УДК 004.942

А.А. САРАНЦЕВА, Е.С. БОРОВИНСКАЯ

A.A. SARANTSEVA, E.S. BOROVINSKAYA

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ ОПТИМИЗАЦИИ В СОБСТВЕННОЙ РЕАЛИЗАЦИИ И ИЗ БИБЛИОТЕК PYTHON ДЛЯ РЕШЕНИЯ ОБРАТНЫХ ЗАДАЧ НА ПРИМЕРЕ ФУНКЦИИ РОЗЕНБРОКА**

**COMPARISON OF OPTIMIZATION METHODS IN OWN IMPLEMENTATION AND FROM PYTHON LIBRARIES FOR SOLVING INVERSE PROBLEMS ON THE EXAMPLE OF THE ROSENBROCK FUNCTION**

*В данной статье освещается проблема эффективности методов оптимизациив собственной реализации по сравнению с встроенными методами на примере функции Розенброка.*

*Ключевые слова: методы оптимизации, Метод Пауэлла, Метод Хука-Дживса, Метод наискорейшего спуска.*

*This paper covers the problem of the efficiency of own-implementation optimisation methods compared to embedded methods, using the Rosenbrock function as an example.*

*Key words: optimization methods, Powell method, Hook-Jeeves method, Steepest descent method.*

Обратные задачи − это задачи, которые требуют идентификации параметров или переменных системы на основе доступных данных. Решение обратных задач в химической кинетике имеет большое значение для понимания и оптимизации химических процессов. В основном обратные задачи возникают в тех случаях, когда необходимо определить параметры реакции на основе экспериментальных данных. Для решения таких задач используются методы оптимизации, которые позволяют находить наилучшие значения параметров реакции, минимизируя отклонение между экспериментальными и расчетными данными [1]. Цель данной работы заключается в сравнении эффективности различных методов оптимизации для их последующего использования при решении обратных задач химической кинетики. На первом этапе исследования рассматривается стандартная функция Розенброка, которая позволяет оценить производительность методов оптимизации на простой задаче [2].

Функция Розенброка − широко известная функция, часто используемая для тестирования алгоритмов оптимизации. Она представляет собой двумерную функцию, определенную на плоскости. Она имеет следующий вид:

 $f\left(x,y\right)=(a-x)^{2}+b(y-x^{2})^{2}$, (1)

где где a и b − константы, обычно принимающие значения a = 1 и b = 100.

Минимум функции Розенброка находится в точке (1, 1) и равен 0. Функция Розенброка имеет узкий и длинный "овраг" с глубоким и узким локальным минимумом внутри. Этот овраг и локальный минимум представляют трудности для некоторых методов оптимизации, включая методы наискорейшего спуска и Пауэлла. Если метод попадает внутрь оврага, он может начать зацикливаться между двумя точками около локального минимума, и таким образом, не сойтись к глобальному минимуму.



Рисунок 1 − График функции Розенброка

Метод Хука-Дживса и некоторые другие методы могут быть более эффективными при поиске глобального минимума функции Розенброка, так как они исследуют всю область определения, а не просто следуют направлению градиента.

Для нахождения минимума функции Розенброка были реализованы три метода оптимизации: градиентный спуск, метод Пауэлла, метод Хука-Дживса. Полученные результаты оптимизации сравнивали с результатами, полученными с помощью библиотеки Python SciPy, в которой также реализованы эти методы.

Метод наискорейшего спуска (англ. Gradient Descent) − это метод оптимизации, который использует направление антиградиента для поиска минимума функции. Он заключается в том, чтобы в каждой точке функции двигаться в направлении, противоположном направлению градиента, с определенным шагом (так называемым learning rate). Этот метод позволяет быстро достигнуть минимума функции, если необходимо найти локальный минимум или если локальный минимум одновременно является глобальным. Алгоритм может застрять в локальном минимуме [3].

