

Методы оптимизации

Отчёт по лабораторной работе №4

Авторы работы:

Галкин Глеб М3234

Дьяков Максим М3234

Кирияк Александр М3234

[Ссылка на репозиторий](#)

Постановка задачи

1. Теоретическое описание алгоритма пчелиной колонии и его реализация;
2. Сравнение алгоритма пчелиной колонии с методами из предыдущих лабораторных работ;
3. Применение библиотечной реализации (optuna) к примерам предыдущих лабораторных;
4. Использование библиотечных методов (optuna) для нахождения оптимальных гиперпараметров.

Основное задание

1 Алгоритм пчелиной колонии

Алгоритм искусственной пчелиной колонии (Artificial Bee Colony, ABC) был предложен для решения задач оптимизации и моделирует поведение пчел при поиске нектара. Этот алгоритм относится к метаэвристическим методам оптимизации и вдохновлен природным поведением пчел. В основе лежит идея распределенного поиска с использованием нескольких агентов (пчел) и их взаимодействий.

Рой пчел отправляет несколько разведчиков в случайных направлениях для поиска нектара. Вернувшись, разведчики сообщают о найденных на поле участках с цветами, содержащими нектар, и на них вылетают остальные пчелы. При этом чем больше на участке нектара, тем больше пчел к нему устремляется. Однако при этом пчелы могут случайным образом отклоняться от выбранного направления. После возвращения всех пчел в улей вновь происходит обмен информацией и отправка пчел-разведчиков и пчел-рабочих. Фактически разведчики действуют по алгоритму случайного поиска.

Тогда, местоположение глобального экстремума - это участок, где больше всего нектара, а количество найденного нектара - это значение целевой функции в данной точке.

Начальные параметры:

1. *num_bees*: Этот параметр указывает количество пчёл в вашем алгоритме. Вероятно, чем больше пчёл, тем более разнообразными будут исследуемые решения.
2. *num_elite*: Количество элитных пчел.
3. *num_best_sites*: Количество лучших участков, которые будут исследоваться элитными пчелами.
4. *neighborhood_size*: Размер окрестности вокруг лучших решений, в которой будут искаться новые решения.
5. *iterations*: Количество последовательных вылетов пчел на поиски.

Реализация метода:

1. **Инициализация пчел**: Пчелы размещаются случайным образом по заданным значениям *low* и *high*, то есть минимальным и максимальным значениям координат в пространстве решений, и оцениваются по значению целевой функции.
2. **Основной цикл**: Повторяется до выполнения критерия остановки.
 - (a) Вычисляется значение целевой функции f для каждой пчелы.
 - (b) Пчелы сортируются по значению целевой функции (чем меньше значение, тем лучше решение).
 - (c) **Разведка новых решений случайным образом**: Пчелы, не входящие в число элитных, заменяются новыми случайными решениями.
 - (d) **Эксплуатация лучших решений**: Для каждой элитной пчелы генерируются новые решения в окрестности текущего решения. Если новое решение лучше текущего (значение целевой функции меньше), то элитная пчела перемещается в новую позицию.

2 Сравнение эффективности разных методов

Для сравнения были выбраны данные методы:

- Градиентный спуск с золотым сечением;
- Метод Нелдера-Мида;
- Метод Ньютона с дихотомией;
- Алгоритм пчелиной колонии.

Функция Химмельблау $f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$ Минимумы:

- $f(3, 2) = 0$;
- $f(-2, 805118..., 3, 131312...) = 0$;
- $f(-3, 779310..., -3, 283186...) = 0$;
- $f(3, 584428..., -1, 848126...) = 0$.

Начальные параметры

- Начальное приближение: $x_0 = (-4, 0)$;
- Критерий останова: $\mathcal{E} = 10^{-5}$;
- Границы в методе золотого сечения/дихотомии: $l = \mathcal{E}$, $r = 100$.
- Алгоритм пчелиной колонии:
 - Количество пчел: $num_bees = 5$;
 - Количество элитных пчел: $num_elite = 3$;
 - Количество лучших участков: $num_best_sites = 10$;
 - Размер окрестности: $neighborhood_size = 0.1$;
 - Количество вылетов: $iterations = 10$;
 - Границы координат: $low = -5, high = 5$.

