

# A GLOBAL OPTIMIZATION ALGORITHM FOR TUNING HYPERPARAMETERS OF MACHINE LEARNING METHODS

M. A. Usova, I. G. Lebedev, A. A. Shtanyuk, K. A. Barkalov

Lobachevsky State University,  
603022, Nizhny Novgorod, Russia

---

DOI: 10.24412/2073-0667-2025-4-52-72

EDN: XGFNEQ

The paper discusses the problem of finding the best combination of hyperparameters of machine learning and artificial intelligence methods. In many cases, the efficiency (in a given metric) of the resulting solution for different values of hyperparameters can be quite different. In such problems, a significant issue is the potential for incorrect operation of the investigated artificial intelligence and machine learning methods within certain (a priori unknown) subregions of the hyperparameters search domain. Furthermore, the computational complexity of tuning makes manual or exhaustive search inappropriate. These characteristics have required the development of various intelligent automatic hyperparameter optimization methods. From a mathematical point of view, such a task can be represented as the problem of finding a global minimum of a function, given in the form of a “black box” and computable only in some part of the search domain. In this case, each computation of the objective function value at some point of the feasible domain may require significant computing resources. The objective function is assumed to satisfy the Lipschitz condition. The existence of subdomains where the objective function is undefined can be interpreted as the existence of some hidden, a priori unknown constraints of the problem. The authors propose an approach to solving this type of problem, which is an extension of the information-statistical global search algorithm (GSA) and takes into account the presence of undefined values of the objective function at some points. The algorithm partitions the search space with trial points and evaluates the characteristics of subregions based on the objective function values computed at their boundaries. If the function value at a point is unknown, the algorithm employs an estimate for this value, considering the size of the subregion under investigation. To minimize the number of redundant trials in subdomains where the function is not defined, the method parameter  $\alpha$  was used that regulates the number of trial points in regions of non-computability. The solution of multidimensional problems implemented through reducing them to one-dimensional optimization problems using space-filling curves (Peano curves). The article provides a detailed description and a flowchart of the operation of the proposed search algorithm. The implementation of the global search algorithm for the case of a not everywhere computable objective function (GSA-N) was based on the iOpt open source framework of intelligent optimization methods. To carry out the experiments, a generator of test problems with hidden constraints GKLS-HC was developed. It is based on the GKLS generator, which allows generating multi-extremal functions with specified properties (number of minima, their regions of attraction, etc.). In the GKLS-HC generator, these functions were spoiled by areas of non-computability in the form of ellipsoids

---

The work was carried out with the support of the Ministry of Science and Higher Education of the Russian Federation (state assignment N FSWR-2023-0034) and Scientific and Educational Mathematical Center “Mathematics of Future Technologies”.

(the coordinates of centers and radii were generated randomly). The experimental results presented in the paper, obtained on a series of GKLS-HC test problems, demonstrate the reliability of global search and the efficiency of the developed algorithm. By adjusting the parameter  $\alpha$ , it was possible to achieve the same performance of GSA-N operation as the basic GSA. The paper also considered the behavior of global optimization algorithms of the `scipy.optimize` library when solving problems with non-computable domains. In this study, the differential evolution and brute force methods showed the worst results, failing to solve this type of problem at all. The DIRECT and SHGO methods, although they solved the problems, were less effective than the developed GSA-N. Experiments were also conducted with hyperparameter tuning problems where undefined values of the quality metric arise. In these problems, certain hyperparameter combinations caused the method to return infinite values for the objective function. The LinearSVC classification algorithm was successfully tuned. GSA-N effectively solved the problem, identifying a better hyperparameter combination compared to standard scikit-learn algorithms. A time series prediction method from the FEDOT framework was also configured. During this experiment, GSA-N was compared to tuning algorithms from the popular Optuna framework. GSA-N achieved comparable performance in terms of the obtained the target metric value, significantly surpassing Optuna in solution time.

**Key words:** machine learning, hyperparameter tuning, global optimization, black-box functions, partially defined functions.

## References

1. Zhou J. , Qiu Y., Zhu S., Armaghani D. J. , Li C., Nguyen H. , Yagiz S. Optimization of support vector machine through the use of metaheuristic algorithms in forecasting TBM advance rate // *Eng. Appl. Artif. Intell.* 2021. V. 97. P. 104015.
2. Yang W., Xia K., Fan S., Wang L., Li T., Zhang J., Feng Y. A Multi-Strategy Whale Optimization Algorithm and Its Application // *Eng. Appl. Artif. Intell.* 2022. V. 108. P. 104558.
3. Frazier P.I. A Tutorial on Bayesian Optimization // *arXiv*. 2018.
4. Archetti F., Candelieri A. Bayesian Optimization and Data Science. Cham: Springer Briefs in Optimization, 2019.
5. Jones D., Martins J. The direct algorithm: 25 years later // *J. Glob. Optim.* 2021. V. 79, N 3. P. 521–566.
6. Paulavicius R. and Zilinskas J. *Simplicial Global Optimization*. New York: Springer, 2014.
7. Paulavicius R., Sergeev Y.D., Kvasov D.E., Zilinskas J. Globally-biased BIRECT algorithm with local accelerators for expensive global optimization // *Expert Syst. Appl.* 2020. V. 144. P. 113052.
8. Sergeev Ya. D., Kvasov D. E. *Diagonalnye metody globalnoj optimizacii*. M.: Fizmatlit, 2008.
9. Liberti L., Kucherenko S. Comparison of deterministic and stochastic approaches to global optimization // *Int. Trans. Oper. Res.* 2005. V. 12. P. 263–285.
10. Sergeev Y. D., Kvasov D. E., Mukhametzhanov M. S. On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget // *Sci. Rep.* 2018. V. 8, N 1. P. 435.
11. Stripinis L., Paulavicius R. A new DIRECT-GLh algorithm for global optimization with hidden constraints // *Optim. Lett.* 2021. V. 15, N 6. P. 1865–1884.
12. Audet C., Batailly A., Kojtych S. Escaping unknown discontinuous regions in blackbox optimization // *SIAM J. Optim.* 2022. V. 32, N 3. P. 1843–1870.
13. Candelieri A. Sequential model based optimization of partially defined functions under unknown constraints // *J. Glob. Optim.* 2019. V. 79, N 2. P. 281–303.
14. Barkalov K. A., Strongin R. G. Metod globalnoj optimizacii s adaptivnym porjadkom proverki ogranichenij // *Zhurn. vy'chisl. matem. i matem. fiz.* 2002. T. 42, N 9. S. 1338–1350.

15. Strongin R. G., Barkalov K. A., Bevzuk S. A. Global optimization method with dual Lipschitz constant estimates for problems with non-convex constraints // *Soft Comput.* 2020. V. 24, N 16. P. 11853–11865.
16. Sergeyev Y. D., Strongin R. G., Lera D. *Introduction to Global Optimization Exploiting Space-Filling Curves*. New York: Springer Briefs in Optimization, 2013.
17. Strongin R. G., Sergeyev Y. D. *Global optimization with non-convex constraints. Sequential and parallel algorithms*. Dordrecht: Kluwer Academic Publishers, 2000.
18. 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. V. 1914. P. 147–161.
19. Barkalov K. A., et al. On solving the problem of finding kinetic parameters of catalytic isomerization of the pentane-hexane fraction using a parallel global search algorithm // *Mathematics*. 2022. V. 10, N 19. P. 3665.
20. Gubaydullin I. M., Enikeeva L. V., Barkalov K. A., Lebedev I. G., Silenko D. G. Kinetic modeling of isobutane alkylation with mixed c4 olefins and sulfuric acid as a catalyst using the asynchronous global optimization algorithm // *Commun. Comput. Inf. Sci.* 2022. V. 1618. P. 293–306.
21. Barkalov K. A., Lebedev I. G., Gergel V. P. Parallel Global Search Algorithm with Local Tuning for Solving Mixed-Integer Global Optimization Problems // *Lobachevskii Journal of Mathematics*. V. 7. N 42. 2021. P. 1492–1503.
22. Sysoev A. V., Kozinov E. A., Barkalov K. A., Lebedev I. G., Karchkov D. A., Rodionov D. M. Frejmvork metodov intelektualnoj evristicheskoj optimizacii iOpt // *V kn.: Superkompyuternye dni v Rossii: Trudy mezhdunarodnoj konferencii*. 2023. S. 179–185.
23. Ishodnyj kod frejmvorka iOpt. [Electron. Res.]: <https://github.com/aimclub/iOpt> (data obrasheniya: 26.01.2025).
24. Dokumentaciya iOpt. [Electron. Res.]: <https://iopt.readthedocs.io/ru/latest/> (data obrasheniya: 26.01.2025).
25. Gaviano M., Kvasov D. E., Lera D., Sergeyev Y. D. Software for generation of classes of test functions with known local and global minima for global optimization // *ACM Trans. Math. Softw.* 2003. V. 29, N 4. P. 469–480.
26. Storn R., Price K., *Differential Evolution — a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces* // *Journal of Global Optimization*. 1997. V. 11. P. 341–359.
27. Xiang Y, Gubian S, Suomela B, Hoeng J. Generalized Simulated Annealing for Efficient Global Optimization: the GenSA Package for R // *The R Journal*. 2013. V. 5, N 1.
28. Gablonsky J., Kelley C. A Locally-Biased form of the DIRECT Algorithm // *Journal of Global Optimization*. 2001. V. 21. P. 27–37.
29. Wales D. J., Doye J. P. K. Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms // *Journal of Physical Chemistry A*. 1997. V. 101. P. 5111.
30. Endres S. C., Sandrock C., Focke W. W. A simplicial homology algorithm for lipschitz optimisation // *Journal of Global Optimization*. 2018.
31. Filippou K., Aifantis G., Papakostas G. A., Tsekouras G. E. Structure learning and hyperparameter optimization using an automated machine learning (AutoML) pipeline // *Information*. 2023. V. 14, N 4. P. 232.
32. Automated modeling and machine learning framework FEDOT. [Electron. Res.]: <https://github.com/aimclub/FEDOT> (data obrashheniya: 25.07.2025).
33. Xu N. Time Series Analysis on Monthly Beer Production in Australia // *Highlights in Science, Engineering and Technology*. 2024. V. 94. P. 392–401.
34. Akiba T., Sano S., Yanase T., Ohta T., Koyama M. Optuna: A Next-Generation Hyperparameter Optimization Framework // *In Proceedings: 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019. P. 2623–2631.

