



Национальный исследовательский Нижегородский
государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

iTmm

XI Международная конференция «Суперкомпьютерные дни в России»
29 - 30 сентября 2025

A PARALLEL ALGORITHM FOR SOLVING GLOBAL OPTIMIZATION PROBLEMS AND ITS APPLICATION FOR TUNING HYPERPARAMETERS OF AI METHODS

Marina Usova, Ilya Lebedev, Anton Shtanyuk,
Evgeny Kozinov and Konstantin Barkalov

This study was supported by the Ministry of Science and Higher Education of the Russian Federation, project No. FSWR-2023-0034.

В качестве базовой постановки задачи рассматривается многомерная задача многоэкстремальной оптимизации без ограничений

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

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

Предположение 1. Функция $\varphi(y)$ удовлетворяет условию Липшица с априори неизвестной константой L :

$$|\varphi(y') - \varphi(y'')| \leq L \|y' - y''\|, \quad y', y'' \in D.$$

Предположение 2. В некоторой подобласти I области поиска D (в частном случае, в одной или нескольких её точках) целевая функция **может быть не определена**. Тогда целевая функция определена и вычислима лишь в подобласти $Q = D \setminus I$ положительного объема.

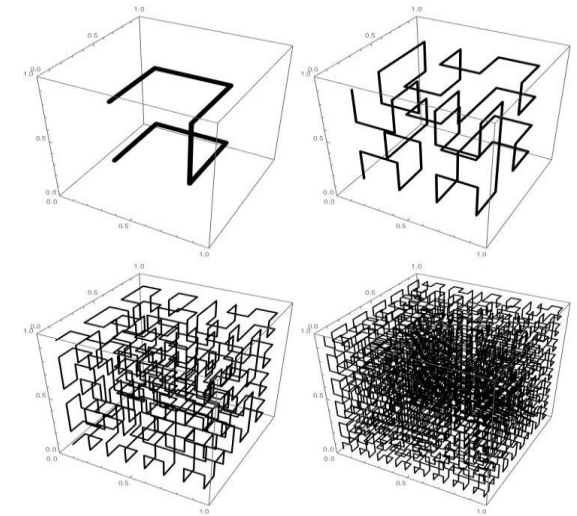


Рис.: Кривые Пеано с различным параметром свертки m

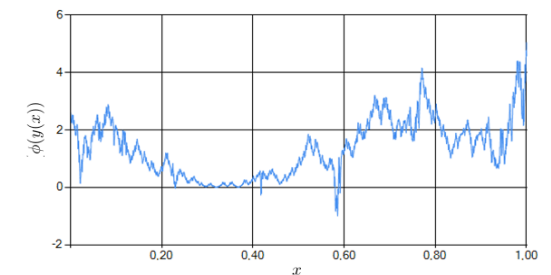
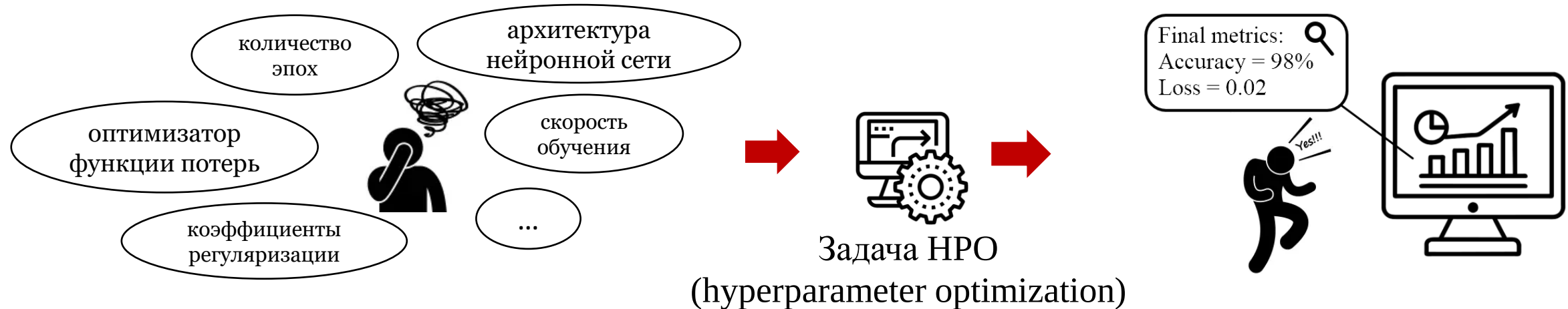


Рис.: Пример профиля редуцированной многомерной задачи

Задача НРО (hyperparameters optimization) [1]

В настоящее время методы машинного обучения и искусственного интеллекта используются во многих научных и прикладных областях. При этом эффективность любого алгоритма машинного обучения на практике зависит не только от самого метода, но и от **настройки гиперпараметров**.



Неверный выбор гиперпараметров может значительно снизить точность предсказаний, увеличить время обучения и даже привести к переобучению (overfitting) или недообучению (underfitting).

Usova M.A., Barkalov K.A. Lipschitz Global Optimization And Machine Learning: Helping Each Other To Solve Complex Problems // TM Web of Conferences. — 2024. — Vol. 59. — P. 01019.

Задача настройки гиперпараметров может быть представлена в виде:

$$\varphi(y^*, u^*) = \min\{\varphi(y, u) = M \circ A(y, u): y \in D, u \in U\},$$

где $A(y, u)$ – настраиваемый метод, M – используемая метрика качества, D – область определения непрерывных параметров метода A , U – область определения дискретных параметров метода A .

Характерные свойства:

- большая размерность,
- большое количество локальных экстремумов,
- трудоемкость вычислений,
- отсутствие аналитического представления целевой функции (т.е. модель представлена в виде «черного ящика»),

$$y = (y_1, y_2, \dots, y_N) \implies \boxed{} \implies \varphi(y)$$

- численная нестабильность.