Golden ratio	Nelder-Mead	Newton dichotomy	ABC
$f(3.584, -1.848) = 5e-10$	$f(-2.805, 3.131) = 2e-10$	$f(-2.805, 3.131) = 0$	$f(3.0077, 1.991) = 0.002065134$
11 iterations	99 iterations	11 iterations	10 iterations
748 function	184 function	616 function	350 function
11 gradient	0 gradient	11 gradient	0 gradient
0 hessian	0 hessian	11 hessian	0 hessian

Таблица 1: Сравнение grad_decent golden ratio, Nelder-Mead, Newton dichotomy, ABC на функции Химмельблау

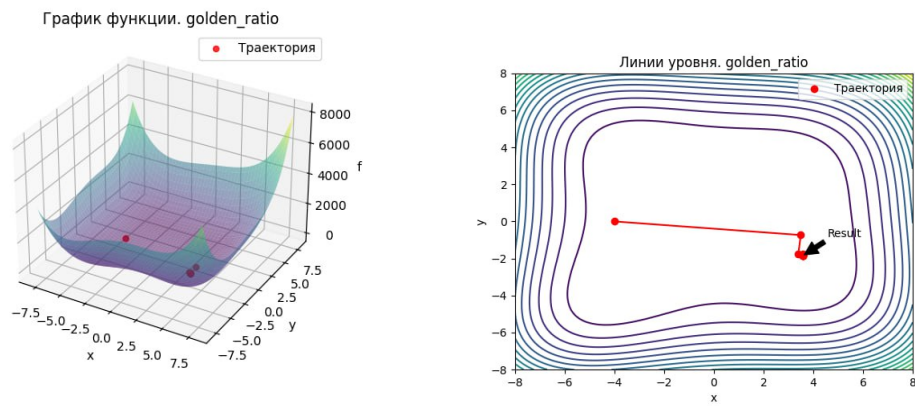


Рис. 1: Golden rate

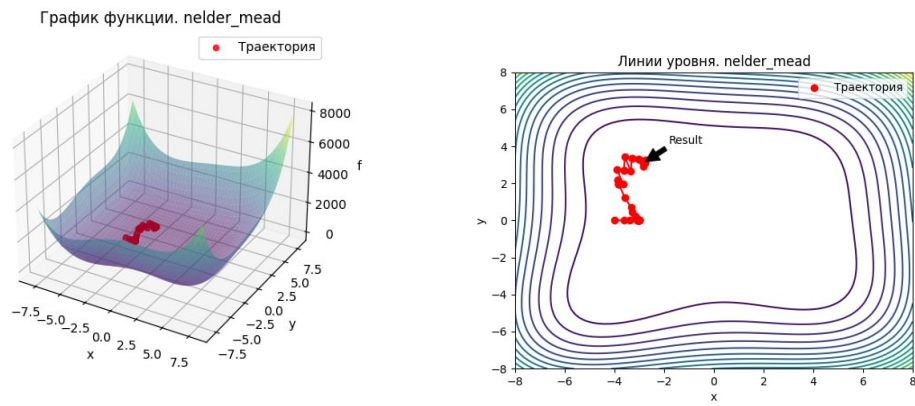


Рис. 2: Nelder-Mead

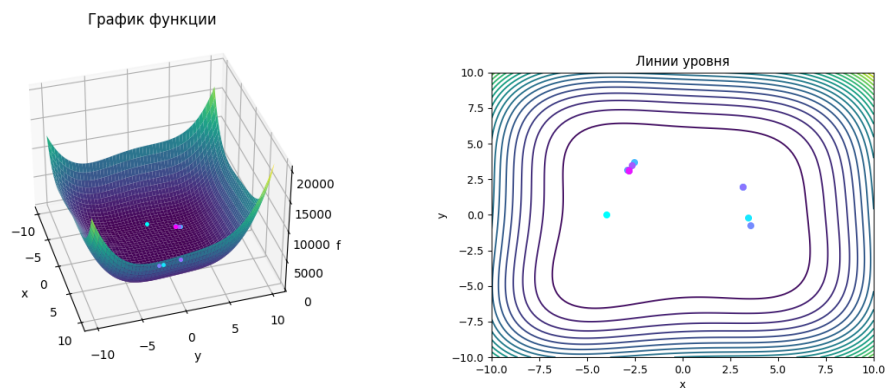


Рис. 3: Newton dichotomy

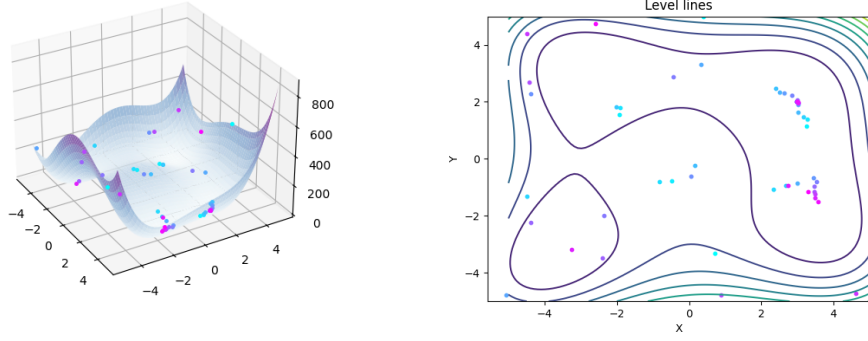


Рис. 4: ABC

Функция Изома $f = -\cos x \cdot \cos y \cdot \exp(-((x - \pi)^2 + (y - \pi)^2))$ Минимум: $f(\pi, \pi) = -1$

Начальные параметры

- Начальное приближение: $x_0 = (2, 2)$;
- Критерий останова: $\mathcal{E} = 10^{-5}$;
- Границы в методе золотого сечения/дихотомии: $l = \mathcal{E}$, $r = 100$.
- Алгоритм пчелиной колонии:
 - Количество пчел: $num_bees = 10$;
 - Количество элитных пчел: $num_elite = 3$;
 - Количество лучших участков: $num_best_sites = 10$;
 - Размер окрестности: $neighborhood_size = 0.1$;
 - Количество вылетов: $iterations = 5$;
 - Границы координат: $low = 0$, $high = 2\pi$.

Golden ratio	Nelder-Mead	Newton dichotomy	ABC
$f(3.141, 3.141) = -1$	$f(3.141, 3.141) = -1$	Минимум не найден	$f(3.15, 3.147) = -0.9998$
2 iterations	42 iterations		5 iterations
136 function	80 function		200 function
2 gradient	0 gradient		0 gradient

Таблица 2: Сравнение grad_decent golden ratio, Nelder-Mead, Newton dichotomy, ABC на функции Изома