# АЛГОРИТМ ГЛОБАЛЬНОЙ ОПТИМИЗАЦИИ ДЛЯ НАСТРОЙКИ ГИПЕРПАРАМЕТРОВ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

М. А. Усова, И. Г. Лебедев, А. А. Штанюк, К. А. Баркалов  
ННГУ им. Н. И. Лобачевского,  
603022, Нижний Новгород, Россия

---

УДК 519.853.4

DOI: 10.24412/2073-0667-2025-4-52-72

EDN: XGFNEQ

В статье рассматриваются задачи поиска наилучшего сочетания гиперпараметров методов машинного обучения и искусственного интеллекта. В таких задачах актуальной является проблема некорректной работы исследуемых методов ИИ и МО в некоторых (заранее неизвестных) подобластях области изменения гиперпараметров. С математической точки зрения такая задача может быть представлена как задача поиска глобального минимума функции, заданной в виде «черного ящика» и не всюду определенной в области поиска. Существование подобластей, где целевая функция является неопределенной, можно интерпретировать как наличие некоторых скрытых, заранее неизвестных ограничений. Предложен подход к решению такого рода задач, который является расширением информационно-статистического алгоритма глобального поиска и учитывает наличие неопределенных значений целевой функции в некоторых точках. В рамках предложенного алгоритма проводится разбиение области поиска точками испытаний и оцениваются характеристики подобластей на основе значений целевой функции, вычисленных на их границах. В случае отсутствия информации о значениях функции в алгоритме используются оценка, учитывающая размер исследуемой подобласти. Для сокращения количества испытаний в подобластях, в которых функция не определена, введен специальный параметр метода, позволяющий регулировать число точек испытаний в области невычислимости. Изложено подробное описание и приведена схема работы модифицированного алгоритма глобального поиска. Продемонстрированы результаты его сравнения с другими известными алгоритмами глобальной оптимизации, полученные при проведении численных экспериментов как с тестовыми функциями, так и с модельными задачами настройки гиперпараметров, в которых возникают неопределенные значения оптимизируемой метрики качества.

**Ключевые слова:** машинное обучение, настройка гиперпараметров, глобальная оптимизация, функции вида «черный ящик», частично определенные функции.

**Введение.** В настоящее время методы искусственного интеллекта (ИИ) и машинного обучения (МО) применяются для решения широкого круга задач. К их числу можно отнести ставшие уже классическими задачи распознавания образов, классификации, регрессии. Однако в большинстве методов ИИ и МО имеются гиперпараметры, от выбора конкретных значений которых может зависеть качество (в некоторой метрике) полученного решения. Тот факт, что разница в качестве получаемого решения при разных значениях

---

Работа выполнена при поддержке Министерства науки и высшего образования РФ (проект № FSWR-2023-0034) и научно-образовательного математического центра «Математика технологий будущего».

гиперпараметров может быть достаточно значительной, привел к возникновению класса задач настройки гиперпараметров (НРО — hyperparameter optimization). С математической точки зрения такие задачи соответствуют задачам глобальной оптимизации с фиксированными границами изменения переменных. При этом часть гиперпараметров может быть категориальной, то есть принимать значения из некоторого дискретного множества, что приводит к необходимости использовать методы оптимизации, способные решать задачи с дискретными параметрами.

Основная сложность настройки гиперпараметров состоит в том, что поиск их оптимального сочетания требует (для каждого выбранного варианта) решения исходной задачи машинного обучения, а значит, может быть довольно длительным по времени. Таким образом, любой подход будет ограничен в числе сочетаний гиперпараметров, которое метод глобальной оптимизации сможет проверить, прежде чем будет исчерпан доступный вычислительный ресурс.

Весьма распространенными методами настройки параметров являются метаэвристические (генетические, имитационные и аналогичные им) алгоритмы [1, 2]. Данные алгоритмы широко применяются при отсутствии формульного описания оптимизируемой функции (функция вида «черный ящик»), что характерно для рассматриваемых задач. Метаэвристические алгоритмы слабо зависят от числа параметров задачи, используют информацию с предыдущих итераций для выполнения текущей, однако в силу заложенной в методы случайности дают гарантию отыскания глобального оптимума только в вероятностном смысле.

Еще один подход к поиску глобального оптимума — байесовская оптимизация, также применяемая для решения задач с функциями вида «черный ящик» [3, 4]. Для своей работы методы байесовской оптимизации используют стохастическую модель оптимизируемой функции. Модель итерационно обновляется на основе накапливаемой в процессе поиска оптимума информации, позволяя на каждой очередной итерации оценить наиболее вероятное положение глобального оптимума. Методы байесовской оптимизации обладают более высокой эффективностью по сравнению с метаэвристическими алгоритмами, но в значительной степени подвержены влиянию «проклятия размерности».

Хорошие результаты при использовании методов байесовской оптимизации достигаются, когда целевая функция задачи соответствует определенной стохастической модели, например, гауссовскому процессу. Однако в глобальной оптимизации существуют и иные предположения о виде функции, которые дают замечательные результаты. Одним из таких допущений о решаемой задаче является предположение об ограниченности относительных изменений целевой функции. В этом случае говорят, что функция удовлетворяет условию Липшица, а решаемая задача называется задачей липшицевой глобальной оптимизации. Для класса задач липшицевой глобальной оптимизации разработан целый ряд эффективных алгоритмов [5–8], которые превосходят многие другие методы глобальной оптимизации [9, 10].

В рассматриваемых задачах настройки гиперпараметров актуальной является проблема некорректной работы исследуемых алгоритмов ИИ и МО в некоторых (заранее неизвестных) подобластях области поиска. При этом программные реализации настраиваемых методов могут вести себя абсолютно по-разному: в лучшем случае информируют исследователя о том, что задача не может быть решена, в худшем — возвращают недопустимое значение (NaN, inf) в качестве найденного решения. Проблема возникновения недопусти-

мых комбинаций, которая не имеет особого значения в случае «ручной» настройки, существенно осложняет работу фреймворков автоматической настройки гиперпараметров.

Указанное свойство с математической точки зрения можно интерпретировать либо как наличие в задаче некоторых скрытых ограничений [11], либо как наличие неизвестных областей, в которых целевая функция не является непрерывной [12], либо как частичную вычислимость целевой функции в области поиска [13–15]. В такой постановке задача оптимизации существенно усложняется, т. к. область допустимых сочетаний параметров является заранее неопределенной.

Настоящая работа продолжает развитие одного из эффективных детерминированных методов решения задач липшицевой глобальной оптимизации — информационно-статистического алгоритма глобального поиска [16, 17]. В статье приведено описание нового алгоритма, адаптированного для работы с частично определенной целевой функцией. Данный алгоритм основан на подходе, предложенном авторами ранее в [18]. Проведено исследование поведения других известных алгоритмов при решении задач с не всюду вычислимой целевой функцией. Продемонстрированы результаты численных экспериментов как с тестовыми функциями, так и с модельными задачами настройки гиперпараметров, в которых возникают неопределенные значения целевой функции. А именно, проведена настройка гиперпараметров метода LinearSVC при решении задачи классификации на наборе данных Iris, а также метода предсказания значений временного ряда для датасета monthly beer production с платформы Kaggle.

