Introduction State of the art Contributions Conclusion References

State of the art in Sequential Decision Making

Bellman Methods [Bellman57]

Bellman Values

The expected Value V^{π} for each state **s** when the agent follow a given (stationary) policy π is

$$V^{\pi}(\mathbf{s}) = E\left[\sum_{t=0}^{\infty} \gamma^{t} r(\mathbf{s}_{t}) | \pi, \mathbf{s}_{0} = \mathbf{s}
ight]$$

16 ▶ 《团 ▶ 《 코 ▶ 《 코 ▶ 로 크 ∽ 익 근 I-Lab: LRI - Inria & Artelys

Hybridization of dynamic optimization methodologies

Bellman Methods

Bellman Values

The *Bellman equation* gives the best value we can expect for any given state (assuming the optimal policy π^* is followed).

$$V^{\pi^*}(\mathbf{s}) = r(\mathbf{s}) + \gamma \max_{\mathbf{a} \in \mathcal{A}} \left[\sum_{\mathbf{s}' \in \mathcal{S}} T(\mathbf{s}, \mathbf{a}, \mathbf{s}') V(\mathbf{s}') \right]$$

The optimal (stationary) policy π^* is defined using the best expected value V^{π^*} and using the principle of *Maximum Expected Utility* as follow

$$\pi^*(\mathbf{s}) = \arg \max_{\mathbf{a} \in \mathcal{A}} \left[\sum_{\mathbf{s}' \in \mathcal{S}} T(\mathbf{s}, \mathbf{a}, \mathbf{s}') V^{\pi^*}(\mathbf{s}') \right]$$

I-Lab: LRI - Inria & Artelvs

Decock

Introduction State of the art Contributions Conclusion References

Bellman Methods

Computing Bellman Values

$$V^{\pi^*}(\mathbf{s}) = r(\mathbf{s}) + \gamma \max_{\mathbf{a} \in \mathcal{A}} \left[\sum_{\mathbf{s}' \in \mathcal{S}} T(\mathbf{s}, \mathbf{a}, \mathbf{s}') V(\mathbf{s}') \right]$$

There are |S| Bellman equations, one for each state. This system of equations cannot be solved analytically because Bellman equations contains non-linear terms (due to the "max" operator).

18

I-Lab: LRI - Inria & Artelvs

Bellman Methods

[Bellman57]

Backward Induction: the non-stationary case Value V^* for each state **s** at the latest time step T is

 $V_T^*(\mathbf{s}) = 0$ (or some valorization)

Best expected value V^* at time t < T computed backwards:

$$V_t^*(\mathbf{s}) = \max_{\mathbf{a} \in \mathcal{A}} r(\mathbf{s}, \mathbf{a}) + \left[\sum_{\mathbf{s}' \in \mathcal{S}} T(\mathbf{s}, \mathbf{a}, \mathbf{s}') V_{t+1}^*(\mathbf{s}') \right]$$

Optimal action (or decision) $\pi_t^*(\mathbf{s})$:

$$\pi_t^*(\mathbf{s}) = \arg \max_{\mathbf{a} \in \mathcal{A}} \left[r(\mathbf{s}, \mathbf{a}) + \sum_{t \in \mathcal{A}} T(\mathbf{s}, \mathbf{a}, \mathbf{s}') V_{t+1}^*(\mathbf{s}') \right]_{t \in \mathcal{A}} \xrightarrow{19} \mathcal{A}_{t+1}^*$$

Decock

Bellman Methods [Bellman57]

Value Iteration: the stationary case

Bellman update is used in Value Iteration to update V at each iteration.

$$V_{i+1}(\mathbf{s}) \leftarrow r(\mathbf{s}) + \gamma \max_{\mathbf{a} \in \mathcal{A}} \left[\sum_{\mathbf{s}' \in \mathcal{S}} T(\mathbf{s}, \mathbf{a}, \mathbf{s}') V_i(\mathbf{s}') \right]$$

20

I-Lab: LRI - Inria & Artelvs

Decock

Introduction State of the art Contributions Conclusion References

Direct Policy Search

Goal

Finds "good" parameters for a given parametric policy π_{θ} : states \rightarrow actions.

Requires a parametric controller (e.g. neural network)

Principle

Decock

Optimize the parameters on simulations (Noisy Black-Box Optimization).

Introduction State of the art Contributions Conclusion References

State of the art in Sequential Decision Making

Parametric policies π_{θ}

Neural Networks: $\boldsymbol{\theta} = \left(w_{10}^{(1)}, \dots, w_{mn}^{(1)}, w_{10}^{(2)}, \dots, w_{km}^{(2)}\right)^T$



22

Hybridization of dynamic optimization methodologies

	State of the art	Contributions	Conclusion	References
	00000000 000 0			
State of the art in Power Systems				

State of the art in Power Systems

Hybridization of dynamic optimization methodologies

State of the art in Power Systems Stochastic Dual Dynamic Programming (SDDP) [Pereira91]

Decompose the problem in instant reward + future reward as:



Approximate V(.) with Bender cuts

Issues

requires convexity of V(.), small state space and Markovian process



24

I-Lab: LRI - Inria & Artelvs

Decock

ntroduction State of the art Contributions Conclusion References

State of the art in Power Systems

State of the art in Power Systems Model-Predictive Control (MPC) [Bertsekas05]



- Rolling planning (closed-loop)
- Replacing the stochastic parts by a pessimistic deterministic forecast

25

I-Lab: LRI - Inria & Artelvs

 \Rightarrow simple but indeed better than SDDP (Zambelli'11): less assumptions

Summary

	Pros	Cons
S(D)DP	large constrained ${\cal A}$	not anytime
	polynomial time decision making	convex problems only (SDDP)
	asymptotically find the optimum	small ${\cal S}$
		Markovian random process
DPS	anytime	slow on large ${\cal A}$
	large ${\cal S}$	hardly handles decision constraints
	works with non linear functions	
	no random process constraint	

26

I-Lab: LRI - Inria & Artelys

Contribution Merge both approaches !

Decock

Introduction	State of the art	Contributions	Conclusion	References
000000000000000000000000000000000000000	000000000000000000000000000000000000000		000000	
Direct Value Searc	h			

Hybridization of dynamic optimization methodologies

Coupling Bellman Methods with Direct Policy Search

- Optimization of Energy Policies
- with Direct Value Search (DVS)
 - Linear Programming (poly-time decision making)

28

I-Lab: LRI - Inria & Artelvs

Direct Policy Search (on real simulations)

Combine instantaneous reward and valorization:

$$\Pi(\mathbf{s}_{t}) = \arg \max_{\mathbf{a}_{t}} r(\mathbf{s}_{t}, \mathbf{a}_{t}) + V(\mathbf{s}_{t+1})$$

$$V(\mathbf{s}_{t+1}) = \underbrace{\alpha_{t} \cdot \mathbf{s}_{t+1}}_{\text{linear}}$$

$$\alpha_{t} = \underbrace{\pi_{\theta}(\mathbf{s}_{t})}_{\text{non linear}}$$

- Given θ , decision making solved as a LP
- Non-linear mapping for choosing the parameters of the LP from the current state

< □ > < 同 >

29

I-Lab: LRI - Inria & Artelvs

Requires the optimization of θ (noisy black-box optimization problem)

Decock

Direct Value Search

Recourse planning



We assume we know $\iota_{t...k}$ the random realizations from current stage to tactical horizon.

