International Scientific Conference "Russian Supercomputing Days"

Lobachevsky State University of Nizhni Novgorod,
Institute of Information Technology, Mathematics and Mechanics

Analysis and elimination of bottlenecks in parallel algorithm for solving global optimization problems

Konstantin Barkalov, Ilya Lebedev, Denis Karchkov

Content

- Global optimization problem statement
- Invers problem of chemical kinetics
- Dimensionality reduction
- Global and local optimization in parallel algorithm
- Results of numerical experiments

Global optimization problem

Problem: to assess the global minimizer y^* of the function $\varphi(y)$

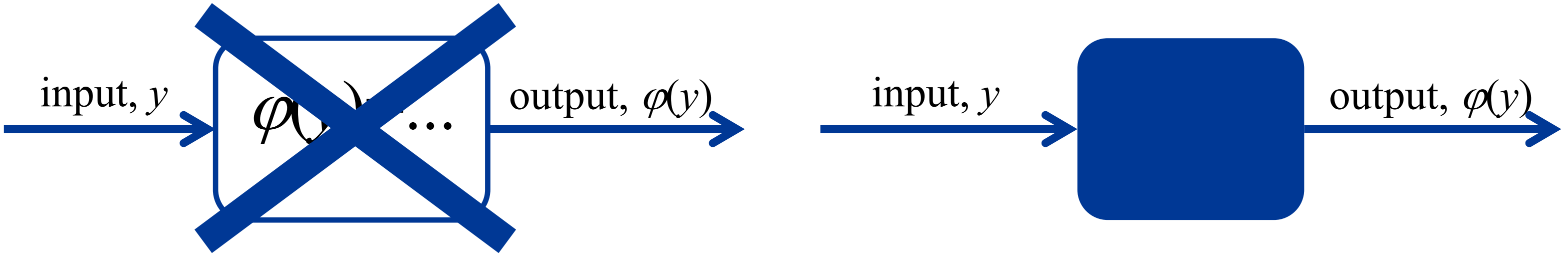
$$\varphi(y^*) = \min\{\varphi(y): y \in D\},$$

$$D = \{y \in R^N: a_i \leq y_i \leq b_i, 1 \leq i \leq N\}.$$

$\varphi(y)$ – objective function, D – search domain, y – variables.

The objective function $\varphi(y)$ can be:

- multiextremal;
- “black-box”;
- hard-to-evaluate.



Parameters identification of chemical reaction*

- ❑ The isobutane alkylation reaction was investigated (alkylate is a high-octane component of automobile fuel).
- ❑ The problem was to develop an adequate mathematical model of the process
- ❑ The mathematical model of the reaction is described by the ODE system

$$\frac{dx_i}{dt} = \sum_{j=1}^{J} v_{ij} w_j \, , i = 1, \ldots, I, \qquad w_j = k_j \prod_{i=1}^{J} x_j^{|v_{ij}|} \qquad k_j = k_j^0 \exp\left(-\frac{E_j}{RT}\right)$$

and includes a priori unknown kinetic parameters:

E_j – the activation energy of *j*-th reaction stage

k_j^0 – pre-exponential factor of *j*-th reaction stage

*The problem was formulated in collaboration with Ufa State Petroleum Technological University and Institute of Petrochemistry and Catalysis of the RAS.

Parameters identification of chemical reaction

- ❑ Invers problem of chemical kinetics: to find the reaction parameters that minimize the deviation between the calculated and experimental data

$$\varphi(k^0, E) = \sum_{i=1}^{M}\sum_{j=1}^{N}\left|x_{ij}^{calc} - x_{ij}^{exp}\right| \to min$$

- ❑ The ODE system is stiff (there are fast and slow reaction stages)

  To solve it we use Radau IIA method
- ❑ One simulation requires from 0.1 to 1 sec. on supercomputer
- ❑ There are 15 unknown constants in the model
- ❑ Global search + local refinement

Lipschitz optimization

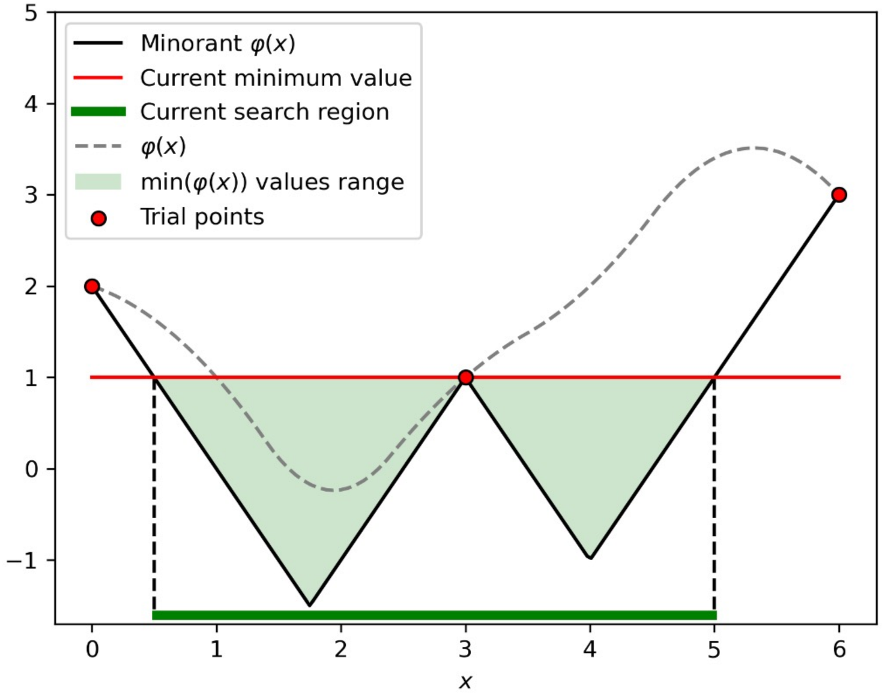
Supposition: the limited change of the argument Δy generates a limited change of the function values $\Delta\varphi$