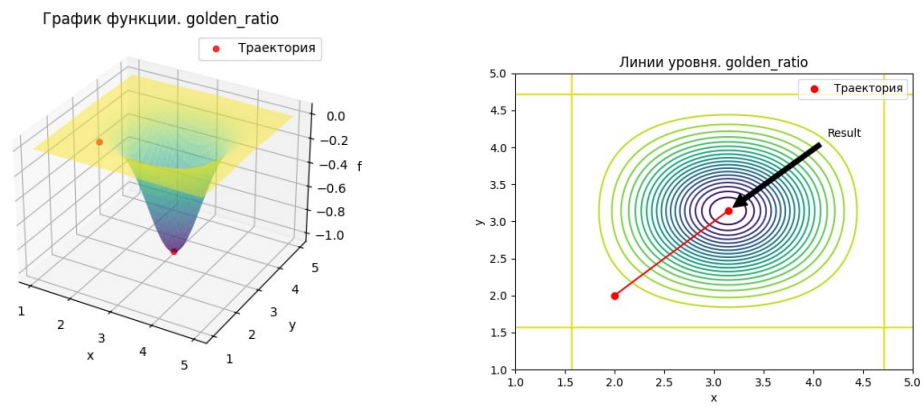


Рис. 5: Golden rate

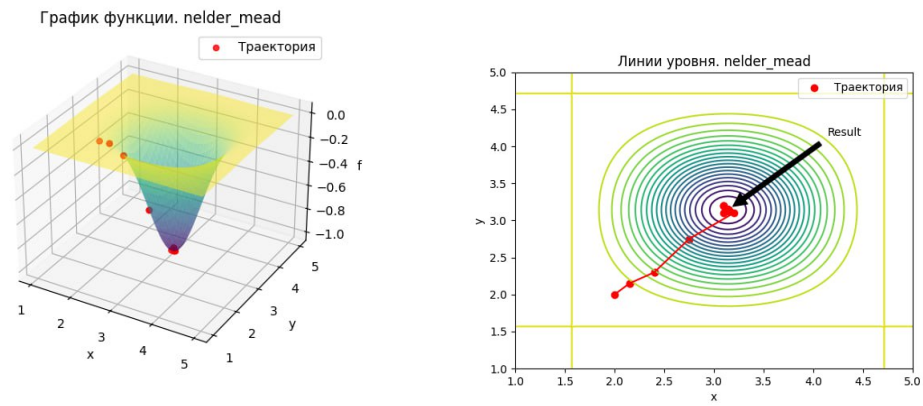


Рис. 6: Nelder-Mead

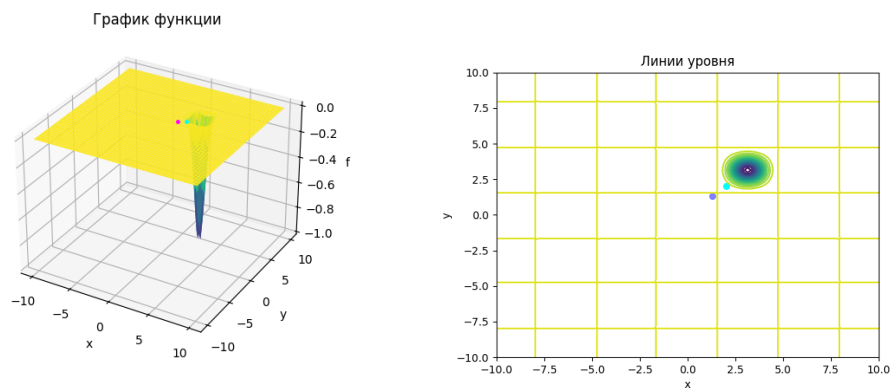


Рис. 7: Newton dichotomy

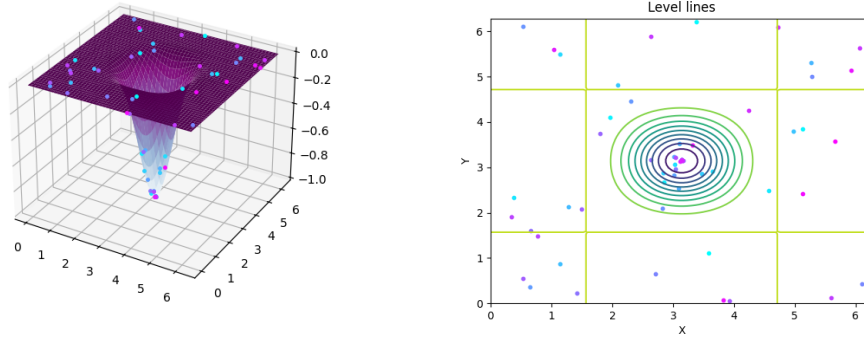


Рис. 8: ABC

Функция Розенброка $f = (1 - x)^2 + 100(y - x^2)^2$ Минимум: $f(1, 1) = 0$

Начальные параметры

- Начальное приближение: $x_0 = (-2, 1)$;
- Критерий останова: $\mathcal{E} = 10^{-5}$;
- Границы в методе золотого сечения/дихотомии: $l = \mathcal{E}$, $r = 100$.
- Алгоритм пчелиной колонии:
 - Количество пчел: $num_bees = 15$;
 - Количество элитных пчел: $num_elite = 5$;
 - Количество лучших участков: $num_best_sites = 10$;
 - Размер окрестности: $neighborhood_size = 0.25$;
 - Количество вылетов: $iterations = 50$;
 - Границы координат: $low = -5$, $high = 5$.