**1. Постановка задачи настройки гиперпараметров.** В общем виде задача настройки гиперпараметров (которой соответствует задача глобальной оптимизации) может быть сформулирована следующим образом:

$$\phi^* = \phi(y^*) = \min_{y \in D} \phi(y), \quad (1)$$

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

где  $y = (y_1, y_2, \dots, y_N)$  — вектор варьируемых параметров,  $D$  —  $N$ -мерный гиперкуб,  $N$  — размерность решаемой задачи. О целевой функции  $\phi(y)$  мы делаем следующие предположения.

1) Целевая функция может быть многоэкстремальной, недифференцируемой и, более того, заданной в форме «черного ящика» (т. е. в виде некоторой подпрограммы, на вход которой подается аргумент, а выходом является соответствующее значение функции).

2) Каждое вычисление функции в некоторой точке допустимой области может требовать значительных вычислительных ресурсов.

3) Целевая функция удовлетворяет условию Липшица

$$|\phi(y') - \phi(y'')| \leq L \|y' - y''\|, \quad y', y'' \in D, \quad (2)$$

где  $0 < L < \infty$  — константа Липшица.

4) В некоторой подобласти  $I$  области поиска  $D$  (в частном случае, в одной или нескольких ее точках) целевая функция может быть не определена. Тогда функция  $\phi(y)$  определена и вычислима лишь в подобласти  $Q = D \setminus I$  (положительного объема). Отметим, что исходя из опыта решения прикладных оптимизационных задач [19, 20], суммарный объем области невычислимости  $I$  составляет небольшую долю объема области поиска  $D$ .

Последнее предположение делает невозможными применение известного информационно-статистического алгоритма глобального поиска [17] или других методов липшицевой

оптимизации [6, 8]. Для решения таких задач нами была разработана модификация алгоритма глобального поиска, основанная на предложенном ранее подходе из [18].

1.1. *Редукция размерности.* Используя кривые типа развертки Пеано, однозначно отображающие отрезок  $[0,1]$  на  $N$ -мерный единичный гиперкуб

$$D = \{y \in R^N : -2^{-1} \leq y_i \leq 2^{-1}, 1 \leq i \leq N\} = \{y(x) : 0 \leq x \leq 1\}, \quad (3)$$

исходную задачу (1) можно редуцировать к одномерной задаче

$$f^*(x) = \phi(y(x^*)) = \min_{x \in [0,1]} \{\phi(y(x))\}, \quad (4)$$

что позволяет применить для ее решения эффективные алгоритмы одномерной оптимизации.

Известно, что схема редукиции размерности с использованием кривых Пеано сопоставляет многомерной задаче с липшицевой целевой функцией (1) задачу (4) с одномерной целевой функцией, удовлетворяющей условию Гельдера

$$|f(x') - f(x'')| \leq K\rho(x', x''), \quad x', x'' \in [0,1], \quad (5)$$

где  $\rho(x', x'') = |x' - x''|^{1/N}$  — метрика Гельдера,  $N$  — размерность исходной задачи, а коэффициент  $K$  связан с константой Липшица  $L$  соотношением  $K \leq 2L\sqrt{N+3}$  [17].

## 2. Алгоритм решения задач с частично определенной целевой функцией.

Алгоритм глобального поиска предполагает построение последовательности точек *поисковых испытаний*  $y^i \in D$ , в которых вычисляются значения целевой функции  $z^i = \varphi(y^i)$ . Согласно используемой схеме редукиции размерности, проведение испытания предполагает вычисление значения  $y^i = y(x^i)$ ,  $x^i \in [0,1]$ . При этом каждой точке испытания  $x^i$  ставится в соответствие индекс  $v^i$ , определяемый по правилу

$$v(x^i) = \begin{cases} -1, & \text{если } x^i \text{ — граничная точка,} \\ 0, & \text{если } x^i \text{ — невычислимая точка,} \\ 1, & \text{если } x^i \text{ — внутренняя точка.} \end{cases} \quad (6)$$

Результатом испытания является набор значений  $(x^i, y^i = y(x^i), z^i = \varphi(y^i), v^i = v(x^i))$ .

Накопленная после проведения  $k$  испытаний поисковая информация хранится в порядке возрастания значений координаты  $x^i$  в множестве  $\Omega_k$ , в котором соответствующие записи для удобства перенумерованы нижним индексом.

На каждом шаге алгоритма производятся вычисление оценки константы Гельдера, определение характеристик поисковых интервалов, выбор наиболее перспективного интервала и вычисление точки  $y^{k+1}$  для проведения очередного испытания. Правила вычисления характеристики интервала для задач с частично определенной целевой функцией представлены в виде Алгоритма 1.

*Алгоритм 1.* Вычисление характеристики  $i$ -го интервала.

**Require:**  $\Omega_k, i, z^*, \mu, r > 1, 0 < \alpha \leq 1$

**Ensure:**  $R$

- 1: **if**  $v(x_{i-1}) = 1$  and  $v(x_i) = 1$  **then** ▷ обе точки являются внутренними
- 2:  $R \leftarrow \Delta_i + \frac{(z_i - z_{i-1})^2}{(r\mu)^2 \Delta_i} - 2 \frac{(z_i + z_{i-1} - 2z^*)}{r\mu}$

```

3: else if  $v(x_i) = 1$  then                                ▷ точка  $x_i$  является внутренней
4:    $R \leftarrow 2\Delta_i - 4\frac{(z_i - z^*)}{r\mu}$ 
5: else if  $v(x_{i-1}) = 1$  then                            ▷ точка  $x_{i-1}$  является внутренней
6:    $R \leftarrow 2\Delta_i - 4\frac{(z_{i-1} - z^*)}{r\mu}$ 
7: else                                                    ▷ обе точки являются невычислимыми или граничными
8:    $R \leftarrow \alpha(1 - \frac{1}{r})^2 \Delta_i$ 
9: end if
10: return R

```

Значения  $r$  и  $\alpha$  являются параметрами алгоритма (их смысл пояснен ниже), а  $\Delta_i$  обозначает длину  $i$ -го интервала в новой метрике,

$$\Delta_i = (x_i - x_{i-1})^{1/N}.$$

Подробное описание правил алгоритма глобального поиска на случай не всюду вычислимой целевой функции для решения задачи (4) приведено в виде Алгоритма 2.

*Алгоритм 2.* Алгоритм глобального поиска для не всюду вычислимой целевой функции (АГП-Н).

**Require:**  $\varphi(y), D, \varepsilon, K_{max}, r > 1, 0 < \alpha \leq 1$

**Ensure:**  $y^{min}$

```

1:  $\Omega_k \leftarrow \{(x^1 = 0.0, y^1 = y(x^1), v^1 = v(x^1)), (x^2 = 1.0, y^2 = y(x^2), v^2 = v(x^2))\}$ 
2:  $\Omega_k \leftarrow \Omega_k \cup \{(x^3 = 0.5, y^3 = y(x^3), z^3 = \varphi(y^3), v^3 = v(x^3))\}$ 
3:  $z^* \leftarrow z^3, k \leftarrow 3; t \leftarrow 2$ 
4: while  $\Delta_t \geq \varepsilon$  and  $k \leq K_{max}$  or  $v(x_t) \neq 1$  and  $v(x_{t-1}) \neq 1$  do
5:    $\mu \leftarrow \max \left\{ \frac{|z_i - z_{i-1}|}{\Delta_i}, v(x_{i-1}) = 1 \text{ and } v(x_i) = 1, 1 \leq i \leq k \right\}$ 
6:   if  $\mu = 0$  then
7:      $\mu = 1$ 
8:   end if
9:   for  $i = 1$  to  $k$  do
10:     $R_i \leftarrow R(\Omega_k, i, z^*, \mu, r, \alpha)$ 
11:   end for
12:    $t \leftarrow \min \{ \arg \max \{ R_i, 1 \leq i \leq k \} \}$ 
13:   if  $v(x_{t-1}) = 1$  and  $v(x_t) = 1$  then
14:      $x^{k+1} \leftarrow \frac{x_t + x_{t-1}}{2} - \text{sign}(z_t - z_{t-1}) \frac{1}{2r} \left[ \frac{|z_t - z_{t-1}|}{\mu} \right]^N$ 
15:   else
16:      $x^{k+1} \leftarrow \frac{x_t + x_{t-1}}{2}$ 
17:   end if
18:    $y^{k+1} \leftarrow y(x^{k+1}); z^{k+1} \leftarrow \varphi(y^{k+1}); v^{k+1} \leftarrow v(x^{k+1})$ 
19:    $z^* = \min \{ z^*, z^{k+1} \}$ 
20:    $\Omega_k \leftarrow \Omega_k \cup \{(x^{k+1}, y^{k+1}, z^{k+1}, v^{k+1})\}$ 
21:    $k \leftarrow k + 1$ 
22: end while
23:  $min \leftarrow \arg \min \{ z^i, 1 < i < k \}$ 
24: return  $y^{min}$ 