Mathematical model: Lipschitz condition

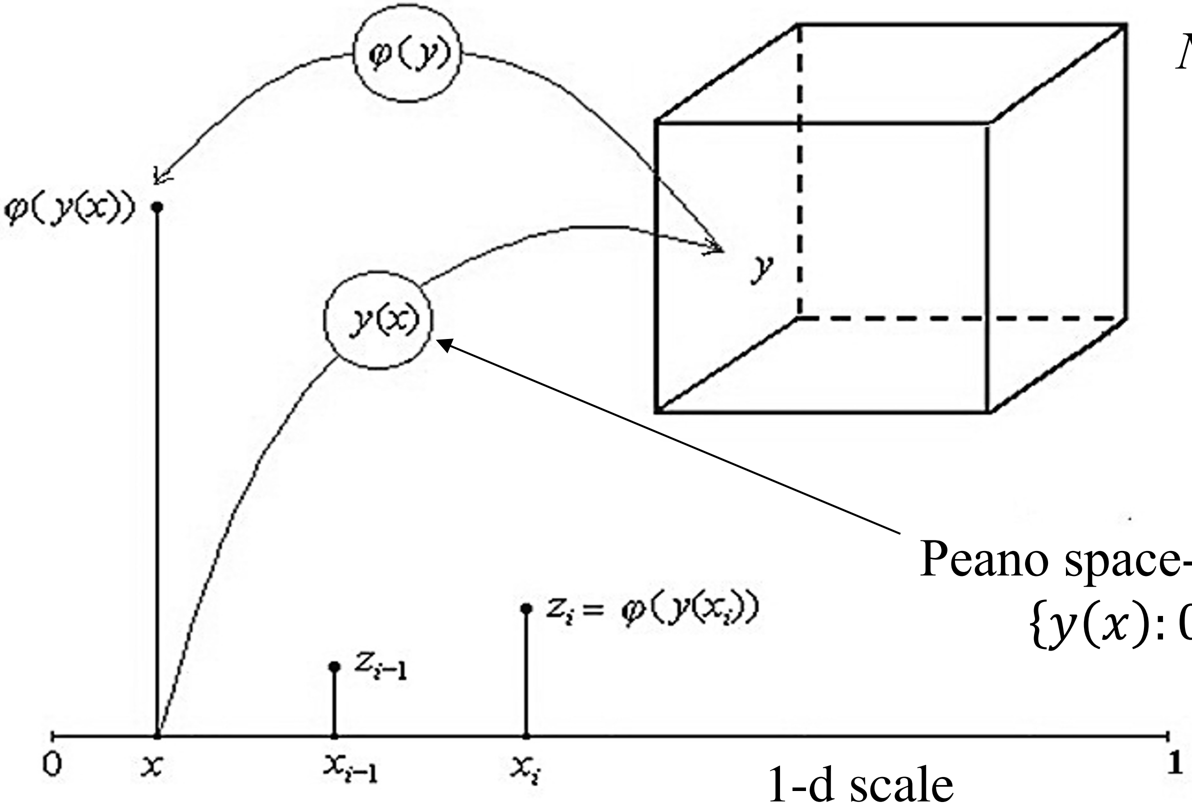
$$|\varphi(y') - \varphi(y'')| \le L\|y' - y''\|, \qquad y', y'' \in D$$

Lipschitz global optimization

S.A. Piyavskii, Yu.G. Evtushenko, M.A. Posypkin, R.G. Strongin, Ya.D. Sergeyev, J. Pinter, P. Hansen, D. Jones, J. Žilinskas, …



Dimensionality reduction [1]



N-d space

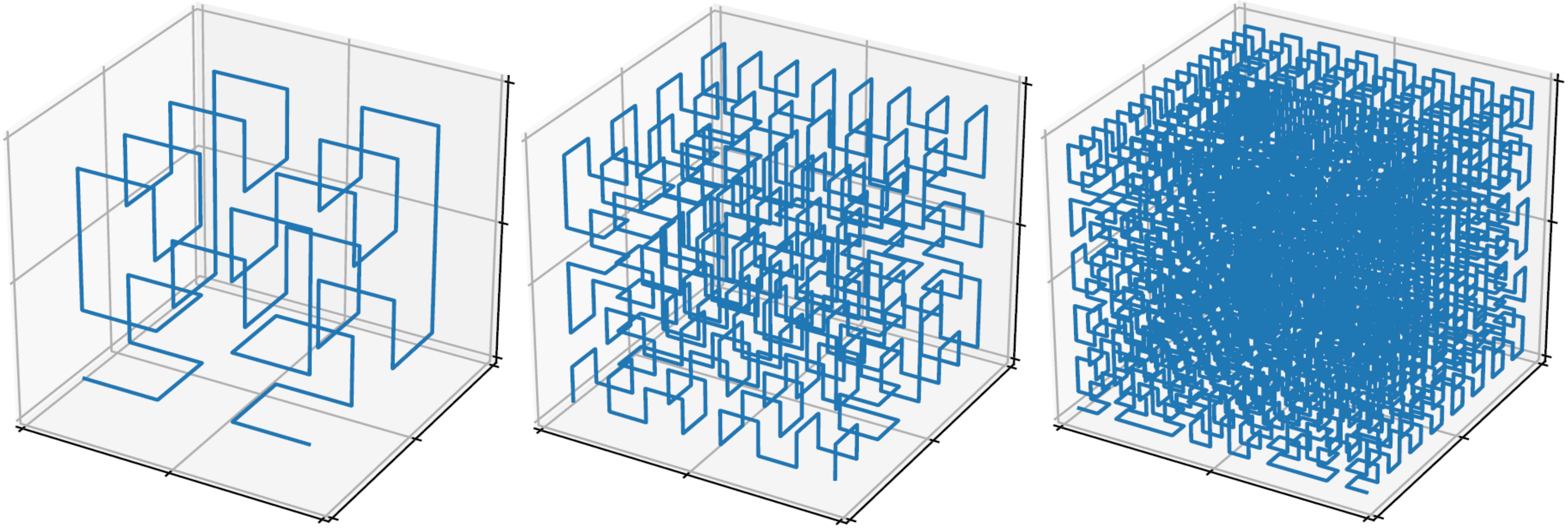
Peano space-filling curve

$$\{y(x): 0 \leq x \leq 1\} = \{y \in R^N: y \in D\}$$

1-d scale

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

Dimensionality reduction [2]



Numerical methods for building approximations of Peano curves with predefined accuracy (*evolvents*) are considered in Strongin, Sergeyev (2000)

If $\varphi(y)$ is Lipschitzian with some constant L then the univariate function $\varphi\left(y(x)\right)$ satisfies Hölder condition

$$\left|\varphi\left(y(x_1)\right) - \varphi\left(y(x_2)\right)\right| \leq 2L\sqrt{N+3}|x_1 - x_2|^{\frac{1}{N}}$$

where $x_1, x_2 \in [0,1]$.

Parallel Global Search Algorithm [1]

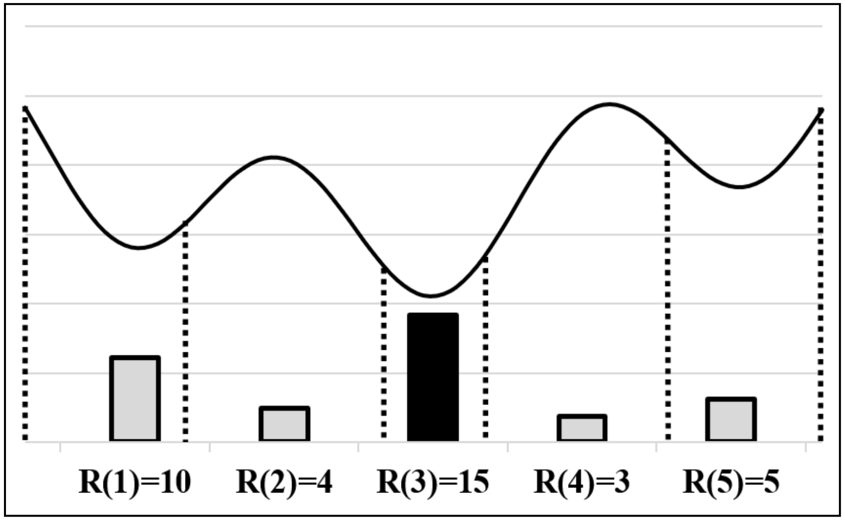
Search trial is calculation of the value of the function $\varphi(y(x))$ at the current point x_i. The general scheme of the global search algorithm:

The trials are carried out at p points $0 = x_0 < x_1 < \cdots < x_{p-1} = 1$

1. For each interval (x_{i-1}, x_i) calculate the value of the characteristic $R(i)$.

2. Sort intervals by characteristic, take p intervals with largest characteristics

$R(t_1) \geq R(t_2) \geq \cdots \geq R(t_p)$



3. Carry out new trials at the internal points of the intervals

$$\left(x_{t_1-1}, x_{t_1}\right), \left(x_{t_2-1}, x_{t_2}\right), \dots, \left(x_{t_p-1}, x_{t_p}\right)$$

4. Check stop condition $\Delta_{t_j} \leq \varepsilon$

Global Search Algorithm [2]

Characteristic

$$R(i) = \Delta_i + \frac{(z_i - z_{i-1})^2}{r^2\mu^2\Delta_i} - 2\frac{(z_i + z_{i-1})}{r\mu},$$

where $\mu = \max\limits_{1\le i\le k}\left\{\frac{|z_i - z_{i-1}|}{\Delta_i}\right\}$ is adaptive estimation of Lipschitz constant L, $r > 1$ is parameter of the algorithm.

New point

$$x^{k+j} = \frac{x_{t_j} + x_{t_j-1}}{2} - \operatorname{sign}\left(z_{t_j} - z_{t_j-1}\right)\frac{1}{2r}\left[\frac{\left|z_{t_j} - z_{t_j-1}\right|}{\mu}\right]^N$$

Theory of convergence of GSA presented in Strongin, Sergeyev (2000).

Convergence conditions

Theorem. Assume that the following conditions are satisfied:

- The objective function $\varphi(y)$ satisfies the Lipschitz condition with the constant L at $y \in D$;
- At some step of the search process

$$r\mu > 2^{3-\frac{1}{N}} L\sqrt{N+3}$$

where r >1 is a parameter, μ is adaptive estimation of the true Lipschitz constant L.

Then any limit point $\bar{y}$ of the sequence $\{y^k = y(x^k)\}$ generated by the algorithm is a global minimizer and any global minimizer y^* is also a limit point of the sequence$\{y^k = y(x^k)\}$.

Non-redundancy of the parallel algorithm

- ❑ $n(p)$ – the number of trials carried out by the parallel algorithm with p processes
  $n(p) \geq n(1)$, if $p \geq 1$.
- ❑ $s(p) = \frac{n(1)p}{n(p)}$ – speedup
- ❑ $\lambda(p) = \frac{n(p)-n(1)}{n(p)}$ – redundancy

Proposition. If the convergence conditions are satisfied and the objective function have Q local minima, then the parallel global search algorithm with parallelization degree Q will be non-redundant, i.e. $\lambda(p) = 0$

Amdal's law. $S(p) = \left(\alpha + \frac{1-\alpha}{p}\right)^{-1}$, where α is the fraction of the search trials necessary for the local refinement of the solution

Hooke-Jeeves method

It combines two moves:

The exploratory move

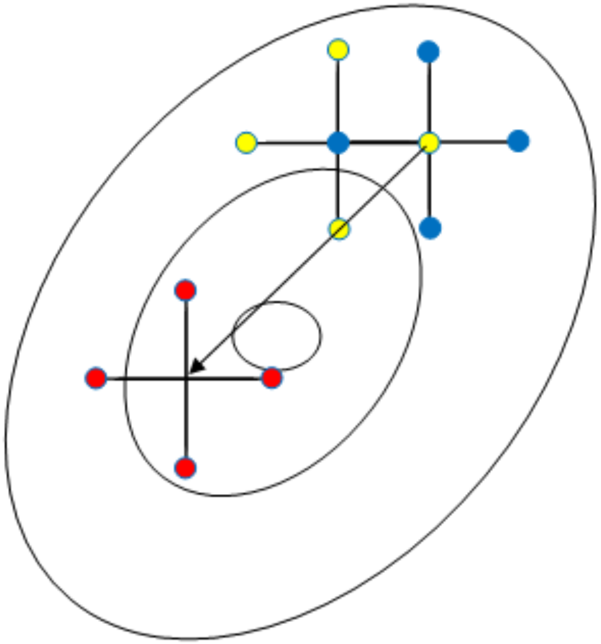
```
Step 0. Set $\bar{y} = y$.
Step 1. For $y$ from 1 to $N$ do:
if $\varphi(\bar{y} + he^i) < \varphi(\bar{y})$ then
    set $\bar{y} = \bar{y} + he^i$
  else if $\varphi(\bar{y} - he^i) < \varphi(y)$ then
    set $\bar{y} = y - he^i$.
```

The pattern move

```
Step 1. Set $y^1 = t^0, k = 0$.
Step 2. Construct a configuration $t^{k+1} = F(y^{k+1})$.
Step 3. Determine the next step
  if $\varphi(t^{k+1}) < \varphi(t^k)$ then
    $k = k + 1$ and go to Step 4,
  else
    if $h \leq \varepsilon$ then STOP
    else if $h > \varepsilon$ then
      if $k = 0$ then
        $h = h / 2$ and go to Step 1.
      else if $k > 0$ then
        set $t^0 = t^k, k = 0$ and go to Step 1.
Step 4. Move by direction $y^{k+1} = t^k + \alpha(t^k - t^{k-1})$ and go to Step 2.
```

Parallel Hooke-Jeeves method

In Hooke-Jeeves method one can parallelize the exploratory move, i.e. parallelize the loop at the second step of the algorithm. In this loop, one may not compute the function values at the points $\bar{y} + he^i$ or $\bar{y} - he^i$ immediately but may store the coordinates of the ones in the buffer, and upon accumulating p points compute the function values at these ones in parallel. Next, the accumulated information is analyzed and the coordinates of the point $\bar{y}$ are changed. This operation is repeated until all N coordinates $\bar{y}$ are computed.



$P = 2$	
🔵	🔵
🔵	🔵
🟡	🟡
🟡	🟡
🔴	🔴
🔴	🔴

BFGS algorithm

BFGS (Broyden-Fletcher-Goldfarb-Shanno) method belongs to the class of quasi-Newton methods using the values of the function $\varphi\left(y^k\right)$ and of its gradient $\nabla\varphi\left(y^k\right)$ to organize the search. In this method, the function $\varphi(y)$ is assumed to have the properties of the quadratic function that corresponds to the existence of a symmetric matrix of the second derivatives (Hessian) $\nabla^2\varphi(y)$.