Метод Пауэлла (англ. Powell's method) − это метод оптимизации, который использует для поиска минимума функции последовательность одномерных оптимизаций в направлениях, заданных базисными векторами. Он применяется для поиска минимума многомерной функции и не требует нахождения производных. Данный позволяет быстро определить минимум функции, если функция не имеет несколько локальных минимумов и имеет симметричную форму.

Метод Хука-Дживса (англ. Hooke-Jeeves method) − это метод оптимизации, который использует пошаговый поиск для поиска минимума функции. Поиск начинается с заданной начальной точки, на каждой итерации совершается два шага: первый шаг − исследовательский, позволяет определить лучшую точку в заданном направлении, второй шаг − поисковый, он осуществляется вокруг точки, найденной на первом шаге. Этот метод позволяет быстро сойтись к минимуму функции, если функция не имеет несколько локальных минимумов и имеет симметричную форму.

В библиотеке SciPy реализованы методы наискорейшего спуска, Пауэлла и Хука-Дживса в модуле scipy.optimize. В частности, для метода наискорейшего спуска используется функция scipy.optimize.minimize() с указанием метода CG (conjugate gradient method), для метода Пауэлла − функция scipy.optimize.minimize() с методом powell, а для метода Хука-Дживса − функция scipy.optimize.minimize() с методом Nelder-Mead. В качестве начального приближения во всех реализациях была взята точка (0;0).

В ходе работы были сформированы две таблицы с результатами применения рассмотренных алгоритмов оптимизации и сравнения их сходимости при различных критериях остановы (Таблица 1-2). В таблице 1 представлены результаты вычислений при разном количестве итераций. В таблице 2 рассматриваются влияние разных критериев остановки поиска − абсолютного и относительного отклонения на результаты нахождения минимума функции. В таблицах fmin − минимум функции и time − время выполнения расчета в секундах.

Таблица 1 − Сравнение методов оптимизации по количеству итераций

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий остановки поиска** | **Кол-во итераций N = 10** | **Кол-во итераций N = 100** | **Кол-во итераций N = 1000000** |
| **Метод оптимизации** | **fmin**  |  **time** | **fmin**  | **time** | **fmin**  | **time** |
| Наискорейший спуск (собственная реализация) | 0,995  | 1,088 | 0,952  | 0,903 | 0,001 | 30,045 |
| Наискорейший спуск (SciPy) | 0,048  | 0,751 | 0,000 | 0,633 | 0,000  | 0,649 |
| Пауэлл (собственная реализация) | - | - | 0,704 | 0,320 | - | - |
|
| Пауэлл (SciPy) | 0,001 | 3,963 | 0,000 | 0,758 | - | - |
| Хука-Дживс (собственная реализация) | 1,107  | 0,265 | 0,001  | 0,290 | 0,000  |  2,223 |
| Хука-Дживс (SciPy)  | 0.948 | 0,720 | 0,000  | 0,686 | -  | - |

Метод Пауэлла в собственной реализации при количестве итераций N = 100 находит минимум fmin = 0,704, то есть в данном случае наблюдается расхождение метода. Этот метод использует направление антиградиента, но вместо того, чтобы выбирать его направление каждую итерацию, он вычисляет матрицу, которая позволяет выбирать направления, соответствующие независимым координатам пространства переменных. Если матрица становится слишком близкой к вырожденной, метод может потерять эффективность и разойтись. Реализация из библиотеки Scipy дала лучший результат, это связано с тем, что в этой реализации метода Пауэлла используется "полное обновление" вектора шага, а не "диагональное обновление", которое используется в классическом алгоритме. Также в SciPy реализована более эффективная процедура поиска начального приближения для метода, что позволяет ускорить сходимость. Однако, при снижении количества итераций до N = 10 реализация с помощью библиотеки SciPy становится менее эффективной и точной.

