LOBACHEVSKY UNIVERSITY

Comparison of Dimensionality Reduction Schemes for Parallel Global Optimization Algorithms

Konstantin Barkalov, **Vladislav Sovrasov** and **Ilya Lebedev**

Lobachevsky State University of Nizhni Novgorod

25 September 2018
Moscow, Russia

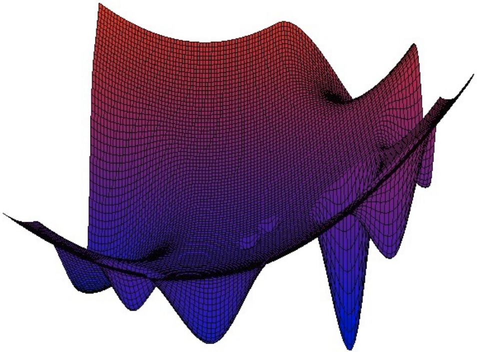
Problem statement

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

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

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

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



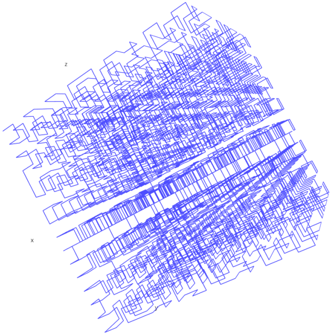
Dimension reduction

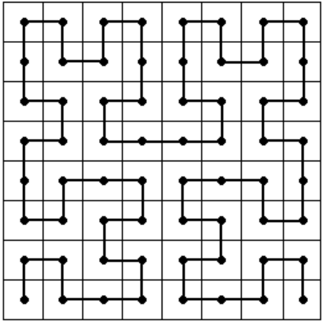
Peano-type curve $y(x)$ allows to reduce the dimension of the original problem:

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

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

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



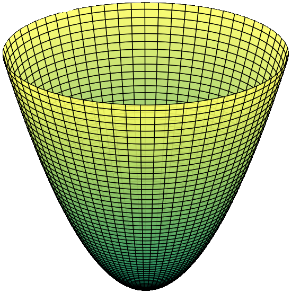


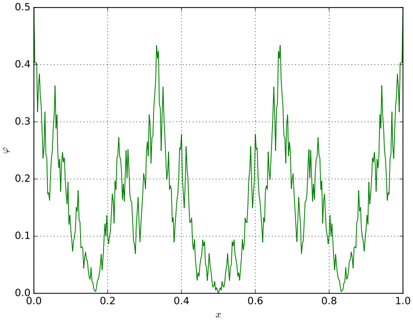
Properties of the reduced problem

After applying the Peano-type evolvent $\varphi(y(x))$ satisfies the uniform Hölder condition:

$$|\varphi(y(x_1)) - \varphi(y(x_2))| \leq H|x_1 - x_2|^{\frac{1}{N}}, x_1, x_2 \in [0, 1],$$

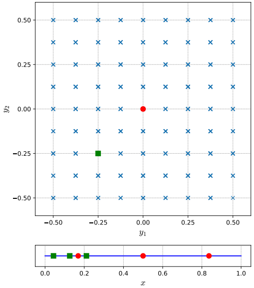
$\varphi(y(x))$ is non-smooth and has multiple local and **global** extremums even if $\varphi(y)$ is unimodal. The latter problem is caused by loss of the information about N-d neighborhood after the transformation to the 1-d space.





Non-univalent evolvent

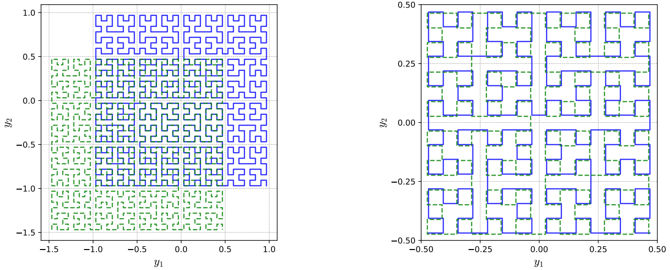
One can try to recover all preimages of $y \in \mathbb{R}^N$ and make optimization method aware of their existence[1]. This allows reducing the effect of growing amount of local minimas after dimension reduction. According to the theory of Peano-type curves, each N-d point could have up to 2^N preimages. For large N such preimages mining would be expensive.



