

MetOpt4

Дроздов Олег, Широков Данил, Исламова Камиля

23 мая 2025 г.

1 Введение

Методы стохастической оптимизации находят широкое применение при решении задач, в которых отсутствует явная производная, присутствует большое количество локальных минимумов или функция обладает шумом. Одним из таких методов является метод имитации отжига (Simulated Annealing), который был вдохновлён физическим процессом охлаждения металлов. В данной лабораторной работе мы реализуем абстрактный алгоритм имитации отжига и протестируем его на различных тестовых функциях, включая мультимодальные и шумные.

Целью работы является:

- Реализация метода имитации отжига в абстрактной форме.
- Применение метода к различным функциям оптимизации.
- Сравнение поведения и эффективности алгоритма на различных типах функций

2 Метод имитации отжига

2.1 Описание алгоритма

Алгоритм имитации отжига основан на идее последовательного "охлаждения" системы: на каждом шаге случайным образом изменяется состояние, и новое состояние принимается либо в случае улучшения, либо с некоторой вероятностью в случае ухудшения. Это позволяет алгоритму

выходить из локальных минимумов, что особенно важно при наличии сложного ландшафта целевой функции.

В данной реализации метод отжига построен над абстрактной моделью `AnnealingModel`, которая реализует следующие действия:

1. Оценка текущего состояния модели — метод `evaluate()`, возвращающий значение целевой функции.
2. Изменение состояния модели — метод `modify()`, принимающий аргумент (масштаб изменения), и случайным образом изменяющий параметры.
3. Откат изменения — метод `rollback()`, позволяющий вернуть предыдущее состояние в случае отклонения изменения.
4. Получение текущего состояния — метод `get_state()`, возвращающий координаты (например,)

В качестве оптимизируемой модели выступает `FunctionModel`, работающая с функцией. Для управления алгоритмом используются следующие параметры:

1. Функция температуры `temperature_function(i)` — определяет "лояльность" системы к ухудшениям:

$$\text{temperature_function}(i) = a \cdot e^{-bi}$$

где a — начальная температура, а b — коэффициент затухания.

2. Функция изменения параметров `get_attrs(i)` определяет масштаб изменения параметров на текущей итерации. Возможны различные реализации, например:

- постоянный масштаб изменения;
- убывающая амплитуда, обратно пропорциональная номеру итерации i :

$$\text{get_attrs}(i) = \frac{c}{\sqrt{i}}, \quad c > 0.$$

3. Функция допусков `get_tolerance(delta, T)` возвращает вероятность принять ухудшающее изменение:

$$\text{get_tolerance}(\text{delta}, T) = e^{-\Delta/T},$$

где $\Delta = \text{after} - \text{before}$ — разность значений целевой функции до и после изменения, а T — температура на данной итерации.

Основной цикл алгоритма работает следующим образом:

- Оценивается текущее значение целевой функции.
- Параметры изменяются с заданным масштабом.
- Оценивается новое значение.
- Если оно лучше — принимается.
- Если оно хуже — принимается с вероятностью, зависящей от температуры и разности значений. В противном случае — откат изменений.

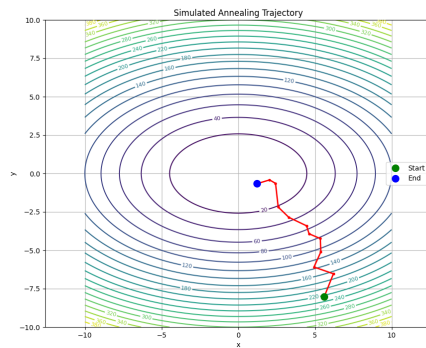
Таким образом, метод имитации отжига балансирует между исследованием пространства решений и постепенным "замораживанием" (по мере убывания температуры), приближаясь к минимуму функции.

