

УДК 519.6

DOI: [10.26102/2310-6018/2025.51.4.037](https://doi.org/10.26102/2310-6018/2025.51.4.037)

Разработка и исследование алгоритмов генерации данных на основе математического ремоделирования

П.В. Сараев, А.С. Тюрин✉

Липецкий государственный технический университет, Липецк, Российская Федерация

Резюме. Математическое ремоделирование представляет собой современный подход в области математического моделирования, суть которого заключается в преобразовании существующей модели одного класса в новую модель, принадлежащую к другому, часто более простому или вычислительно эффективному классу. В отличие от классического процесса моделирования, где модель создается «с нуля» на основе первичных данных, ремоделирование исходит из того, что уже существует некая адекватная исходная модель f_1 , достаточно точно описывающая объект или процесс. Эта модель, однако, может быть слишком сложной для практического применения, требовать значительных вычислительных ресурсов или быть представленной в форме, неудобной для дальнейшего использования, например, в системах реального времени или в устройствах с ограниченной производительностью. Ключевой задачей в процессе ремоделирования является генерация репрезентативного обучающего множества данных, на основе которого будет строиться новая модель f_2 . Именно от качества и структуры этого синтезированного набора данных напрямую зависит точность и адекватность вновь получаемой модели. Традиционные методы генерации, такие как равномерное случайное распределение точек в заданной области или использование методов планирования эксперимента, зачастую оказываются неэффективными: они либо не учитывают особенности поведения исходной функции, либо становятся вычислительно нереализуемыми при высокой размерности задачи. В связи с этим возникает необходимость в разработке интеллектуальных алгоритмов адаптивной генерации данных, которые могли бы целенаправленно размещать точки в тех областях пространства входных переменных, где исходная функция f_1 демонстрирует наибольшую вариативность и нелинейность. Данная работа посвящена разработке и исследованию именно такого подхода, основанного на принципах интервального анализа и последовательной бисекции области определения, что позволяет оптимально распределять ограниченный объем генерируемых данных и существенно повышать точность математического ремоделирования.

Ключевые слова: математическое моделирование, ремоделирование, генерация данных, интервальный анализ, численные методы.

Благодарности: Исследование выполнено за счет гранта Российского научного фонда № 24-21-00474, <https://rscf.ru/project/24-21-00474/>.

Для цитирования: Сараев П.В., Тюрин А.С. Разработка и исследование алгоритмов генерации данных на основе математического ремоделирования. *Моделирование, оптимизация и информационные технологии*. 2025;13(4). URL: <https://moitvvt.ru/ru/journal/pdf?id=2058> DOI: 10.26102/2310-6018/2025.51.4.037

Development and research of data generation algorithms based on mathematical remodelling

P.V. Saraev, A.S. Tyurin✉

Lipetsk State Technical University, Lipetsk, the Russian Federation

Abstract. Mathematical remodeling is a modern approach in the field of mathematical modeling, the essence of which is the transformation of an existing model of one class into a new model belonging to

a different, often simpler or computationally more efficient class. Unlike the classical modeling process, where a model is created "from scratch" based on primary data, remodeling starts from the premise that there already exists some adequate initial model f_1 that describes an object or process accurately enough. However, this model may be too complex for practical application, require significant computational resources, or be presented in a form that is inconvenient for further use, for example, in real-time systems or on devices with limited performance. The key task in the remodeling process is the generation of a representative training dataset on which the new model f_2 will be built. The accuracy and adequacy of the newly obtained model directly depend on the quality and structure of this synthesized dataset. Traditional generation methods, such as uniform random distribution of points in a given domain or using design of experiments methods, often prove to be ineffective: they either do not account for the behavioral features of the original function or become computationally infeasible in high-dimensional problems. Consequently, there is a need to develop intelligent algorithms for adaptive data generation that could purposefully place points in those regions of the input variable space where the original function f_1 demonstrates the greatest variability and nonlinearity. This work is devoted to the development and research of precisely such an approach, based on the principles of interval analysis and sequential bisection of the domain. This allows for the optimal distribution of a limited volume of generated data and significantly improves the accuracy of mathematical remodeling.

Keywords: mathematical modeling, remodeling, data generation, interval analysis, numerical methods.

Acknowledgements: The study was supported by the Russian Science Foundation grant No. 24-21-00474, <https://rscf.ru/project/24-21-00474/>.

For citation: Saraev P.V., Tyurin A.S. Development and research of data generation algorithms based on mathematical remodelling. *Modeling, Optimization and Information Technology*. 2025;13(4). (In Russ.). URL: <https://moitvvt.ru/ru/journal/pdf?id=2058> DOI: 10.26102/2310-6018/2025.51.4.037

Введение

Математическое ремоделирование – процесс преобразования модели одного класса в другой. Отличием данного подхода от классического математического моделирования является то, что уже известна некоторая модель f_1 , которая может быть использована для генерации необходимого множества данных для формирования модели новой структуры [1].

