

Лабораторная работа №4	2024
<b><i>Метод симуляции отжига</i></b>	<b>Артемьев Иван Вадимович</b>
	<b>Никитин Михаил Алексеевич</b>
	<b>Павлов Евгений Андреевич</b>

**Цель работы:** Реализация стохастического метода - имитации отжига для поиска минимумов функций на языке программирования Python и исследование полученных результатов

**Инструментарий:** Python 3.12 - библиотеки: NumPy, SciPy, Matplotlib

#### **Постановка задачи:**

1. Разберите теоретическое описание и реализуйте метод стохастической оптимизации (к примеру, метод имитации отжига);
2. Сравните его эффективность на методах и примерах из лаб. 1 и/или лаб 2. Приведите примеры, иллюстрирующие разницу в результатах и эффективности (количество вызовов функции, скорости) применения методов из лаб. 1 и/или лаб 2 и методов стохастической оптимизации.

#### **Описание метода отжига:**

**Метод имитации отжига** — это вероятностная эвристическая техника для решения задач глобальной оптимизации, которая особенно эффективна в случаях, когда пространство поиска велико и содержит множество локальных экстремумов. Метод вдохновлен физическим процессом отжига, который заключается в нагреве и медленном

охлаждении материала для достижения его наиболее стабильного состояния.

### Основные этапы метода имитации отжига

#### 1. Инициализация:

- Выбирается начальное состояние (начальная точка поиска) и начальная температура.

#### 2. Итерационный процесс:

- *Генерация соседнего состояния:* На каждом шаге выбирается новое состояние в окрестности текущего состояния путем небольшого изменения последнего.

- *Оценка состояния:* Вычисляется значение целевой функции для нового состояния.

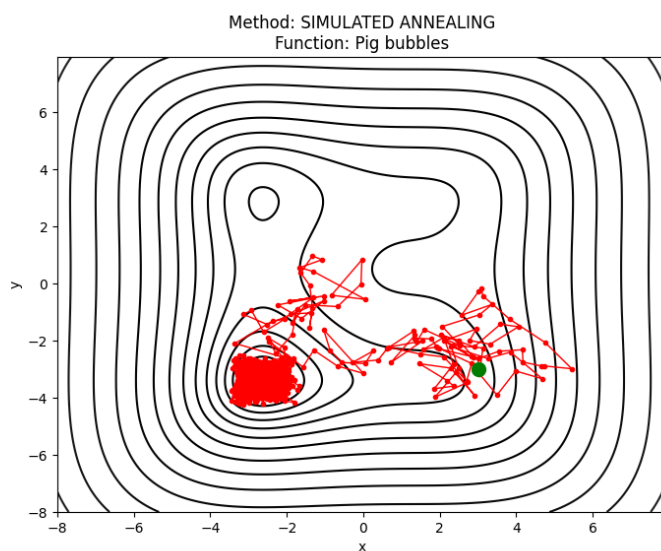
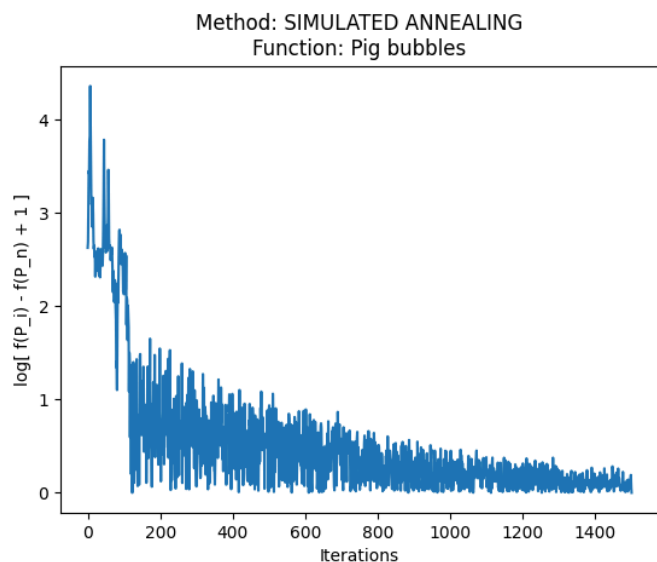
- *Принятие решения:* Новое состояние принимается, если оно улучшает значение целевой функции. Если же новое состояние хуже текущего, оно может быть принято с некоторой вероятностью, зависящей от разницы значений целевой функции и текущей температуры. Вероятность принятия ухудшающего состояния определяется формулой:

$$P = \exp\left(-\frac{\Delta E}{T}\right)$$

где  $(\Delta E)$  — разница значений целевой функции, а  $T$  — текущая температура.

