???

LOBACHEVSKY UNIVERSITY

Comparison of Several Stochastic and Deterministic Derivative-free Global Optimization Algorithms

Vladislav Sovrasov

Lobachevsky State University of Nizhni Novgorod

9 July 2019
Ekaterinburg, Russia

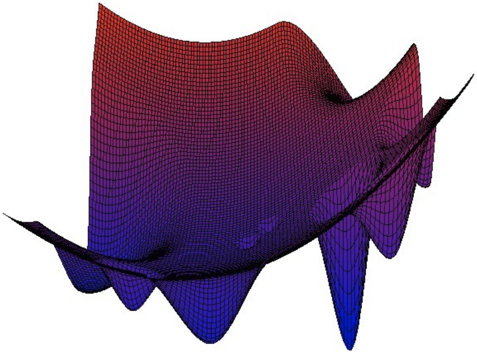
Problem statement

$$\varphi(y^*) = \min\{\varphi(y) : y \in D\},$$
$$D = \{y \in \mathbb{R}^N : a_i \leq y_i \leq b_i, 1 \leq i \leq N\}$$

$\varphi(y)$ is multiextremal objective function, which satisfies the Lipschitz condition:

$$|\varphi(y_1) - \varphi(y_2)| \leq L\|y_1 - y_2\|, y_1, y_2 \in D,$$

where $L > 0$ is the Lipschitz constant, and $||\cdot||$ denotes l_2 norm in $\mathbb{R}^N$ space.



Methods

- **Deteriminstic**:
  - Have complicated internal structure in multidimensional case;
  - Usually store and use the whole history of trials accumulated during search;
  - Under some assumptions convergence to the global solution is guaranteed.
- **Stochastic**:
  - Have relative simple internal structure;
  - Require constant amount of memory to store internal state of some random process or individuals of population;
  - Convergence is guaranteed in probabilistic sense only.

Methods

In this work considered the following solvers available in open-source:

- **Deteriminstic**:
  - DIRECT, DIRECTl;
  - AGS, AGSl.
- **Stochastic**:
  - Multi Level Single Linkage;
  - StoGO;
  - Differential Evolution;
  - Controlled Random Search;
  - Dual Simulated Annealing;
  - Ant Colony Optimization.

Univariate Algorithm of Global Search

Optimization method generates search sequence $\{x_k\}$ and consists of the following steps:

Step 1. Sort the search information (one-dimensional points) in increasing order.

Step 2. For each interval (x_{i-1}, x_i) compute quantity $R(i)$, called characteristic.

Step 3. Choose p intervals (x_{t_j-1}, x_{t_j}) with the greatest characteristics and compute objective $\varphi(y(x^{k+j}))$ in points chosen using the decision rule d:

$$x^{k+1+j} = d(t) \in (x_{t_j-1}, x_{t_j}),\ j = \overline{1, p}$$

Step 4. If $x_{t_j} - x_{t_j-1} < \varepsilon$ for one of $j = \overline{1, p}$, stop the method.

Detailed description: Strongin R.G., Sergeyev Ya.D.: Global optimization with non-convex constraints. Sequential and parallel algorithms (2000), Chapter 7

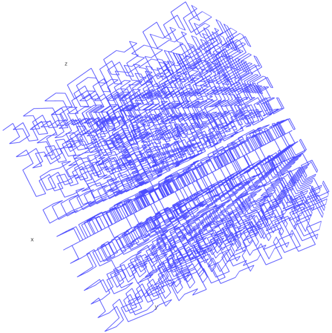
Dimension reduction

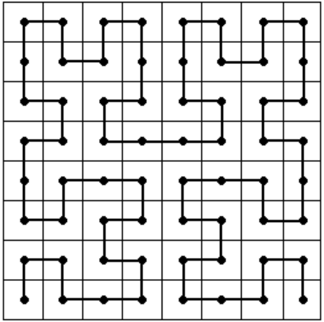
Peano-type curve $y(x)$ allows to reduce the dimension of the original problem:

$$\{y \in \mathbb{R}^N : -2^{-1} \leqslant y_i \leqslant 2^{-1}, 1 \leqslant i \leqslant N\} = \{y(x) : 0 \leqslant x \leqslant 1\}$$

$$\min\{\varphi(y) : y \in D\} = \min\{\varphi(y(x)) : x \in [0,1]\}$$

$y(x)$ is non-smooth function which continuously maps the segment $[0,1]$ to the hypercube D.





Hyperparameters tuning in AGS

AGS selects the best intervals according to the following characteristic:

$$R(i) = \Delta_i + \frac{(z_i - z_{i-1})^2}{(rH)^2\Delta_i} - 2\frac{z_i + z_{i-1}}{rH}, 1 < i < k+1$$

where L is current estimation on Holder constant, r is reliability coefficient. AGS is very sensitive to choice of r:

- low value can lead to fast convergence to a local minima
- high value leads to very slow convergence

Hyperparameters tuning in AGS

In order to resolve the problem of choosing r to some extent, let us use the following scheme:

- execute q iterations of AGS with $r = r_{max}$;
- execute q iterations of AGS with $r = r_{min}$;
- repeat the above steps either until convergence or until the allowed number of iterations are exhausted.