При построении новой модели возникает вопрос о потере точности при трансформации моделей. Данный вопрос следует разделить на две части: оценка адекватности новой модели f_2 , а также точность ремоделирования, т. е. степень близости вновь полученной модели исходной. Для ответа на первый вопрос разумно обратиться к тому, как оценивалась адекватность модели f_1 . Если, как принято в задачах анализа данных, это основывается на оценке точности модели на выделенном тестовом множестве данных, то, если такое множество данных осталось, целесообразно провести аналогичную оценку с использованием того же функционала качества для новой модели f_2 . Если же тестовое множество отсутствует, то после формирования набора данных можно выделить множество для оценки степени близости f_2 к модели f_1 . Для этого также потребуется выбор функционала качества близости этих моделей. Новое тестовое множество объемом m_t для f_2 может быть сгенерировано и отдельно на основе равномерного закона распределения. Это может быть полезно при оценивании точности различных функций f_2 для разных классов, построенных, например, при разных способах формирования обучающих множеств.

Материалы и методы

Промежуточным этапом в процессе трансформации моделей из f_1 в f_2 выступает промежуточное множество данных. Это множество данных генерируется на основе

функции $f1$ и используется в качестве обучающего множества для модели $f2$. Множество данных содержит n входных переменных, количество которых определяется либо исходя из вида функции $f1$, либо формируется на основе перехода от динамической модели. Оно может иметь произвольную размерность в зависимости от частоты дискретизации и необходимости учитывать количество значений входной переменной в предыдущие моменты времени. Во втором случае значение n требует определения. Количество выходов r определяется решаемой задачей и обычно, но не всегда, равно единице. Для динамических систем это значение r также может потребоваться для определения.

Объем данных m , то есть количество примеров в промежуточном обучающем множестве, представляет собой один из ключевых гиперпараметров процесса математического ремоделирования. Его определение является нетривиальной задачей, поскольку m непосредственно влияет на качество итоговой модели $f2$ и вычислительную стоимость. В отличие от классических задач машинного обучения, где объем данных ограничен исходной выборкой, в случае ремоделирования мы обладаем уникальной возможностью генерировать сколь угодно большое множество данных на основе известной модели $f1$.

Слишком малое значение m может привести к недообучению модели $f2$, поскольку недостаточное количество примеров не позволит адекватно отразить сложность и особенности исходной функции $f1$. В этом случае модель может оказаться неспособной к обобщению и продемонстрирует низкую точность на тестовых данных. Напротив, чрезмерно большое значение m хотя и способно потенциально улучшить точность, ведет к закономерному росту вычислительной нагрузки на этапах генерации данных и обучения модели. Это особенно критично в случаях сложных моделей $f2$ (например, глубоких нейронных сетей) или при высокой размерности входного пространства, где время обучения может расти непропорционально объему данных. Кроме того, существует асимптотический предел, после которого добавление новых примеров перестает существенно улучшать качество модели, но продолжает увеличивать затраты.

Таким образом, определение оптимального объема m является задачей нахождения баланса между точностью модели и ресурсными ограничениями. Наиболее рациональным подходом к решению этой задачи представляется итеративный процесс. Он начинается с выбора некоторого начального значения m , основанного на априорных оценках сложности модели $f2$. На следующем шаге формируется обучающее множество и проводится идентификация модели $f2$. Затем оценивается ее адекватность на отдельном тестовом множестве, сгенерированном с помощью $f1$. Если точность модели признается недостаточной, значение m увеличивается (например, по правилу геометрической прогрессии), и весь цикл – генерация данных, обучение и оценка – повторяется. Процесс продолжается до достижения требуемого уровня точности либо до исчерпания выделенных ресурсов (временных или вычислительных).

После определения числа m требуется непосредственно генерация объема данных. Для этого нужно дополнительно определить ограничения на изменения значений переменных $[a_i, b_i] \ni x_i$ для всех $i = 1, \dots, n$. Эти значения могут быть произвольно большими, но они не должны быть бесконечными, а должны быть заданы конечными значениями. В связи с тем, что во многих задачах используются нормализованные входные значения, границы a_i и b_i иногда могут быть заданы в виде значений того диапазона, в который осуществляется нормировка данных, например, $[-1, 1]$.

