МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
«Национальный исследовательский   
Нижегородский государственный университет им. Н.И. Лобачевского»

(ННГУ)

Институт информационных технологий, математики и механики

Передовая инженерная школа

Направление подготовки: «Прикладная информатика»

Отчет по лабораторной работе

на тему:

**«Метод Нелдера – Мида»**

**Выполнили:**

студенты группы 3821Б1ПИмэ

Волков Д.А.

Бузникова М.О

Гурьянов И.А

И студенты группы 3821Б1ПИсм

Стаценко А.П

Трошанин Ф.И

**Преподаватель:**

Сморякова Валентина Михайловна

Нижний Новгород

2024

Оглавление

[Введение 3](#_Toc166532953)

[Цель работы: 4](#_Toc166532954)

[Теоретический материал 5](#_Toc166532955)

[Алгоритм 6](#_Toc166532956)

[Идентификация функций, проходящих через метод 7](#_Toc166532957)

[Вывод 10](#_Toc166532958)

[Примечание 11](#_Toc166532959)

[Источники: 12](#_Toc166532960)

# Введение

Метод Нелдера — Мида, также известный как метод деформируемого многогранника и симплекс-метод, — метод безусловной оптимизации функции от нескольких переменных, не использующий производной градиентов функции.

Метод Нелдера – Мида был разработан в 1965 году Джоном Нелдером и Роджером Мидом. Они искали метод оптимизации, который мог бы работать с функциями, не имеющими производных или не имеющими аналитических формул для производных. Они также хотели разработать метод, который был бы прост в реализации и эффективен для использования на вычислительных машинах того времени. Исследования привели их к идее использования симплекса - многогранника в пространстве параметров функции.

Симплекс представляет собой набор точек, образующих многогранник, где каждая точка – это набор значений параметров оптимизируемой функции. Идея заключается в том, чтобы изменять и перемещать симплекс в пространстве параметров, чтобы найти оптимальное значение функции.

# Цель работы:

Изучить метод Нелдера-Мида, научиться применять его на практике посредством реализации на языке программирования Python.

# Теоретический материал

Метод Нелдера – Мида применяется для нахождения решения задачи оптимизации вещественных функций многих переменных

причем функция f(x) не является гладкой.

Другой особенностью метода является то, что на каждой итерации вычисляется значение функции f(x) не более чем в трех точках. Это особенно важно в случае сложно-вычислимой функции f(x). Метод Нелдера – Мида прост в реализации и полезен на практике, так как не подразумевает поиск решения через производные. Но, с другой стороны, для него не существует теории сходимости[[1]](#footnote-1) — алгоритм может расходиться даже на гладких функциях.

Симплексом (n-симплексом) называется выпуклая оболочка множества[[2]](#footnote-2) независимых (n + 1) точек (вершин симплекса).

# Алгоритм

Алгоритм метода Нелдера – Мида включает следующие шаги:

1. Инициализация: строится симплекс - многогранник в n-мерном пространстве, где n - количество переменных функции оптимизации;
2. Оценка значений функции: вычисляются значения функции в вершинах симплекса;
3. Сортировка: вершины сортируются в порядке убывания значений функции;
4. Центр тяжести: вычисляется центр тяжести симплекса без наихудшей вершины;
5. Отражение: проводится отражение на основе центра тяжести и лучшей вершины симплекса;
6. Оценка нового значения функции в отраженной точке;
7. Расширение: если новая точка лучше, чем лучшая вершина симплекса, происходит расширение в этом направлении;
8. Сжатие: если новая точка хуже, чем наихудшая вершина, происходит сжатие симплекса в сторону лучшей вершины;
9. Метод средней точки: если ни одно из вышеупомянутых действий не улучшает значение функции, то сжимаем симплекс в сторону центра тяжести без наихудшей вершины;
10. Условие останова: алгоритм завершается, если выполнено условие сходимости, например, при достижении заданной точности или после определенного числа итераций.

Алгоритм метода Нелдера – Мида повторяется до достижения оптимального значения функции или условия останова.

# Идентификация функций, проходящих через метод

Основная идея метода заключается в том, что он использует только значения функции в различных точках пространства параметров, чтобы определить, в каком направлении двигаться к минимуму.

Рассмотрим функции, которые проходят тесты.

Функция Химмельблау — функция двух переменных, используемая для проверки эффективности алгоритмов оптимизации.

Она определяется формулой:

Функция Химмельбау содержит четыре равнозначных локальных минимума:

f(3, 2)=0;

f(-2,80511…, 3,13131…)=0;

f(-3,77931…, -3,28318…)=0;

f(3,58442…, -1,84812…)=0.

Создается лямбда-функция f, которая вычисляет значение функции в точке .

