

Методы оптимизации

Отчёт по лабораторной работе №4

Выполнили:

Евсеева Арина М3234

Шульпин Егор М3232

Гаврилюк Виталий М3232

Постановка задачи

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

Алгоритм пчелиной колонии

Описание

Алгоритм роя пчел (Artificial Bee Colony Algorithm или Bees Algorithm) разработан в 2005 году. Метод основан на симуляции поведения пчел при поиске нектара. 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.

А теперь представьте, что местоположение глобального экстремума - это участок, где больше всего нектара, и этот участок единственный, то есть в других местах нектар есть, но меньше. Причем пчелы живут не на плоскости, где достаточно знать две координаты, чтобы определить расположение участков, а в многомерном пространстве, где каждая координата представляет собой один параметр функции, которую нужно оптимизировать. Количество найденного нектара - это значение целевой функции в данной точке.

Начальные параметры

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

Основной процесс

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.

Сравнение эффективности разных методов

Выбранные методы

Для сравнения были выбраны данные методы:

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

Функция Химмельблау

Функция Химмельблау $f = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$ имеет 4 локальных минимума:

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

Метод градиентного спуска с золотым сечением

Начальные параметры:

- $\mathcal{E} = 10^{-5}$;
- Начальные координаты: $x_0 = (-4, 0)$;
- Границы золотого сечения: $l = \mathcal{E}, r = 50$.

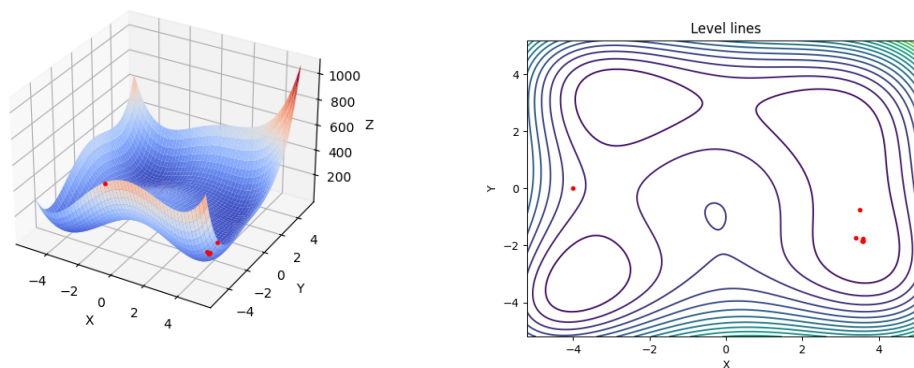


Рис. 1: $f(3, 58437, -1, 84809) = 0$

Метод Нелдера-Мида

Начальные параметры:

- $\mathcal{E} = 10^{-5}$;
- Начальные параметры: $x_0 = (-4, 0)$.

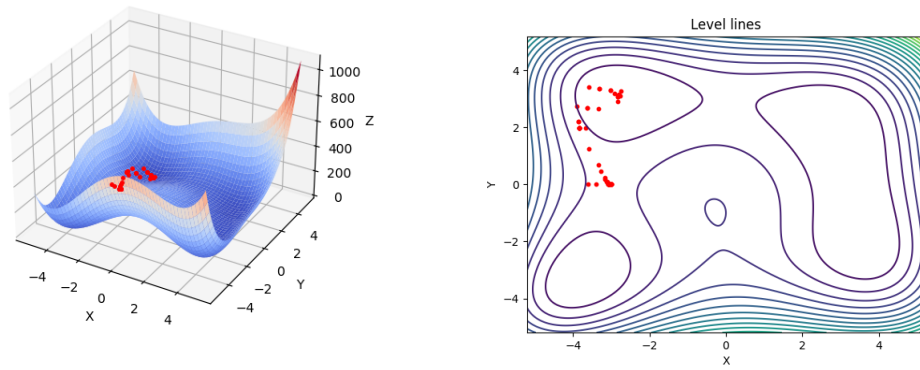


Рис. 2: $f(-2, 8051, 3, 1313) = 0$

Метод Ньютона с дихотомией

Начальные параметры:

- $\mathcal{E} = 10^{-5}$;
- Начальные координаты: $x_0 = (-4, 0)$;
- Границы дихотомии: $l = \mathcal{E}, r = 30$.