Могут быть предложены следующие способы генерации набора данных:

1. На основе вычисления значений на множестве граничных точек аналогично тому, как это делается при решении задач планирования эксперимента. Такой способ возможен только при небольшом количестве переменных, потому что даже при рассмотрении крайних точек для 100 переменных всего таких комбинаций будет $10^{30} \sim 2^{100}$.

2. На основе равномерного в области $[a_i, b_i] \ni x_i \dots$ случайного закона распределения в количестве m . При этом требуется определение шага дискретизации пространства изменения входных переменных. Если входные величины находятся в диапазоне $[-1, 1]$, то может быть принято значение шага 10^{-3} для определения точности генерации значений обучающего множества.

3. На основе оценки диапазона изменения функции $f1$ на рассматриваемой области пространства $[x] \subset R^n$. Основная идея заключается в генерации большего количества точек в той части пространства, где функция сильнее изменяет свои значения. Для адаптивной оценки диапазона изменения значения функции можно использовать внешнюю оценку образа функции с помощью методов интервального оценивания. Такая оценка производится с использованием естественной функции включения $[f1]_n$. Если исходный брус $[x]$ подвергается бисекции (его разделению на две части – два подбруса $[x_1]$ и $[x_2]$ по центру координаты, объединение которых дает исходный брус – область изменения входных данных), то после оценки образа функции $f1$ на этих подбрусах получатся образы $[f1]([x_1])$ и $[f1]([x_2])$. Вычислив ширину $wf1$ и $wf2$ соответственно (как max_i), количество генерируемых точек m распределяем пропорционально этим значениям: m_1 точка будет генерироваться на $[x_1]$, а m_2 точек – на брусе $[x_2]$. Выбор координаты l , по которой пройдет бисекция, может быть выбрана случайно или в ее качестве взята первая, по которой ширина стороны бруса является максимальной величиной (это соответствует выбору брусков для бисекций в интервальных алгоритмах глобальной оптимизации). Такой алгоритм может выполняться рекурсивно, например, для брусков, имеющих наибольший диапазон изменения значений функции. Если задать количество d таких разбиений, то мы получим список из d упорядоченных пар $([x_i], m_i)$, первый элемент которых является бруском – диапазоном изменения входных переменных, а m_i – количество точек, генерируемых в этой области [2, 3].

Если отсутствует тестовое множество для оценки адекватности модели, то перед идентификацией новой модели $f2$ требуется выделение из сгенерированного тестового множества и определение функционала качества для оценки точности приближения моделей. В этом качестве логично выглядит выбор одной из норм – MAE, MAPE, RMSE или равномерная норма. Таким образом, обобщенный алгоритм математического ремоделирования может быть сформулирован следующим образом [4, 5]:

1. Определение множества входных переменных n и выходных переменных r .
Определение границ изменения переменных $[a_i, b_i]$.

2. Определение класса моделей $f2$ – некоторой структуры моделей, зависящих от набора неизвестных параметров. Как правило, более простой вид, чем $f1$.

3. При наличии тестового множества данных и метрики для оценивания адекватности модели $f1$ сохранить их для использования $f2$. При отсутствии – определение метрики Q , объема обучающего множества m_t .

4. Определение объема данных m для генерации данных.

5. Выбор стратегии генерации данных, формирование обучающего множества.

6. Идентификация модели $f2$ на основе сгенерированного множества данных, оценка точности математического ремоделирования на основе построенной модели $f2$ и оцененных параметров в п. 3 алгоритма.

Представленный алгоритм может выполняться для разных моделей f_2 и разных значений параметров работы алгоритма трансформации моделей (гиперпараметров в терминах машинного обучения).

Так как алгоритм 3 опирается на изменение функции внутри бруса, ожидается, что количество точек внутри каждого бруса будет зависеть от вида функции.

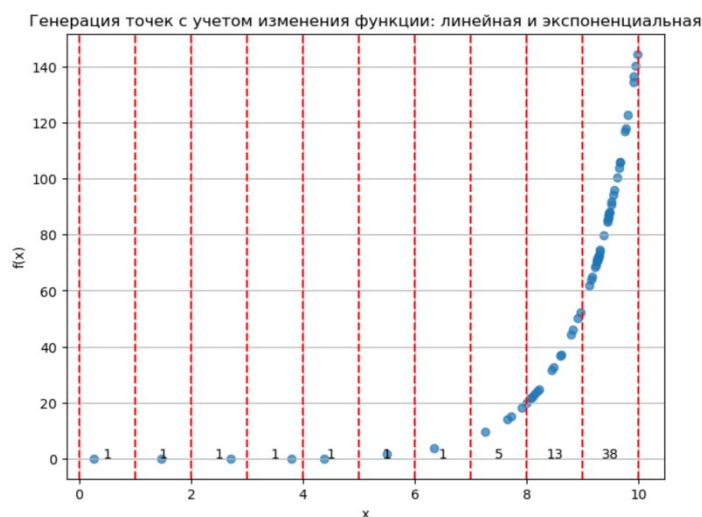


Рисунок 1 – Визуализация генерации точек с разделением пространства на брусы

Figure 1 – Visualization of point generation with space division into bars

На Рисунке 1 представлен пример генерации точек, где каждый брус выделен вертикальной пунктирной линией. Внутри каждого бруса под осью x также указано количество точек, которое было сгенерировано. Для функции вида

