LOBACHEVSKY UNIVERSITY

Comparison Of Several Sequential And Parallel Derivative-free Global Optimization Algorithms

Vladislav Sovrasov and **Semen Bevzuk**

Lobachevsky State University of Nizhni Novgorod

3 April 2019
Kaliningrad, Russia

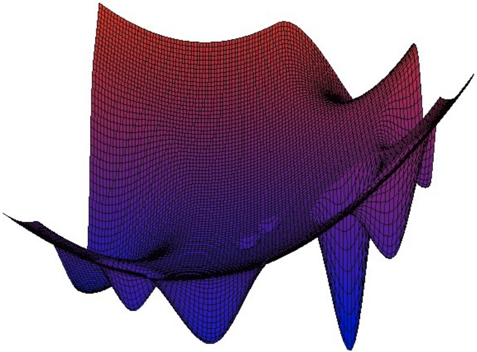
Problem statement

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

$\varphi(y)$ 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;
  - Require non-trivial parallelization schemes;
  - 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;
  - In most cases allows wide-scale parallelization by trials or by running many copies of the same method with different seeds in parallel.
  - 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;

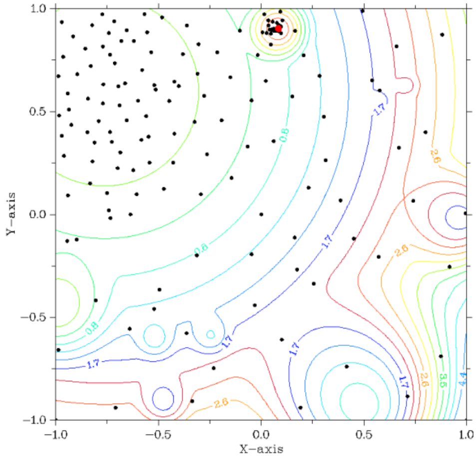
Test problems

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



Results of sequential methods

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

	AGS	AGSl	CRS	DIRECT	DIRECTl	MLSL	SDA	DE	StoGO
F_{GR}	193.1	**158.3**	400.3	182.3	214.9	947.2	691.2	1257.3	1336.8
GKLS 2d Simple	254.9	217.6	510.6	**189.0**	255.2	556.8	356.3	952.2	1251.5
GKLS 2d Hard	728.7	**488.0**	844.7	985.4	1126.7	1042.5	1637.9	1041.1	2532.2
GKLS 3d Simple	1372.1	1195.3	4145.8	**973.6**	1477.8	4609.2	2706.5	5956.94	3856.1
GKLS 3d Hard	3636.1	**1930.5**	6787.0	2298.7	3553.3	5640.1	4708.4	6914.3	7843.2
GKLS 4d Simple	26654.1	11095.7	37436.8	**7824.3**	15994.1	41514.3	21417.9	19157.7	59895.4
GKLS 4d Hard	54536.8	**23167.8**	73779.3	23204.4	54489.9	80247.2	68815.5	27466.1	109328.1
GKLS 5d Simple	29810.0	11529.0	143575.0	**7166.5**	13970.5	52647.6	34255.3	73074.5	91580.4
GKLS 5d Hard	113129.1	67652.7	165192.8	**66327.4**	164390.6	138766.2	116973.1	105496.9	155123.8

Results of sequential methods

Table: Number of test optimization problems solved by sequential methods

	AGS	AGSl	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	100	74	100	100	100	93	85	77
GKLS 3d Simple	100	97	75	100	100	100	89	86	44
GKLS 3d Hard	100	99	72	100	99	100	88	77	43
GKLS 4d Simple	100	100	46	100	100	94	78	59	16
GKLS 4d Hard	100	100	47	99	97	94	72	32	10
GKLS 5d Simple	100	100	68	100	100	98	100	77	9
GKLS 5d Hard	97	99	42	100	90	79	84	48	8

Basic 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. 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