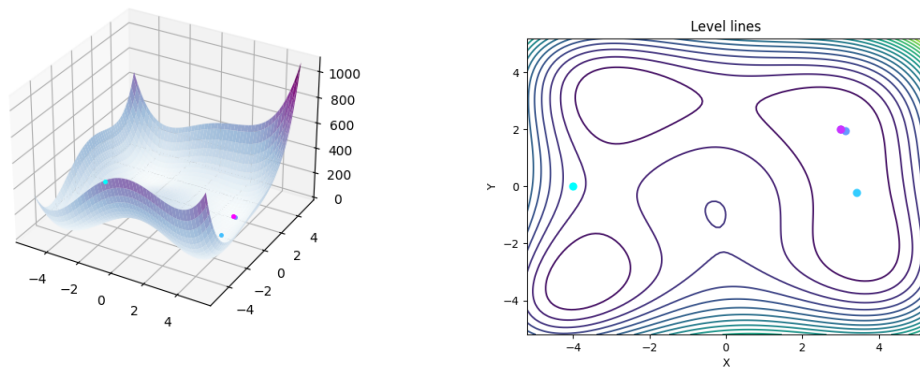


Рис. 3: $f(3, 2) = 0$

Метод пчелиной колонии

Начальные параметры:

- 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: $low = -5, high = 5$.

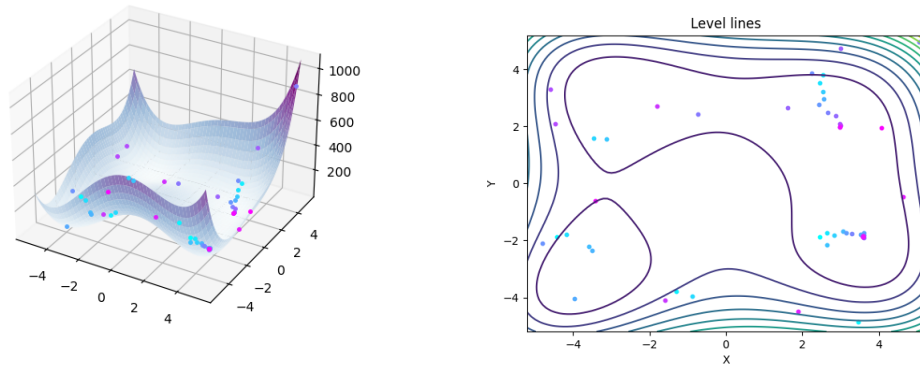


Рис. 4: $f(3, 5926, -1.8557) = 0,0039$

Полученные результаты

	Итерации	Вычисления функции	Вычисления производных	Вычисления вторых производных
Gradient descent	8	544	16	0
Nelder-Mead	99	184	0	0
Newton	5	300	10	20
Bees algorithm	10	350	0	0

Функция Изома

Функция Изома $f = -\cos x \cdot \cos y \cdot \exp(-((x - \pi)^2 + (y - \pi)^2))$ имеет глобальный минимум в точке $f(\pi, \pi) = -1$.

Метод градиентного спуска с золотым сечением

Начальные параметры:

- $\mathcal{E} = 10^{-5}$;
- Начальные координаты: $x_0 = (1, 61, 1, 61)$ - максимально удаленное значение, при котором верно находится минимум;

- Границы золотого сечения: $l = \mathcal{E}, r = 50$.