$$f(x) = \begin{cases} 0, & x \leq 5, \\ \exp(x - 5), & x > 5 \end{cases} \quad (1)$$

очевидно, что аппроксимация тем или иным способом требует большего количества точек при значениях $x > 5$ – там, где функция изменяется более существенно. На гистограмме (Рисунок 2) представлено распределение точек по брусам.

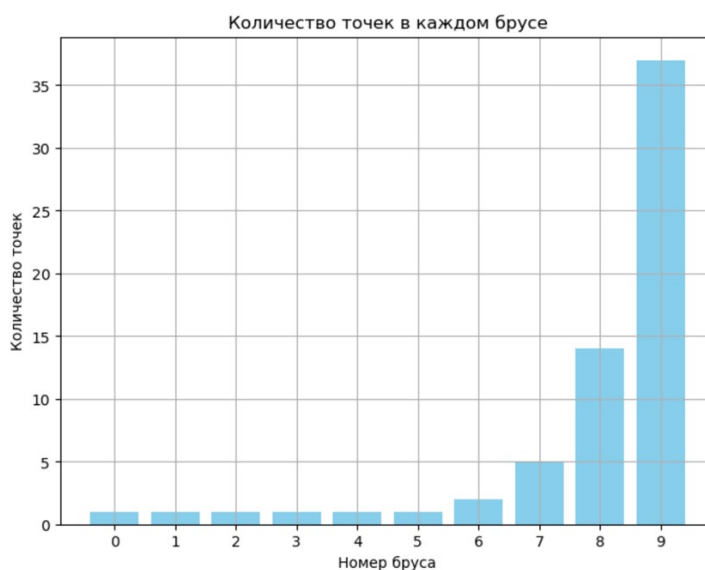


Рисунок 2 – Гистограмма распределения точек по брусам

Figure 2 – Histogram of point distribution by bars

Для более сложных многомерных функций распределение точек по брусам также зависит от изменения значений на границах бруса. Как показано на Рисунке 3, в квадрантах (брусах) 1 и 3 наибольшее изменение функции, что подтверждается гистограммой распределения точек по брусам.