Step 0. Initialize the starting point y^0. Set the precision $\varepsilon > 0$;

Step 1. Determine the initial approximation H^0 (either by the unit matrix E or by $\nabla^2\varphi(y^0)$);

Step 2. Determine the search direction: $d^k = -H^k\nabla\varphi\left(y^k\right)$;

Step 3. Compute $y^{k+1} = y^k + \alpha_k d^k$;

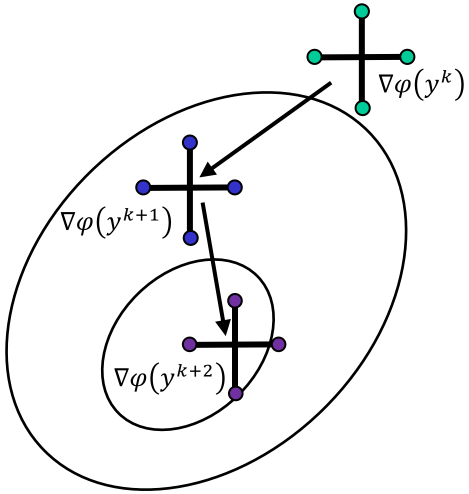
Step 4. Determine the vectors $s^k\ and\ z^k$;

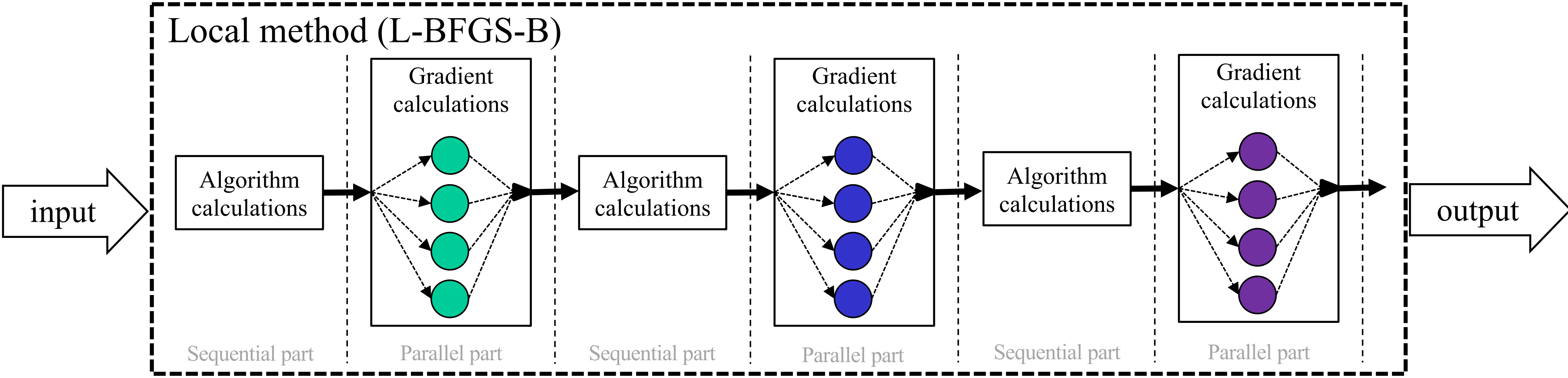
Step 5. Compute the Hessian H^{k+1};

Step 6. Stop, if the condition $|\nabla\varphi\left(y^k\right)| <$ is fulfilled.

Parallel BFGS algorithm

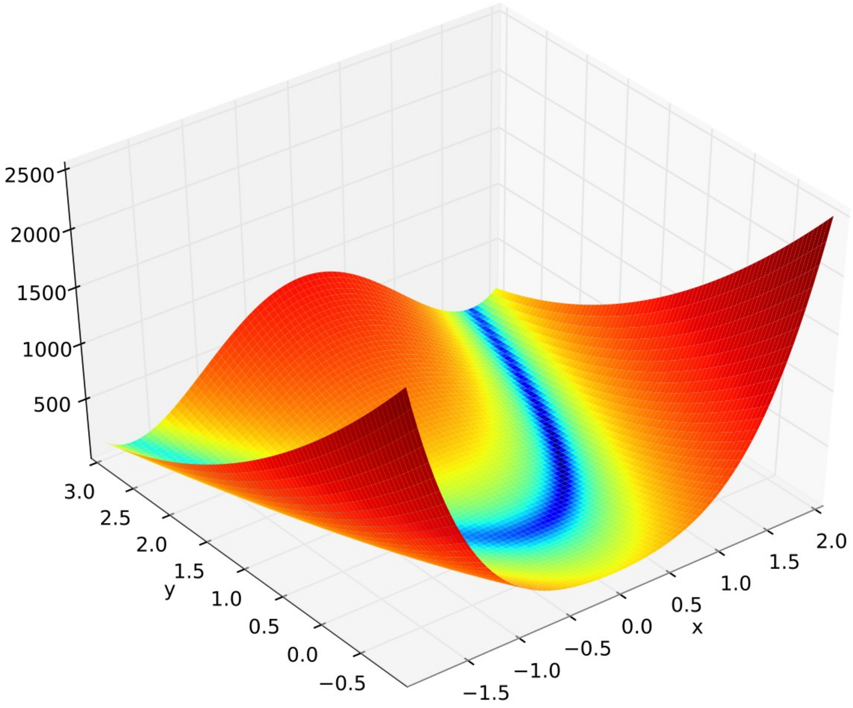
In the L-BFGS-B algorithm, one can parallelize the computing of gradient at the second step of the algorithm. To do so, one needs to store all points, in which the computing of the function values for the gradient estimate are required, into an intermediate buffer, and then compute the function values at these points in parallel.