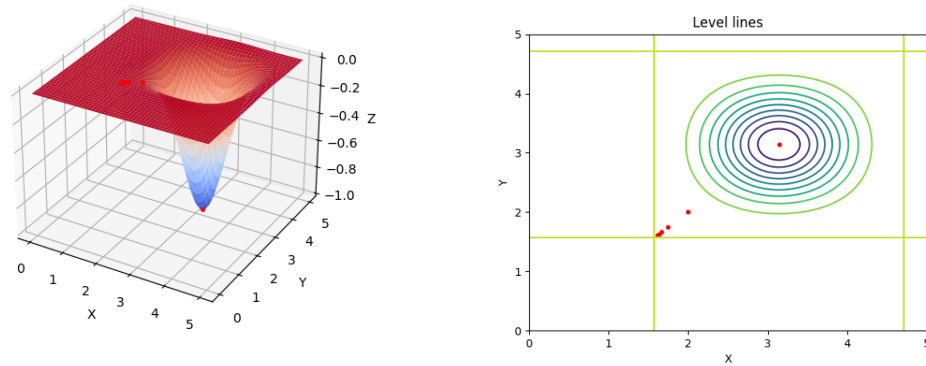


Рис. 5: $f(3, 1415, 3, 1415) = -1$

Метод Нелдера-Мида

Начальные параметры:

- $\mathcal{E} = 10^{-5}$;
- Начальные координаты: $x_0 = (1, 56, 1, 56)$ - максимально удаленное значение, при котором верно находится минимум.

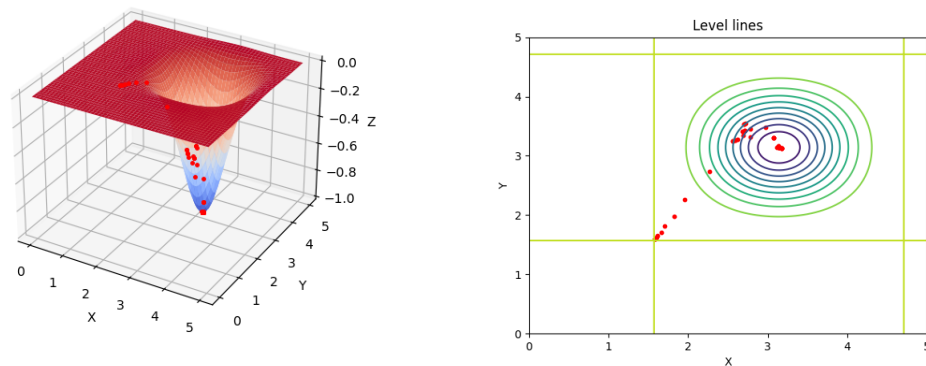


Рис. 6: $f(3, 1415, 3, 1415) = -0,9999$

Метод Ньютона с дихотомией

Начальные параметры:

- $\mathcal{E} = 10^{-5}$;
- Начальные координаты: $x_0 = (2, 14, 2, 14)$;
- Границы дихотомии: $l = \mathcal{E}, r = 30$.

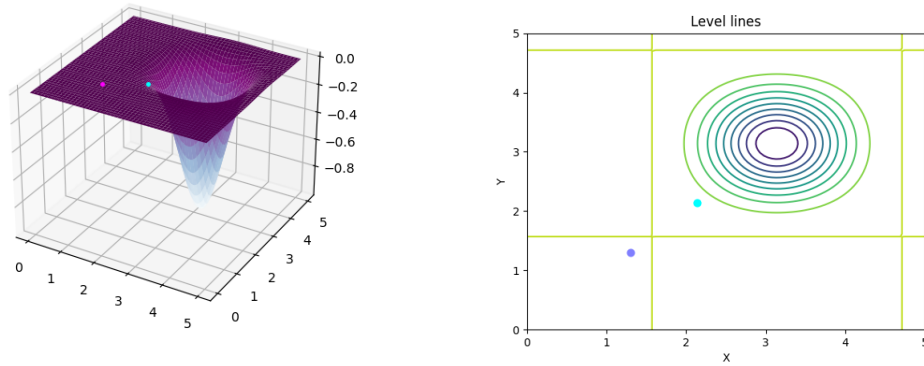


Рис. 7: $f(1, 3049, 1, 3049) = 0$

Метод пчелиной колонии

Начальные параметры:

- 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: $low = 0, high = 2\pi$.