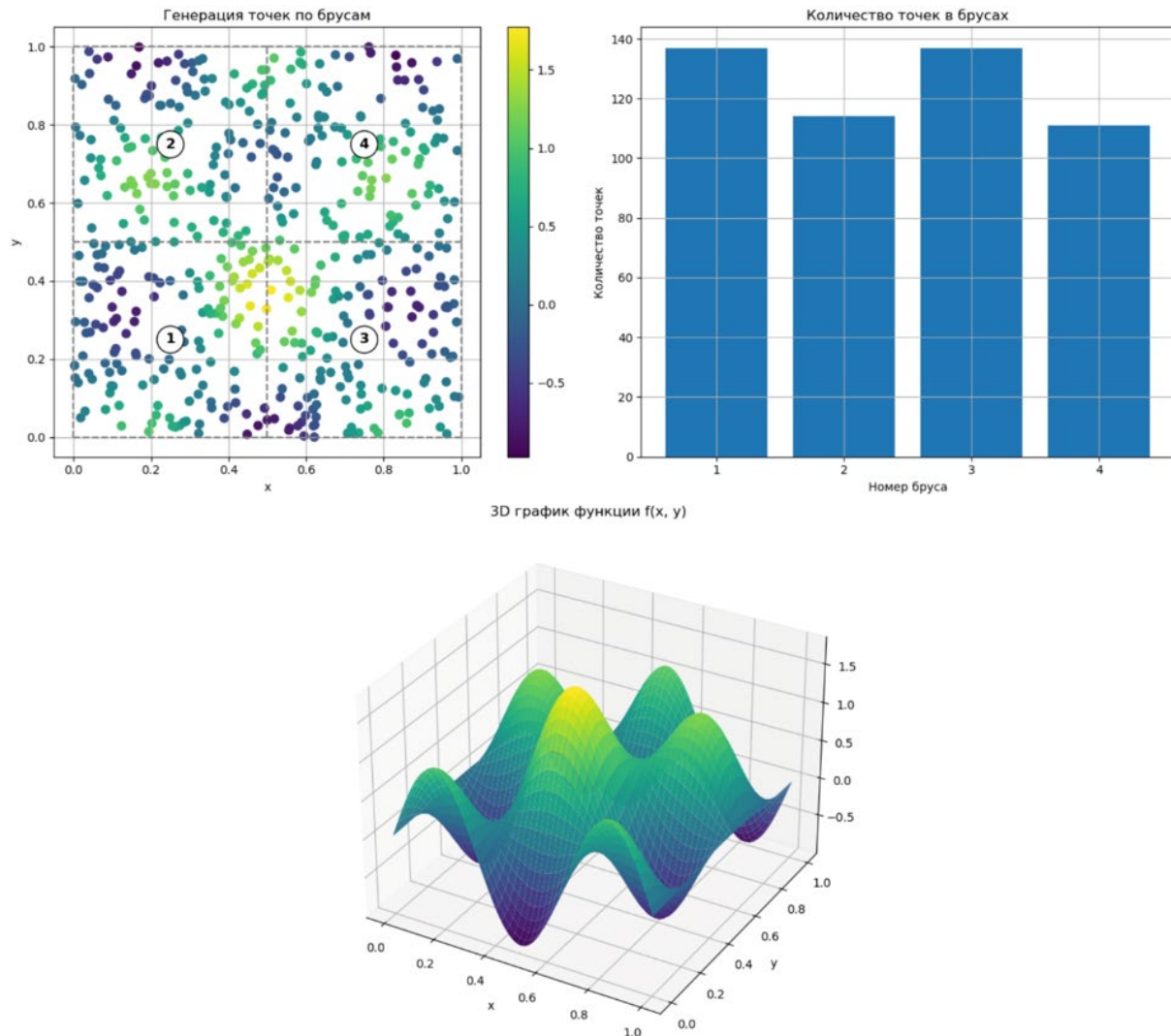


Рисунок 3 – Гистограмма распределения точек по брусам для трехмерной функции
Figure 3 – Histogram of point distribution over bars for the three-dimensional function

Для проведения вычислительных экспериментов реализованы случайный и адаптивный алгоритмы из раздела выше. При реализации использовался язык Python и библиотеки numpy, matplotlib и pandas. Структура программы реализована следующим образом – класс MathRemodeller инициализируется:

- функцией f , которая описывает исходную модель;
- параметром n , который задает количество генерируемых строк в обучающем множестве;
- функцией g , которая описывает обучающую модель;
- параметром num_boxes , который имеет вид списка и описывает количество интервалов, на которых будет генерироваться обучающее множество;
- параметром $params_ranges$, который описывает интервалы значений факторов.

Также класс реализует стандартный интерфейс для работы с моделями машинного обучения – методы *fit* и *predict*. Метод *fit* принимает на вход массивы значений x и y для обучения модели g , а метод *predict* принимает на вход только массив значений x для генерации массива y на основе функции g . Отдельно в классе реализован метод *generate_data* для генерации массива обучающих данных согласно приведенному ранее алгоритму. В качестве метрик оценки ремоделирования будут использованы MAE, RMSE и R^2 .

Исследуем, как стратегия генерации точек влияют на качество аппроксимации функции g при разных стратегиях обучения [6, 7]. В качестве стратегий обучения модели (аппроксимации функции) используется метод наименьших квадратов и градиентный спуск. Следует отметить, что в данных экспериментах мы используем известный нам вид функции $f(x)$, хотя реализованный код поддерживает применение модели f – черного ящика, где, например, есть только значение весов для нейросети и т. п.

Таким образом, важно только наличие информации об обучающем и тестовом наборе данных для функции $f(x)$. При вычислении метрик качества обучения мы используем именно тестовый набор данных для функции $f(x)$, что дополнительно усложняет задачу, так как в случае аппроксимации новая модель $g(x)$ учитывает ошибку $f(x)$.

В простейшем случае, когда $f(x)$ и $g(x)$ имеют одинаковую структуру вида:

$$f(x, a) = g(x, a) = x \cdot a + \sum_{i=1}^n x_i^2, \quad (2)$$

где i – номер фактора, a – вектор коэффициентов, x – вектор факторов, тогда оценки точности имеют следующие значения в зависимости от количества брусков (Рисунок 4):