30

I-Lab: LRI - Inria & Artelvs

Decock

Step 1 (offline): compute $\pi_{\theta}(.)$

```
Build parametric policy \pi_{\theta}
```

Require:

```
a parametric policy \pi_{\theta}(.) where \pi_{\theta} is a mapping from S to A,
```

```
a Stochastic Decision Process SDP,
```

```
an initial state {\boldsymbol{s}}
```

Ensure:

```
a parameter \hat{\theta}^* leading to a policy \pi_{\hat{\theta}^*}(.)
```

```
Find a parameter \hat{\theta}^* maximizing the expectation of the following fitness function \theta \mapsto \text{Simulate}(\mathbf{s}, \text{SDP}, \pi_{\theta})
with a given par linear point entimization algorithm (a.g. SA ES, CMA ES)
```

< 17 >

31

ELE NOR

I-Lab: LRI - Inria & Artelvs

with a given non-linear noisy optimization algorithm (e.g. SA-ES, CMA-ES)

return $\hat{\theta}^*$

	State of the art	Contributions	Conclusion	References	
		000000000000000000000000000000000000000			
Direct Value Security					

Direct Value Search Step 1 (offline): compute $\pi_{\theta}(.)$

 $\begin{aligned} \text{Simulate}(\mathbf{s}_{0}, \text{ SDP}, \pi_{\theta}) \\ & r \leftarrow 0 \\ & \text{for } t \leftarrow t_{0}, t_{0} + h, t_{0} + 2h, ..., T \text{ do} \\ & \iota_{t...k} \leftarrow \text{get_forecast}(.) \\ & \alpha \leftarrow \pi_{\theta}(\mathbf{s}_{t}^{+}) \\ & \mathbf{a}_{t...k} \leftarrow \text{arg max}_{\mathbf{a}} r(\mathbf{a}_{t...k}, \iota_{t...k}, \mathbf{s}_{t}) + \alpha_{\mathbf{s}}^{\top} \cdot \mathbf{s}_{t+k-1} \\ & r \leftarrow r + \text{SDP_reward}(\mathbf{s}_{t}, \mathbf{a}_{t...h}, \iota_{t...h}) & \leftarrow \text{ not necessarily linear} \\ & \mathbf{s}_{t+h} \leftarrow \text{ SDP_transition}(\mathbf{s}_{t}, \mathbf{a}_{t...h}, \iota_{t...h}) & \leftarrow \text{ not necessarily linear} \\ & \text{end for} \end{aligned}$

32 ≡ ∽) Q (?

I-Lab: LRI - Inria & Artelvs

return r

Decock

Step 2 (online): use $\pi_{\theta}(.)$ to solve the actual problem

The offline optimisation of π_{θ} can be stopped at any time.

Then

- we have π_{θ} , an approximation of state's marginal value
- we can use it to solve the actual problem, the same manner as in *Simulate*

DVS: A proof of concept



Goal: find a policy which maximises gains

Buy (and store) when the market price is low, sell when the market price is high.

34

I-Lab: LRI - Inria & Artelvs

- The market price is stochastic.
- 10 constrained batteries.

DVS: A proof of concept

- State vector s⁺ = stock level of the 10 batteries + handcrafted inputs
- 10 decision variables at each time step = the quantity of energy to buy/sell for each battery

35

I-Lab: LRI - Inria & Artelvs

Baselines

Recourse planning without final valorization Bellman values ightarrow null ($lpha=\mathbf{0}$)

$$\Pi(\mathbf{s},t) = \arg \max_{\mathbf{a}_t} \max_{\mathbf{a}_{t+1},\dots,\mathbf{a}_{t+k-1}} \mathbb{E}\left[r_t + \dots + r_{t+k-1}\right]$$

Recourse planning with constant marginal valorization Bellman values \rightarrow linear function of total stock ($\alpha = constant$)

$$\Pi(\mathbf{s},t) = \arg \max_{\mathbf{a}_t} \max_{\mathbf{a}_{t+1},\dots,\mathbf{a}_{t+k-1}} \mathbb{E}\left[r_t + \dots + r_{t+k-1} + \alpha.\mathbf{s}_{t+k-1}\right]$$

36

I-Lab: LRI - Inria & Artelvs

with optimized constant marginal valorization α .

Decock
DVS: experimental setting

Parametric policy π_θ = a neural network with N neurons in a single hidden layer and weights vector θ;

37

I-Lab: LRI - Inria & Artelvs

 θ is optimized by maximizing θ → Simulate(θ) with a Self-Adaptive Evolution Strategy (SA-ES).

Introduction 00000000000	State of the art	Contributions 000000000000000000000000000000000000	Conclusion	References
Direct Value Searc	:h			

Results

Decock



I-Lab: LRI - Inria & Artelys

38

Noisy Optimization: Lower Bounds on Runtimes

Hybridization of dynamic optimization methodologies

Introduction

The aim of this 2nd contribution

 Study convergence rate for numerical optimization with noisy objective function

40

I-Lab: LRI - Inria & Artelvs

- Reduce the gap between upper and lower bounds
 - for a given family of noisy objective functions
 - under some assumptions

Introduction State of the art contributions Conclusion Conclusion Conclusion Conclusion Conclusion

Noisy Optimization: Lower Bounds on Runtimes

Considered family F of objective functions

$$f_{\mathbf{x}^*,\beta,\gamma}: \quad [0,1]^d \to \{0,1\}$$
$$x \mapsto \mathcal{B}\left(\underbrace{\gamma\left(\frac{\|\mathbf{x} - \mathbf{x}^*\|}{\sqrt{d}}\right)^\beta + (1-\gamma)}_p\right)$$

*f*_{x*,β,γ} ∈ *F* : a stochastic objective function (a random variable)

41

I-Lab: LRI - Inria & Artelvs

- $\mathcal{B}(p)$: the Bernoulli distribution with parameter p
- $d \in \mathbb{N}^*$: the dimension of the domain of $f_{\mathbf{x}^*,eta,\gamma}$
- $\mathbf{x}^* \in [0,1]^d$: the optimum
- $\beta \in \mathbb{R}^*_+$: the "flatness" of the expectation $\mathbb{E}f$ around \mathbf{x}^*
- ▶ $\gamma \in [0,1]$: a noise parameter (variance at x^*)

 Introduction
 State of the art
 Contributions
 Conclusion
 References

 accordsecodo
 accordsecodo

Considered objective functions



I-Lab: LRI - Inria & Artelys

42

Hybridization of dynamic optimization methodologies

Introduction State of the art contributions Conclusion Conclusion Conclusion Conclusion