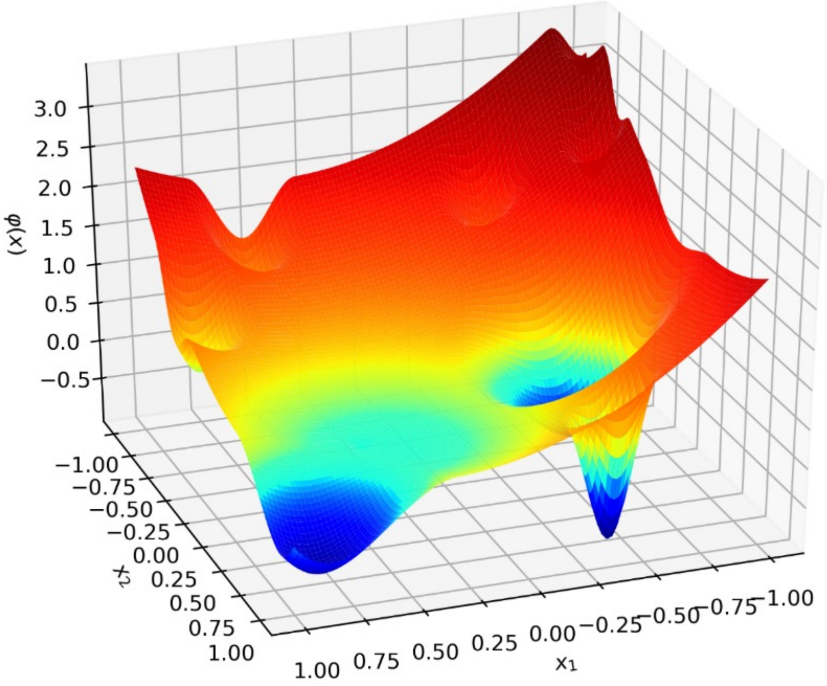


Test problems

Rosenbrock function

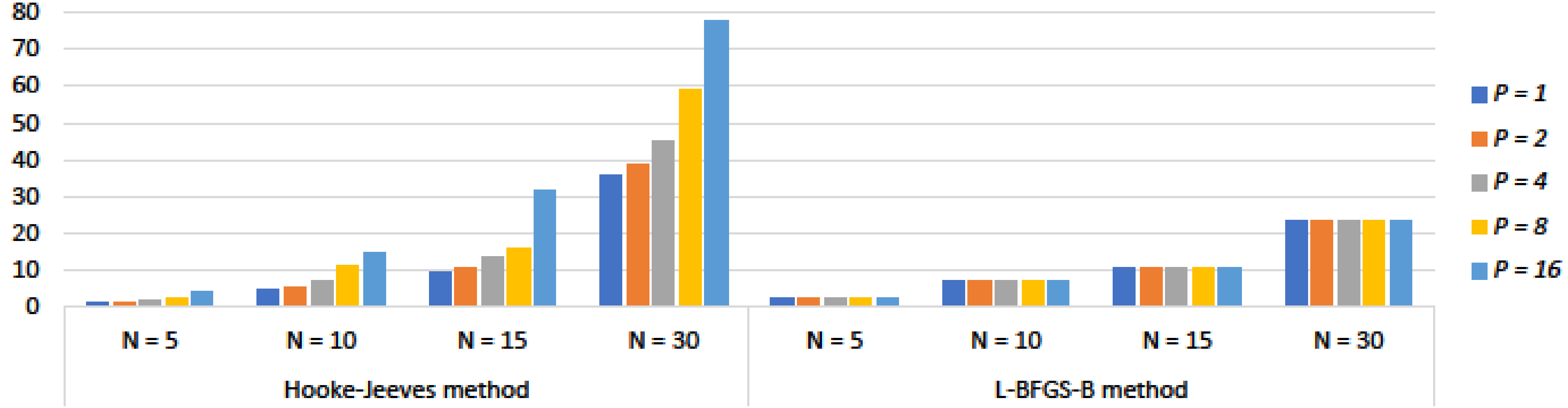


GKLS functions

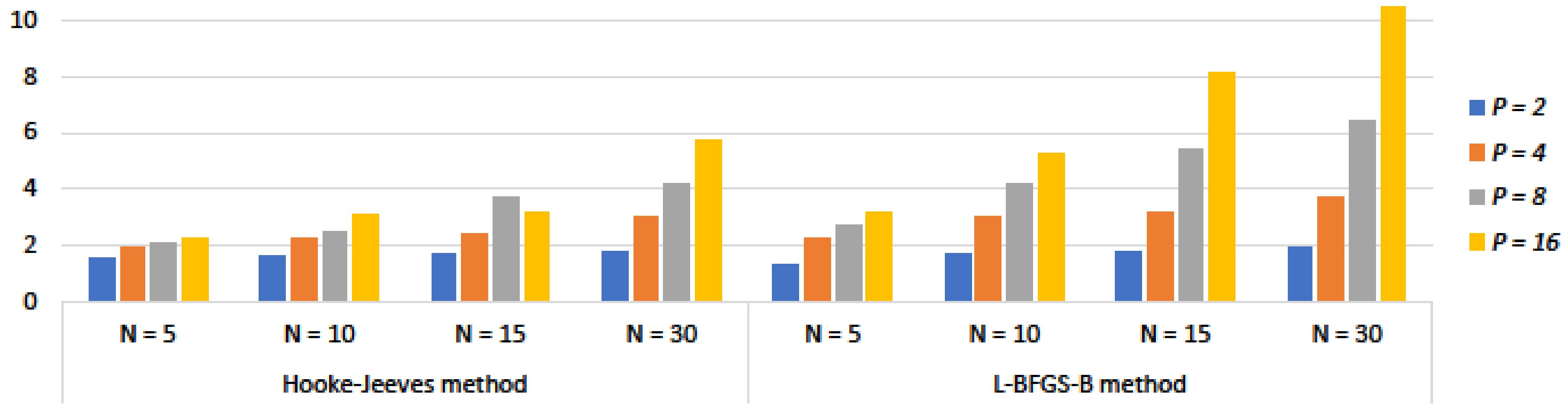


Results of numerical experiments

Number of trials of the local search algorithms, Rosenbrock function

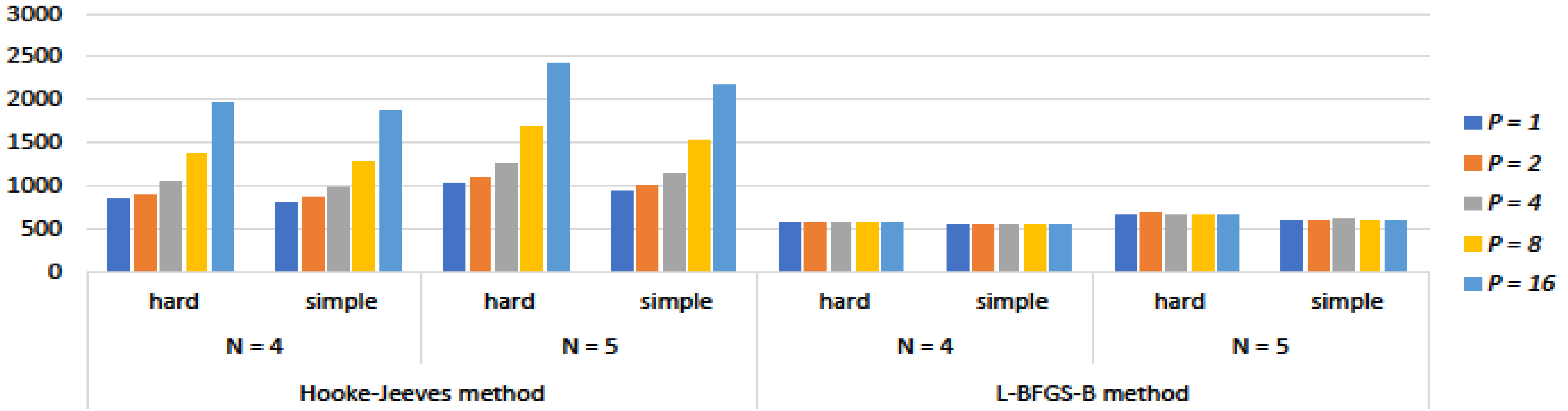


Speedup of the local search algorithms, Rosenbrock function

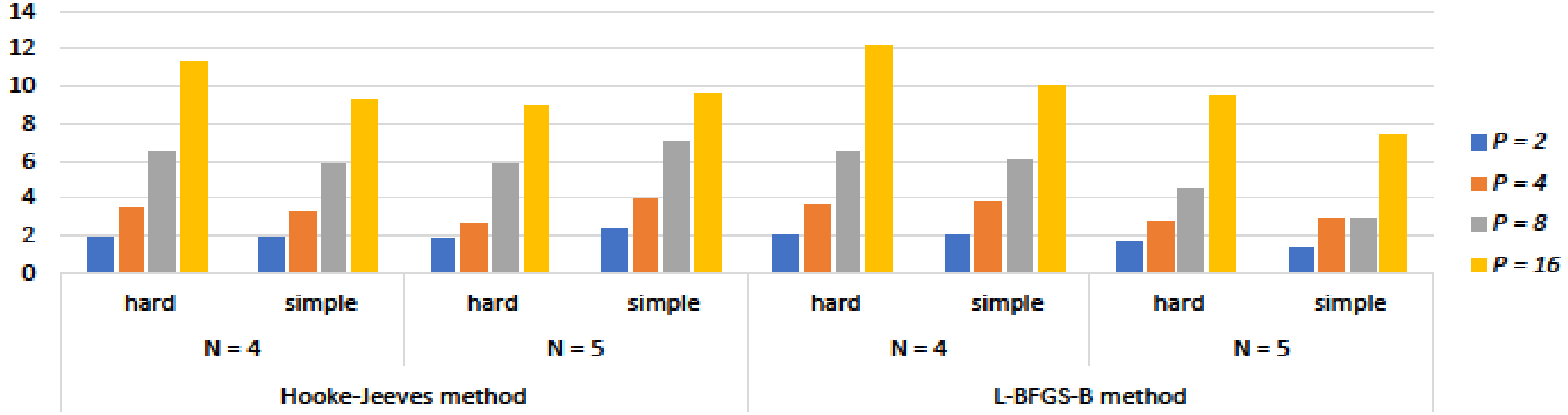


Results of numerical experiments

Global search + local search, average number of trials, 100 GKLS functions



Global search + local search, average number of trials, 100 GKLS functions



Invers problem of chemical kinetics, $N = 15$

pGSA – parallel global search algorithm
pGSA-l – parallel global + serial local search