Golden ratio	Nelder-Mead	Newton dichotomy	ABC
$f(0.996, 0.992) = 1e-5$	$f(1, 1) = 5e-12$	$f(1, 1) = 7e-25$	$f(1.024, 1.049) = 5e-5$
410 iterations	116 iterations	15 iterations	50 iterations
27880 function	219 function	840 function	3250 function
410 gradient	0 gradient	15 gradient	0 gradient
0 hessian	0 hessian	15 hessian	0 hessian

Таблица 3: Сравнение grad_decent golden ratio, Nelder-Mead, Newton dichotomy, ABC на функции Розенброка

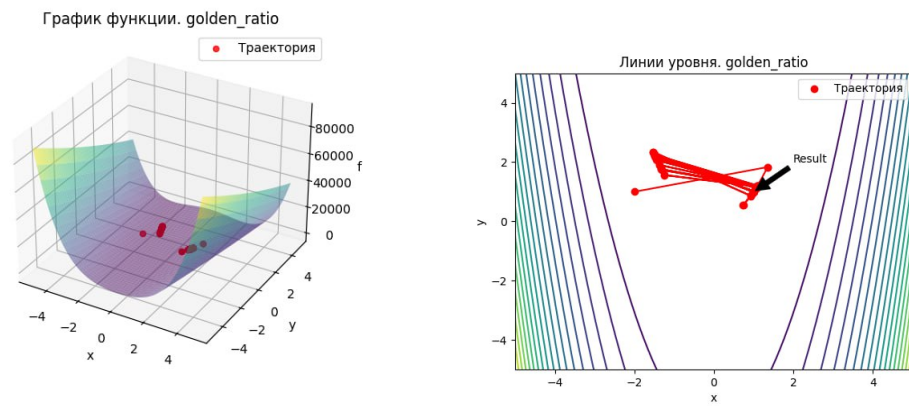


Рис. 9: Golden rate

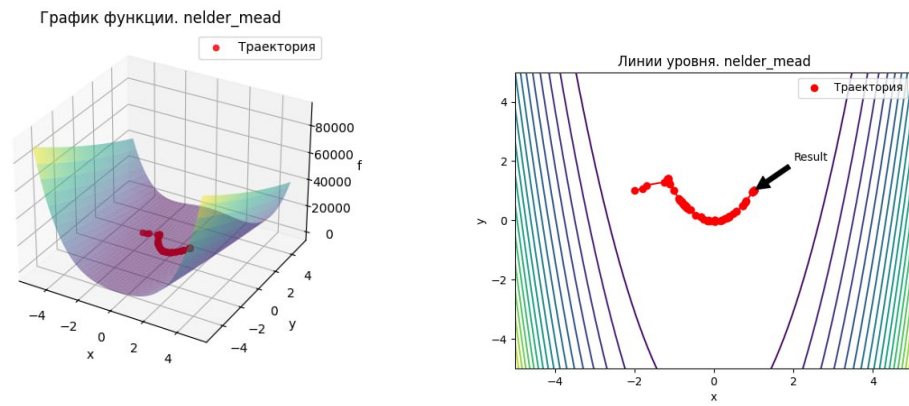


Рис. 10: Nelder-Mead

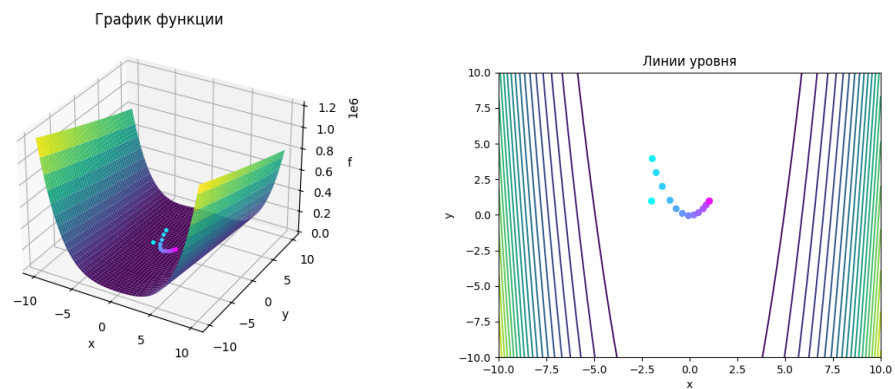


Рис. 11: Newton dichotomy

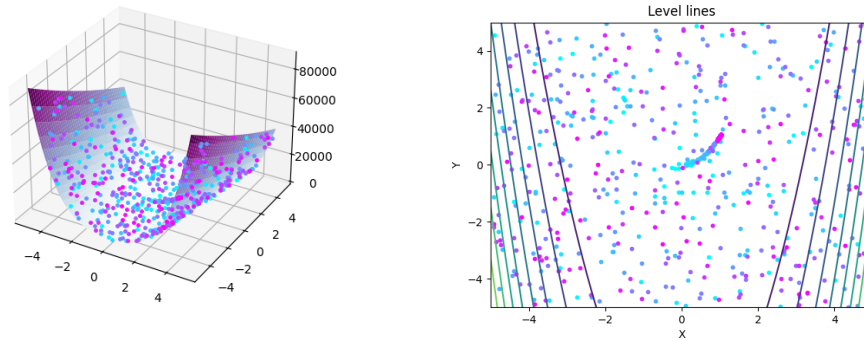


Рис. 12: ABC

Результат:

Метод пчелиного роя способен с большой вероятностью найти локальный экстремум функции, имеющей один минимум на заданной области поиска. В случае функции с большим количеством экстремумов результат сильно зависит от начальных параметров: количества пчел и итераций, размера области локального поиска, начального случайного распределения пчел. Решением данной проблемы стало увеличение начальных параметров или последовательный запуск кода несколько раз, что во всех случаях помогало найти минимум. Из-за наличия случайных значений результаты алгоритма могут варьироваться между запусками, что требует многократного запуска для повышения надежности результатов. Плюсом является и то, что данный алгоритм не требует вычисления производных.

3 Применение optuna к примерам предыдущих лабораторных

Optuna - это библиотека для автоматизации гиперпараметрического поиска и их оптимизации. В основном, ее используют уже с готовыми методами оптимизации. Однако данную библиотеку можно использовать и для поиска оптимумов функции.

Сравнение Для сравнения работоспособности и эффективности реализации с помощью optuna возьмем те же функции, что были рассмотрены в предыдущем пункте. Отметим начальные параметры нашей реализации:

- Границы координат: l и r ;
- Количество испытаний: n_trials ;
- Алгоритм оптимизации гиперпараметров: Алгоритм TPE (Tree-structured Parzen Estimator).

Функция Химмельблау Начальные параметры:

- $l = -6, r = 6$ и $l = -5, r = 5$;
- Количество испытаний: $n_trials = 50$ и $n_trials = 350$.