```

**Замечание 1 (о связи с АГП).** В случаях, когда задача не содержит скрытых ограничений или проводимые алгоритмом испытания не попадают в существующие области

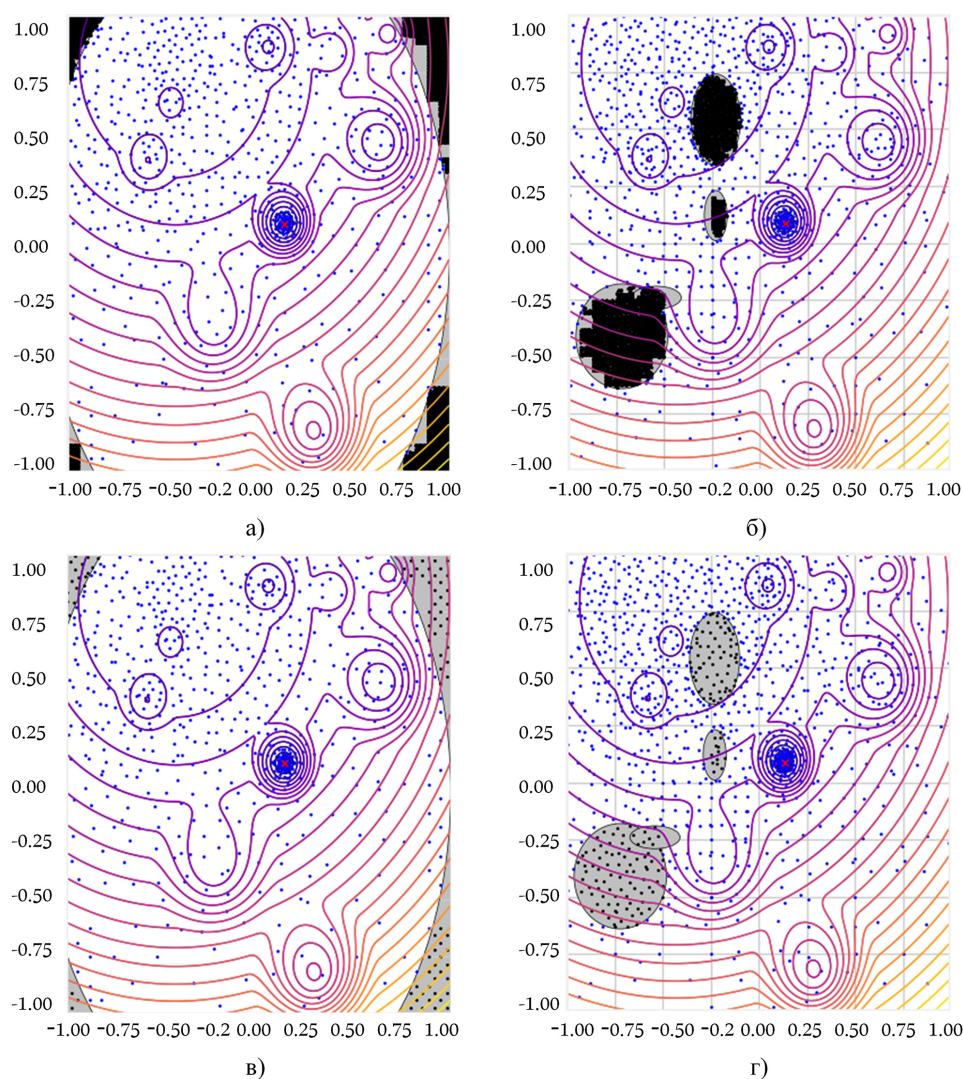


Рис. 1. Распределение точек испытаний с различными видами невычислимых областей и настройкой параметра  $\alpha$ : (а) область невычислимости на границе области поиска,  $\alpha = 1.0$ ; (б) случайно расположенные области невычислимости,  $\alpha = 1.0$ ; (в) область невычислимости на границе области поиска,  $\alpha = 0.08$ ; (г) случайно расположенные области невычислимости,  $\alpha = 0.08$

невычислимости, АГП-Н полностью повторяет работу своего прототипа — алгоритма глобального поиска.

**Замечание 2 (о параметре  $\alpha$ ).** Параметр  $0 < \alpha \leq 1.0$  позволяет регулировать плотность сетки испытаний в области невычислимости  $I$ ,  $I \subset D$ . Чем меньше параметр, тем менее перспективными для алгоритма будут интервалы с граничными точками в областях невычислимости. При максимально допустимом параметре  $\alpha = 1.0$  метод густо усеивает точками испытаний подобласти, в которых функция не вычислима, возникает эффект «черного пятна» (точки испытаний в таких подобластях на рисунке обозначены черным цветом), см. рис. 1, а, и рис. 1, б. При уменьшении параметра  $\alpha$  точек этих испытаний становится значительно меньше, что наглядно видно на рис. 1, в, и рис. 1, г.

**Замечание 3 (о существовании вычислимого отрезка на кривой Пеано).** В случае попадания точек испытаний в область невычислимости на первых итерациях, ал-

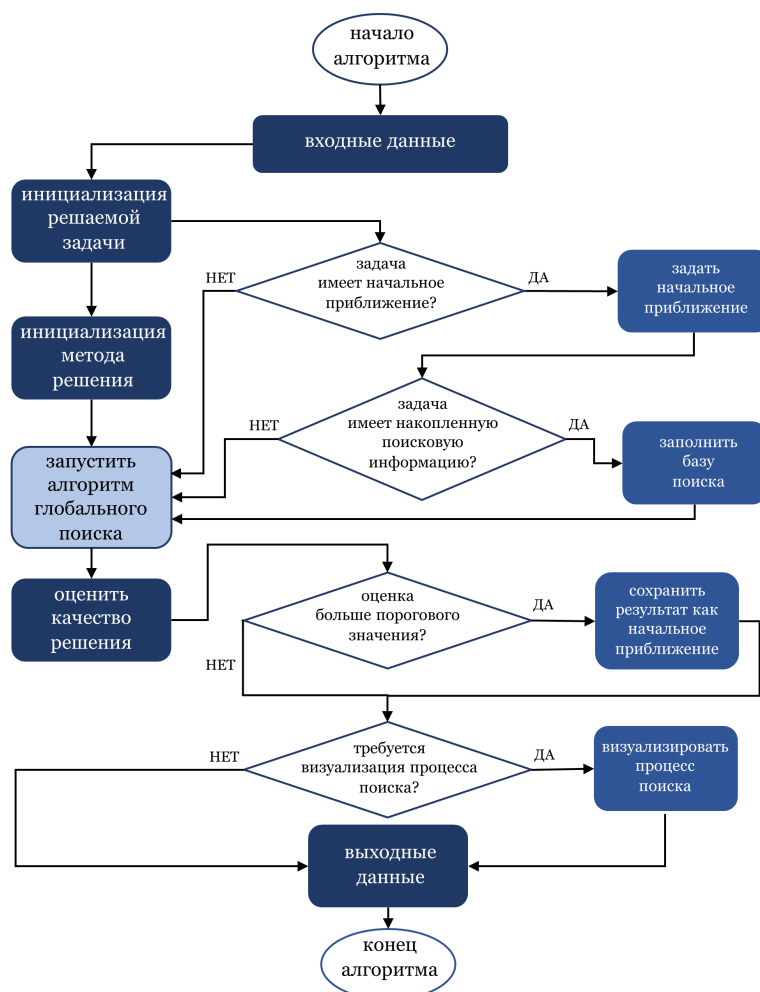


Рис. 2. Блок-схема общего алгоритма функционирования фреймворка iOpt

горитм будет выбирать лучший интервал на основе длин интервалов (т. е. будет строиться равномерная сетка). Так как мы предполагаем, что область вычислимости целевой функции  $Q = D \setminus I$  — положительного объема, то последующее сгущение сетки приведет к попаданию очередной точки испытания в область  $Q$ .

**Замечание 4 (о решении задач с дискретными параметрами).** Алгоритм глобального поиска допускает обобщение для решения задач, в которых часть переменных являются непрерывными, а часть может принимать только дискретные значения [21]. Это позволяет применять АГП-Н в задачах, где часть параметров объекта оптимизации может принадлежать некоторому дискретному множеству. Подобные задачи часто возникают при настройке гиперпараметров методов машинного обучения.

**3. Программная реализация.** Алгоритм глобального поиска, предназначенный для решения задач с не всюду вычислимой целевой функцией (АГП-Н), был реализован на базе открытого фреймворка методов интеллектуальной оптимизации iOpt [22, 23].

Фреймворк позволяет проводить точную настройку параметров моделей и методов, используемых в прикладных исследованиях в различных научных областях. Характерными примерами задач, решаемых фреймворком iOpt, являются задачи настройки гиперпараметров методов машинного обучения. В таких задачах часто возникает рассматриваемая

в данном исследовании проблема частичной определенности критериев в отсутствии формульного описания исследуемой модели вида «черный ящик».