Функция calculate\_neldermead вызывается для поиска минимума функции Химмельблау в двумерном пространстве. Полученные координаты точки значение функции сохраняются.

Ожидаемая точка минимума равна [3, 2], а ожидаемое значение функции в этой точке равно 0.

Тест проверяет, что полученная точка близка к ожидаемой и что значение функции близко к нулю. Но мы проверяем приблизительные значения с точностью до пяти знаков после запятой.

Следующая функция – функция Функция Розенброка. Невыпуклая функция, используемая для оценки производительности алгоритмов оптимизации, предложенная Ховардом Розенброком в 1960 году. Считается, что поиск глобального минимума для данной функции является нетривиальной задачей.

Является примером тестовой функции для локальных методов оптимизации. Имеет минимум 0 в точке (1,1).

Функция определена следующим образом:

И также мы проверили простую квадратичную функцию:

Сначала в тесте определяется функция.

Затем создается лямбда-функция, которая вычисляет значение функции в точке x.

Функция calculate\_neldermead вызывается для поиска минимума этой функции в одномерном пространстве. Полученные координаты точки и значение функции сохраняются.

Ожидаемая точка минимума равна 1, а ожидаемое значение функции в этой точке равно 0.

Тест проверяет, что полученная точка близка к ожидаемой и что значение функции близко к нулю с помощью методов assert\_array\_almost\_equal и assertAlmostEqual из библиотеки NumPy.

При написании математических тестов стало ясно, что не все функции идеально подходят для его применения. Для повышения эффективности и увеличения шансов на точное нахождение минимума, нужно обратиться к различным алгоритмам. Например, в таких случаях часто применяют метод Монте-Карло или эволюционно-генетический алгоритм. Эти методы позволяют дополнительно оптимизировать процесс поиска минимума и обеспечивают более надежные результаты. Комбинирование различных подходов подтверждает значимость и многообразие методов оптимизации, их применение становится ключевым фактором в достижении желаемых результатов в решении сложных задач оптимизации.

Пример такой функции – функция Растригина. Нахождение минимума этой функции является достаточно трудной задачей из-за большой области поиска и большого количества локальных минимумов. Метод Нелдера-Мида является простым и эффективным методом оптимизации для невыпуклых функций без ограничений, но он не подходит для задач оптимизации с многими локальными минимумами. В таких случаях метод Нелдера-Мида может застрять и не дойти до глобального минимума.

Тест проверяет работу метода Нелдера-Мида для функции Растригина.

Вначале определяется функция Растригина, которая имеет формулу

Затем создается лямбда-функция, которая вычисляет значение функции в точке.

Далее вызывается функция по нахождению минимум функции Растригина в двумерном пространстве. Полученные координаты точки и значение функции сохраняются.

Ожидаемая точка минимума равна [5, 5], а ожидаемое значение функции в этой точке равно 0. Тест проверяет, что полученная точка близка к ожидаемой и что значение функции близко к нулю с помощью методов из библиотеки NumPy.

Если тест не проходит, он выводит сообщение об ошибке.

# Вывод

Метод Нелдера-Мида является эффективным инструментом для оптимизации функций без использования градиента. Его простота и эффективность делают его популярным выбором для различных задач оптимизации.

# Примечание

Ссылка на наш проект:

<https://www.google.com/url?q=https://github.com/DanilVolkov/lab-for-MO&sa=D&source=editors&ust=1715633126836508&usg=AOvVaw1khmjVrFlzfHyCjJDIqZyA>

# Источники:

1. <https://ru.wikipedia.org/wiki/Метод_Нелдера_—_Мида>
2. <https://www.mql5.com/ru/articles/13805>
3. <https://mathprofi.com/uploads/files/1156_f_41_metody-optimizacii-negladkih-funkcii.pdf?key=a94b32268f6014b85c9c48ef842cc5db/>
4. [https://ru.wikipedia.org/wiki/Функция\_Растригина](https://vk.com/away.php?to=https%3A%2F%2Fru.wikipedia.org%2Fwiki%2F%D4%F3%ED%EA%F6%E8%FF_%D0%E0%F1%F2%F0%E8%E3%E8%ED%E0&cc_key=)
5. <https://ru.wikipedia.org/wiki/Функция_Химмельблау>
6. https://ru.wikipedia.org/wiki/Функция\_Розенброка

1. Функция f(x) сходится к предельному значению L при x→a, если для любого положительного числа существует положительное число , такое что для всех x удовлетворяющих условиям 0<|x−a|<,, выполнено неравенство |f(x)−L|<. [↑](#footnote-ref-1)
2. Выпуклая оболочка множества — это наименьшее выпуклое множество, которое его содержит. Им будет пересечение всех выпуклых множеств, содержащих множество. [↑](#footnote-ref-2)