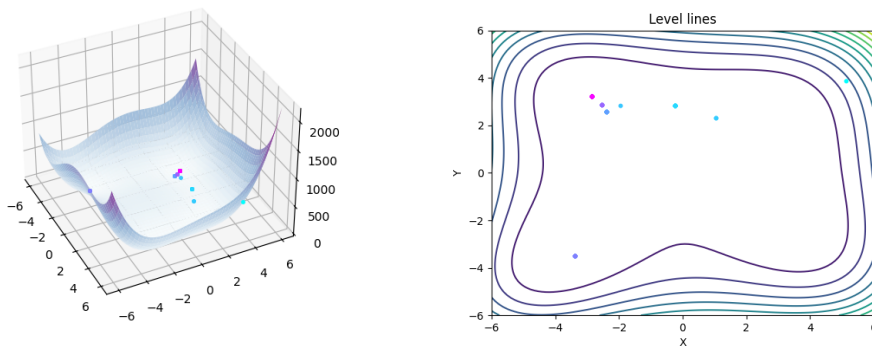


Рис. 13: $f(-2.8468, 3.2248) = 0.4141$

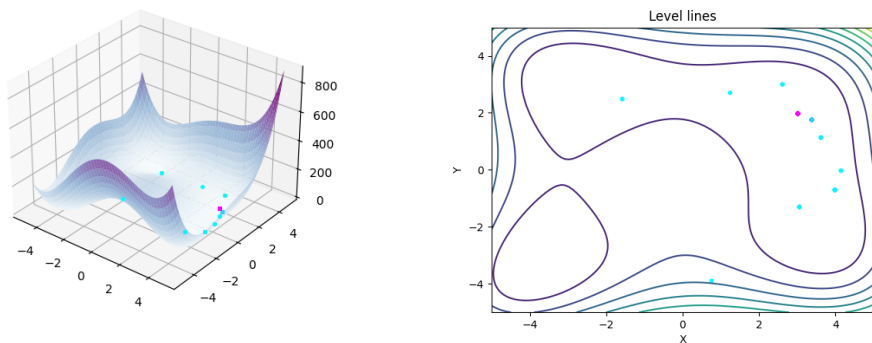


Рис. 14: $f(3.0076, 1.9884) = 0.0026$

Функция Изома Начальные параметры:

- $l = 0, r = 2\pi$ и $l = 0, r = 5$;
- Количество испытаний: $n_trials = 50$ и $n_trials = 350$.

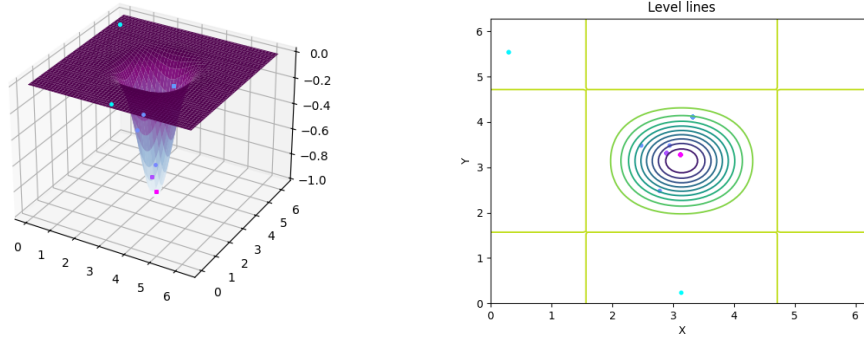


Рис. 15: $f(3.1103, 3.2819) = -0.9694$

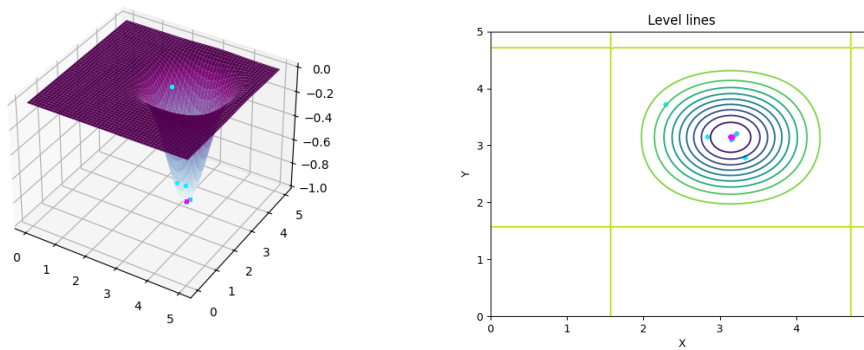


Рис. 16: $f(3.1409, 3.1392) = -0.999991$

Функция Розенброка Начальные параметры:

- $l = -4, r = 4$;
- Количество испытаний: $n_trials = 50$ и $n_trials = 350$.