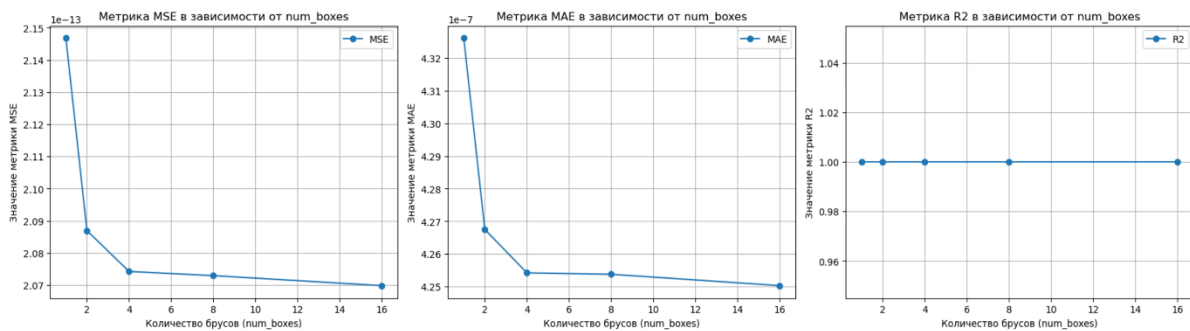


Рисунок 4 – Оценки точности для модели (2)

Figure 4 – Accuracy estimates for the model (2)

Далее определим более сложную функцию f так, чтобы функция g лишь отчасти описывала ее. Зададим в функции f более сложные компоненты нелинейности. Данный эксперимент будет описывать случай, когда исходная модель имеет более сложную структуру:

$$f(x, a) = a \cdot x \cdot \sum_{i=1}^n x_i^2 + \sum_{i=1}^n x_i^3, \quad (3)$$

$$g(x, a) = a \cdot x + \sum_{i=1}^n x_i^2. \quad (4)$$

Тогда оценки точности имеют значения в зависимости от количества брусков, показанные на Рисунке 5.

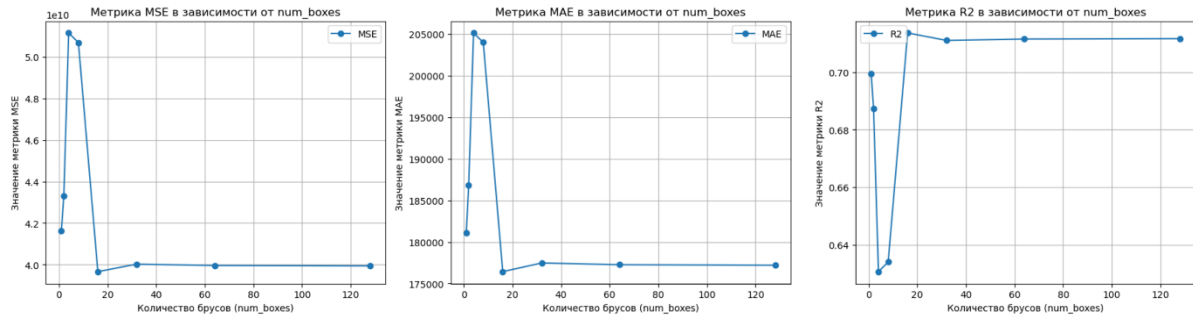


Рисунок 5 – Оценки точности для модели (3), (4)
Figure 5 – Accuracy estimates for the model (3), (4)

Как видно, наиболее оптимальное количество брусков равняется 5. Далее, после значения 0,71 метрика R^2 меняется несущественно. Это обстоятельство связано с тем, что границы брусков удачно разделяют пространство точек так, что экстремальные значения функции f находятся на границах этих брусков.

Далее применим разработанный алгоритм генерации обучающих данных на наборе California House Prices. В приведенных далее примерах используется ограниченное количество брусков по каждой из переменной, так как с ростом количества переменных количество многомерных брусков значительно увеличивается, что значительно повышает время генерации данных.

Пространство по двум осям на n_1 и n_2 частей, то количество брусков (прямоугольников или гиперпрямоугольников) будет равно произведению числа частей по каждой оси: $n_1 \cdot n_2$. В случае с двумя осями и двумя брусками по каждой оси, получается $2 \cdot 2 = 4$ брусков. В используемом далее наборе данных мы имеем 8 переменных. В случае, если мы делим каждую их переменных на 8 брусков, то итоговое количество брусков возрастает до 8^8 .

Для регрессионной модели, построенной при помощи градиентного спуска, функции g и f будут иметь следующий вид:

$$f(x, a) = g(x, a) = ax + b, \quad (5)$$

где b – коэффициент смещения. В качестве исходной модели используется модель градиентного спуска, построенная при помощи библиотеки sklearn с гиперпараметрами по умолчанию. Полученная матрица коэффициентов a задается как известный параметр для функции f . Как было показано ранее, в случае с такой относительно простой моделью мы получим достаточно точное приближение g к f (Рисунок 6).