2.2 Результаты исследования на различных функциях

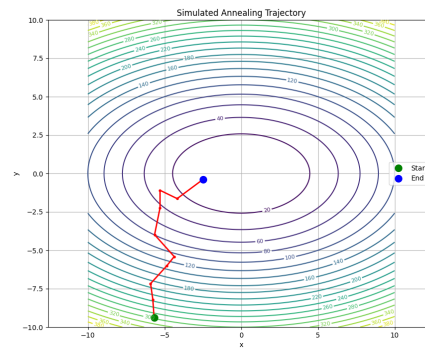
2.2.1 Гладкая функция

$$f(x, y) = x^2 + 3y^2$$

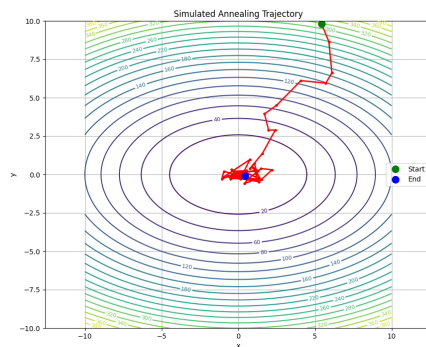
Если, увеличить число итераций до 500, то сходимость улучшается, а если ещё уменьшать шаг, то станет ещё лучше.



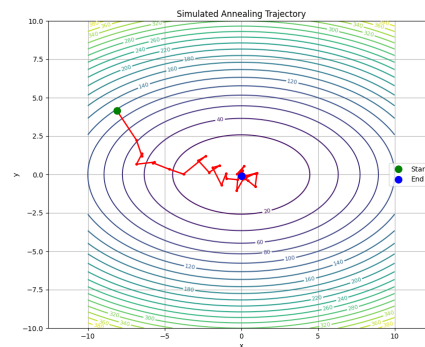
(a) 50 итераций, обычная температура



(b) 50 итераций, температура 10^{-8}



(c) 500 итераций, обычная температура



(d) 500 итераций, уменьш. шаг, обычная температура

2.2.2 Функция с шумом

$$f(x, y) = x^2 + 3y^2 + M \cdot \sin(5x) \sin(3y) \sin(\pi x) \sin\left(\frac{\pi y}{2}\right)$$

Параметр M определяет амплитуду шумовой компоненты. При больших значениях M функция приобретает множество локальных минимумов, затрудняя поиск глобального минимума. Тем не менее, за счёт вероятностного допуща ухудшающих шагов алгоритм имитации отжига способен преодолевать локальные ямы и находить области, близкие к глобальному минимуму.

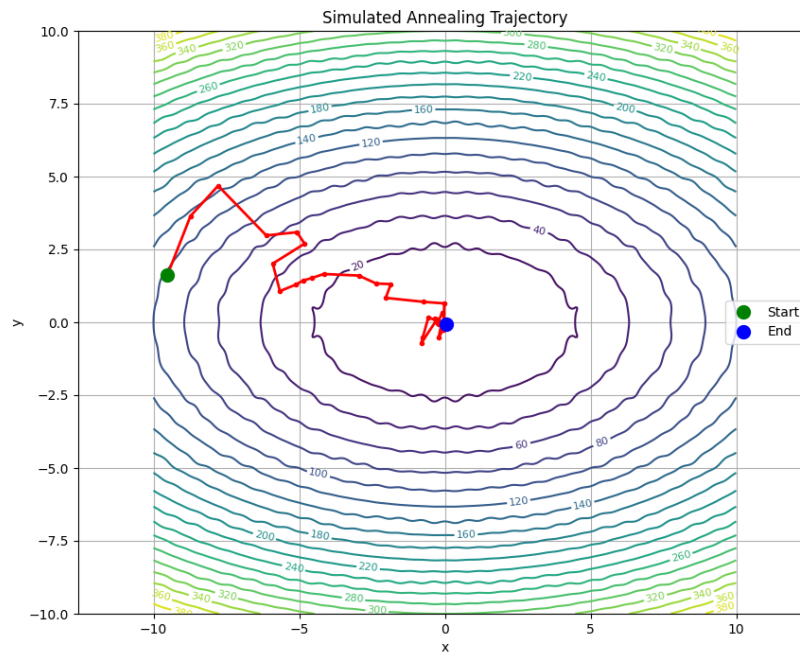


Рис. 2: Небольшой шум, 500 итераций, обычная функция температуры, уменьшающийся шаг