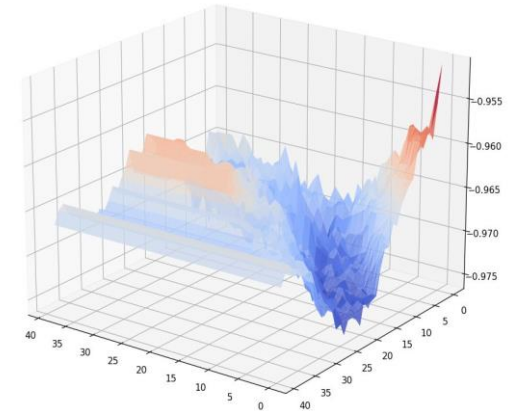


Рис.: Настройка гиперпараметров метода прогнозирования по временным рядам типа неисправности трансформатора

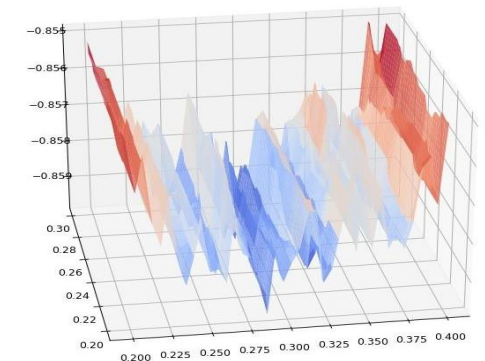
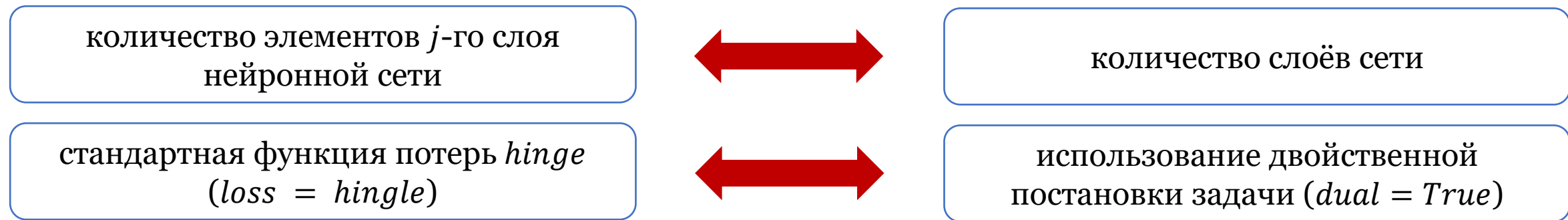


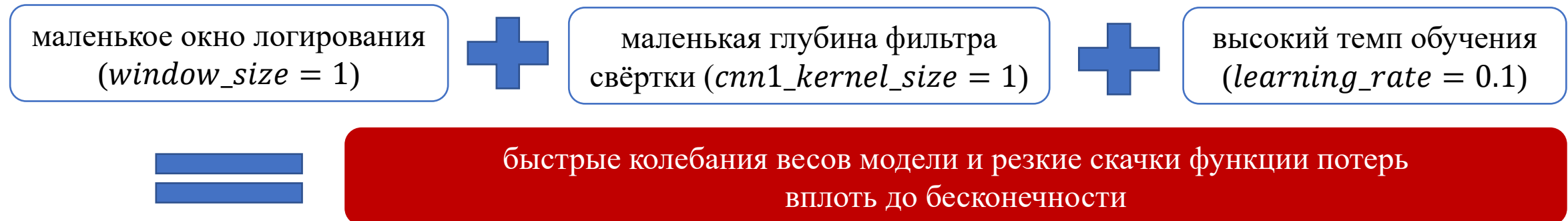
Рис.: Настройка гиперпараметров метода прогнозирования концентрации оксидов азота в рабочей зоне газотурбины

Численная нестабильность задач НРО

1. **Условные** гиперпараметры – релевантны только в зависимости от значений, принимаемых другими параметрами.



2. **Несочетаемые** гиперпараметры.



- Методы, основанные на идеях **мультистарта** (А. Жилинскас, А.А. Жиглявский, ...).
- Методы использующие **построение покрытий** (случайных или равномерных) области поиска (Л.А. Растрингин, И.М. Соболь, ...)
- **Стохастические, эвристические и метаэвристические** подходы, например:
 - имитация отжига и её модификации (S. Kirkpatrick, G. Gelatt, M. Vecchi, ...),
 - генетические алгоритмы (J. Holland, R. Storn, K. Price, А.П. Карпенко, ...),
 - роевые алгоритмы (J. Kennedy, R. Eberhart, Y. Shi, А.А. Кажаров, В.М. Курейчик, ...).
- **Детерминированные** подходы, например:
 - d.c.-оптимизация (А.С. Стрекаловский, Н. Туу, ...),
 - методы отсечений (В.П. Булатов, О.В. Хамисов, ...),
 - методы ветвей и границ, в том числе интервальный анализ (E. Hansen, Ю.Г. Евтушенко...),
 - байесовские методы (D.R. Jones, J. Mockus, ...),
 - методы липшицевой оптимизации (Р.Г. Стронгин, Я.Д. Сергеев, J. Pinter, P. Hansen, ...).

- Методы, основанные на идеях **мультистарта** – **требуют начальных точек для запуска локального метода.**
- Методы использующие **построение покрытий** (случайных или равномерных) области поиска – **решение гарантировано только при уплотнении сетки.**
- **Стохастические, эвристические и метаэвристические** подходы, например:
 - имитация отжига и её модификации,
 - генетические алгоритмы,
 - роевые алгоритмы,
 - **содержат случайную компоненту, параметры запуска могут существенно влиять на решение.**
- **Детерминированные** подходы, например:
 - d.c.-оптимизация,
 - методы отсечений,
 - методы ветвей и границ, в том числе интервальный анализ,
 - **не применимы к задачам вида «черный ящик»;**
 - байесовские методы,
 - методы липшицевой оптимизации.
 - **практически неприменимы к задачам большого размера ($N \gg 10$).**

- Методы, основанные на идеях **мультистарта**.
- Методы использующие **построение покрытий** (случайных или равномерных) области поиска.
- **Стохастические, эвристические и метаэвристические** подходы, например:
 - имитация отжига и её модификации,
 - генетические алгоритмы,
 - роевые алгоритмы. + стохастический градиентный спуск и др. комбинации алгоритмов
- **Детерминированные** подходы, например:
 - d.c.-оптимизация,
 - методы отсечений,
 - методы ветвей и границ, в том числе интервальный анализ,
 - байесовские методы, в частности EGO (efficient global optimization)
 - методы липшицевой оптимизации.