[1] R.G. Strongin. Numerical Methods in Multiextremal Problems (in Russian), 1978

Shifted and rotated evolvents

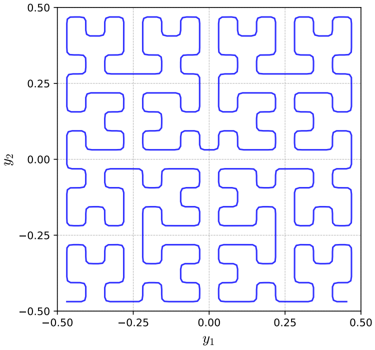
To create a fixed amount of preimages one can use a pre-defined set of different evolvents. These evolvents could be shifted or rotated versions of the original one. Set of shifted evolvents[2] is theoretically proven to generate at least one pair of close preimages if images are close and it perform better than the set of rotated curves.



[2] Strongin, R.G., Gergel, V.P., Barkalov, K.A. Parallel methods for global optimization problem solving (in Russian), 2009

Smooth evolvent

Smooth functions are more predictable for optimizer, so smooth approximation of the Peano-like $y(x)$ curve could improve convergence rate [3].



[3] Goryachih, A. A class of smooth modification of space-filling curves for global optimization problems, NET 2016

Basic parallel optimization method

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. Compute the evolvent $y(x^k)$ and the function $\varphi(y(x^k))$.

Step 3. For each interval (x_{i-1}, x_i) compute quantity $R(i)$, called characteristic.

Step 4. Choose an interval (x_{t-1}, x_t) with the greatest characteristic and compute objective $f(y(x))$ in the point chosen using the decision rule d:

$$x^{k+1} = d(t) \in (x_{t-1}, x_t)$$

Step 5. If $x_t - x_{t-1} < \varepsilon$ stop the method.

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

Parallel optimization method with multiple evolvents

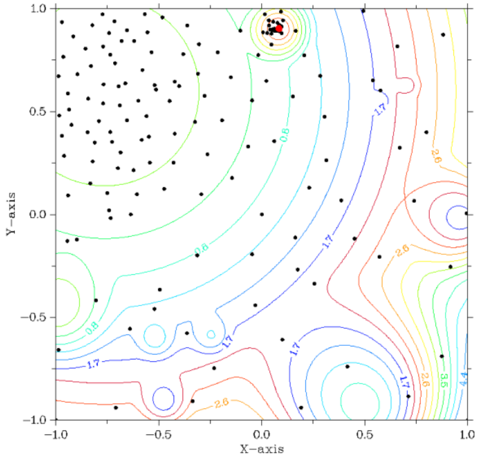
Test problems

Generator GKLS was employed to construct the sets of test problems:

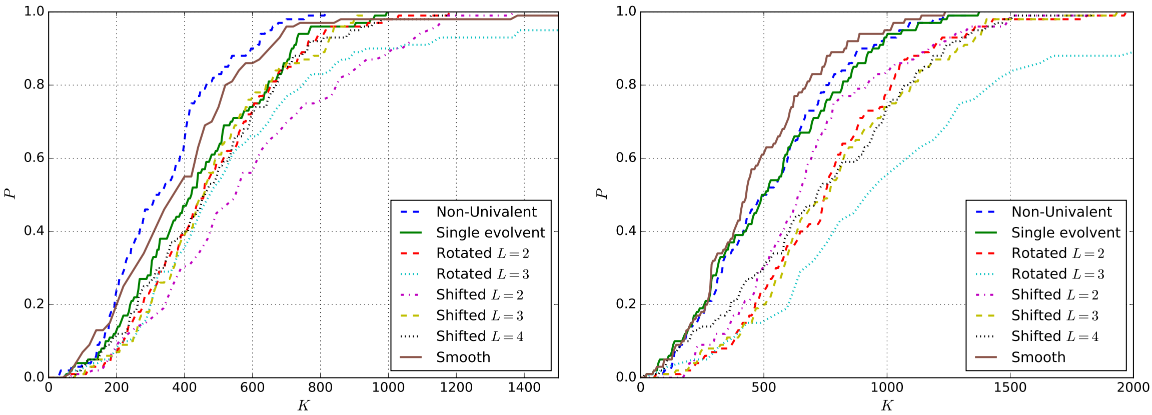
$$f(x) = \begin{cases} C_i(x), x \in S_i, i \in 2, \dots, m \\ \|x - T\|^2 + t, x \notin S_2, \dots, S_m \end{cases}$$

The generator allows to adjust:

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



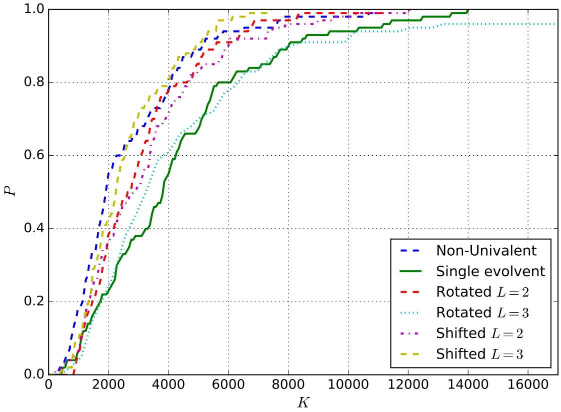
Evolvents comparison



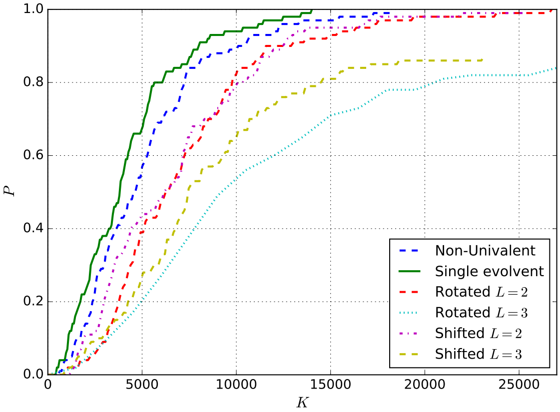
Minimal r $r = 5.0$

Operating characteristics on GKLS 2d Simple class

Evolvents comparison



Minimal r



$r = 4.5$

Operating characteristics on GKLS 3d Simple class

Choice of evolvent for the parallel algorithm

Results of applying the parallel algorithm

Table: Averaged numbers of iterations executed by the parallel algorithm for solving the test optimization problems

		p	$N = 4$		$N = 5$	
			Simple	*Hard*	*Simple*	*Hard*
I	**1 cluster node**	*1*	12167	25635	20979	187353
		32	328	1268	898	12208
II	**4 cluster nodes**	*1*	25312	11103	1472	17009
		32	64	913	47	345
III	**8 cluster nodes**	*1*	810	4351	868	5697
		32	34	112	35	868

Conclusions

- the smooth evolvent and the non-univalent one demonstrate the best result in the problems of small dimensionality and can be applied successfully in solving the problems with the computational costly objective functions.
- the shifted evolvents introduce large overhead costs on the execution of the method due to the requirement to adding an auxiliary constraint. About 95% of iterations are overhead to fight the auxiliary constraint.
- rotated evolvents perform almost the same as the shifted ones but without overhead.

Q&A

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