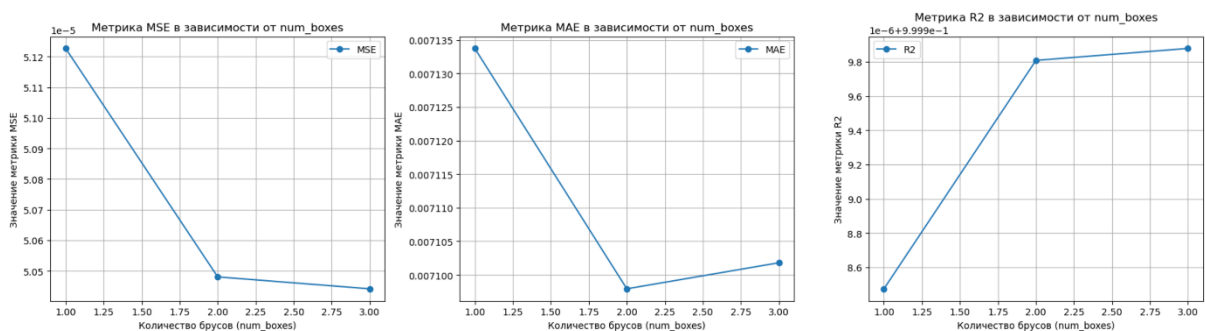


Рисунок 6 – Оценки точности для модели (5)
Figure 6 – Accuracy estimates for the model according (5)

Как видно на иллюстрации выше, модель точно приближается к исходной с постепенным улучшением метрик, но данные изменения достаточно малы в рамках рассматриваемой задачи. В Таблице 1 представлено более подробное сравнение полученной модели g с моделью f и исходной моделью.

Таблица 1 – Сравнение точности моделей

Table 1 – Comparison of model accuracy

Модель	MSE	MAE	R ²
g (1 брус)	0,56	0,007134	0,5766
g (2 бруса)	0,55	0,007098	0,5769
g (3 бруса)	0,55	0,007102	0,5770
f	0,55	0,007090	0,5768

Как видно из Таблицы 1, модель g достаточно точно приближается к f при всех значениях количества брусков, однако наиболее точным является решение с количеством брусков 3. Следует отметить, что уже даже при использовании всего одного бруса (что эквивалентно равномерному распределению точек) модель g демонстрирует высокую степень приближения к модели f . Значения метрик MSE (0,56 против 0,55) и R² (0,5766 против 0,5768) очень близки, что подтверждает принципиальную возможность успешного ремоделирования даже простыми методами генерации данных. Важно подчеркнуть, что модель g , построенная путем ремоделирования, не просто приближается к f , но в одной из метрик (R²) даже незначительно превосходит ее. Это может быть объяснено тем, что процесс обучения g на сгенерированном множестве данных действует как своего рода регуляризация, позволяя модели избежать переобучения на шумах, которые могли присутствовать в исходных данных, использованных для построения f .

Заключение

Проведенный анализ подтверждает, что разработанный алгоритм адаптивной генерации данных на основе бисекции является эффективным. Он позволяет целенаправленно управлять процессом создания обучающего множества, что приводит к построению модели g , практически идентичной исходной модели f по своей предсказательной силе. Результаты также демонстрируют важность правильного выбора гиперпараметра алгоритма – количества брусков, – указывая на существование оптимального значения, которое обеспечивает баланс между точностью и вычислительными затратами.

Как было показано ранее, при высокой размерности данных значительно возрастает вычислительная сложность, так как приходится обрабатывать большое количество многомерных брусков. Кроме того, гладкие области функций также попадают под оценку изменений значения функции на границах брусков. Так как для небольших наборов данных целесообразно оставить хотя бы одну точку в такой брус, то в случае большой размерности данных требуется объединять брусья, чтобы искусственно не увеличивать количество сгенерированных примеров.

Так как описанный алгоритм предполагает лишь оценку на границах брусков, есть вероятность, что локальные экстремумы не попадут на эти границы. В качестве развития предложенного алгоритма может быть применен метод постепенного разделения брусков пополам с ограничением глубины, подобно тому, как выполняется построение бинарных деревьев. Также в качестве развития алгоритма может быть применено вычисление производных для каждого бруса, что позволит определить не только само изменение

функции, но и скорость этого изменения и дополнительно сбалансировать распределение точек по различным участкам функции. Отдельно стоит рассмотреть применимость метода для ансамблевых методов [8, 9] и интервального анализа [10].