АГП для решения задач НРО [1]

Алгоритм глобального поиска (АГП) для решения НРО-задач объединяет внутри себя несколько модификаций АГП:

1. Многомерный алгоритм глобального поиска (использующий схему редукции размерности на основе кривых Пеано);
2. Алгоритм глобального поиска со смешанными переменными;
3. Алгоритм глобального поиска с частично определенной целевой функцией*.

Strongin R.G., Sergeyev Ya.D. Global Optimization with Non-convex Constraints: Sequential and Parallel Algorithms. Dordrecht : Kluwer Academic Publishers, 2000. 704 p.

Gergel V., Barkalov K., Lebedev I. A Global Optimization Algorithm for Non-Convex Mixed-Integer Problems // In: Lecture Notes in Computer Science. 2018. Vol. 11353. P. 78-81.

Usova M.A., Barkalov K.A. An Algorithm For Finding The Global Extremum Of A Partially Defined Function // Communications in Computer and Information Science. 2024. P. 147–161.



АГП для решения задач НРО [2]

Шаг 1. Исходную многомерную задачу НРО свести к множеству подзадач вида

$$\varphi(y^*, u^*) = \min_{s \in \{1, \dots, S\}} \{M \circ A(y, u_s) : y \in D\},$$

где число S , т.е. количество решаемых подзадач, соответствует числу всех возможных комбинаций целочисленных (дискретных) переменных.

Шаг 2. Описанное множество многомерных подзадач свести к одномерной задаче вида

$$f^*(x) = M \circ A(Y(x^*), u_{E(x^*)+1}) = \min \{M \circ A(Y(x), u_{E(x)+1}) : x \in [0, S]\},$$

где $Y(x)$ – отображение на основе развертки $y(x)$, переводящее любую точку интервала $[0, S]$ на область D (обозначение $E(x)$ – целая часть числа x), т.е. $Y(x) = y(x - E(x))$, $x \in [0, S]$.

Шаг 3. Применить к задаче алгоритм глобального поиска обобщенный на случай частично определенной целевой функции.



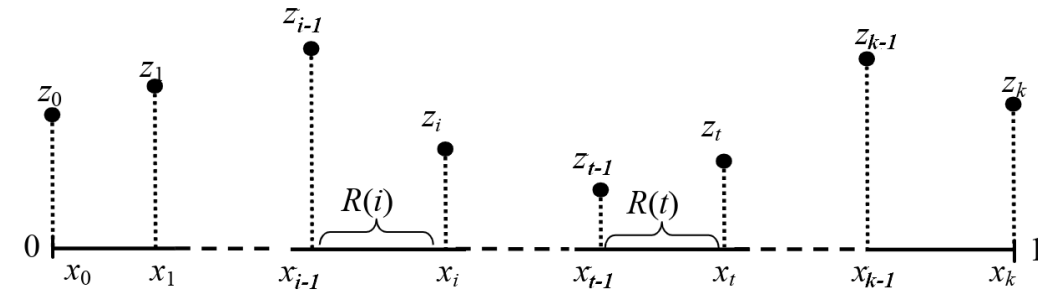
АГП для решения задач НРО [3]

Каждой точке ставится в соответствие индекс: $v_i = v(x_i) = \begin{cases} 1, & \text{если } x_i \in I, \\ 0, & \text{если } x_i \in Q. \end{cases}$

Шаг 1. На основе накопленной информации произвести оценку константы μ .

Шаг 2. Для каждого поискового интервала **вычислить значение характеристики $R(i)$**

$$R(i) = \begin{cases} \alpha \left(1 - \frac{1}{r}\right)^2 \Delta_i, & v(x_i) = v(x_{i-1}) = 0, \\ \Delta_i + \frac{(z_i - z_{i-1})^2}{(r\mu)^2 \Delta_i} - 2 \frac{(z_i + z_{i-1} - 2z^*)}{r\mu}, & v(x_i) = v(x_{i-1}) = 1, \\ 2\Delta_i - 4 \frac{(z_i - z^*)}{r\mu}, & v(x_{i-1}) = 0, v(x_i) = 1, \\ 2\Delta_i - 4 \frac{(z_{i-1} - z^*)}{r\mu}, & v(x_{i-1}) = 1, v(x_i) = 0. \end{cases}$$



$$0 < \alpha \leq 1, \\ r > 1$$

Шаг 3. Выбрать интервал t с лучшим (наибольшим) значением характеристики $R(i)$.

Шаг 4. Провести очередное испытание во внутренней точке лучшего интервала

$$x^{k+1} = \begin{cases} \frac{x_t + x_{t-1}}{2} - \text{sign}(z_t - z_{t-1}) \frac{(|z_t - z_{t-1}|)^N}{2r\mu^N}, & v(x_i) = v(x_{i-1}) = 1, \\ \frac{x_t + x_{t-1}}{2}, & \text{в противном случае.} \end{cases}$$

Условие остановки. Достижение максимального числа шагов или заданной точности решения.

При этом решение задач НРО требует значительных вычислений, и для работы с реалистичными временными затратами, может осуществляться только при использовании суперкомпьютерных систем с использованием высокоэффективных параллельных алгоритмов глобальной оптимизации.

Известные способы или не обеспечивают должного эффекта, или не применимы. Так, например, блочная схема неэффективна, ведь только небольшая часть процессов будет работать в подобласти с искомым глобальным минимумом.

Более продуктивный подход к распараллеливанию основан на изменении схемы последовательного алгоритма для **параллельного выполнения нескольких испытаний**. При этом для решения задач НРО предпочтение следует отдать асинхронной схеме организации вычислений.