Now we have 3 parameters instead of one, but the method is not so sensitive to them. Let $q = 50 \cdot \log_2(N-1) \cdot N^2$, $r_{min} = 3$, $r_{max} = 2 \cdot r_{min}$ in all the future experiments.

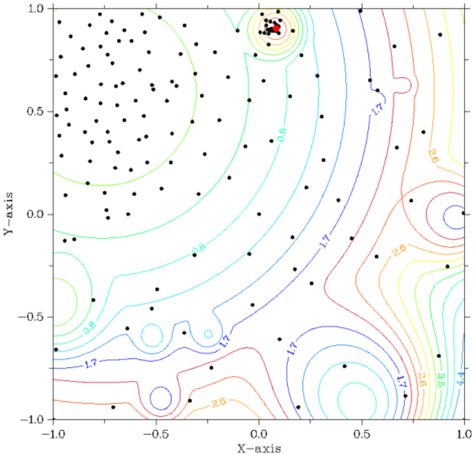
Test problems

Generator GKLS was employed to construct 8 sets of 100 test problems:

$$f(x) = \begin{cases} C_i(x), x \in S_i, i \in 2, \dots, m \\ \|x - T\|^2 + t, x \notin S_2, \dots, S_m \end{cases}$$

The generator allows to adjust:

- the number of local minimums;
- the size of the global minima attraction region;
- the space dimension.



Also one set of 100 pre-defined highly multiextremal functions (with 10-30 local minimums) was used.

Experimental setup

Table: Trials limits and relative precision for the test problem classes

Problems class	Trials limit	α
F_{GR}	5000	0.01
GKLS 2d Simple	8000	0.01
GKLS 2d Hard	9000	0.01
GKLS 3d Simple	15000	0.01
GKLS 3d Hard	25000	0.01
GKLS 4d Simple	150000	$\sqrt[4]{10^{-6}}$
GKLS 4d Hard	250000	$\sqrt[4]{10^{-6}}$
GKLS 5d Simple	350000	$\sqrt[5]{10^{-7}}$
GKLS 5d Hard	600000	$\sqrt[5]{10^{-7}}$

Experimental setup

Table: Class-specific parameters of the optimization algorithms

	AGS	CRS	DE
F_{GR}	$r = 3$	popsize=150	mutation=(1.1,1.9), popsize=60
GKLS 2d Simple	$r = 4.6$	popsize=200	mutation=(1.1,1.9), popsize=60
GKLS 2d Hard	$r = 6.5$	popsize=400	mutation=(1.1,1.9), popsize=60
GKLS 3d Simple	$r = 3.7$	popsize=1000	mutation=(1.1,1.9), popsize=70
GKLS 3d Hard	$r = 4.4$	popsize=2000	mutation=(1.1,1.9), popsize=80
GKLS 4d Simple	$r = 4.7$	popsize=8000	mutation=(1.1,1.9), popsize=90
GKLS 4d Hard	$r = 4.9$	popsize=16000	mutation=(1.1,1.9), popsize=100
GKLS 5d Simple	$r = 4$	popsize=25000	mutation=(1.1,1.9), popsize=120
GKLS 5d Hard	$r = 4$	popsize=30000	mutation=(1.1,1.9), popsize=140

- in the DIRECT and DIRECTl methods, the parameter $\epsilon = 10^{-4}$;
- in the SDA method, the parameter $visit = 2.72$.

Results of different methods

Table: Averaged number of trials executed by the methods for solving the test optimization problems

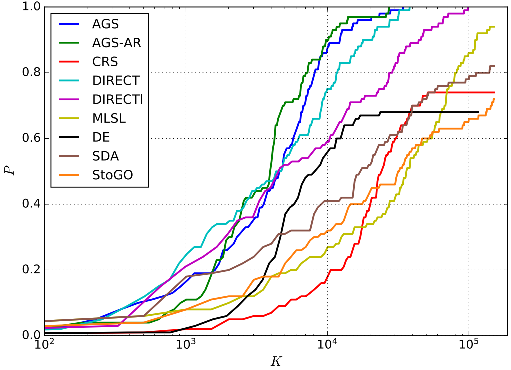
	AGS	AGS-AR	CRS	DIRECT	DIRECTl	MLSL	SDA	DE	StoGO
F_{GR}	193.1	248.3	400.3	**182.2**	214.9	947.2	691.2	1257.3	1336.8
GKLS 2d Simple	254.9	221.6	510.6	**189.0**	255.2	556.8	356.3	952.2	1251.5
GKLS 2d Hard	**728.7**	785.0	844.7	985.4	1126.7	1042.5	1637.9	1041.1	2532.2
GKLS 3d Simple	1372.1	1169.5	4145.8	**973.6**	1477.8	4609.2	2706.5	5956.9	3856.1
GKLS 3d Hard	3636.1	**1952.1**	6787.0	2298.7	3553.3	5640.1	4708.4	6914.3	7843.2
GKLS 4d Simple	5729.8	**4919.1**	19883.6	7328.8	15010.0	41484.8	22066.0	6271.2	29359.2
GKLS 4d Hard	13113.4	**12860.1**	27137.4	22884.4	55596.1	80220.1	68048.0	12487.6	58925.5
GKLS 5d Simple	**5821.5**	6241.3	62921.7	5966.1	10795.5	52609.2	34208.8	20859.4	69206.8
GKLS 5d Hard	**17008.6**	21555.1	87563.9	61657.3	148637.8	138011.8	115634.6	26850.0	141886.5