На рис. 2 приведена блок-схема общего алгоритма функционирования фреймворка.

Фреймворк `iOpt` написан на Python в виде системы взаимосвязанных классов. Реализация описанного в разделе 2 алгоритма содержится в базовом классе `Method`. Описанный далее в разделе 4.1.1 генератор задач GKLS со скрытыми ограничениями (GKLS-НС) реализован в виде класса `GKLSHiddenConstraint` — наследника интерфейса `Problem`. В фреймворк добавлена возможность визуализации процесса решения задач со скрытыми ограничениями с использованием `StaticPainterNDListener`. Доступны различные способы отрисовки линий уровня целевой функции: по равномерной сетке, только по точкам поисковых испытаний, по точкам испытаний с использованием интерполяции. Подробное описание возможностей фреймворка представлено в документации [24].

#### 4. Результаты вычислительных экспериментов.

4.1. *Решение серий тестовых задач.* Решение серии задач с известными свойствами является одним из традиционных способов оценки качества работы методов глобального поиска. Однако доступные наборы тестовых задач не содержат областей невычислимости целевой функции, поэтому для тестирования разработанного алгоритма АГП-Н был предложен способ генерации тестовых задач со скрытыми ограничениями.

В данном разделе дано краткое описание тестовых задач, а также приведены результаты экспериментов, демонстрирующие надежность глобального поиска и эффективность разработанного алгоритма оптимизации.

4.1.1. *Описание тестовых задач.* Для создания тестовых задач GKLS-НС использовался генератор GKLS [25], позволяющий порождать задачи многоэкстремальной оптимизации с заранее известными свойствами.

Генерация задачи состоит в определении выпуклой квадратичной функции, дополненной полиномами более высокого порядка для введения локальных минимумов. Каждый тестовый класс, предоставляемый генератором GKLS, состоит из 100 функций, построенных случайным образом, и определяется размером области поиска, количеством локальных минимумов, значением глобального минимума, радиусом области притяжения глобального минимума, расстоянием от глобального минимума до вершины квадратичной функции. Другие необходимые параметры, такие как точки всех локальных минимумов, их области притяжения и значения, выбираются генератором случайным образом.

Данные функции были дополнены областями невычислимости в форме эллипсоидов (координаты центров и длины радиусов генерировались случайно), при попадании в которые функция возвращает исключение, сообщающее о невозможности вычислить ее значение.

В текущей версии фреймворк `iOpt` использует генератор GKLS для формирования задач с размерностью от 2 до 5, числом локальных минимумов, равным 10, и областью поиска от  $-1$  до  $1$  по каждой переменной. Радиус области притяжения глобального минимума и расстояние от глобального минимума до вершины квадратичной функции варьируется в зависимости от размерности задачи. Количество генерируемых областей невычислимости задавалось равным 4. Области невычислимости могут пересекаться, при этом не включают точку известного глобального минимума функций GKLS по построению. Радиусы областей невычислимости выбираются случайно из диапазона  $[0.05, 0.25]$ .

Если говорить о программной реализации, то генератор задач реализован в виде класса `GKLSHiddenConstraint`, конструктор которого принимает размерность генерируемой

Таблица 1

Результаты решения задач GKLS и GKLS-НС средствами фреймворка iOpt

класс задач	$\alpha$	среднее число итераций
GKLS-НС	0.08	2151
	0.008	1635
GKLS	–	1510

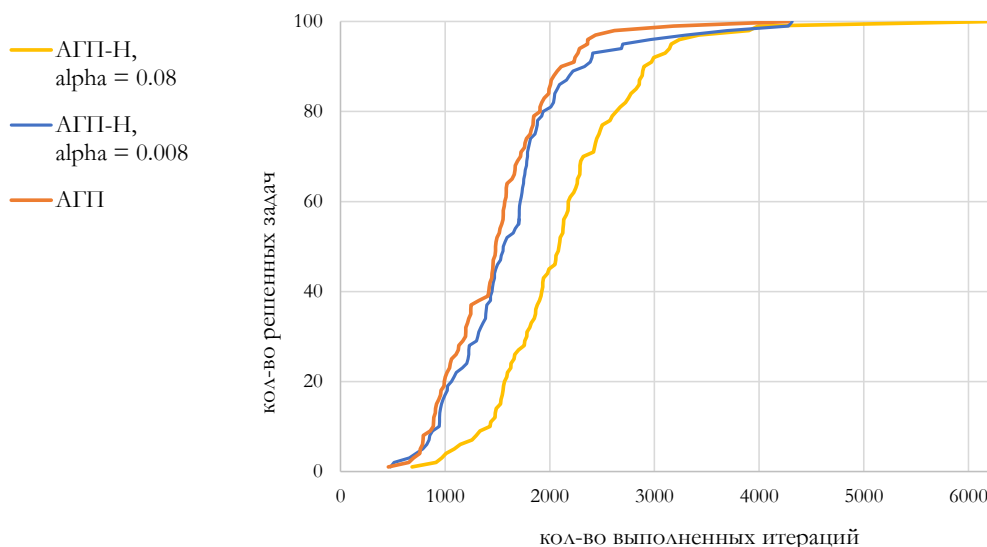


Рис. 3. Операционные характеристики алгоритмов АГП-Н и АГП

задачи (от 2 до 5), номер функции (от 1 до 100), а также параметр для настройки генератора случайных чисел для воспроизведения экспериментов, после чего автоматически устанавливает остальные параметры генератора.

4.1.2. *Результаты решения тестовых задач с помощью АГП-Н.* Вычислительный эксперимент был проведен с использованием фреймворка iOpt, описанного в разделе 3. АГП-Н запускался на серии задач GKLS-НС с точностью поиска минимума  $\varepsilon = 0.001$ , параметром надежности  $r = 4.2$  и двумя значениями параметра  $\alpha$ :  $\alpha = 0.08$  и  $\alpha = 0.008$ . Для полноценного сравнения алгоритм АГП-Н также был запущен на серии задач GKSL (т.е. на задачах без подобластей, в которых целевая функция является неопределенной), что соответствует поведению исходного АГП.

Результаты эксперимента приведены в табл. 1.

На рис. 3 показаны операционные характеристики, которые демонстрируют для каждого количества итераций число задач из тестовой выборки, решенное с заданной точностью  $\varepsilon = 0.001$ .

Примеры визуализации процесса решения задачи GKLS-НС № 38 представлены на рис. 4. Черными точками отмечены точки испытаний, попавшие в подобласти неопределенности функции. Красной точкой отмечено найденное алгоритмом решение.

Результаты проведенных экспериментов показывают, что после выполнения 2000 итераций АГП-Н с параметром  $\alpha = 0.008$  будет получена оценка оптимума с требуемой точностью для 80 % задач. Для решения оставшихся 20 % задач требуется (в худшем случае)

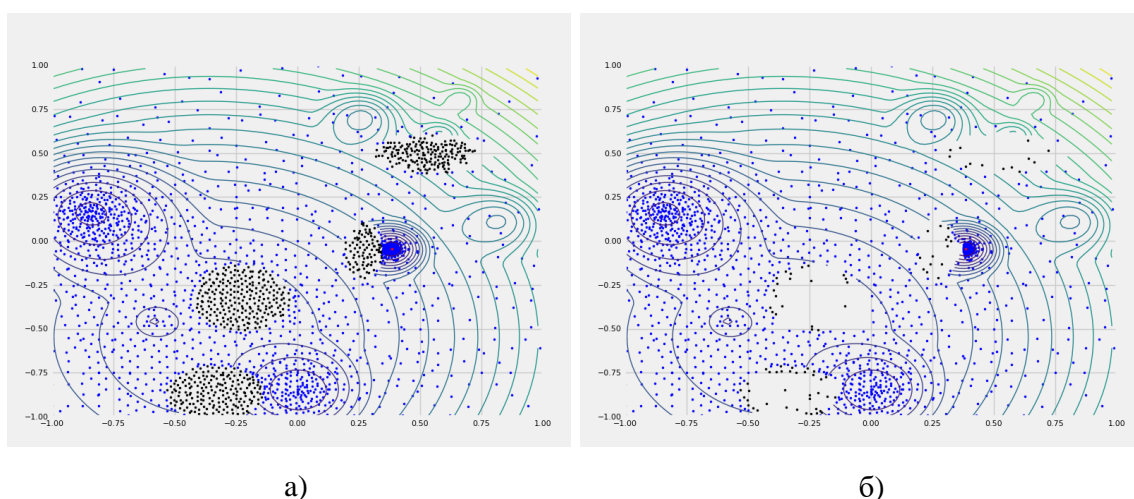


Рис. 4. Графики линий уровня задачи GKLС-НС № 38 с точками проводимых испытаний, полученные средствами фреймворка iOpt: (а)  $\alpha = 0.08$ , (б)  $\alpha = 0.008$

4315 поисковых испытаний. Отметим, что за счет регулировки параметра  $\alpha$  получилось добиться такой же скорости работы, как у АГП при решении серии задач GKLС (в терминах числа испытаний, требующихся для решения задач с заданной точностью).