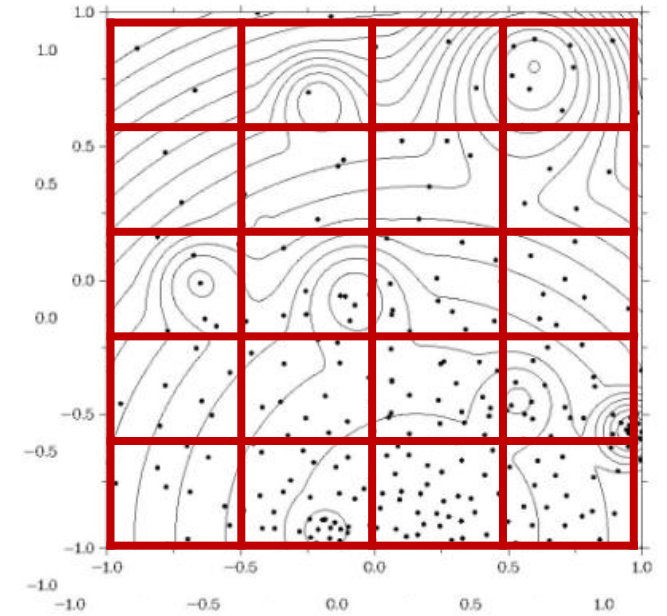


Рис.: Неудачное разбиение задачи на блоки

На первой итерации мастер-процесс распределяет координаты p испытаний по процессам-рабочим.

Каждый процесс-рабочий проводит одно испытание (вычисляет значение целевой функции) и передаёт результаты проведенного испытания мастер-процессу.

Полученные мастером-процессом результаты обрабатываются **немедленно**, и на данном узле запускается проведение очередного испытания.

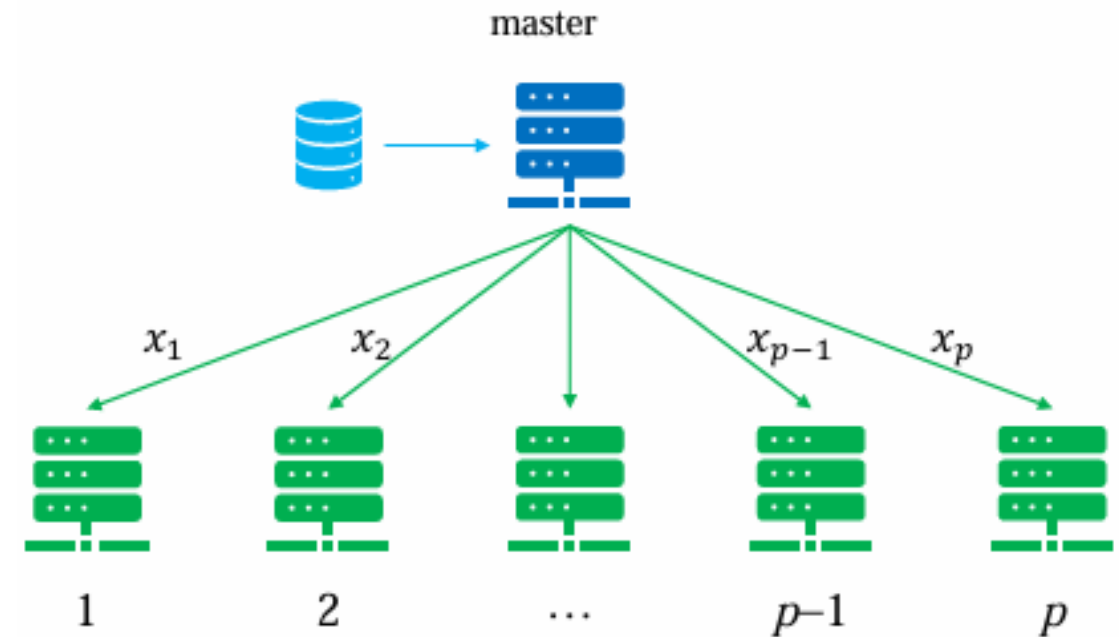


Рис.: Модель мастер/рабочие

Шаги последовательной версии алгоритма выполняет мастер-процесс.



Реализация алгоритма и проведение экспериментов

Описанный алгоритм был реализован на базе фреймворка методов интеллектуальной оптимизации iOpt, разрабатываемого в ННГУ им. Н.И. Лобачевского.



Для распараллеливания вычислений в iOpt мы использовали библиотеку multiprocessing, которая хорошо работает при передаче небольших объемов данных между процессами.

Исходный код:

<https://github.com/aimclub/iOpt>

Документация:

<https://iopt.readthedocs.io/>

Вычислительные эксперименты были проведены на суперкомпьютере «Лобачевский» Нижегородского государственного университета им. Н.И. Лобачевского (операционная система CentOS 7, SLURM, каждый узел имеет 2 процессора Intel Xeon Silver 4310T CPU@ 2.30GHz (20 ядер), 64Gb RAM).

Решение задач производилось с использованием Python 3.9, FEDOT 0.7.5, iOpt 0.5.0 and Optuna 4.2.1.

Рассматривалась задача настройки гиперпараметров метода предсказания значений временного ряда (time series forecasting).



Предсказание значений производилось с использованием фреймворка FEDOT (фреймворк с открытым исходным кодом для задач автоматизированного моделирования и машинного обучения, AutoML).