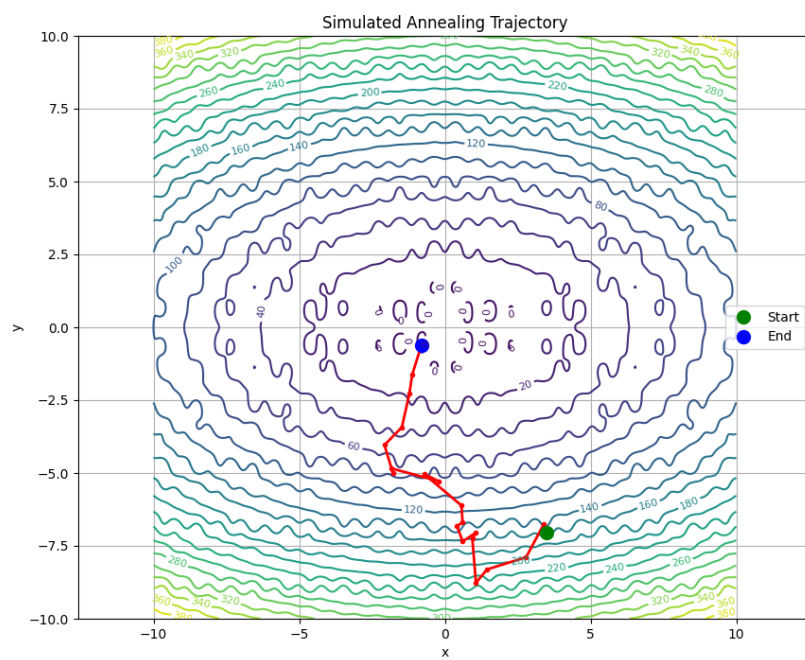


Рис. 3: Сильный шум, 500 итераций, обычная функция температуры, уменьшающийся шаг

2.2.3 Мультимодальная функция

$$f(x, y) = 0.1x^2 + 0.1y^2 + M \cdot \left(\sin \left(x - \frac{\pi}{2} \right) + \sin \left(y - \frac{\pi}{2} \right) \right)$$

Данная функция обладает чётко выраженными периодическими потенциальными ямами. При малых значениях параметра M доминирует квадратичная часть, и функция остаётся относительно простой для оптимизации. При увеличении M возрастает глубина ложных минимумов, однако при правильно подобранной температурной функции алгоритм продолжает успешно избегать преждевременной сходимости и находит решения, близкие к глобальному минимуму.

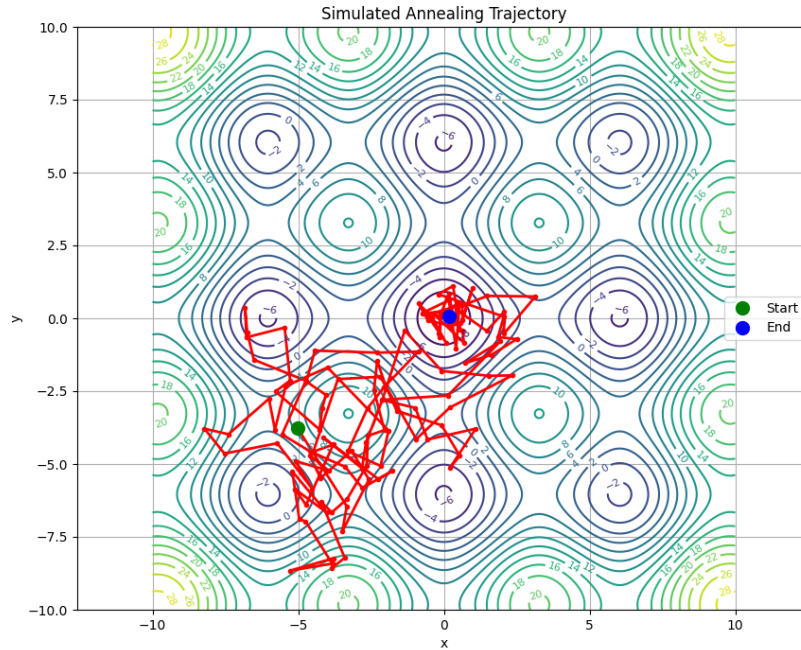


Рис. 4: $M = 5$, 500 итераций

2.2.4 Выводы

Метод имитации отжига показал высокую устойчивость к наличию локальных минимумов и шумов в целевой функции. Особенно хорошо он проявил себя на зашумленных и мультимодальных функциях при корректной настройке параметров температуры и масштаба изменений. Это делает его удобным инструментом для стохастической оптимизации в задачах с неизвестной или сложной ландшафтной структурой.