4.2. *Сравнение АГП-Н с известными алгоритмами оптимизации.* АГП-Н может быть использован не только в задаче настройки гиперпараметров различных методов, но и для решения рядовых оптимизационных задач из различных прикладных областей. В связи с этим имеет смысл привести сравнение реализованного метода с существующими оптимизаторами на задачах с не всюду вычислимой целевой функцией.

4.2.1. *Описание алгоритмов оптимизации.* В данной статье сравнение проводилось с алгоритмами глобальной оптимизации библиотеки *scipy.optimize*, а именно с методами *differential\_evolution* [26], *dual\_annealing* [27], *direct* [28], *basinhopping* [29], *shgo* [30], *brute*.

Метод дифференциальной эволюции (differential evolution) — один из самых известных генетических алгоритмов вещественной оптимизации. Во многих практических задачах метод обеспечивает приемлемое решение, т.к. не использует информацию о градиенте функции, а схемы мутации и рекомбинации позволяют ему «выходить» из локальных минимумов.

Двойной отжиг (dual annealing) — алгоритм глобальной оптимизации, основанный на комбинировании алгоритмов классического и быстрого имитационного отжига. Алгоритм имитирует процесс отжига в металлургии. Отжиг заключается в нагревании и контролируемом охлаждении металла для формирования устойчивой кристаллической решетки: при высокой температуре системы алгоритм использует более широкий диапазон решений, при снижении температуры диапазон поиска становится все меньше, пока не будет найден глобальный оптимум.

Direct (Dlviding RECTangles), как и АГП, предназначен для решения задач липшицевой глобальной оптимизации. Особенностью метода является отказ от оценивания константы Липшица. В процессе своей работы алгоритм разбивает область поиска на гиперинтервалы, среди которых выбираются потенциально оптимальные (подлежащие дальнейшему разбиению), после чего в их центрах проводится вычисление значений целевой функции.

Перепрыгивание через бассейн (basin-hopping) — алгоритм глобальной оптимизации, осуществляющий поиск оптимума путем последовательного выполнения локальной оптимизации (по умолчанию используя метод Монте-Карло) с последующим принятием или отклонением новых координат на основе текущего лучшего значения функции.

SHGO (simplicial homology global optimization) — сравнительно новый алгоритм глобальной оптимизации, основанный на применении симплициальной гомологии. Алгоритм использует концепции из комбинаторной интегральной теории гомологии для поиска под областей, которые приблизительно локально выпуклы. В отличие от многих других алгоритмов глобальной оптимизации, использующих теорию графов и методы кластеризации, SHGO создает начальные точки, которые сходятся к уникальным локальным минимумам, что улучшает его производительность.

Метод brute реализует метод «грубой силы» (brute force), т. е. полный перебор.

Алгоритмы *differential\_evolution*, *dual\_annealing*, *basinhopping* относятся к классу стохастических алгоритмов (т. е. каждый запуск поиска может найти другое решение), потому поиск глобального минимума с использованием данных алгоритмов требует многократных запусков.

4.2.2. *Результаты проведенного сравнения.* В рамках проведенного эксперимента были получены следующие результаты.

При наличии у задачи скрытых ограничений худшее поведение демонстрируют алгоритмы *differential\_evolution* и *brute*. Алгоритм *differential\_evolution* полностью исчерпывает все выделенные ему итерации, не находя при этом корректный ответ. Полный перебор в свою очередь в качестве результата работы может вернуть что угодно.

Алгоритмы *dual\_annealing* и *basinhopping* в ряде случаев справляются с поставленной оптимизационной задачей. При этом в рамках эксперимента количество вычислений целевой функции алгоритмом *dual\_annealing* составило порядка 4000, в то время как *basinhopping* мог выполнять и 140000 вычислений для одной задачи.

Алгоритмы *direct* и *shgo* демонстрируют хорошие результаты, хоть и не всегда обладают сходимостью. Среднее число итераций алгоритма *direct* составило 94 (соответствует 1981 вычислениям целевой функции). Алгоритм *shgo* в каждой задаче полностью исчерпал выделенные ему 200 итерации, выполняя в среднем 4482 вычисления целевой функции. В сравнении с данными методами АПП-Н работает эффективнее (отметим, что одна итерация АПП-Н соответствует одному вычислению значения целевой функции) и обладает лучшей сходимостью к глобальному оптимуму.

В табл. 2 собрана информация о возможном поведении исследованных алгоритмов с задачами с не всюду определенной целевой функцией, обнаруженная при решении задач GKLS-НС. Алгоритм АПП-Н всегда находит глобальный минимум, тогда как *differential\_evolution* всегда возвращает значение nan. Остальные алгоритмы могут вернуть как глобальный, так и локальный минимум, при этом алгоритмы *dual\_annealing* и *basinhopping* в ряде случаев могут вернуть значения nan и  $1e+100$  соответственно, а алгоритм *brute*, как уже было отмечено, и вовсе может вернуть в качестве ответа что угодно.

4.3. *Решение задач настройки методов машинного обучения.*

4.3.1. *Настройка гиперпараметров метода LinearSVC.* В качестве первой модельной задачи оптимизации, в которой встречается недопустимое сочетание параметров, рассмотрим настройку гиперпараметров метода LinearSVC из библиотеки *scikit-learn* при решении задачи классификации.

Таблица 2

Возможные выявленные результаты работы алгоритмов глобальной оптимизации библиотеки *scipy.optimize* на задачах GKLS-НС

алгоритм	возможный результат поиска			
	глобальный минимум	локальный минимум	nan	1e+100
АГП-Н	●			
differential_evolution			●	
dual_annealing	●		●	
basinhopping	●	●		●
direct	●	●		
shgo	●	●		
brute	●	●	●	●

Исследование проводилось на классическом датасете Iris, который содержит характеристики трех различных видов ириса и часто используется в качестве тестового примера для алгоритмов машинного обучения. Набор данных Iris состоит из 150 записей (по 50 записей на каждый вид ириса), содержащих по 5 атрибутов. Датасет включен в библиотеку машинного обучения *scikit-learn*.

Задача заключалась в максимизации метрики  $F1$ , характеризующей качество построения классификатора. Для LinearSVC варьировались 2 дискретных и 1 непрерывный параметры: тип функции потерь  $loss$ ,  $loss \in \{hinge, squared\_hinge\}$ ; параметр  $dual$ ,  $dual \in \{true, false\}$ , который позволяет выбирать между решением двойственной или основной задачи оптимизации; коэффициент регуляризации  $C$ ,  $C \in [1,6]$ .

Для возможности воспроизведения результатов параметр  $random\_state$  был зафиксирован в значении 10. Остальные параметры метода LinearSVC рассматривались заданными по умолчанию.

Отметим, что сочетание гиперпараметров  $loss = hinge$  и  $dual = False$  приводит к «падению» метода LinearSVC (метод выбрасывает исключение и не выдает значения целевой метрики).

Поставленная задача решалась в области изменения гиперпараметров с использованием следующих алгоритмов: АГП-Н (предложенный в данной статье) с использованием параметра надежности  $r = 3.5$  и ограничением на число испытаний  $K_{max} = 100$ ; GridSearchCV (метод полного перебора по равномерной сетке); RandomizedSearchCV (метод случайного поиска).

Полученные результаты представлены в табл. 3. Отметим, что в процессе своей работы оба алгоритма библиотеки *scikit-learn* решили задачу с выдачей предупреждения о возникновении недопустимого сочетания параметров. Алгоритм АГП-Н нашел лучшее значение  $F1 = 0.973$  при сочетании значений гиперпараметров  $C = 1.3125$ ,  $loss = squared\_hinge$ ,  $dual = False$ .

4.3.2. *Настройка гиперпараметров метода предсказания значений временного ряда.* В качестве второй модельной задачи рассмотрим задачу настройки гиперпараметров методов предсказания значений временного ряда (time series forecasting). Предсказание значений производилось с использованием фреймворка FEDOT [31, 32] (фреймворк с открытым исходным кодом для задач автоматизированного моделирования и машинного обучения, AutoML). FEDOT поддерживает полный цикл решения задачи, включающий препроцес-

Таблица 3

Результаты настройки гиперпараметров метода LinearSVC  
при решении задачи классификации на датасете Iris

метод	финальная F1
АГП-Н	0.973
GridSearchCV	0.960
RandomizedSearchCV	0.960

синг, выбор модели, настройку, кроссвалидацию и т. д. Фреймворк использует ML-модели, в основном из стандартных библиотек *sklearn*, *statsmodels* и *keras*.