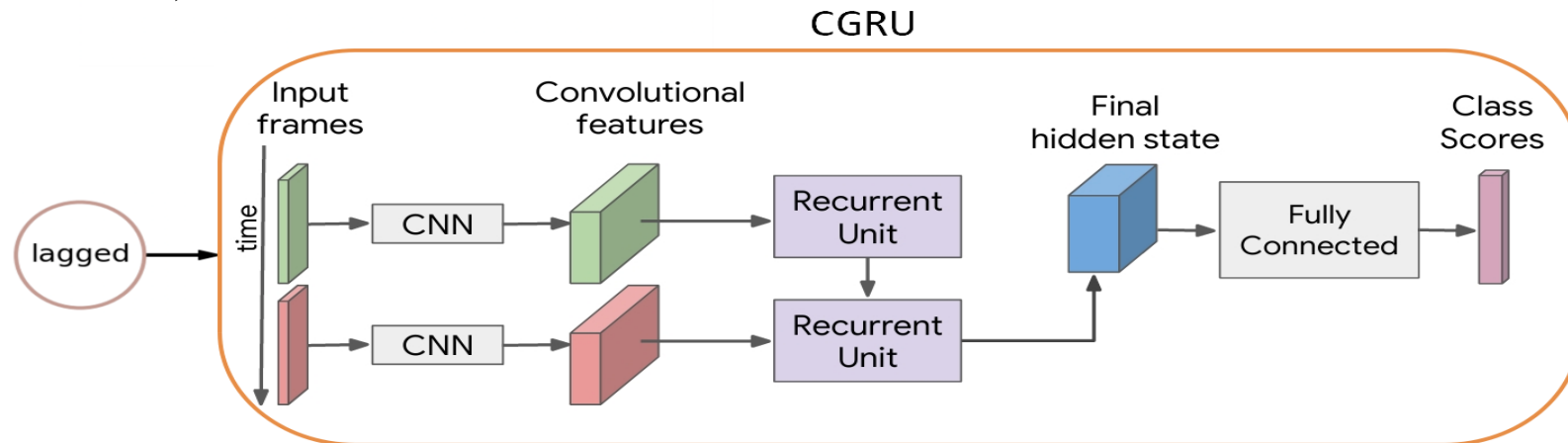


Рис.: Визуализация пайплайна со схемой архитектуры сети CGRU

* Фреймворк FEDOT использует ML-модели, в основном из стандартных библиотек sklearn, statsmodels и keras.

У lagged-преобразования варьировался параметр $window_size \in \{5, 6, 7, \dots, 254, 250\}$ (был рассмотрен как непрерывный параметр с использованием округления до единиц).

У CGRU варьировались 4 дискретных и 2 непрерывных параметра:

- $cnn1_kernel_size \in \{4, 5\}$,
 - $cnn2_kernel_size \in \{5, 6\}$,
 - $cnn1_output_size \in \{32, 64\}$,
 - $cnn2_output_size \in \{32, 64\}$,
 - $hidden_size \in [20, 200]$,
 - $learning_rate \in [0.0005, 0.005]$.
- всего 16 комбинаций

Остальные параметры были зафиксированы в значениях по умолчанию: $batch_size = 64$, $num_epochs = 50$, $optimizer = 'adamw'$, $loss = 'mse'$.

Эксперимент проводился на датасете `montly beer production`.

Для сравнения поставленная задача также была решена при помощи известного фреймворка оптимизации гиперпараметров Optuna.

Фреймворк	Время решения, сек.	Финальная MSE
iOpt	6574.2	13.5
Optuna	9554.1	13.4

Число точек, в которых не получилось вычислить критерий оптимизации, составило порядка 10% от общего числа точек поисковых испытаний!

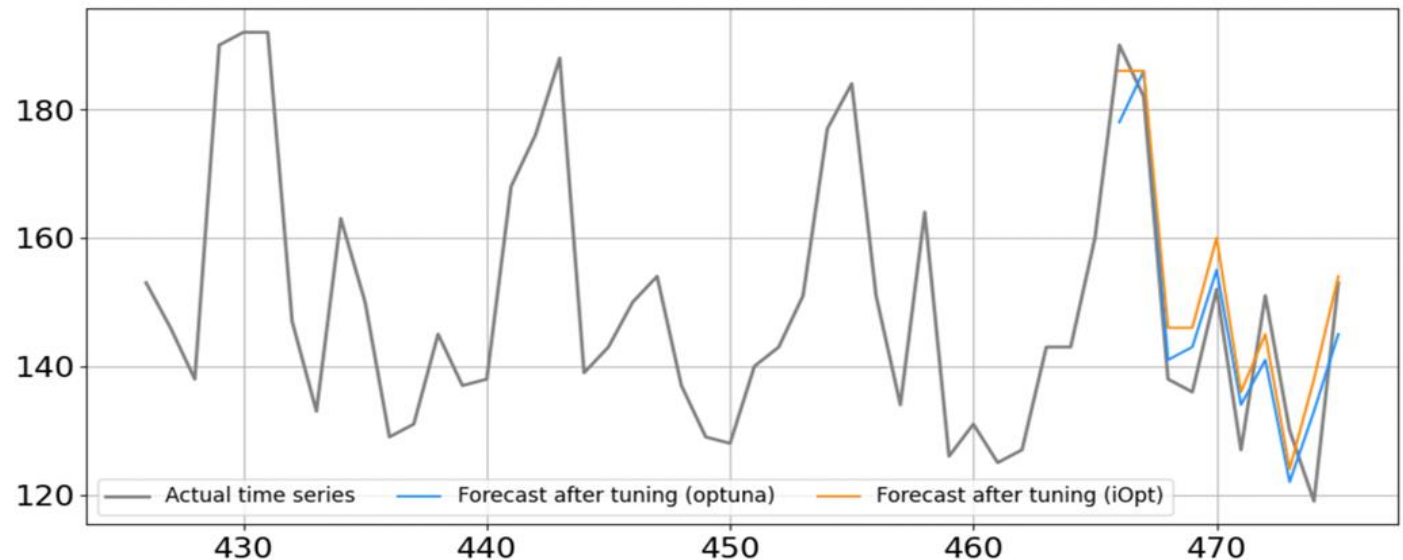


Рис.: Визуализация предсказания и истинного значения временного ряда

* В эксперименте было выставлено ограничение на 1000 испытаний, в iOpt использовалось локальное уточнение.

P	iOpt		Optuna	
	Время решения, сек.	Финальная MSE	Время решения, сек.	Финальная MSE
1	6574.2	13.5	9554.1	13.4
2	3579.2	13.9	5127.5	13.6
4	1998.7	13.6	2598.6	13.2
8	1331.7	13.2	1275.4	13.2
16	846.6	13.3	1148.5	13.3
20	699.9	13.9	1578.6	13.3

Увеличение числа процессов положительно влияет на общее ускорение обоих инструментов, однако на больших количествах процессов (от 16 и выше) фреймворк iOpt демонстрирует лучшие результаты по сравнению с Optuna. Время выполнения задачи в фреймворке iOpt растет медленнее с увеличением числа процессов, что свидетельствует о лучшей масштабируемости и эффективном распределении нагрузки.