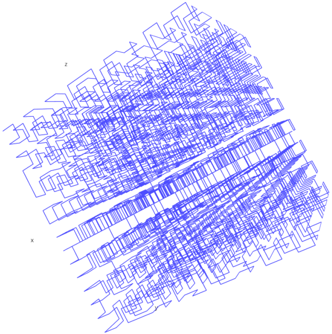
Dimension reduction

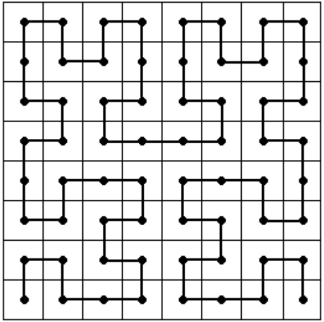
Peano-type curve $y(x)$ allows to reduce the dimension of the original problem:

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

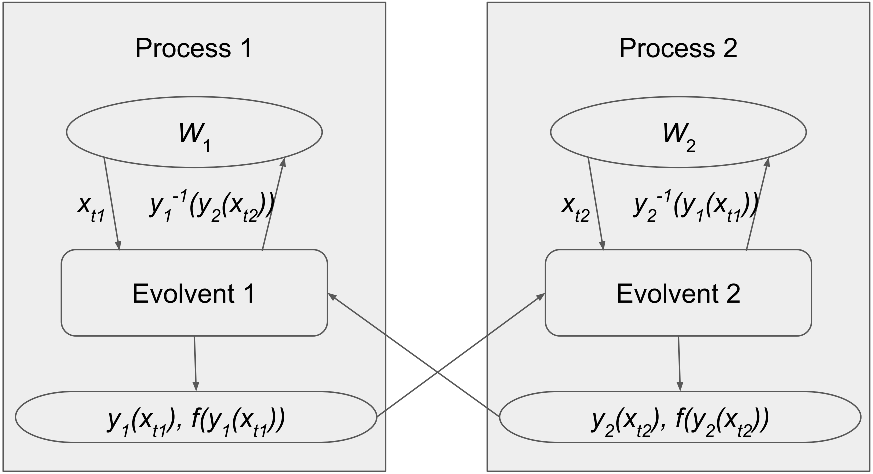
$$\min\{\varphi(y) : y \in D\} = \min\{\varphi(y(x)) : x \in [0,1]\}$$

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





Parallel Globalizer solver



Parallel MIDACO solver

Cluster environment

The computational experiments have been carried out on the Lobachevsky supercomputer at State University of Nizhni Novgorod. One node includes 2 Intel Sandy Bridge E5-2660 2.2 GHz processors and 64 GB RAM.
Each node runs under CentOS 7 Linux with GCC 4.8 compiler and Intel MPI library.

Results of applying the parallel algorithm

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

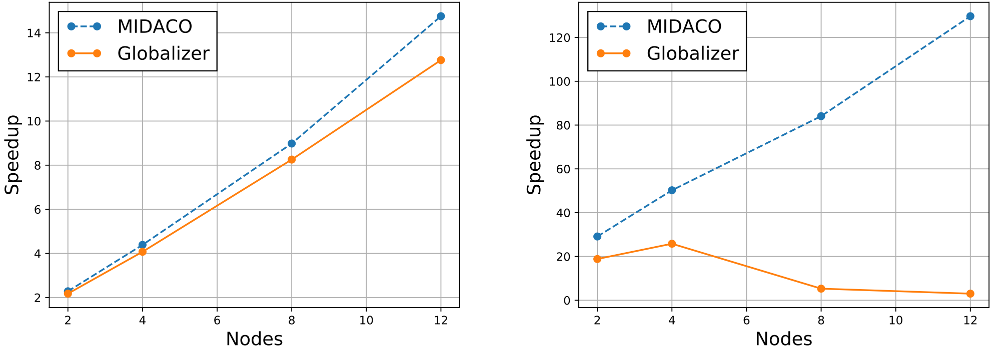
		p	Globalizer		MIDACO	
			Simple	*Hard*	*Simple*	*Hard*
I	**1 cluster node**	*1*	25270 (100)	55180 (99)	27645 (98)	72068 (71)
		16	1765 (100)	3714 (100)	1640 (97)	4304 (70)
II	**2 cluster nodes**	*1*	13056 (100)	22938 (99)	10558 (89)	27128 (73)
		16	732 (100)	1759 (100)	1130 (92)	2254 (73)
III	**4 cluster nodes**	*1*	5016 (100)	12703 (100)	5777 (94)	15980 (75)
		16	367 (100)	776 (100)	529 (88)	1264 (66)
VI	**8 cluster nodes**	*1*	2103 (100)	5063 (100)	2847 (97)	9853 (83)
		16	145 (100)	310 (100)	272 (83)	774 (57)
V	**12 cluster nodes**	*1*	1155 (100)	2399 (100)	2233 (98)	6022 (86)
		16	76 (100)	159 (100)	168 (87)	393 (53)