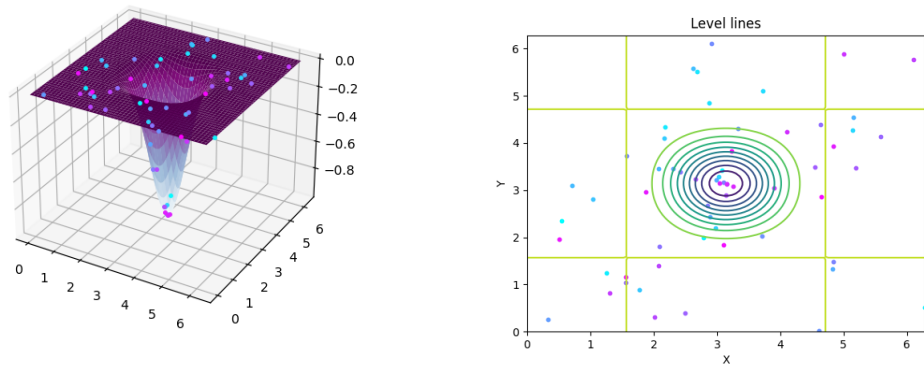


Рис. 8: $f(3, 1548, 3, 1382) = -0,9997$

Полученные результаты

	Итерации	Вычисления функции	Вычисления производных	Вычисления вторых производных
Gradient descent	6	402	12	0
Nelder-Mead	65	125	0	0
Newton	Не находит!			
Bees algorithm	5	200	0	0

Функция Розенброка

Каноническое представление функция Розенброка $f = (1 - x)^2 + 100(y - x^2)^2$ имеет глобальный минимум $f(1, 1) = 0$.

Метод градиентного спуска с золотым сечением

Начальные параметры:

- $\mathcal{E} = 10^{-5}$;
- Начальные координаты: $x_0 = (-2, 3)$;
- Границы золотого сечения: $l = \mathcal{E}, r = 50$.

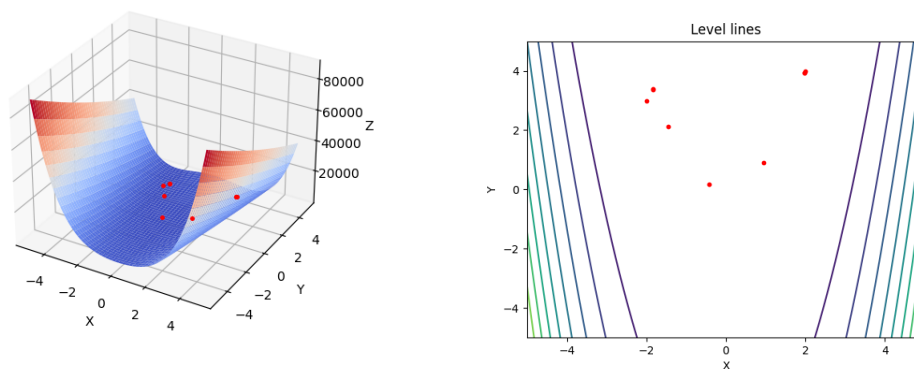


Рис. 9: $f(0,9457, 0,8942) = 0,0029$

Метод Нелдера-Мида

Начальные параметры:

- $\mathcal{E} = 10^{-5}$;
- Начальные координаты: $x_0 = (-2, 3)$.

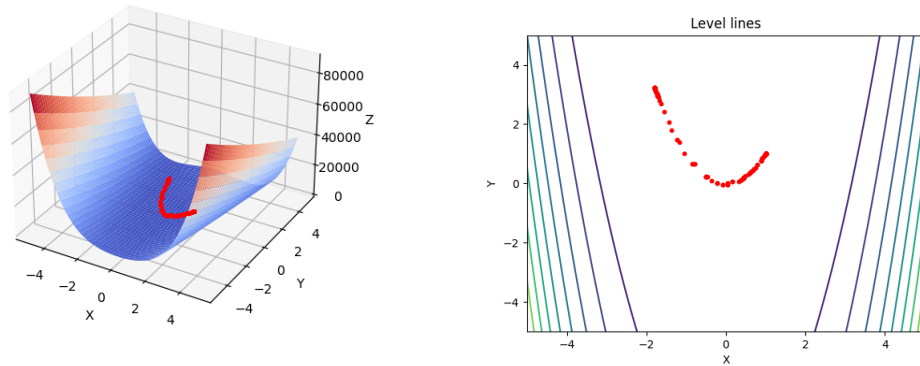


Рис. 10: $f(1,1) = 0$

Метод Ньютона с дихотомией

Начальные параметры:

- $\mathcal{E} = 10^{-5}$;
- Начальные координаты: $x_0 = (1,1) = 0$

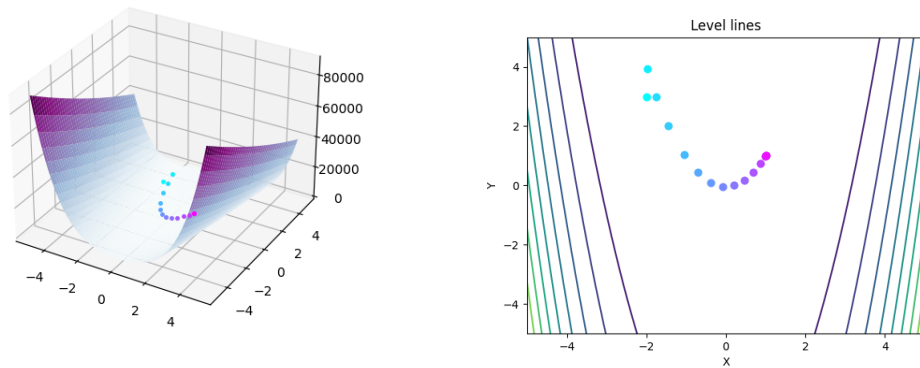


Рис. 11: $f(1,1) = 0$

Метод пчелиной колонии

Начальные параметры:

- 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: $low = -5, high = 5$.

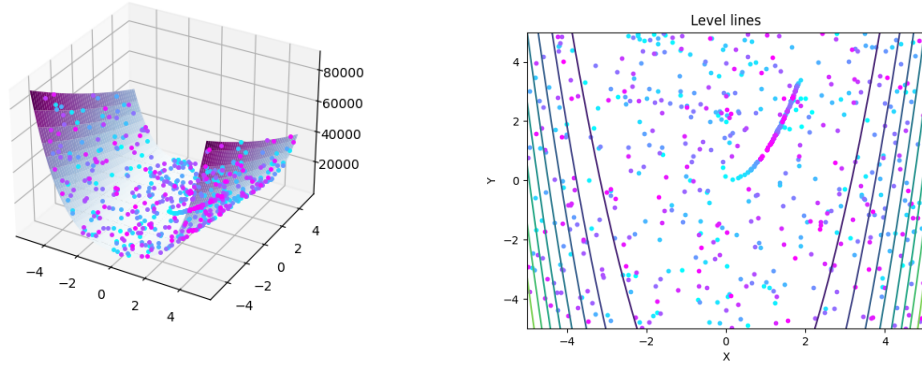


Рис. 12: $f(1, 0201, 1, 0441) = 0, 0016$

Полученные результаты

	Итерации	Вычисления функции	Вычисления производных	Вычисления вторых производных
Gradient descent	9	612	18	0
Nelder-Mead	116	223	0	0
Newton	15	900	30	60
Bees algorithm	50	3250	0	0

Вывод

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. Также при рассмотрении функции Изома, которая является функцией плоского вида, методу пчелиного роя в 10% случаев не удалось найти минимум из-за неудачно подобранных начальных параметров. 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.

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 $n_trials = 250$.