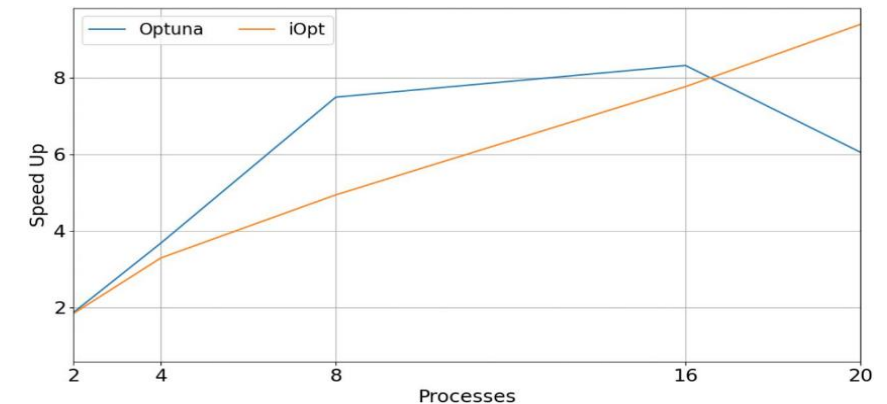


Рис.: Зависимость ускорения от числа процессов для фреймворков iOpt и Optuna

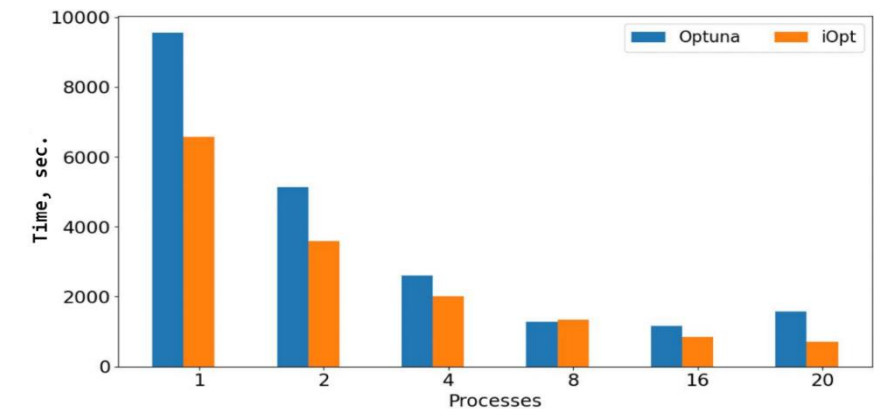


Рис.: Зависимость времени решения задачи от числа процессов для фреймворков iOpt и Optuna

Экспериментально подтверждено преимущество нового подхода над существующими методами при работе с моделями, характеризующимися большими областями неопределенности.

Результаты параллельных экспериментов показывают, что iOpt способен обеспечить большее ускорение и меньшую задержку по сравнению с Optuna при увеличении числа параллельно работающих процессов, делая его предпочтительным выбором для задач, требующих интенсивного использования параллельности.

Однако при увеличении числа параллельных процессов объем данных, передаваемых между процессами, будет линейно расти, что может привести к снижению ускорения с ростом числа используемых процессов. Дальнейшие исследования будут направлены на использование более эффективного алгоритма копирования и передачи данных в iOpt с целью повышения ускорения.

Спасибо за внимание!

Институт информационных технологий, математики и механики
ННГУ им. Н.И. Лобачевского

Усова Марина Андреевна
usova@itmm.unn.ru

This study was supported by the Ministry of Science and Higher Education of the Russian Federation, project No. FSWR-2023-0034.

Согласно непрерывному алгоритму первые два испытания проводятся в крайних точках исследуемого отрезка $x^0 = 0$, $x^1 = 1$. Выбор точки x^{k+1} , $k \geq 1$ очередного $(k + 1)$ -го испытания осуществляется на основе следующих правил.

Правило 1. Перенумеровать (нижним индексом) точки x^i , $1 \leq i \leq k$ предшествующих испытаний в порядке возрастания значений координаты, то есть

$$0 = x_0 < x_1 < \dots < x_i < \dots < x_k = 1 \quad (1)$$

и сопоставить им значения $z_i = f(x_i)$, $1 \leq i \leq k$, вычисленные в этих точках.

Правило 2. Вычислить текущую нижнюю оценку

$$\mu = \max \left\{ \frac{|z_i - z_{i-1}|}{\Delta_i}, 1 \leq i \leq k \right\}, \quad \Delta_i = (x_i - x_{i-1})^{1/N}, \quad (2)$$

для константы Гельдера, где r , $r > 1$ – параметр алгоритма; если $\mu = 0$, то принять $\mu = 1$.

Правило 3. Для каждого интервала (x_{i-1}, x_i) , $1 \leq i \leq k$ вычислить значение $R(i)$ по формуле

$$R(i) = \Delta_i + \frac{(z_i - z_{i-1})^2}{(r\mu)^2 \Delta_i} - 2 \frac{(z_i + z_{i-1})}{r\mu}. \quad (3)$$

Правило 4. Выбрать интервал t с наибольшим значением характеристики

$$R(i): R(t) = \max_{1 \leq i \leq k} R(i).$$

Если максимальная характеристика соответствует нескольким интервалам, то в качестве t выбирается минимальное число, удовлетворяющее описанному условию.