pGSA-pl – parallel global + parallel local search

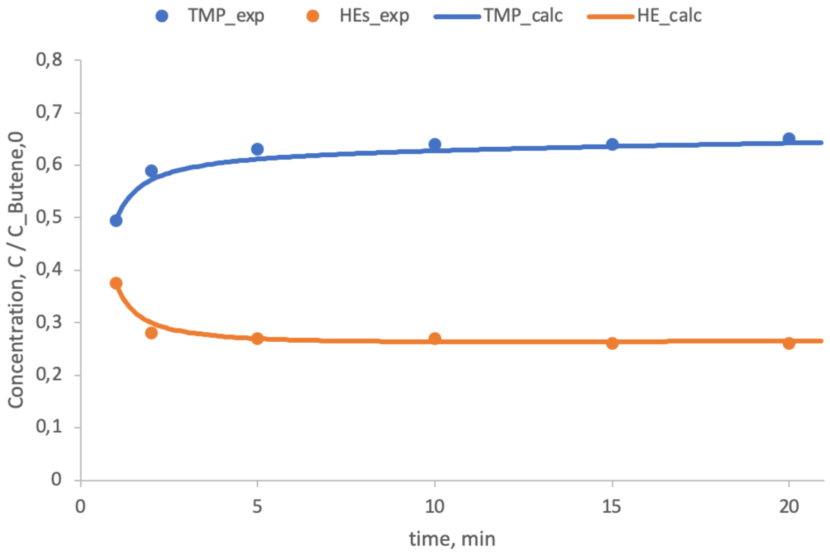
8 nodes (16 processors, 128 cores)

Based on the identified model, the concentrations of key components were calculated (lines)

The model agrees well with the experimental data (dots)

Time, min

	pGSA	pGSA-l	pGSA-pl
Hooke-Jeeves	158.1	299.8	177.4
L-BFGS-B	158.1	252.5	165.6



Lobachevsky supercomputer (CentOS 7.2, Intel Sandy Bridge E5-2660 2.2 GHz processors, 64 Gb RAM)

Thank you for attention

Konstantin Barkalov

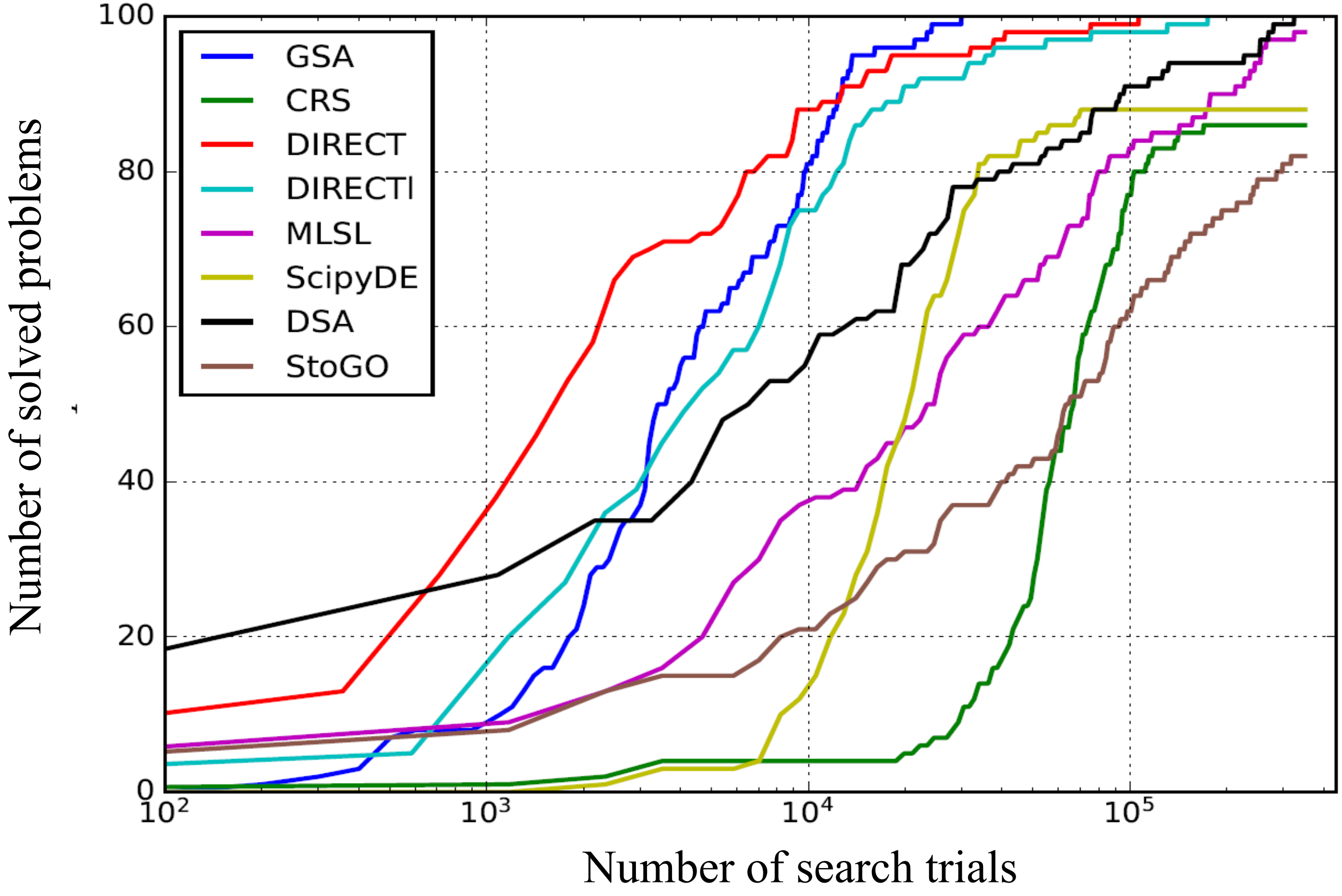
konstantin.barkalov@itmm.unn.ru

Lobachevsky State University of Nizhni Novgorod,
Institute of Information Technology, Mathematics and Mechanics

This study was supported by the Russian Science Foundation, project No 21-11-00204.