Results of applying the parallel algorithm

Table: Speedup of parallel computations executed by the parallel algorithms

		p	Globalizer		MIDACO	
			Simple	*Hard*	*Simple*	*Hard*
I	**1 cluster node**	*1*	25270 (27.3s)	55180 (61.8s)	27645 (32.2s)	72068 (132.5s)
		16	14.3(11.7)	14.9(12.6)	16.9(14.4)	16.7(14.4)
II	**2 cluster nodes**	*1*	1.9(1.9)	2.4(2.2)	2.6(1.7)	2.7(2.3)
		16	34.5(20.4)	31.4(18.8)	24.5(18.8)	32.0(29.1)
III	**4 cluster nodes**	*1*	5.0(4.6)	4.3(4.1)	4.8(3.9)	4.5(4.4)
		16	68.8(23.7)	71.2(25.8)	52.3(32.9)	57.0(50.2)
VI	**8 cluster nodes**	*1*	12.0(8.7)	10.9(8.3)	9.7(8.2)	7.3(9.0)
		16	174.1(4.7)	177.8(5.3)	101.6(51.6)	93.1(84.1)
V	**12 cluster nodes**	*1*	21.9(11.4)	23.0(12.8)	12.4(11.1)	12.0(14.8)
		16	333.5(2.7)	347.9(3.0)	164.5(82.0)	183.2(129.7)

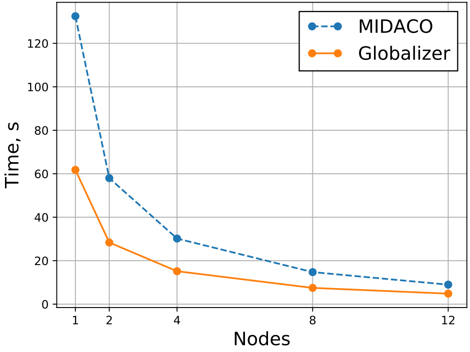
Results of applying the parallel algorithm



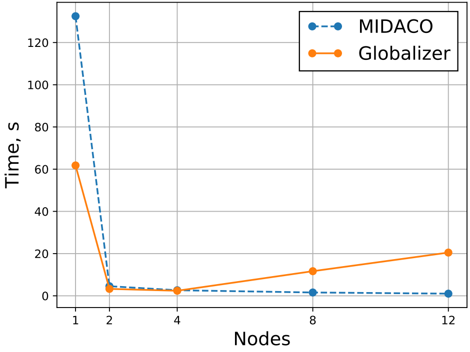
$p = 1$ $\qquad$ $p = 16$

Speedup demonstrated by the parallel algorithms when solving problems from the GKLS 4d Hard class

Results of applying the parallel algorithm



$p = 1$



$p = 16$

Averaged execution time of the parallel algorithms when solving problems from the GKLS 4d Hard class

Conclusions

- 1

Q&A

Contacts:

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