Results of different methods

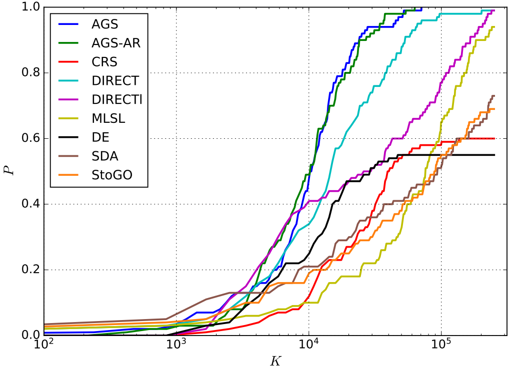
Table: Number of test optimization problems solved by the methods

	AGS	AGS-AR	CRS	DIRECT	DIRECTl	MLSL	SDA	DE	StoGO
F_{GR}	100	100	76	100	100	97	96	96	67
GKLS 2d Simple	100	100	85	100	100	100	100	98	90
GKLS 2d Hard	100	97	74	100	100	100	93	85	77
GKLS 3d Simple	100	100	75	100	100	100	89	86	44
GKLS 3d Hard	100	100	72	100	99	100	88	77	43
GKLS 4d Simple	100	100	74	100	100	94	82	68	72
GKLS 4d Hard	100	100	60	99	99	94	73	55	69
GKLS 5d Simple	100	100	86	100	100	98	100	88	82
GKLS 5d Hard	100	100	77	100	93	79	86	77	78

Results of different methods

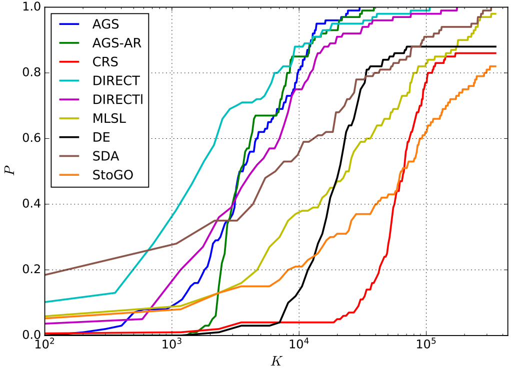


4d Simple

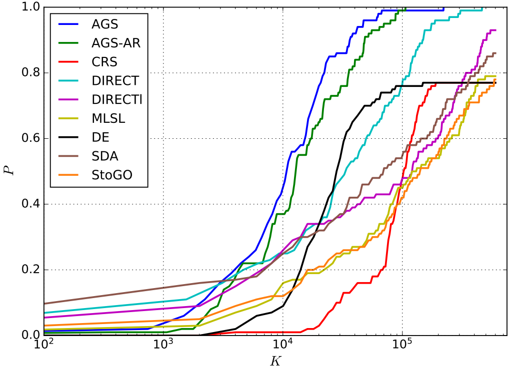


4d Hard

Results of different methods

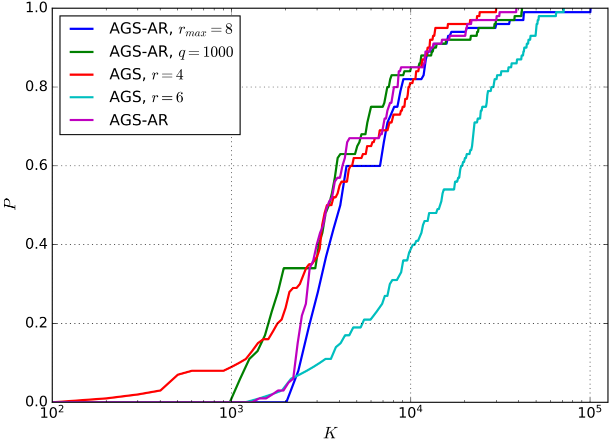


5d Simple



5d Hard

Robustness of the AGS-AR to hyperparameters



Operating characteristics of AGS and AGS-AR with different hyperparameters when solving problems from the GKLS 5d Simple classe.

Conclusions

- The proposed modification of the stock AGS, AGS-AR allows to pay less attention to initial hyperparameter tuning and performs on-par with properly tuned AGS;
- AGS-AR method has demonstrated the convergence speed and reliability at the level of DIRECT and exceeds many other algorithms, the open-source implementations of which are available in web;
- the stochastic optimization methods inferior to the deterministic ones in the convergence speed and in reliability. It is manifested especially strongly on more complex multiextremal problems.

Q&A

Contacts:

sovrasov.vlad@gmail.com
https://github.com/sovrasov