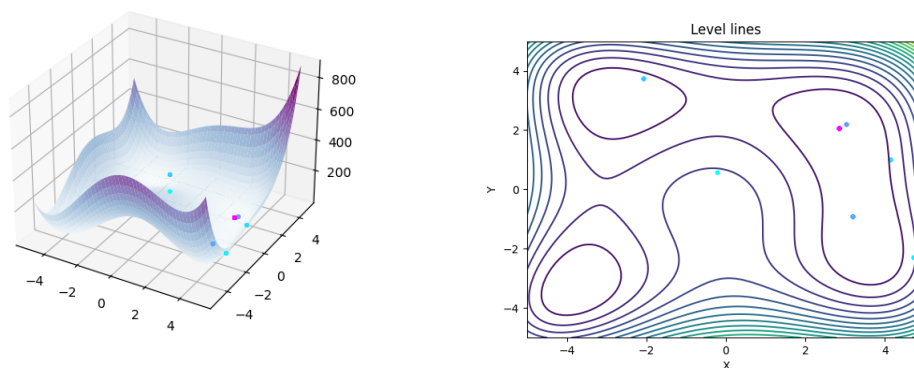


Рис. 13: $f(2,8424,2,0652) = 0.7428$

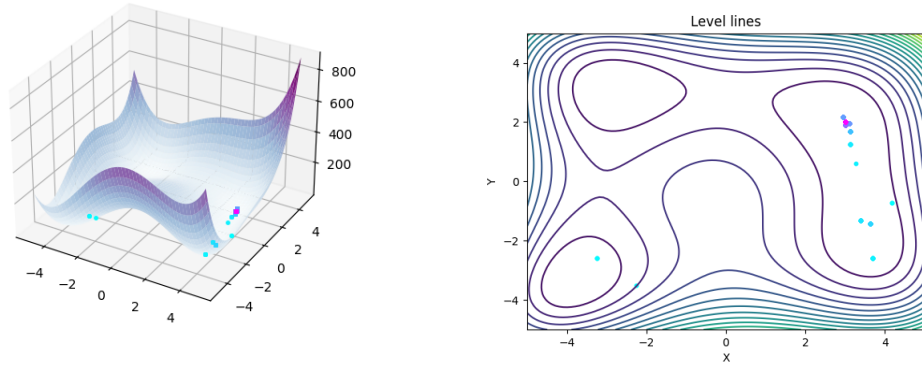


Рис. 14: $f(3,003,2) = 0,0003$

Функция Изома

Начальные параметры:

- $l = 0, r = 2\pi$ и $l = 0, r = 5$;
- Количество испытаний: $n_trials = 50$ и $n_trials = 250$.

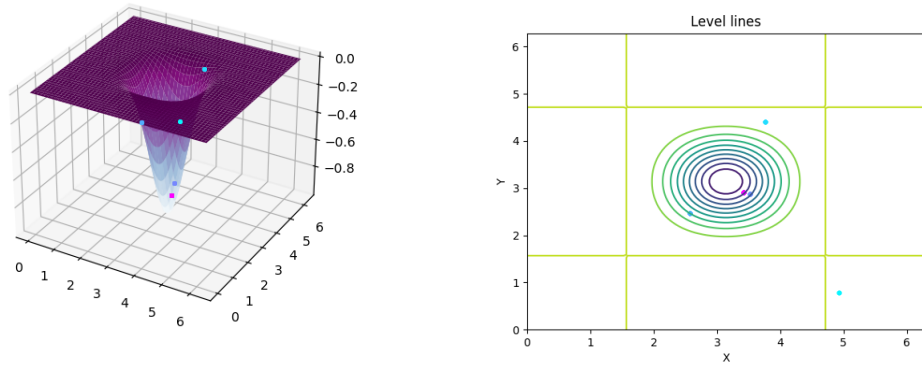


Рис. 15: $f(3,416,2,9083) = -0,8225$

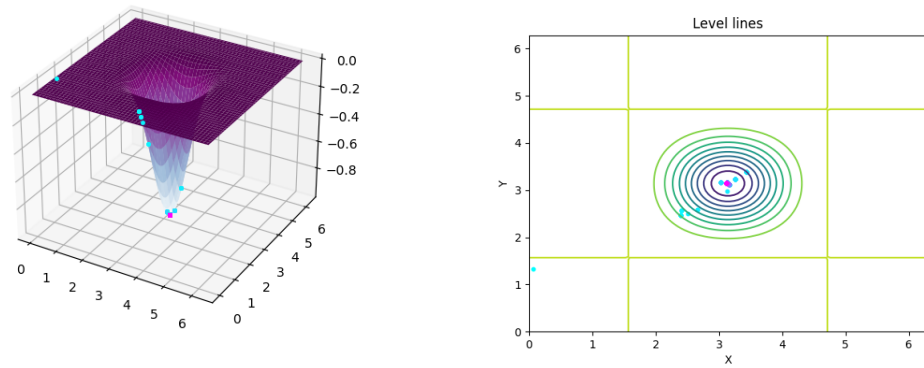


Рис. 16: $f(3, 1224, 3, 1456) = -0,9994$

Функция Розенброка

Начальные параметры:

- $l = -4, r = 4$;
- Количество испытаний: $n_trials = 50$ и $n_trials = 250$.