Noisy Optimization: Lower Bounds on Runtimes

Considered objective functions

- we assume that the optimum is unique
- we consider noisy optimization in the case of local convergence

Lower bound \rightarrow concerns all families including *F*

43

Hybridization of dynamic optimization methodologies

Contributions References Noisy Optimization: Lower Bounds on Runtimes

The experimental setting for our noisy optimization setting

Require:
$$\omega, \omega', \mathbf{x}^*, \beta, \gamma$$
 and (unknown).
for all $n = 1, 2, 3, ...$ do
 $\mathbf{x}_{\mathbf{x}^*, n, \omega, \omega'} = \text{Optimize}(\mathbf{x}_{\mathbf{x}^*, 1, \omega, \omega'}, \dots, \mathbf{x}_{\mathbf{x}^*, n-1, \omega, \omega'}, y_1, \dots, y_{n-1}, \omega')$
if $\omega_n \leq \mathbb{E} f_{\mathbf{x}^*, \beta, \gamma}(\mathbf{x}_{\mathbf{x}^*, n, \omega, \omega'})$ then
 $y_n = 1$
else
 $y_n = 0$
end if
end for
return $\mathbf{x}_{\mathbf{x}^*, n, \omega, \omega'}$

The framework requires:

- $\triangleright \omega$ uniform random variable for simulating the Bernoulli in f
- ω' a random seed of the algorithm (optimizer's internal randomness)

44

I-Lab: LRI - Inria & Artelvs

- ▶ \mathbf{x}^* the optimum (that minimizes $\mathbb{E}_{\omega} f_{\mathbf{x}^*,\beta,\gamma}(\mathbf{x},\omega)$)
- \triangleright β and γ two fixed parameters of f

The experimental setting for our noisy optimization setting

Require:
$$\omega$$
, ω' , x^* , β , γ and (unknown).
for all $n = 1, 2, 3, ...$ do
 $\mathbf{x}_{x^*, n, \omega, \omega'} = \text{Optimize}(\mathbf{x}_{x^*, 1, \omega, \omega'}, \dots, \mathbf{x}_{x^*, n-1, \omega, \omega'}, y_1, \dots, y_{n-1}, \omega')$
if $\omega_n \leq \mathbb{E} f_{x^*, \beta, \gamma}(\mathbf{x}_{x^*, n, \omega, \omega'})$ then
 $y_n = 1$
else
 $y_n = 0$
end if
end for
return $\mathbf{x}_{x^*, n, \omega, \omega'}$

It's an iterative process. For each iteration:

Optimize (the optimization algorithm) returns the next point to visit

- looking for x*
- according to some inputs
- This point is evaluated by the *fitness function* (if-then-else statement)

45

I-Lab: LRI - Inria & Artelvs

Decock

F

The experimental setting for our noisy optimization setting

Require:
$$\omega$$
, ω' , \mathbf{x}^* , β , γ and (unknown).
for all $n = 1, 2, 3, ...$ do
 $\mathbf{x}_{\mathbf{x}^*, n, \omega, \omega'} = \text{Optimize}(\mathbf{x}_{\mathbf{x}^*, 1, \omega, \omega'}, \dots, \mathbf{x}_{\mathbf{x}^*, n-1, \omega, \omega'}, y_1, \dots, y_{n-1}, \omega')$
if $\omega_n \leq \mathbb{E} f_{\mathbf{x}^*, \beta, \gamma}(\mathbf{x}_{\mathbf{x}^*, n, \omega, \omega'})$ then
 $y_n = 1$
else
 $y_n = 0$
end if
end for
return $\mathbf{x}_{\mathbf{x}^*, n, \omega, \omega'}$

46

I-Lab: LRI - Inria & Artelvs

Optimize makes its recommendation according to:

- the sequence of former visited points \mathbf{x}_n
- their binary noisy fitness values y_n
- \blacktriangleright the optimizer's internal randomness ω'

F

The experimental setting for our noisy optimization setting

$$\begin{array}{l} \mbox{Require: } \omega, \; \omega', \; x^*, \; \beta, \; \gamma \; \mbox{and} \; (\mbox{unknown}). \\ \mbox{for all } n = 1, 2, 3, \ldots \; \mbox{do} \\ & \mbox{$x_{x^*,n,\omega,\omega'} = Optimize($x_{x^*,1,\omega,\omega'}, \; \ldots \;, \; $x_{x^*,n-1,\omega,\omega'}, \; y_1, \; \ldots \;, \; y_{n-1}, \; \omega'$) \\ & \mbox{if } \omega_n \leq \mathbb{E} f_{x^*,\beta,\gamma}($x_{x^*,n,\omega,\omega'}$) \; \mbox{then} \\ & y_n = 1 \\ & \mbox{else} \\ & y_n = 0 \\ & \mbox{end if} \\ & \mbox{end for} \\ & \mbox{return } $x_{x^*,n,\omega,\omega'}$ \\ \end{array}$$

47

I-Lab: LRI - Inria & Artelvs

The fitness function $f_{\mathbf{x}^*,\beta,\gamma}$ outputs

- ▶ 1 if random (= ω_n) is less than $\mathbb{E}f_{\mathbf{x}^*,\beta,\gamma}(\mathbf{x}_{\mathbf{x}^*,n,\boldsymbol{\omega},\boldsymbol{\omega}'})$
- 0 otherwise

Decock

The experimental setting for our noisy optimization setting

$$\begin{array}{l} \mbox{Require: } \omega, \, \omega', \, x^*, \, \beta, \, \gamma \mbox{ and } (unknown). \\ \mbox{for all } n = 1, 2, 3, \dots \mbox{ do} \\ x_{x^*, n, \omega, \omega'} = \mbox{Optimize}(x_{x^*, 1, \omega, \omega'}, \ \dots, \ x_{x^*, n-1, \omega, \omega'}, \ y_1, \ \dots, \ y_{n-1}, \ \omega') \\ \mbox{if } \omega_n \leq \mathbb{E} f_{x^*, \beta, \gamma}(x_{x^*, n, \omega, \omega'}) \mbox{ then } \\ y_n = 1 \\ \mbox{else} \\ y_n = 0 \\ \mbox{end if } \\ \mbox{end for } \\ \mbox{return } x_{x^*, n, \omega, \omega'} \end{array}$$

48

I-Lab: LRI - Inria & Artelvs

Eventually, the estimation of the optimum is returned

Decock

Sampling strategies

Sampling close to the current estimation of the optimum

most evolution strategies do that

Sampling far from the current estimation of the optimum

- when f is learnable
- the optimizer can use a model of f to sample far from x*
- optimize's outputs can have different meanings
 - be the most informative points to sample (exploration)

I-Lab: LRI - Inria & Artelvs

• provide an estimate of arg min $\mathbb{E}f$ (recommendation)

These strategies lead to different convergence rates.