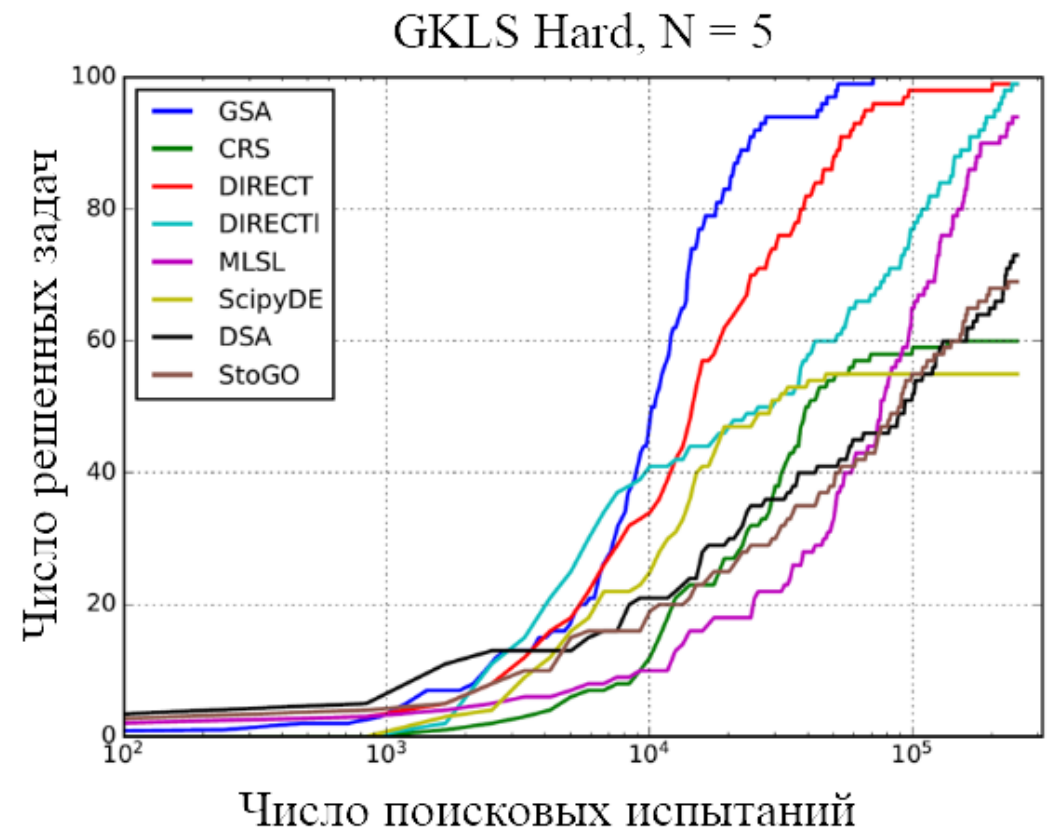
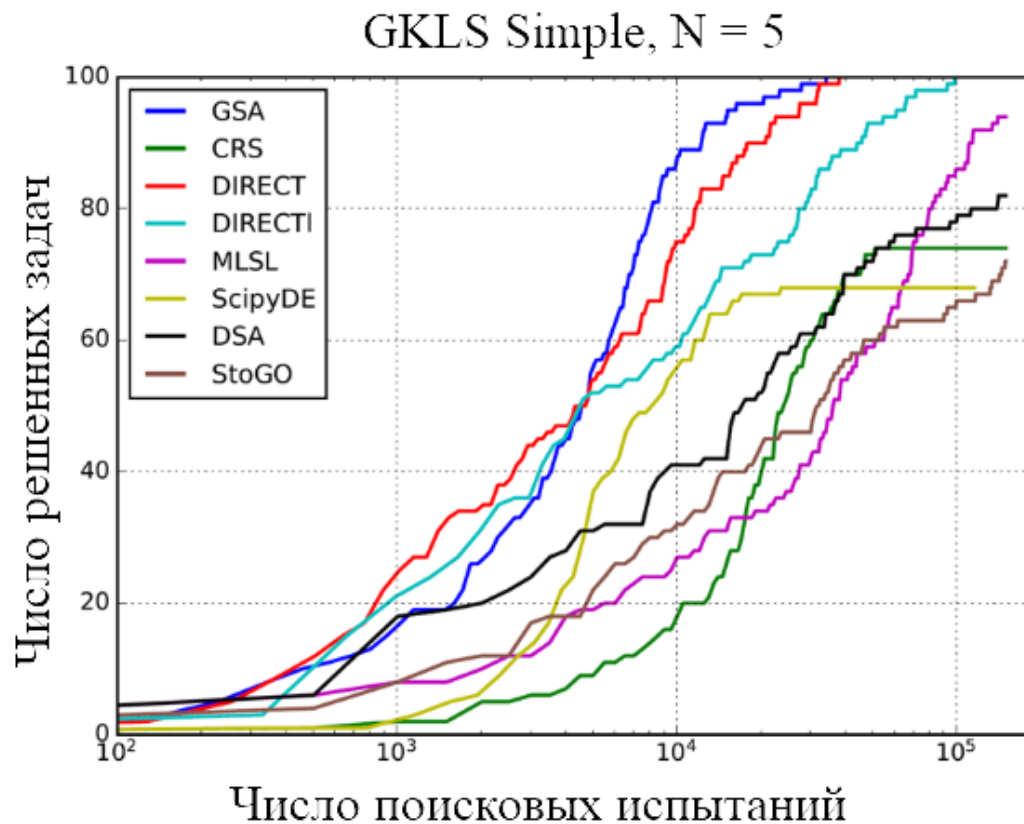
Правило 5. Провести очередное испытание во внутренней точке интервала (x_{t-1}, x_t) по правилу

$$x^{k+1} = \frac{x_t + x_{t-1}}{2} - \text{sign}(z_t - z_{t-1}) \frac{(|z_t - z_{t-1}|)^N}{2r\mu^N}. \quad (4)$$

Условие остановки. При достижении максимального количества итераций метода $k > K_{max}$ или заданной точности $\Delta_t \leq \varepsilon$ завершить работу алгоритма.