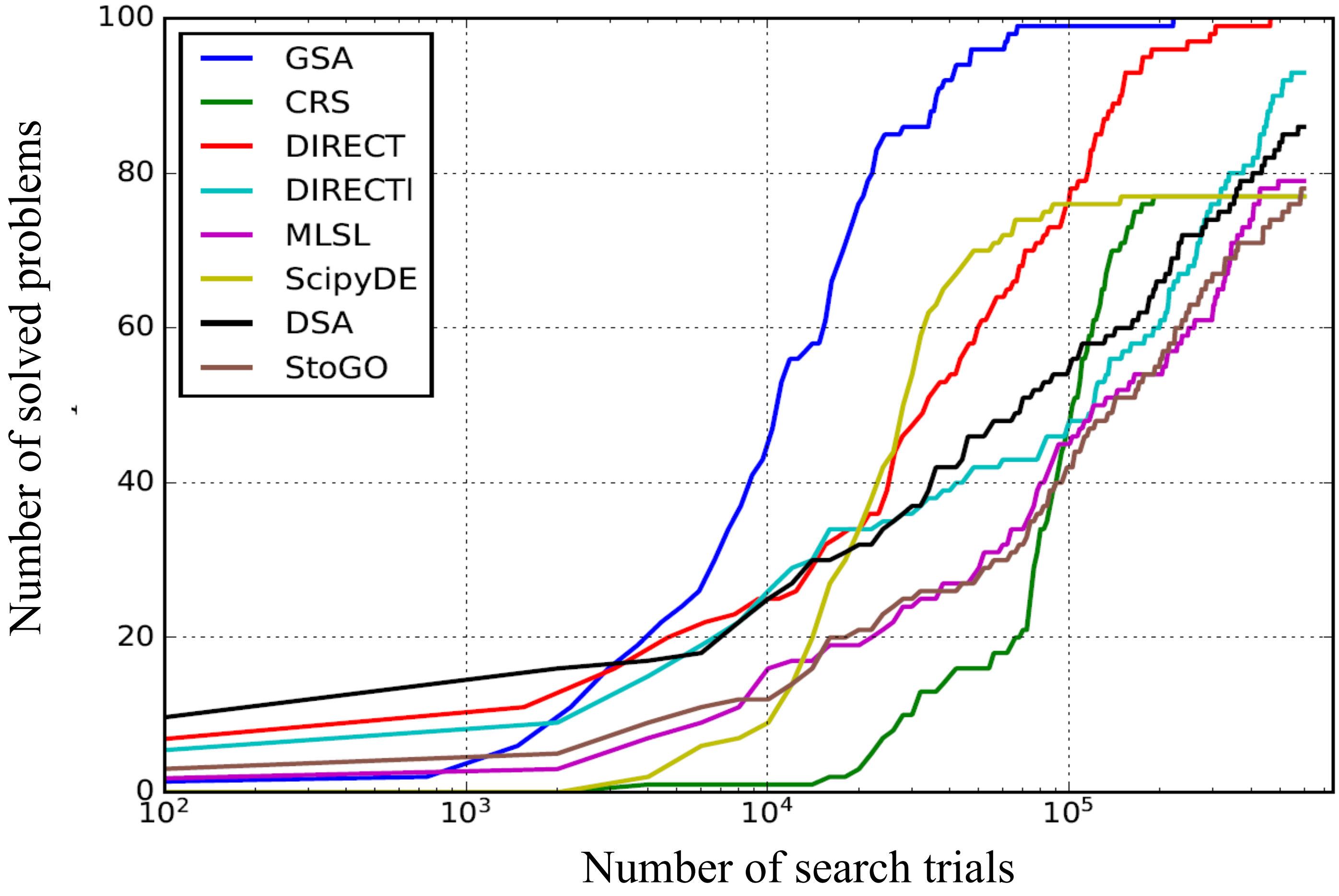
Comparison with other methods

Operation characteristics, 100 GKLS problems, Simple class, *N*=5

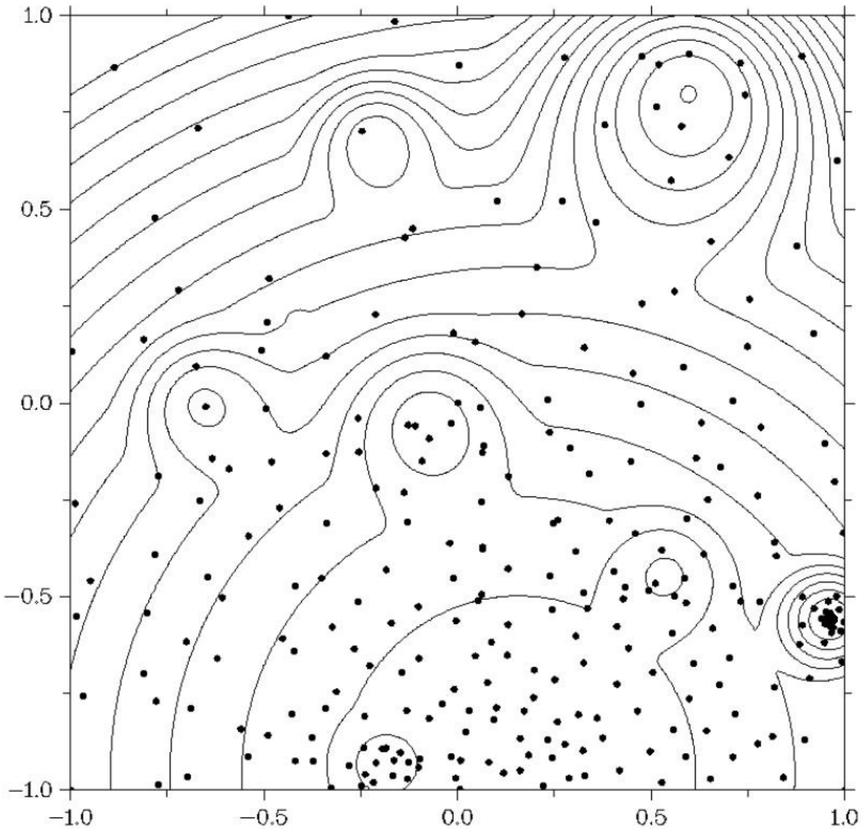


Comparison with other methods

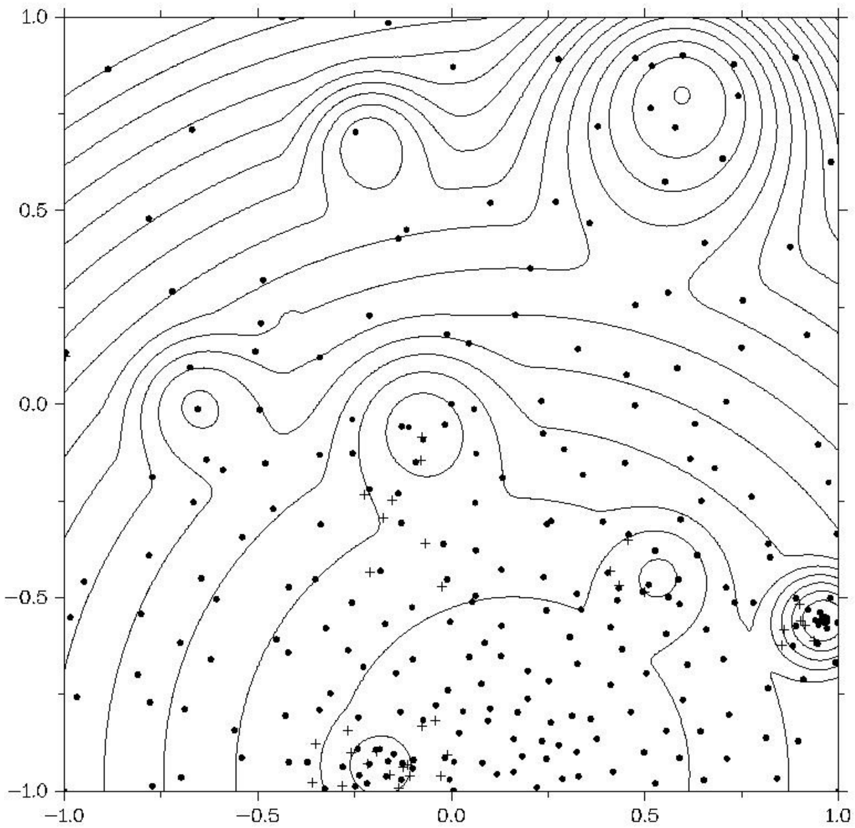
Operation characteristics, 100 GKLS problems, Hard class, *N*=5



Parallel global search: example