Our assumptions

We assume that the optimization algorithm

- does not require a model of the objective function
- samples close to the optimum (first strategy)

The *locality* assumption

$$\forall f \in F, \quad P\left(\forall i \leq n, \quad \|\mathbf{x}_i - \mathbf{x}^*\| \leq \frac{C(d)}{i^{\alpha}}\right) \geq 1 - \frac{\delta}{2}$$

- for some $0 < \delta < 1/2$
- ► C(d) > 0: a constant depending on d only
- $\alpha > 0$: convergence speed (large α implies a fast convergence)

< 17 >

50

= 200

I-Lab: LRI - Inria & Artelvs

Noisy optimization complexity

What is the best theoretical convergence rate of an optimization algorithm on $f \in F$ assuming this *locality assumption* ?

$$\forall f \in F, \quad P\left(\forall i \leq n, \quad \|\mathbf{x}_i - \mathbf{x}^*\| \leq \frac{C(d)}{i^{\alpha}}\right) \geq 1 - \frac{\delta}{2}$$

51

I-Lab: LRI - Inria & Artelvs

Question

What is the supremum of possible α ?

Introduction State of the art Contributions Conclusion Conclusion Conclusion Conclusion

Noisy Optimization: Lower Bounds on Runtimes

State of the art: upper and lower bounds [Roley2010,Shamir,Chen88

Consider α such that

$$\forall f \in F, \quad P\left(\forall i \leq n, \quad \|\mathbf{x}_i - \mathbf{x}^*\| \leq \frac{C(d)}{i^{\alpha}}\right) \geq 1 - \frac{\delta}{2}$$

	$\gamma=1$ (small noise)	$\gamma < 1$ (large noise)
Proved rate for R-EDA	$rac{1}{eta} \leq lpha$	$rac{1}{2eta}\leq lpha$
Former lower bounds	$lpha \leq 1$	$lpha \leq rac{1}{2}$
R-EDA experimental rates	$lpha=rac{1}{eta}$	$lpha=rac{1}{2eta}$
Rate by active learning	$lpha = rac{1}{2}$	$lpha=rac{1}{2}$

I-Lab: LRI - Inria & Artelys

ъ

52

Hybridization of dynamic optimization methodologies

What we prove

Consider α such that

$$\forall f \in F, \quad P\left(\forall i \leq n, \quad \|\mathbf{x}_i - \mathbf{x}^*\| \leq \frac{C(d)}{i^{\alpha}}\right) \geq 1 - \frac{\delta}{2}$$

	$\gamma=1$ (small noise)	$\gamma < 1$ (large noise)
Proved rate for R-EDA	$rac{1}{eta} \leq lpha$	$rac{1}{2eta}\leq lpha$
Former lower bounds	$lpha \leq 1$	$lpha \leq rac{1}{2}$
This proof	$lpha \leq rac{1}{eta}$	$lpha \leq rac{1}{eta}$
(lower bounds)		(recently: 1/4 for $\beta = 2$)

< □ > < 同 >

53 |≡ ∽⊙⊙

Hybridization of dynamic optimization methodologies

Key ideas

Key point

if $\omega_i < \mathbb{E}f(x^*)$ or $\omega_i > \mathbb{E}f(x_i)$, the function evaluation is the same for all $x^* \Rightarrow$ let us show that this happens for most *i*.

Definitions

- ▶ N = number of ω_i between $\mathbb{E}f(x^*)$ and $\mathbb{E}f(x_i)$ for $i \leq n$.
- X_{n,Ω} the set of points which can be chosen as nth sampled point if the random sequence is Ω = (ω₁,..., ω_n,...).

Card $X_{n,\Omega} \leq 2^N$ We will show that locality $\Rightarrow N$ small. (intuition: N (depends on n) is the number of i which bring information; the locality assumption implies that N is small, i.e. we receive little information)

I-Lab: LRI - Inria & Artelvs

(日) (同) (三) (三)

Lemma

- *N* is probably upper bounded by $O(\sum_{i=1}^{n} 1/i^{\alpha\beta})$.
- (i.e. for most evaluations, the result does not depend on x^* !)

Proof

Decock

probability of $\omega_i \in (\mathbb{E}f(x^*), \mathbb{E}f(x_i))$ is $O(1/i^{\alpha\beta})$ by locality assumption.

- \Rightarrow hence the result by summation (for expectations).
- \Rightarrow hence the result by Chebyshev (with large probability).

Theorem

Theorem

Assume the objective function $f \in F$

Assume the locality assumption

$$\forall f \in F, \quad P\left(\forall i \leq n, \quad \|\mathbf{x}_i - \mathbf{x}^*\| \leq \frac{C(d)}{i^{\alpha}}\right) \geq 1 - \frac{\delta}{2}$$

56

I-Lab: LRI - Inria & Artelys

Then $\alpha \leq 1/\beta$.

Decock

Noisy Optimization: Lower Bounds on Runtimes

Proof of the theorem (1)

Let us show that $\alpha\beta \leq 1$

In order to do so, let us assume, in order to get a contradiction, that $\alpha\beta>1$

57

I-Lab: LRI - Inria & Artelvs

Noisy Optimization: Lower Bounds on Runtimes

Proof of the theorem (2)

Definition

Decock

Consider R a set of points with lower bounded distance to each other:

 two distinct elements of R are at distance greater than 2
 e from each other with

 $\epsilon = \frac{C(d)}{i^{\alpha}}$



58

Noisy Optimization: Lower Bounds on Runtimes

Proof of the theorem (3)

If x^* is uniformly drawn in R...

► optimize should have the opportunity to choose points which at distance ≤ ε of (almost) each possible x*

That is to say:

Decock

X_{n,Ω} has points at distance ≤ ε of a certain percentage

 (1 − δ/2) of elements in R



59

Noisy Optimization: Lower Bounds on Runtimes

Proof of the theorem (4)

But...

- ▶ if e decreases (x_i approach x^{*}) then |R| increases
- and X_{n,Ω} is finite (with great probability, by Lemma)

Up to a certain timestep, the locality assumption can't be respected.

We have a contradiction on our assumption.

 $\alpha\beta>1$ is wrong

Decock

QED



60

	State of the art	Contributions	Conclusion	References
		000000000000000000000000000000000000000		
Noisy Optimization	n: Variance Reduction			

Noisy Optimization: Variance Reduction

Hybridization of dynamic optimization methodologies

Variance Reduction

Aim

Improve convergence rate in *Noisy optimization problems* (e.g. DVS) using traditional *variance reduction* techniques

62

I-Lab: LRI - Inria & Artelvs

Framework

Grey-box: manipulate random seeds

Decock

Noisy Optimization: Variance Reduction

Common Random Numbers (CRN)

Relevant for population-based noisy optimization:

- we want to know $\mathbb{E}f(\mathbf{x}, \boldsymbol{\omega})$ for several \mathbf{x} ,
- we approximate $\mathbb{E}f(\mathbf{x}, \boldsymbol{\omega}) \simeq \frac{1}{n} \sum_{i=1}^{n} f(x, \boldsymbol{\omega}_i)$

 \Rightarrow CRN = use the same samples ω_1,\ldots,ω_n for all x in a same optimization iteration

Stratified Sampling

Each ω_i is randomly drawn conditionally to a stratum. The domain of ω is partitioned into disjoint strata S_1, \ldots, S_N . The stratification function $i \mapsto s(i)$ is chosen by the noisy optimization algorithm.