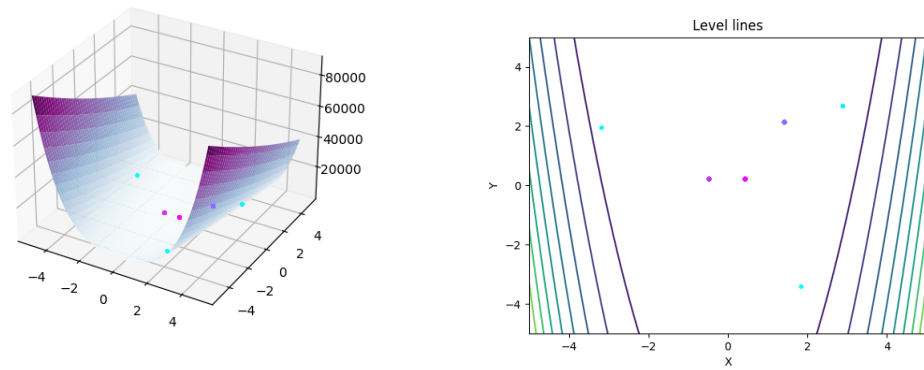


Рис. 17: $f(0, 4284, 0, 2219) = 0,4738$

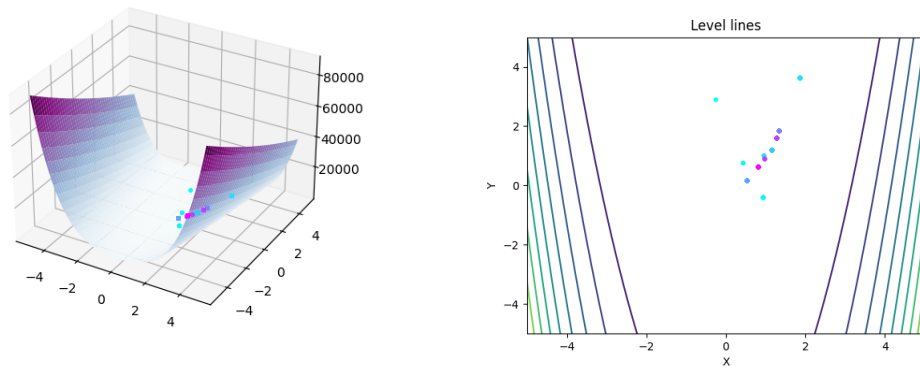


Рис. 18: $f(0,7996, 0,6388) = 0,0401$

Вывод

Исходя из полученных результатов, можно сделать вывод, что реализация с помощью библиотеки `optuna` практически всегда находит верный минимум, но зачастую требуется большое количество итераций. Например, при количестве итераций меньше 250 метода часто не сходилась к нужному оптимуму. Также стоит отметить, что на работоспособность сильно влияют значения начальных точек: при большем расстоянии от точки экстремума требовалось большее количество вычислений.

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 моего исследования были получены следующие оптимальные значения гиперпараметров и соответствующие значения функции потерь:

- **SGD:**
 - **Best parameters:**
 - * `'learning_rate'`: 0.036957644796091156
 - * `'eps'`: 1.2658880982575924e-05
 - **Best loss:** 0.9703574489061714
- **Newton:**
 - **Best parameters:**
 - * `'eps'`: 1.0677599746102762e-08
 - **Best loss:** 0.9680457060081309

Данные результаты практически совпали с используемыми нами ранее значениями в предыдущих лабораторных работах: `'eps'` = $1e-7$ для метода Ньютона и `'eps'` = $1e-5$ для стохастического градиентного спуска.

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.