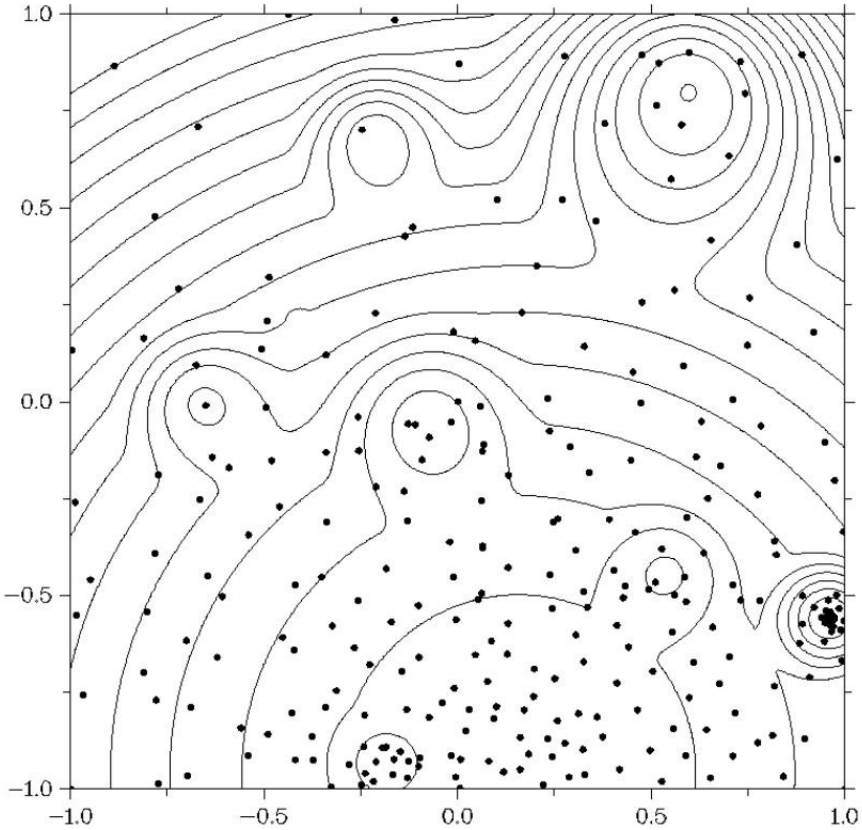
Serial algorithm
261 iterations → 261 trials



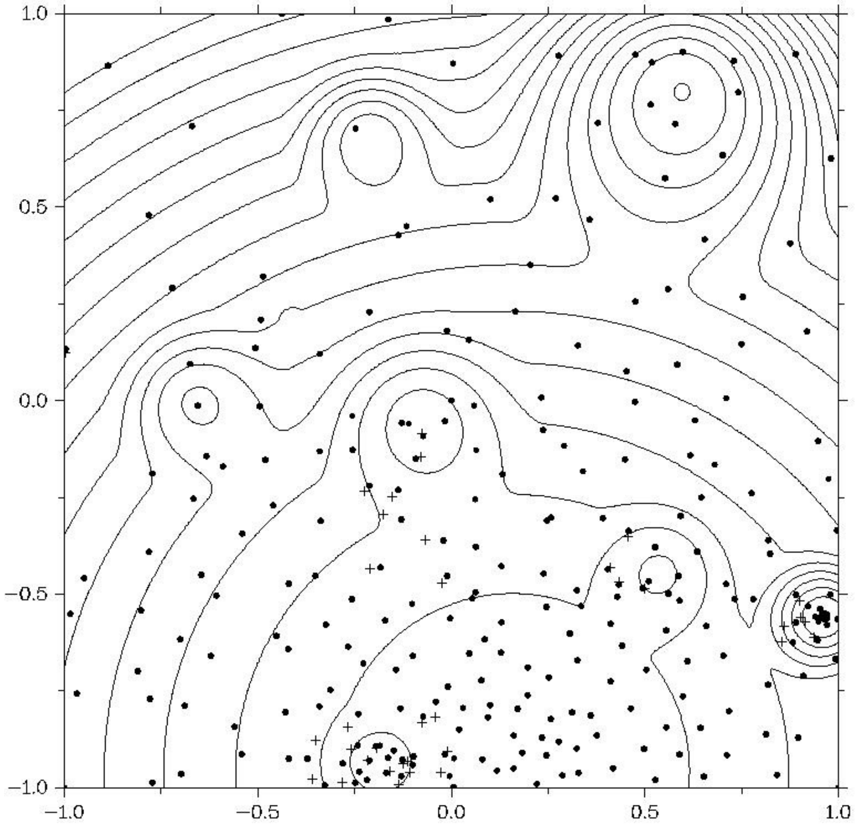
Parallel algorithm, p=4
76 iterations → 304 trials

Speedup – 3.4 times

Параллельные вычисления: пример работы



Последовательный алгоритм
261 итераций → 261 испытаний



Параллельный алгоритм, *p*=4
76 итераций → 304 испытания

Ускорение (по итерациям) – 3.4 раза