 ω_i is randomly drawn conditionally to $\omega_i \in S_{s(i)}$.

$$\hat{\mathbb{E}}f(\mathbf{x}, \boldsymbol{\omega}) = \sum_{i \in \{1, \dots, n\}} \frac{P(\boldsymbol{\omega} \in \boldsymbol{s}(i))f(\mathbf{x}, \boldsymbol{\omega}_i)}{\mathsf{Card} \ \{j \in \{1, \dots, n\}; \boldsymbol{\omega}_j \in \boldsymbol{s}(i)\}}$$

I-Lab: LRI - Inria & Artelys

64

Decock

65

ъ

I-Lab: LRI - Inria & Artelvs

Experiments

- Known:
 - CRN can be detrimental
 - Stratification rarely (never ?) detrimental
- Here: experimental test



Experiments



66

I-Lab: LRI - Inria & Artelvs

Decock

Introduction	State of the art	Contributions	Conclusion	References
		000000000000000000000000000000000000000	00000	
Conclusions and n	erspectives			

Conclusions and perspectives

Hybridization of dynamic optimization methodologies

68

I-Lab: LRI - Inria & Artelvs

Conclusions

- DVS: combining DPS and SDP for
 - fast decision making (polynomial)
 - anytime optimization
 - initialized at MPC (which is great)
 - no need for tree representation
- ► CRN:
 - no proof of good properties
 - but empirically really good

69

I-Lab: LRI - Inria & Artelvs

Future work

- Mathematical analysis of DVS
- More intensive testing of DVS
- Optimization at the level of investments
- Non-stochastic uncertainties

Introduction 0000000000	State of the art	Contributions	Conclusion 000●00	References
Conclusions and perspectives				

Bibliography

- An Introduction to Direct Value Search, 9èmes Journées Francophones de Planification, Décision et Apprentissage (JFPDA'14), Liège (Belgique), May 2014.
- Direct model predictive control. Jean-Joseph Christophe, Jérémie Decock, and Olivier Teytaud. In European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), Bruges, Belgique, April 2014.
- Linear Convergence of Evolution Strategies with Derandomized Sampling Beyond Quasi-Convex Functions. Jérémie Decock and Olivier Teytaud. In EA - 11th Biennal International Conference on Artificial Evolution - 2013, Lecture Notes in Computer Science, Bordeaux, France, August 2013. Springer.
- Noisy Optimization Complexity Under Locality Assumption I-Lab: LRI - Inria & Artelys

70

Bibliography

- NEWAVE versus ODIN: comparison of stochastic and deterministic models for the long term hydropower scheduling of the interconnected brazilian system. M. Zambelli et al., 2011.
- Dynamic Programming and Suboptimal Control: A Survey from ADP to MPC. D. Bertsekas, 2005. (MPC = deterministic forecasts)
- Astrom 1965
- Renewable energy forecasts ought to be probabilistic! P. Pinson, 2013 (WIPFOR talk)
- Training a neural network with a financial criterion rather than a prediction criterion Y. Bengio, 1997 (quite practical application of direct policy search, convincing experiments)

71

Thank you for your attention

Questions ?



Hybridization of dynamic optimization methodologies
References I

- S. Astete-Morales, J. Liu, and O. Teytaud, *log-log convergence for noisy optimization*, Proceedings of EA 2013, LLNCS, Springer, 2013, p. accepted.
- alexander shapiro, Analysis of stochastic dual dynamic programming method, European Journal of Operational Research 209 (2011), no. 1, 63–72.
- R. Bellman, *Dynamic programming*, Princeton Univ. Press, 1957.
- Solution Sensition of the sensition of t

73

I-Lab: LRI - Inria & Artelvs

References II

- Dimitri P. Bertsekas, Dynamic programming and suboptimal control: A survey from ADP to MPC, Eur. J. Control 11 (2005), no. 4-5, 310–334.
- H.-G. Beyer, *The theory of evolutions strategies*, Springer, Heidelberg, 2001.
- D.P. Bertsekas and J.N. Tsitsiklis, *Neuro-dynamic programming*, Athena Scientific, 1996.
- Zhihao Cen, J. Frederic Bonnans, and Thibault Christel, Energy contracts management by stochastic programming techniques, Annals of Operations Research 200 (2011), no. 1, 199–222 (Anglais), RR-7289 RR-7289.

References III

- Gérard Cornuéjols, *Revival of the gomory cuts in the 1990's*, Annals OR **149** (2007), no. 1, 63–66.
- Z. L. Chen and W. B. Powell, Convergent cutting-plane and partial-sampling algorithm for multistage stochastic linear programs with recourse, J. Optim. Theory Appl. 102 (1999), no. 3, 497–524.
- Christopher J. Donohue and John R. Birge, The abridged nested decomposition method for multistage stochastic linear programs with relatively complete recourse, Algorithmic Operations Research 1 (2006), no. 1.

75

I-Lab: LRI - Inria & Artelvs

References IV

- Vaclav Fabian, Stochastic approximation of minima with improved asymptotic speed, Ann. Math. Statist. 38 (1967), no. 1, 191–200.
- S. Hong, P.O. Malaterre, G. Belaud, and C. Dejean, *Optimization of irrigation scheduling for complex water distribution using mixed integer quadratic programming (MIQP)*, HIC 2012 – 10th International Conference on Hydroinformatics (Hamburg, Allemagne), IAHR, 2012, pp. p. – p. (Anglais).

76

I-Lab: LRI - Inria & Artelvs

References V

- Verena Heidrich-Meisner and Christian Igel, Hoeffding and bernstein races for selecting policies in evolutionary direct policy search, ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning (New York, NY, USA), ACM, 2009, pp. 401–408.
- Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es), Evolutionary Computation 11 (2003), no. 1, 1–18.
- Holger Heitsch and Werner Römisch, *Scenario tree reduction for multistage stochastic programs*, Computational Management Science **6** (2009), no. 2, 117–133.

References VI

- L. G. Khachiyan, A polynomial algorithm in linear programming, Doklady Akademii Nauk SSSR 244 (1979), 1093–1096.
- K. Linowsky and A. B. Philpott, On the convergence of sampling-based decomposition algorithms for multistage stochastic programs, J. Optim. Theory Appl. 125 (2005), no. 2, 349–366.
- A.B. Philpott and Z. Guan, On the convergence of stochastic dual dynamic programming and related methods, Operations Research Letters 36 (2008), no. 4, 450 – 455.
- P. Pinson, *Renewable energy forecasts ought to be probabilistic*, 2013, WIPFOR seminar, EDF.