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Сараев П.В., Блюмин С.Л., Галкин А.В. Нейросетевое и нейро-нечеткое ремоделирование в управлении металлургическими процессами. В сборнике: *Современные проблемы горно-металлургического комплекса. Наука и производство: Материалы XIII Всероссийской научно-практической конференции с международным участием, 23–25 ноября 2016 года, Старый Оскол, Россия*. Старый Оскол: Национальный исследовательский технологический университет «МИСИС»; 2016. С. 102–105.
Saraev P.V., Blyumin S.L., Galkin A.V. Neural and Neuro-Fuzzy Modelling in Control of Metallurgical Processes. In: *Sovremennye problemy gorno-metallurgicheskogo kompleksa. Nauka i proizvodstvo: Materialy XIII Vserossiiskoi nauchno-prakticheskoi konferentsii s mezhdunarodnym uchastiem, 23–25 November 2016, Stary Oskol, Russia*. Stary Oskol: National University of Science and Technology "MISIS"; 2016. P. 102–105. (In Russ.).
2. Huang Ch., Radi B., Hami A.E. Uncertainty Analysis of Deep Drawing Using Surrogate Model Based Probabilistic Method. *The International Journal of Advanced Manufacturing Technology*. 2016;86(9):3229–3240. <https://doi.org/10.1007/s00170-016-8436-4>
3. Jansson T., Nilsson L., Redhe M. Using Surrogate Models and Response Surfaces in Structural Optimization – With Application to Crashworthiness Design and Sheet Metal Forming. *Structural and Multidisciplinary Optimization*. 2003;25(2):129–140. <https://doi.org/10.1007/s00158-002-0279-y>
4. Burnaev E., Grihon S. Construction of the Metamodels in Support of Stiffened Panel Optimization. In: *Proceedings of the VI International Conference on Mathematical Methods in Reliability: Theory. Methods. Applications, 22–29 June 2009, Moscow, Russia*. Moscow: PFUR; 2009. P. 124–128.
5. Zhao D., Xue D. A Multi-Surrogate Approximation Method for Metamodeling. *Engineering with Computers*. 2011;27(2):139–153. <https://doi.org/10.1007/s00366-009-0173-y>
6. Сараев П.В. Концепция математического ремоделирования. В сборнике: *Нано-биотехнологии. Теплоэнергетика. Математическое моделирование, 27–28 февраля 2024 года, Липецк, Россия*. Липецк: Липецкий государственный технический университет; 2024. С. 150–154.
Saraev P. Conception of Mathematical Remodeling. In: *Nano-bio-tekhnologii. Teploenergetika. Matematicheskoe modelirovanie, 27–28 February 2024, Lipetsk, Russia*. Lipetsk: Lipetsk State Technical University; 2024. P. 150–154. (In Russ.).
7. Mienye I.D., Sun Y. A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects. *IEEE Access*. 2022;10:99129–99149. <https://doi.org/10.1109/ACCESS.2022.3207287>
8. Kunapuli G. *Ensemble Methods for Machine Learning*. Shelter Island: Manning Publications Co.; 2023. 352 p.
9. Жолен Л., Кифер М., Дидри О., Вальтер Э. *Прикладной интервальный анализ*. Москва, Ижевск: Институт компьютерных исследований; 2019. 468 с.

- Jaulin L., Kieffer M., Didrit O., Walter E. *Applied Interval Analysis: With Examples in Parameter and State Estimation, Robust Control and Robotics*. Moscow, Izhevsk: Institut komp'yuternykh issledovaniy; 2019. 468 p. (In Russ.).
10. Hansen E.R., Walster G.W. *Global Optimization Using Interval Analysis*. New York: Marcel Dekker; 2004. 489 p.

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Сараев Павел Викторович, доктор технических наук, доцент, профессор, Липецкий государственный технический университет, Липецк, Российская Федерация.

e-mail: psaraev@yandex.ru

ORCID: [0000-0002-1373-2521](https://orcid.org/0000-0002-1373-2521)

Pavel V. Saraev, Doctor of Engineering Sciences, Docent, Professor, Lipetsk State Technical University, Lipetsk, the Russian Federation.

Тюрин Алексей Сергеевич, доцент, кафедра автоматизированных систем управления Липецкого государственного технического университета, Липецк, Российская Федерация.

e-mail: leha2148@gmail.com

ORCID: [0009-0000-1313-5826](https://orcid.org/0009-0000-1313-5826)

Alexey S. Tyurin, Associate Professor, Department of Automated Control Systems, Lipetsk State Technical University, Lipetsk, the Russian Federation.

Статья поступила в редакцию 28.08.2025; одобрена после рецензирования 07.11.2025; принята к публикации 12.11.2023.

The article was submitted 28.08.2025; approved after reviewing 07.11.2025; accepted for publication 12.11.2025.