Исследование проводилось на датасете *monthly beer production*, который содержит 476 записей. Указанный датасет представлен на платформе Kaggle и часто используется для тестирования алгоритмов машинного обучения (см., например, [33]). Построенный для решения поставленной задачи в FEDOT пайплайн включает в себя комбинацию из двух преобразований: *lagged* и *cgru*. Lagged-преобразование выполняется путем взятия значения переменной в предыдущий момент времени и включения его в модель в качестве признака в текущий момент времени. Для этого данные временного ряда сдвигаются на определенное количество шагов, которые называются лагом или временной задержкой. В качестве используемой нейронной сети выступает рекуррентная сеть (GRU), использующая свертку (CGRU).

У lagged-преобразования имеется один параметр *window\_size*, который изменяется от 5 до 250 с шагом 1. Поскольку этот параметр имеет большое количество значений, при оптимизации будем рассматривать его как непрерывный (с округлением 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*  $\in [0.0005, 0.005]$ .

Остальные параметры были зафиксированы в значениях по умолчанию: *batch\_size* = 64, *num\_epochs* = 50, *optimizer* = 'adamw', *loss* = 'mse'.

Таким образом, фреймворк iOpt решал задачу минимизации метрики *MSE*, варьируя 3 непрерывных и 4 дискретных параметра, каждый из которых принимал 2 значения (всего 16 комбинаций). Отметим, что при некоторых сочетаниях параметров *window\_size*, *cnn1\_kernel\_size*, *cnn1\_output\_size*, *cnn2\_kernel\_size* и *cnn2\_output\_size* невозможно вычислить значение критерия (метод возвращает бесконечное значение функции). Такие точки рассматривались как невычислимые.

Для сравнения поставленная задача также была решена при помощи известного фреймворка оптимизации гиперпараметров Optuna [34]. Оба решателя были интегрированы в FEDOT и задействованы в вычислительном эксперименте.

Для решения поставленной задачи было выставлено ограничение на 1000 поисковых испытаний. В iOpt после 950 «глобальных» итераций глобального метода запускалось локальное уточнение с использованием метода Хука-Дживса. Результаты проведенного эксперимента отражены в табл. 4.

Число точек, в которых не получилось вычислить критерий оптимизации, составило порядка 10 % от общего числа точек поисковых испытаний. Кривые, соответствующие полученному предсказанию, приведены на рис. 5. Темная линия соответствует истинным значениям временного ряда, голубая — предсказанным значениям, полученным после ис-

Таблица 4

Результаты решения задачи настройки гиперпараметров метода предсказания значений временного ряда на датасете *monthly beer production* с использованием фреймворков *iOpt* и *Optuna*

фреймворк	время решения, сек.	финальная MSE
<i>iOpt</i>	6574.2	13.5
<i>Optuna</i>	9554.1	13.4

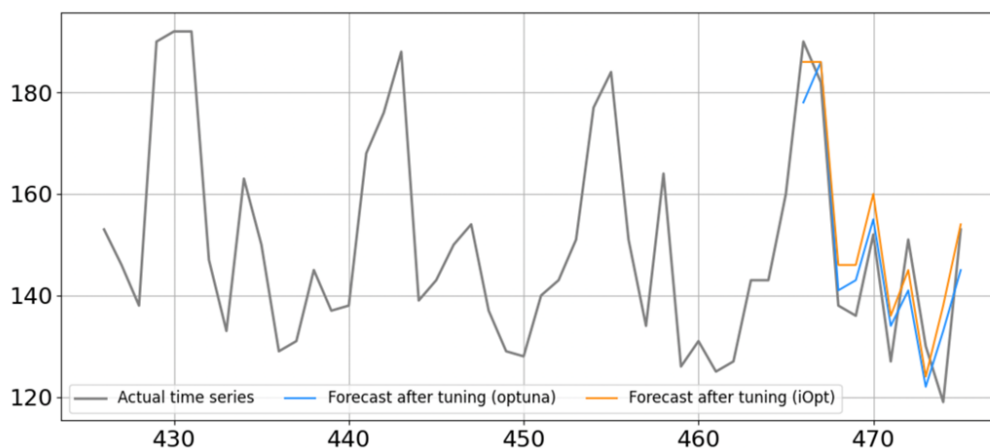


Рис. 5. Визуализация предсказания и истинного значения временного ряда

пользования *Optuna*, желтая линия — предсказанным значениям, полученным после использования *iOpt*.

**Заключение.** В настоящей работе рассматривалась актуальная проблема решения задач настройки гиперпараметров методов искусственного интеллекта (ИИ) и машинного обучения (МО). Сложность таких задач заключается в большой трудоемкости оценки метрик качества методов ИИ и МО для заданного сочетания гиперпараметров, что ограничивает возможное число проверяемых комбинаций. Одновременно с этим исследуемые методы ИИ и МО могут работать некорректно в некоторых заранее неизвестных подобластях всей области изменения гиперпараметров; фактически, область допустимых сочетаний гиперпараметров является неопределенной. Проблема возникновения некорректных сочетаний гиперпараметров, которая ранее не имела особого значения при «ручной» настройке, при использовании автоматической настройки стала весьма актуальной.

Для решения указанного класса задач была разработана модификация одного из эффективных детерминированных методов решения задач глобальной оптимизации — информационно-статистического алгоритма глобального поиска. Предложенный алгоритм глобального поиска для не всюду вычислимых функций (АГП-Н) реализован на базе фреймворка методов интеллектуальной оптимизации с открытым исходным кодом *iOpt*.

В статье подробно описаны вычислительные правила и приведена схема работы АГП-Н. Представлены результаты экспериментов на нескольких сотнях тестовых задач, демонстрирующие надежность предложенного подхода к обработке неопределенных значений оптимизируемой функции. Также приведены результаты сравнения работы АГП-Н и стандартных алгоритмов оптимизации из библиотеки *SciPy*, которые подтверждают эффективность разработанного метода при решении тестового класса задач.

В качестве модельных примеров были решены несколько задач настройки гиперпараметров, где возникает проблема возникновения недопустимых сочетаний гиперпараметров, при которых невозможно оценить целевую метрику качества. В первом примере была проведена настройка гиперпараметров классического алгоритма классификации LinearSVC. АГП-Н успешно решил поставленную задачу, найдя лучшее сочетание гиперпараметров по сравнению со стандартными алгоритмами из библиотеки scikit-learn. В качестве второй модельной задачи была рассмотрена настройка гиперпараметров метода предсказания значений временного ряда. В ходе эксперимента было проведено сравнение АГП-Н с алгоритмами настройки из широко известного фреймворка Optuna. АГП-Н показал сравнимый результат (в терминах полученного значения целевой метрики), значительно обогнав Optuna по времени, затраченному на решение задачи.

### Список литературы

1. Zhou J. , Qiu Y., Zhu S., Armaghani D. J. , Li C., Nguyen H. , Yagiz S. Optimization of support vector machine through the use of metaheuristic algorithms in forecasting TBM advance rate // Eng. Appl. Artif. Intell. 2021. V. 97. P. 104015. DOI: 10.1016/j.engappai.2020.104015.
2. Yang W., Xia K., Fan S., Wang L., Li T., Zhang J., Feng Y. A Multi-Strategy Whale Optimization Algorithm and Its Application // Eng. Appl. Artif. Intell. 2022. V. 108. P. 104558. DOI: 10.1016/j.engappai.2021.104558.
3. Frazier P.I. A Tutorial on Bayesian Optimization // arXiv. 2018. DOI: 10.48550/arXiv.1807.02811.
4. Archetti F., Candelieri A. Bayesian Optimization and Data Science. Cham: Springer Briefs in Optimization, 2019. DOI: 10.1007/978-3-030-24494-1.
5. Jones D., Martins J. The direct algorithm: 25 years later // J. Glob. Optim. 2021. V. 79, № 3. P. 521–566. DOI: 10.1007/s10898-020-00952-6.
6. Paulavicius R. and Zilinskas J. Simplicial Global Optimization. New York: Springer, 2014. DOI: 10.1007/978-1-4614-9093-7.
7. Paulavicius R., Sergeyev Y.D., Kvasov D.E., Zilinskas J. Globally-biased BIRECT algorithm with local accelerators for expensive global optimization // Expert Syst. Appl. 2020. V. 144. P. 113052. DOI: 10.1016/j.eswa.2019.113052.
8. Сергеев Я. Д., Квасов Д. Е. Диагональные методы глобальной оптимизации. М.: Физматлит, 2008.
9. Liberti L., Kucherenko S. Comparison of deterministic and stochastic approaches to global optimization // Int. Trans. Oper. Res. 2005. V. 12. P. 263–285.
10. Sergeyev Y.D., Kvasov D.E., Mukhametzhanov M.S. On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget // Sci. Rep. 2018. V. 8, № 1. P. 435.
11. Stripinis L., Paulavicius R. A new DIRECT-GLh algorithm for global optimization with hidden constraints // Optim. Lett. 2021. V. 15, № 6. P. 1865–1884. DOI: 10.1007/s11590-021-01726-z.
12. Audet C., Batailly A., Kojtych S. Escaping unknown discontinuous regions in blackbox optimization // SIAM J. Optim. 2022. V. 32, № 3. P. 1843–1870. DOI: 10.1137/21m1420915.
13. Candelieri A. Sequential model based optimization of partially defined functions under unknown constraints // J. Glob. Optim. 2019. V. 79, № 2. P. 281–303. DOI: 10.1007/s10898-019-00860-4.
14. Баркалов К. А., Стронгин П. Г. Метод глобальной оптимизации с адаптивным порядком проверки ограничений // Журн. вычисл. матем. и матем. физ. 2002. Т. 42, № 9. С. 1338–1350.