78

References VII

- W.-B. Powell, *Approximate dynamic programming*, Wiley, 2007.
- M. V. F. Pereira and L. M. V. G. Pinto, *Multi-stage stochastic optimization applied to energy planning*, Math. Program. 52 (1991), no. 2, 359–375.
- A. Ruszczynski, vol. 10, ch. Decomposition methods, North-Holland Publishing Company, Amsterdam, 2003.
- S. Uryasev and R.T. Rockafellar, Optimization of conditional value-at-risk, Research report (University of Florida. Dept. of Industrial & Systems Engineering), Department of Industrial & Systems Engineering, University of Florida, 1999.

I-Lab: LRI - Inria & Artelvs

Parametric policies



 $z = W_0 + W_1 \tanh(W_2 x + W_3)$

80

I-Lab: LRI - Inria & Artelys

Decock

Parametric policies

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \ \mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \end{pmatrix}, \ \mathbf{W}_3 = \begin{pmatrix} w_{10}^{(1)} \\ w_{20}^{(1)} \\ \vdots \\ w_{m1}^{(1)} \end{pmatrix}, \ \mathbf{W}_0 = \begin{pmatrix} w_{10}^{(2)} \\ w_{20}^{(2)} \\ \vdots \\ w_{20}^{(2)} \end{pmatrix},$$
$$\mathbf{W}_2 = \begin{pmatrix} w_{11}^{(1)} & w_{12}^{(1)} & \cdots & w_{1n}^{(1)} \\ w_{11}^{(1)} & w_{12}^{(1)} & \cdots & w_{1n}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & \cdots & w_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1}^{(1)} & w_{m2}^{(1)} & \cdots & w_{mn}^{(1)} \end{pmatrix}, \ \mathbf{W}_1 = \begin{pmatrix} w_{11}^{(2)} & w_{12}^{(2)} & \cdots & w_{1m}^{(2)} \\ w_{21}^{(2)} & w_{22}^{(2)} & \cdots & w_{2m}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1}^{(1)} & w_{m2}^{(1)} & \cdots & w_{mn}^{(1)} \end{pmatrix}$$

 $z = W_0 + W_1 \tanh(W_2 x + W_3)$

I-Lab: LRI - Inria & Artelys

81

.

Self-adaptive Evolution Strategy (SA-ES) with revaluations

Require:

 $K > 0, \lambda > \mu > 0$, a dimension $d > 0, \tau$ (usually $\tau = \frac{1}{\sqrt{2d}}$).

Initialize parent population $P_{\mu} = \{(\mathbf{x}_1, \sigma_1), (\mathbf{x}_2, \sigma_2), \dots, (\mathbf{x}_{\mu}, \sigma_{\mu})\}$ with $\forall i \in \{1, \dots, \mu\}$, $\mathbf{x}_i \in \mathbb{R}^d$ and $\sigma_i = 1$.

while stop condition do

Generate the offspring population $P_{\lambda} = \{(\mathbf{x}'_1, \sigma'_1), (\mathbf{x}'_2, \sigma'_2), \dots, (\mathbf{x}'_{\lambda}, \sigma'_{\lambda})\}$ where each individual is generated by:

- 1. Select (randomly) ρ parents from P_{μ} .
- 2. Recombine the ρ selected parents to form a recombinant individual (\mathbf{x}', σ') .

82

EL OQO

I-Lab: LRI - Inria & Artelys

(日) (同) (三) (三)

- 3. Mutate the strategy parameter: $\sigma' \leftarrow \sigma' e^{\tau \mathcal{N}(0,1)}$.
- 4. Mutate the objective parameter: $\mathbf{x}' \leftarrow \mathbf{x}' + \sigma' \mathcal{N}(\mathbf{0}, \mathbf{1})$.

Select the new parent population P_{μ} taking the μ best form $P_{\lambda} \cup P_{\mu}$. end while

Fabian

- 1: Input: an initial $x_1 = 0 \in \mathbb{R}^d$, $\frac{1}{2} > \gamma > 0$, a > 0, c > 0, $m \in \mathbb{N}$, weights $w_1 > \cdots > w_m$ summing to 1, scales $1 \ge u_1 > \cdots > u_m > 0$. 2: $n \leftarrow 1$
- 3: while (true) do
- 4: Compute $\sigma_n = c/n^{\gamma}$.
- 5: Evaluate the gradient g at x_n by finite differences, averaging over 2m samples per axis:

$$\begin{aligned} \forall i, j \in \{1, \dots, d\} \times \{1 \dots m\}, \ x_n^{(i,j)+} &= x_n + u_j e_i, \\ \forall i, j \in \{1, \dots, d\} \times \{1 \dots m\}, \ x_n^{(i,j)-} &= x_n - u_j e_i, \\ \forall i \in \{1, \dots, d\}, g^{(i)} &= \frac{1}{2\sigma_n} \sum_{j=1}^m w_j \left(f(x_n^{(i,j)+}) - f(x_n^{(i,j)-}) \right). \end{aligned}$$

- 6: Apply $x_{n+1} \leftarrow x_n \frac{a}{n}g$ 7: $n \leftarrow n+1$
- 8: end while

 Fabian's stochastic gradient algorithm with finite differences. Several variants have

 been defined, in particular versions in which only one point (or a constant number of

 points, independently of the dimension) is evaluated at each iteration.

 Decock

Lemmas - Introduction (1)

The objective function

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

and the locality assumption

$$\forall f \in F, P\left(\forall i \leq n, \underbrace{\|\mathbf{x}_i - \mathbf{x}^*\| \leq \frac{C(d)}{i^{\alpha}}}_{i^{\alpha}}\right) \geq 1 - \frac{\delta}{2}$$

imply

Decock

$$\underbrace{\mathbb{E}f(\mathbf{x}^*)}_{1-\gamma} \leq \overline{\mathbb{E}f(\mathbf{x}_n)} \leq \underbrace{\mathbb{E}f(\mathbf{x}^*)}_{1-\gamma} + \frac{\gamma}{d^{\beta/2}} \frac{C(d)^{\beta}}{n^{\alpha\beta}}$$

with probability at least $1-\delta/2$

(f and $\mathbb{E}f(\mathbf{x})$ are short notations for $f_{\mathbf{x}^*,\beta,\gamma}$ and $\mathbb{E}_{\boldsymbol{\omega}}f(\mathbf{x},\boldsymbol{\omega})$)