3 Дополнительное задание 1: Генетический алгоритм

3.1 Описание алгоритма

Генетический алгоритм (ГА) имитирует процессы естественного отбора и эволюции для поиска глобального минимума заданной функции $f(x, y)$. Алгоритм работает следующим образом:

1. **Инициализация:** создаётся популяция моделей с параметрами, случайно сгенерированными в области $[-10, 10]^2$.
2. **Оценка приспособленности:** для каждой особи вычисляется значение целевой функции f , выступающее в роли "приспособленности".
3. **Селекция:** отбирается часть лучших особей (по критерию наименьшего значения f) в соответствии с коэффициентом селекции.
4. **Скращивание:** с заданной вероятностью производится скращивание случайных пар выживших особей путём усреднения их параметров.
5. **Мутация:** к потомкам применяется мутация, добавляющая шум с заданным масштабом.
6. **Обновление популяции:** новая популяция формируется из выживших и потомков.
7. **Повтор:** шаги 2–6 повторяются в течение фиксированного числа поколений.

Визуализация включает отображение:

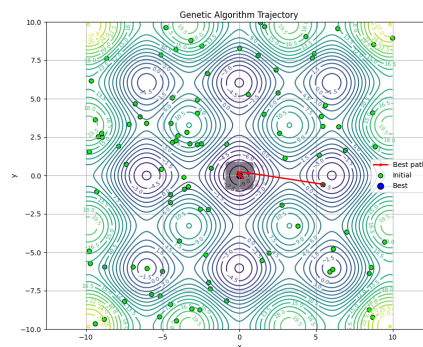
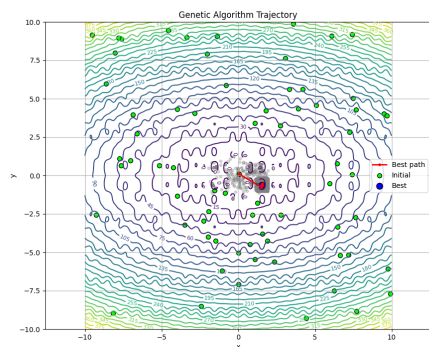
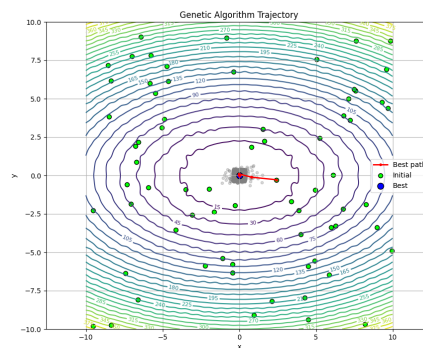
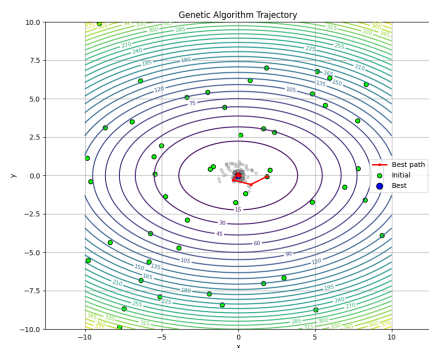
- траектории лучших особей по поколениям (красная линия),
- облаков популяции через заданные интервалы поколений (серые точки),
- начальной популяции (зелёные точки),
- лучшей найденной особи (синяя точка).

3.2 Выводы по экспериментам

Алгоритм был протестирован на трёх функциях:

1. Эллиптическая функция: $f(x, y) = x^2 + 3y^2$
2. Шумная функция: $f(x, y) = x^2 + 3y^2 +$
3. Мультимодальная функция: $f(x, y) = 0.1(x^2 + y^2) + 5 \cdot (\sin(x - \pi/2) + \sin(y - \pi/2))$

3.2.1 Графики



Подбор параметров показал следующее:

- Для гладкой эллиптической функции генетический алгоритм быстро сходится при умеренной мутации и средней популяции: `mutation_scale=0.3`, `population_size=50`.
- Для шумной функции необходимы более высокие значения мутации, чтобы избежать преждевременной сходимости: `mutation_scale=0.5-0.7`.
- Для мультимодальной функции требуется большая популяция и высокая мутация, чтобы избежать локальных минимумов: `population_size=100`, `mutation_scale=0.8`.

Генетический алгоритм показал устойчивую способность находить хорошие решения даже в условиях множественных локальных минимумов и шумов, особенно при корректной настройке параметров мутации, селекции и кроссовера.

4 Конец

Спасибо за внимание!