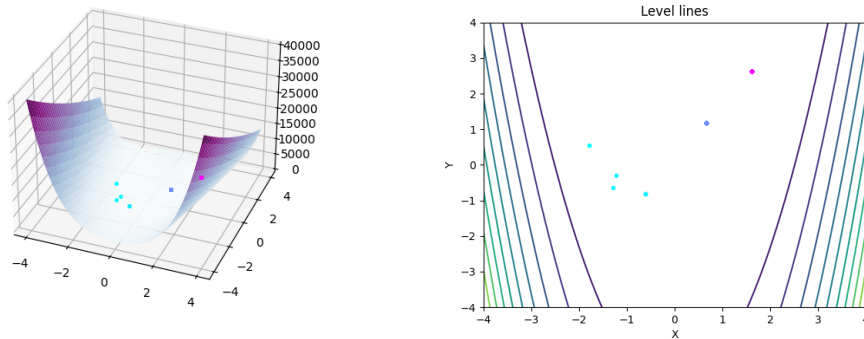


Рис. 17: $f(1.6109, 2.6360) = 0.5407$

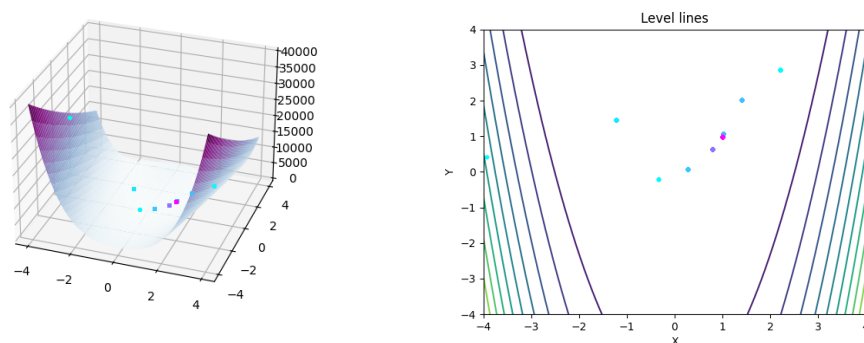


Рис. 18: $f(0.9962, 0.9826) = 0.0096$

Результат: Исходя из полученных результатов, можно сделать вывод, что реализация с помощью библиотеки `optuna` практически всегда находит верный минимум, но зачастую требуется большое количество итераций. Также стоит отметить, что на работоспособность сильно влияют значения начальных точек: при большем расстоянии от точки экстремума требовалось большее количество вычислений.

4 Применение optuna к гиперпараметрам предыдущих лабораторных

Основные компоненты Optuna

study:

- **Описание:** центральный объект в Optuna, который управляет процессом оптимизации.
- **Основные параметры:**
 - `study_name (str)`: имя исследования. Может быть полезно для идентификации исследования.
 - `direction (str)`: направление оптимизации ('minimize' или 'maximize'). Указывает, нужно ли минимизировать или максимизировать целевую функцию.
 - `storage (str или optuna.storages.BaseStorage)`: место для хранения истории исследований. Может быть локальная база данных или URL к удаленной базе данных.

trial:

- **Описание:** объект, представляющий одно испытание (evaluation) в процессе оптимизации.
- **Основные параметры:**
 - `suggest_float(name, low, high)` (метод): предлагает значение для непрерывного гиперпараметра.
 - `suggest_int(name, low, high)` (метод): предлагает значение для целочисленного гиперпараметра.
 - `suggest_categorical(name, choices)` (метод): предлагает значение для категориального гиперпараметра.

Функция потерь

Функция потерь (loss function) — это математическая функция, которая измеряет, насколько хорошо модель машинного обучения предсказывает целевые значения. Она оценивает разницу между предсказанными моделью значениями и фактическими значениями. Целью является минимизация значения функции потерь, что означает улучшение точности предсказаний модели. Мы используем MSE (Mean Squared Error): средняя квадратичная ошибка, от 0 до бесконечности. Значение 0 означает, что модель предсказывает точные значения без ошибки.

Результаты исследования

В результате нашего исследования были получены следующие оптимальные значения гиперпараметров и соответствующие значения функции потерь:

- **Newton:** Bounds: $(1e-8, 1e-1)$
 - **Best parameters:**
 - * **'eps':** $1.0664057405282948e-08$
 - **Best loss:** 0.9680457060081309

Bounds: $(1e-12, 1e-1)$

- **Best parameters:**
 - * **'eps':** $1.9190953966900052e-09$
- **Best loss:** 0.9680457060081308

Данные результаты практически совпали с используемыми нами ранее значениями в предыдущих лабораторных работах: **'eps'** = $1e-6$ для метода Ньютона. Это значение даёт не сильно худшее значение функции потерь и в нашем исследовании было вполне достаточно.

Вывод

Из полученных результатов можно сделать вывод, что данная библиотека больше подходит для оптимизации начальных параметров, нежели для полноценного решения задачи поиска экстремума. В предыдущих лабораторных работах нам приходилось запускать реализованные методы большое количество раз для нахождения оптимальных гиперпараметров. Данную проблему решает optuna: из проведенного исследования можно выделить, что найденные параметры схоже, и даже лучше, чем найденные нами ранее.