84

Decock

Lemmas - Introduction (2)

$$\begin{aligned} \|\mathbf{x}_{i} - \mathbf{x}^{*}\| &\leq \frac{C(d)}{i^{\alpha}} \quad \text{(the locality assumption)} \\ \Leftrightarrow & \underbrace{\gamma\left(\frac{\|\mathbf{x}_{i} - \mathbf{x}^{*}\|}{\sqrt{d}}\right)^{\beta} + (1 - \gamma)}_{\langle \sqrt{d}} \leq \underbrace{(1 - \gamma)}_{i^{\alpha}\sqrt{d}} + \gamma\left(\frac{C(d)}{i^{\alpha}\sqrt{d}}\right)^{\beta} \\ \Leftrightarrow & \mathbb{E}f(\mathbf{x}_{i}) \leq \mathbb{E}f(\mathbf{x}^{*}) + \gamma\left(\frac{C(d)}{i^{\alpha}\sqrt{d}}\right)^{\beta} \\ \Leftrightarrow & \mathbb{E}f(\mathbf{x}_{i}) \leq \mathbb{E}f(\mathbf{x}^{*}) + \frac{\gamma C(d)^{\beta}}{i^{\alpha\beta}\sqrt{d}^{\beta}} \end{aligned}$$

I-Lab: LRI - Inria & Artelys

ъ

85

Proof

Consider $f_{\mathbf{x}^*} = f_{\mathbf{x}^*,\beta,\gamma}$ with \mathbf{x}^* uniformly distributed in R. Then:

$$P(\|\mathbf{x}_{n} - \mathbf{x}^{*}\| \leq \epsilon)$$

$$\leq \mathbb{E}_{\Omega} P_{\mathbf{x}^{*}} (\mathbf{x}^{*} \in \mathsf{Enl}(X_{n,\Omega}, \epsilon))$$

$$\leq P(\#X_{n,\Omega} \leq C) P_{\mathbf{x}^{*}} (\mathbf{x}^{*} \in \mathsf{Enl}(X_{n,\Omega}, \epsilon) | \#X_{n,\Omega} \leq C)$$

$$+P(\#X_{n,\Omega} > C)$$

$$\leq (1 - \frac{\delta}{2}) \frac{C}{C'} + \frac{\delta}{2}$$

$$< 1 - \frac{\delta}{2}$$

where $Enl(U, \epsilon)$ is the ϵ -enlargement of U defined as:

$$\mathsf{Enl}(U,\epsilon) = \left\{ \mathbf{x}; \exists \mathbf{x}' \in U, \|\mathbf{x} - \mathbf{x}'\| \le \epsilon \right\}.$$

Image: A math a math

86

I-Lab: LRI - Inria & Artelvs

This contradicts the locality assumption.

This concludes the proof of $\alpha\beta \leq 1$.

Decock

R-EDA (1)

Algorithm 2 R-EDA: algorithm for optimizing noisy fitness functions. Bernstein denotes a Bernstein race, as defined in Algorithm 3. The initial domain is $[x_0^-, x_0^+] \in \mathbb{R}^D$, δ is the confidence parameter. This algorithm goes back to [15, 16].

 $n \leftarrow 0$ while True do // Pick the coordinate with highest uncertainty $c_n = \arg\max_i (x_n^+)_i - (x_n^-)_i$ $\delta_n^{\max} = (x_n^+)_{c_n} - (x_n^-)_{c_n}$ for $i \in [1, 3]$ do // Consider the middle point $x'_n^i \leftarrow \frac{1}{2}(x_n^- + x_n^+)$ // The c_n^{th} coordinate may take $3 \neq$ values $(x'_{n}^{i})_{c_{n}} \leftarrow (x_{n}^{-})_{c_{n}} + \frac{i-1}{2}(x_{n}^{+} - x_{n}^{-})_{c_{n}}$ end for $(good_n, bad_n) = Bernstein(x'_n^1, x'_n^2, x'_n^3, \frac{6\delta}{\pi^2(n+1)^2}).$ // A good and a bad point Let H_n be the halfspace $\{x \in \mathbb{R}^D; ||x - qood_n|| \le ||x - bad_n||\}$ Split the domain: $[x_{n+1}^{-}, x_{n+1}^{+}] = H_n \cap [x_n^{-}, x_n^{+}]$ $n \leftarrow n + 1$ end while ∃ ► ▲ ∃ ► ∃ =

87

I-Lab: LRI - Inria & Artelvs

Decock

R-EDA(2)

Algorithm 3 Bernstein race between 3 points. Eq. 3 is Bernstein's inequality to compute the precision for empirical estimates (see e.g. [18, p124]); $\hat{\sigma}_i$ is the empirical estimate of the standard deviation of point x_i 's associated random variable $F_t(x_i)$ (it is 0 in the first iteration, which does not alter the algorithm's correctness); $\hat{f}(x)$ is the average of the fitness measurements at x.

 $Bernstein(x_1, x_2, x_3, \delta')$ T = 0

repeat

 $T \leftarrow T + 1$

Evaluate the fitness of points x_1, x_2, x_3 once, *i.e.* evaluate the noisy fitness at each of these points. Evaluate the precision:

$$\epsilon_{(T)} = 3 \log\left(\frac{3\pi^2 T^2}{6\delta'}\right) / T + \max_i \hat{\sigma}_i \sqrt{2 \log\left(\frac{3\pi^2 T^2}{6\delta'}\right) / T}.$$
(3)

until Two points (good ,bad) satisfy $f(bad) - f(good) > 2\epsilon$ - return (good, bad)

> < ロ > < 同 > < 三 > < 三 I-Lab: LRI - Inria & Artelvs

88

= 200

Decock

R-EDA(3)



Fig. 1. Noisy optimization algorithm (cf Algorithm 2). At each iteration, a main axis is selected (the one on which the domain has maximum range). Three equally spaced points are generated in the domain on this axis (this is the offspring). Then, a Bernstein race is applied for choosing a "good" and a "bad" arm among these points. The domain is reduced thanks to this knowledge, removing one fourth or one half of the domain (depending on the position of the good arm and of the bad arm - the best case is when the good and the bad arm are diametrically opposed: see Fig. 2).

Decock

89

ELE DOG

I-Lab: LRI - Inria & Artelvs

Image: A math a math

Decock

Theorem's proof Step 2

$$(??) \iff \ln\left(\frac{f(\mathbf{x})}{f(\mathbf{x}')}\right) \ge \max\left(\ln\left(\frac{1}{\eta}\right), \ln\left(\frac{K'}{K''}\right) + \ln\left(\frac{||\mathbf{x}||}{||\mathbf{x}'||}\right)\right)$$

$$f(\mathbf{x}') \le \eta f(\mathbf{x}) \iff \frac{1}{\eta} f(\mathbf{x}') \le f(\mathbf{x})$$

$$\Leftrightarrow \frac{1}{\eta} \le \frac{f(\mathbf{x})}{f(\mathbf{x}')}$$

$$\Leftrightarrow \ln\left(\frac{f(\mathbf{x})}{f(\mathbf{x}')}\right) \ge \ln\left(\frac{1}{\eta}\right)$$

$$f(\mathbf{x}') \le \frac{K''}{K'} \frac{||\mathbf{x}'||}{||\mathbf{x}||} f(\mathbf{x}) \iff f(\mathbf{x}') \frac{K'}{K''} \frac{||\mathbf{x}||}{||\mathbf{x}'||} \le f(\mathbf{x})$$

$$\Leftrightarrow \frac{K'}{K''} \frac{||\mathbf{x}||}{||\mathbf{x}'||} \le \frac{f(\mathbf{x})}{f(\mathbf{x}')}$$

$$\Leftrightarrow \ln\left(\frac{K'}{K''} \frac{||\mathbf{x}||}{||\mathbf{x}'||}\right) \le \ln\left(\frac{f(\mathbf{x})}{f(\mathbf{x}')}\right)$$

$$\Leftrightarrow \ln\left(\frac{f(\mathbf{x})}{f(\mathbf{x}')}\right) \ge \ln\left(\frac{K'}{K''}\right) + \ln\left(\frac{||\mathbf{x}||}{||\mathbf{x}'||}\right)$$