- *Обновление температуры:* Температура постепенно уменьшается согласно заданному правилу (например, экспоненциальное уменьшение:  $T = T_0 * \alpha^{step}$ , где  $0 < \alpha < 1$  - *cooling rate*, *step* - номер итерации).

3. Завершение процесса: Алгоритм останавливается, когда достигается заданное число итераций или температура становится очень низкой.



### ***Преимущества и особенности***

- *Избежание локальных экстремумов:* Возможность принимать ухудшающие состояния позволяет алгоритму выходить из локальных минимумов и продолжать поиск глобального оптимума.
- *Гибкость:* Метод применим к широкому спектру задач оптимизации и может быть легко адаптирован под конкретную задачу путем выбора подходящих функций соседства и правил охлаждения.
- *Параметры:* Важными параметрами метода являются начальная температура, правило охлаждения и механизм генерации соседних состояний. Эти параметры могут существенно влиять на эффективность и результативность алгоритма.

**Сравнение метода имитации отжига с методами из первых двух лаб:**

**Анализ сходимости:**

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$
Градиентный спуск	90%	88%	7%	0%	0%	100%
Градиентный спуск с дихотомией	89%	87%	7%	0%	0%	100%
Нелдер-Мид	100%	100%	100%	91%	100%	99.7%
Покоординатный спуск	100%	100%	100%	10%	100%	100%
Метод Ньютона	100%	80%	0%	0%	100%	100%
Метод Ньютона с дихотомией	100%	80%	0%	0%	100%	100%
Newton-CG	100%	100%	28%	2%	84%	100%
scipy BFGS	100%	100%	16%	22%	100%	100%
JBFGS	82%	57%	0%	0%	25%	88%
ИМИТАЦИЯ ОТЖИГА	100%	100%	100%	100%	96%	100%

**Количества вызовов:**

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$
Градиентный спуск	263.55	150.7	314.3	-	-	563.568
Градиентный спуск с дихотомией	429.46	268.8	526.6	-	-	870.552
Нелдер-Мид	93.431	95.06	88.37	98.55	178.4	98.798
Покоординатный спуск	182.53	196.1	147.1	133.9	44626	235.699
Метод Ньютона	330.76	780.4	61.29	-	906.6	310.032
Метод Ньютона с дихотомией	427.48	1152	65.58	-	1001	408.052
Newton-CG	149.37	191.7	102.3	474.8	1924	102.694
scipy BFGS	30.048	50.78	8.364	257.1	249.4	17.7
JBFGS	121.80	174.6	21.62	-	723.3	50.585

ИМИТАЦИЯ ОТЖИГА	1502	7502	1502	6002	6502	1502
-----------------	------	------	------	------	------	------

**Время работы:**

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$
Градиентный спуск	0.33	0.25	0.06	-	-	0.67
Градиентный спуск с дихотомией	0.56	0.43	0.1	-	-	1.17
Нелдер-Мид	1.37	0.74	0.69	0.74	1.21	0.84
Покоординатный спуск	0.04	0.06	0.08	0.13	10.93	0.06
Метод Ньютона	0.28	0.91	0.07	-	0.67	0.27
Метод Ньютона с дихотомией	0.37	1.28	0.08	-	0.82	0.4
Newton-CG	0.38	0.5	0.39	1.12	4.76	0.28
scipy BFGS	1.03	1.7	0.25	8.22	9.51	0.61
JBFGS	0.21	0.42	0.02	-	1.52	0.08
ИМИТАЦИЯ ОТЖИГА	17.36	92.19	21.51	79.76	86.93	21.2

Вывод: как можно наблюдать: имитация отжига довольно довольно точна при правильно подобранных гиперпараметрах, однако за точность приходится платить большим количеством итераций, а следовательно большим количеством вызовов и временем работы.

**Ссылка на репозиторий с кодом:**

<https://github.com/Sedromun/lab4-MetOpt>