15. Strongin R. G., Barkalov K. A., Bevzuk S. A. Global optimization method with dual Lipschitz constant estimates for problems with non-convex constraints // *Soft Comput.* 2020. V. 24, № 16. P. 11853–11865. DOI: 10.1007/s00500-020-05078-1.
16. Sergeyev Y. D., Strongin R. G., Lera D. *Introduction to Global Optimization Exploiting Space-Filling Curves*. New York: Springer Briefs in Optimization, 2013. DOI: 10.1007/978-1-4614-8042-6.
17. Strongin R. G., Sergeyev Y. D. *Global optimization with non-convex constraints. Sequential and parallel algorithms*. Dordrecht: Kluwer Academic Publishers, 2000.
18. 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. V. 1914. P. 147–161. DOI: 10.1007/978-3-031-52470-7\_13.
19. Barkalov K. A., et al. On solving the problem of finding kinetic parameters of catalytic isomerization of the pentane-hexane fraction using a parallel global search algorithm // *Mathematics*. 2022. V. 10, № 19. P. 3665. DOI: 10.3390/math10193665.
20. Gubaydullin I. M., Enikeeva L. V., Barkalov K. A., Lebedev I. G., Silenko D. G. Kinetic modeling of isobutane alkylation with mixed C4 olefins and sulfuric acid as a catalyst using the asynchronous global optimization algorithm // *Commun. Comput. Inf. Sci.* 2022. V. 1618. P. 293–306. DOI: 10.1007/978-3-031-11623-0\_20.
21. Barkalov K. A., Lebedev I. G., Gergel V. P. Parallel Global Search Algorithm with Local Tuning for Solving Mixed-Integer Global Optimization Problems // *Lobachevskii Journal of Mathematics*. V. 7. № 42. 2021. P. 1492–1503.
22. Сысоев А. В., Козинев Е. А., Баркалов К. А., Лебедев И. Г., Карчков Д. А., Родионов Д. М. Фреймворк методов интеллектуальной эвристической оптимизации iOpt // В кн.: *Суперкомпьютерные дни в России: Труды международной конференции*. 2023. С. 179–185.
23. Исходный код фреймворка iOpt. [Электрон. рес.]: <https://github.com/aimclub/iOpt> (дата обращения: 26.01.2025).
24. Документация iOpt. [Электрон. рес.]: <https://iopt.readthedocs.io/ru/latest/> (дата обращения: 26.01.2025).
25. Gaviano M., Kvasov D. E., Lera D., Sergeyev Y. D. Software for generation of classes of test functions with known local and global minima for global optimization // *ACM Trans. Math. Softw.* 2003. V. 29, № 4. P. 469–480.
26. Storn R., Price K., *Differential Evolution — a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces* // *Journal of Global Optimization*. 1997. V. 11. P. 341–359.
27. Xiang Y, Gubian S, Suomela B, Hoeng J. Generalized Simulated Annealing for Efficient Global Optimization: the GenSA Package for R // *The R Journal*. 2013. V. 5, № 1.
28. Gablonsky J., Kelley C. A Locally-Biased form of the DIRECT Algorithm // *Journal of Global Optimization*. 2001. V. 21. P. 27–37.
29. Wales D. J., Doye J. P. K. Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms // *Journal of Physical Chemistry A*. 1997. V. 101. P. 5111.
30. Endres S. C., Sandrock C., Focke W. W. A simplicial homology algorithm for Lipschitz optimisation // *Journal of Global Optimization*. 2018.
31. Filippou K., Aifantis G., Papakostas G. A., Tsekouras G. E. Structure learning and hyperparameter optimization using an automated machine learning (AutoML) pipeline // *Information*. 2023. V. 14, № 4. P. 232.
32. Automated modeling and machine learning framework FEDOT. [Электрон. рес.]: <https://github.com/aimclub/FEDOT> (дата обращения: 25.07.2025).
33. Xu N. Time Series Analysis on Monthly Beer Production in Australia // *Highlights in Science, Engineering and Technology*. 2024. V. 94. P. 392–401. DOI: 10.54097/4z3krj13.

34. Akiba T., Sano S., Yanase T., Ohta T., Koyama M. Optuna: A Next-Generation Hyperparameter Optimization Framework // In Proceedings: 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019. P. 2623–2631. DOI: 10.1145/3292500.3330701.



**Усова Марина Андреевна** — ассистент кафедры Математического обеспечения и суперкомпьютерных технологий Института информационных технологий, математики и механики ННГУ им. Н. И. Лобачевского.

Область научных интересов: модели и методы решения задач глобальной оптимизации, разработка средств визуализации данных. E-mail: [usova@itmm.unn.ru](mailto:usova@itmm.unn.ru).

**Marina Usova** — Assistant of the Department of Mathematical Software and Supercomputer Technologies of the Institute of Information Technologies, Mathematics and Mechanics, Lobachevsky State University of Nizhny Novgorod. Research interests: models and methods for solving global optimization problems, development of data visualization tools. E-mail: [usova@itmm.unn.ru](mailto:usova@itmm.unn.ru).



**Лебедев Илья Геннадьевич** — заведующий лабораторией суперкомпьютерных технологий и высокопроизводительных вычислений кафедры Математического обеспечения и суперкомпьютерных технологий Института информационных технологий, математики и механики

Национального исследовательского нижегородского государственного университета им. Н. И. Лобачевского, Область научных интересов: методы глобальной и локальной оптимизации, параллельные алгоритмы, программирование для графических процессоров, CUDA. Является автором и соавтором 61 научной работы. E-mail: [ilya.lebedev@itmm.unn.ru](mailto:ilya.lebedev@itmm.unn.ru).

**Ilya Lebedev** — PhD., Head of the Laboratory of Supercomputer Technologies and High-Performance Computing, Department of Mathematical Software and Supercomputing Technologies, Institute of Information Technology, Mathematics and Mechanics, Nizhny Novgorod

State University. N. I. Lobachevsky. His research interests include algorithms of global and local optimization, parallel algorithms, GPU programming, CUDA. He is the author or coauthor of more than 30 papers in these areas. E-mail: [ilya.lebedev@itmm.unn.ru](mailto:ilya.lebedev@itmm.unn.ru).



**Штанюк Антон Александрович** — кандидат технических наук, доцент, доцент кафедры информатики и автоматизации научных исследований Национального исследовательского нижегородского государственного университета им. Н. И. Лобачевского.

Количество печатных работ 53. Область научных интересов: моделирование и проектирование сложных систем, языки и технология программирования, обучение программированию, алгоритмы и структуры данных. E-mail: [anton.shtanyuk@itmm.unn.ru](mailto:anton.shtanyuk@itmm.unn.ru).

**Anton Shtanyuk** — PhD., Associate Professor at the Department of Informatics and Research Automation of the Institute of Information Technology, mathematics and mechanics at Lobachevsky State University of Nizhny Novgorod. The number of publications: 50. Research interests: modelling and design of complex systems, programming languages and technology, programming teaching, algorithms and data structures. E-mail: [anton.shtanyuk@itmm.unn.ru](mailto:anton.shtanyuk@itmm.unn.ru).



**Баркалов Константин Александрович** — доктор технических наук, доцент, заведующий кафедрой Математического обеспечения и суперкомпьютерных технологий Института информационных технологий, математики и механики

ННГУ им. Н. И. Лобачевского. Область научных интересов: математические модели, методы и программные средства для решения задач гло-

бальной оптимизации; параллельные вычисления. Является автором и соавтором более 140 научных работ в указанных областях. E-mail: [konstantin.barkalov@itmm.unn.ru](mailto:konstantin.barkalov@itmm.unn.ru).

**Konstantin Barkalov** — Doctor of Technical Sciences, Professor, Head of the Department of Mathematical Software and Supercomputer Technologies, Institute of

Information Technologies, Mathematics and Mechanics, Lobachevsky State University of Nizhny Novgorod. Research interests: mathematical models, methods and software for solving global optimization problems; parallel computing. E-mail: [konstantin.barkalov@itmm.unn.ru](mailto:konstantin.barkalov@itmm.unn.ru).

*Дата поступления* — 04.04.2025