Сравнение с известными алгоритмами

На классах многоэкстремальных задач алгоритм глобального поиска превосходит иные алгоритмы аналогичного назначения по числу решенных задач:





Методы и подходы

Популярные алгоритмы решения задач **НРО** (настройки гиперпараметров):

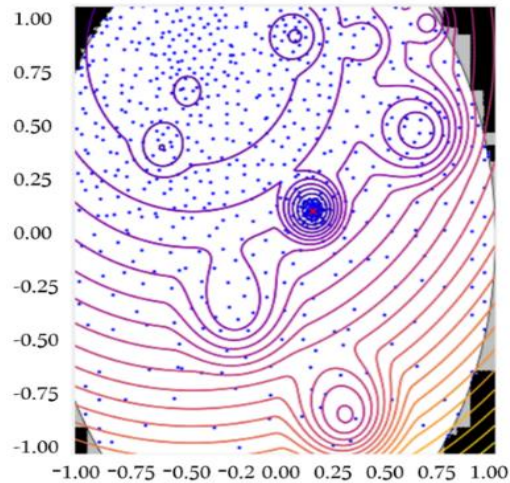
- генетические алгоритмы,
- метод роя частиц,
- комбинации различных подходов (генетические + градиентные методы),
- стохастический градиентный спуск;
- байесовские методы (например, EGO).

Алгоритм	Возможный результат поиска			
	Глобальный минимум	Локальный минимум	nan	1e+100
differential_evolution			●	
dual_annealing	●		●	
basinhopping	●	●		●
direct	●	●		
shgo	●	●		
brute	●	●	●	●

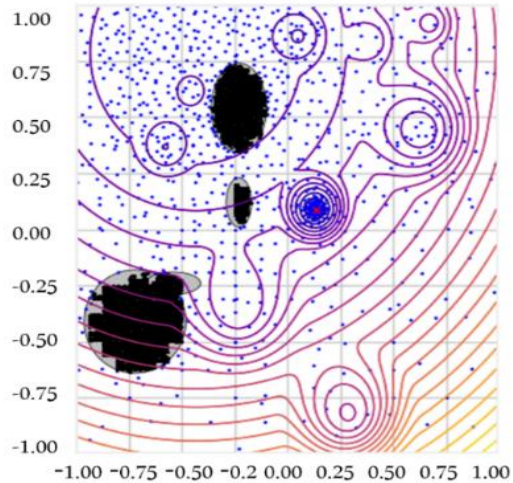
Решение задач с **численной неопределенностью** могут быть использованы следующие подходы:

- фильтрация недопустимых точек (точки исключаются перед началом процесса оптимизации);
- использование штрафных функций (в том числе регуляризация в МО для устранения «артефактов» и стабилизации процесса оптимизации);
- аппроксимация поверхности целевой функции гладкой поверхностью;
- повторный запуск (с обходом проблемных областей);
- использование подходов из Data Science для обработки «missing data».

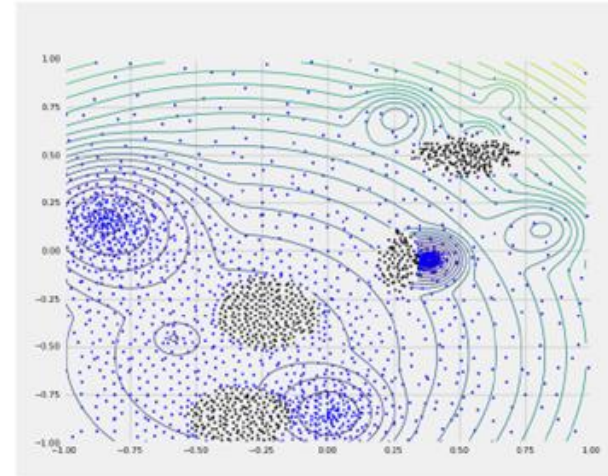
Влияние параметра α на распределение точек испытаний



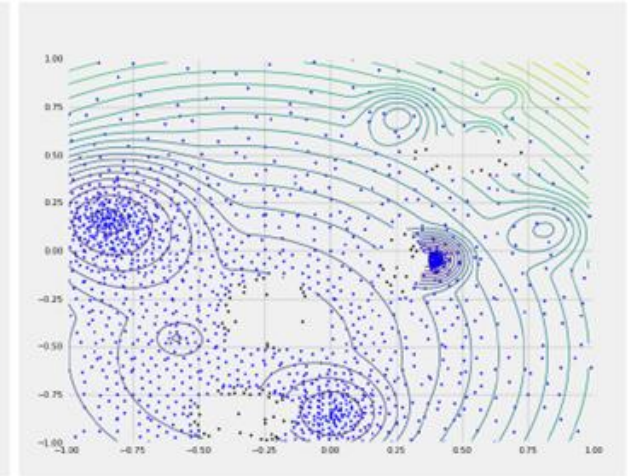
а)



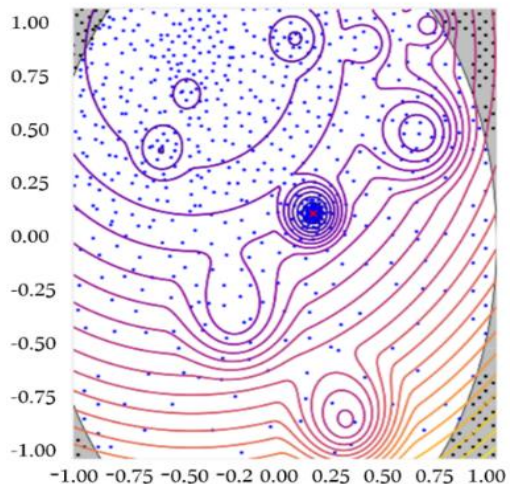
б)



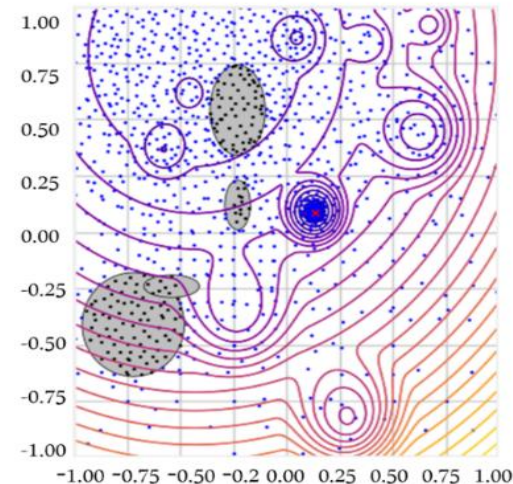
а)



б)



в)



г)

- АГПИ-Н,
alpha = 0.08
- АГПИ-Н,
alpha = 0.008
- АГПИ