90

I-Lab: LRI - Inria & Artelys

ъ

Theorem's proof

Step 2 if $l' \ge \ln(c')$ then $l < \ln(c)$ (otherwise it couldn't be a "win")

$$\begin{aligned} & ' - \ln(c') & \leq & \ln(c + \max_{i} ||\delta_{i}||) - \ln(c') \\ & \leq & \ln(c) + \ln(1 + \max_{i} ||\delta_{i}||/c) - \ln(c') \\ & \leq & \ln(c/c') + \max_{i} ||\delta_{i}||/c \\ & \leq & \max_{i} ||\delta_{i}||/c \end{aligned}$$

$$l' - \ln(c') = \ln(\frac{||x + \sigma \delta_i||}{\sigma}) - \ln(c') \leq \ln(\frac{||x||}{\sigma} + \delta_i) - \ln(c')$$

$$\leq \ln(c * (1 + \frac{\delta_i}{c})) - \ln(c')$$

$$\leq \ln(c) + \ln(1 + \max_i ||\delta_i||/c) - \ln(c')$$

$$\leq \ln(c/c') + \max_i ||\delta_i||/c$$

$$\leq \max_i ||\delta_i||/c$$

I-Lab: LRI - Inria & Artelys

91

Decock

Theorem's Proof Step 3

Summing iterations

Hence if $t > n_0$,

$$\begin{split} \ln(f(\mathcal{X}_{t})) &\leq \ln(f(\mathcal{X}_{1})) - (t - n_{0}) \times \sum_{i} \frac{\max\left(\ln\left(\frac{1}{\eta}\right), \ln\left(\frac{K'}{K''}\right) + \Delta\right)}{\min\left(1 + \ln\left(\frac{c}{b}\right)\frac{\Delta}{\ln(2)}, 1 + \ln\left(\frac{c}{b}\right)\frac{\max_{j}\ln(||\delta_{j}||)}{\ln(2)}\right)} \\ &\iff \quad \ln(f(\mathcal{X}_{t})) - \ln(f(\mathcal{X}_{1})) \leq -(t - n_{0}) \times C \\ &\iff \quad \frac{\ln\left(\frac{f(\mathcal{X}_{t})}{f(\mathcal{X}_{1})}\right)}{t - n_{0}} \leq -C \\ &\Rightarrow \quad \frac{\ln(||\mathcal{X}_{t}||)}{t} \leq K < 0 \quad \text{(theorem)} \end{split}$$

with C a positive constant.

Decock

Corollary's proof Step 2 (1/4)

Step 2: showing that σ small leads to high acceptance rate and σ high leads to small acceptance rate.

Thanks to the bounded conditioning, there exists $\epsilon > 0$ s.t.

$$s' < \frac{1}{2}s$$
with $s = \sup\left\{\frac{\sigma}{||\mathbf{x}||}; \sigma, \mathbf{x}, f \text{ s.t. } p \ge \frac{\epsilon}{2}\right\}$
and $s' = \inf\left\{\frac{\sigma}{||\mathbf{x}||}; \sigma, \mathbf{x}, f \text{ s.t. } p < \frac{1}{2} - \frac{\epsilon}{2}\right\}$

because $s' \to 0$ and $s \to \infty$ as $\epsilon \to 0$.

I-Lab: LRI - Inria & Artelys

93

Decock

Corollary's proof Step 2 (2/4)

Notes

$$\hat{\mathbf{s}} = \sup\left\{\frac{\sigma}{||\mathbf{x}||}; \sigma, \mathbf{x}, f \text{ s.t. } \hat{p} \ge \epsilon\right\}$$
$$\hat{\mathbf{s}'} = \inf\left\{\frac{\sigma}{||\mathbf{x}||}; \sigma, \mathbf{x}, f \text{ s.t. } \hat{p} < \frac{1}{2} - \epsilon\right\}$$

Then

$$\sup_{\mathbf{x},f,\sigma>0}|\hat{p}-p|\leq\epsilon/2$$

implies

$$rac{1}{2}\hat{s}\geq rac{1}{2}s$$
 and $s'\geq \hat{s'}$

< • • • • • •

94

ъ

I-Lab: LRI - Inria & Artelys

Decock

Decock

Corollary's proof Step 2 (3/4)

So $\hat{s'} \leq s' < rac{1}{2}s \leq rac{1}{2}\hat{s}$ and $\hat{s'} \leq rac{1}{2}\hat{s}$

I-Lab: LRI - Inria & Artelys

ъ

95

Decock

Corollary's proof Step 2 (4/4)

This provides k_1 , k_2 , c' and b' as follows for Eqs. **??** and **??**:

$$\begin{aligned} \frac{1}{b'} &= \hat{s} &= \sup\left\{\frac{\sigma}{||\mathbf{x}||}; \sigma, \mathbf{x}, f \text{ s.t. } \hat{p} \ge \epsilon\right\} \\ \frac{1}{c'} &= \hat{s'} &= \inf\left\{\frac{\sigma}{||\mathbf{x}||}; \sigma, \mathbf{x}, f \text{ s.t. } \hat{p} < \frac{1}{2} - \epsilon\right\} \\ k_1 &= \lfloor \epsilon k \rfloor \\ k_2 &= \left\lceil (\frac{1}{2} - \epsilon) k \right\rceil \end{aligned}$$

Eqs. above imply $c' \ge 2b'$

96

Corollary's proof Step 3

k large enough yield

$$b^{-1} = \sup \left\{ \frac{\sigma}{||\mathbf{x}||}; \sigma, \mathbf{x}, f \text{ s.t. } \hat{p} > k_1/k \right\},$$

$$c^{-1} = \inf \left\{ \frac{\sigma}{||\mathbf{x}||}; \sigma, \mathbf{x}, f \text{ s.t. } \hat{p} < k_2/k \right\},$$

97

I-Lab: LRI - Inria & Artelvs

which provide assumptions 2 and 5 with b < cAssumptions 2 and 5 then imply b < b' and c' < c

Non quasi-convex functions





Decoc<u>k</u>