Метод наискорейшего спуска сошелся при количестве итераций N = 1000000 и выдал результат fmin = 0,001 за 30,045 секунд. При критериях остановки поиска количество итераций N = 10 и N = 100, этот метод оказался неэффективен. Метод, реализованный в библиотеке Scipy оказался также более эффективным. При малом количестве итераций он является менее точным и работает медленнее, возвращая результат fmin = 0,048 за 0,751 секунды.

Метод Хука-Дживса в собственной реализации сошелся при всех критериях остановки поиска, кроме малого количества итераций N = 10. Также видно, что этот метод разошелся при малом количестве итераций и в реализации из библиотеки SciPy при fmin = 0,948.

Из таблицы 2 также видно, что метод Пауэлла в собственной реализации расходится, чего не происходит при использовании библиотеки SciPy. Метод наискорейшего спуска, реализованный в библиотеке SciPy сходится и время его выполнения составляет примерно 0,6 секунд. Метод Хука-Дживса сходится в обоих случаях, причем в собственной реализации он эффективнее в среднем на 0,3 секунды.

Таблица 2 − Сравнение методов оптимизации по критериям остановы: абсолютное отклонение, относительное отклонение

|  |  |  |  |
| --- | --- | --- | --- |
| **Метод оптимизации** | **Критерий остановы** | **Полученный минимум fmin** | **Время выполнения, сек** |
| Наискорейший спуск (собственная реализация) | **Абсолютное отклонение xtol = 1·10-6** | 0,981 | 1,102 |
| Наискорейший спуск (SciPy) | **Абсолютное отклонение xtol = 1·10-6** | 0,000  | 0,678 |
| Пауэлл (собственная реализация) | **Абсолютное отклонение xtol = 1·10-6** | 0,704  | 0,278 |
| Пауэлл (SciPy) | **Абсолютное отклонение xtol = 1·10-6** | 0,000 | 0,658 |
| Хука-Дживс (собственная реализация) | **Абсолютное отклонение xtol = 1·10-6** |  0,001  | 0,326 |
| Хука-Дживс (SciPy) | **Абсолютное отклонение xtol = 1·10-6** |  0,000  | 0,718 |
| Наискорейший спуск (собственная реализация) | **Относительное отклонение ftol = 1·10-6** |  0,999  | 1,057 |
| Наискорейший спуск (SciPy) | **Относительное отклонение ftol = 1·10-6** |  0,000  | 0,674 |
| Пауэлл (собственная реализация) | **Относительное отклонение ftol = 1·10-6** |  0,704  | 0,259 |
| Пауэлл (SciPy) | **Относительное отклонение ftol = 1·10-6** |  0,000  | 0,753 |
| Хука-Дживс (собственная реализация) | **Относительное отклонение ftol = 1·10-6** |  0,000  | 0,265 |
| Хука-Дживс (SciPy) | **Относительное отклонение ftol = 1·10-6** |  0,000  | 0,705 |

В данной работе проанализированы такие методы, как: метод наискорейшего спуска, метод Пауэлла и метод Хука-Дживса и их признаки сходимости на примере функции Розенброка. Численные эксперименты подтверждают, что метод Хука-Дживса в собственной реализации имеет самую быструю скорость сходимости для функции Розенброка. Метод Хука-Дживса из библиотеки Scipy является менее эффективным, это может быть связано с тем, что он реализован для общего случая и может быть использован для решения широкого спектра задач оптимизации. Это делает его универсальным, но также накладывает определенные ограничения на его эффективность для конкретных задач.

Выбор оптимального метода для решения обратной задачи зависит от многих факторов, таких как размерность пространства параметров, форма функции ошибок, наличие локальных экстремумов и шума в данных.

Метод Пауэлла хорошо работает в случаях, когда функция ошибки гладкая и имеет несколько локальных экстремумов, но нет больших узких локальных минимумов. Он может быть эффективным для поиска глобального минимума в многомерном пространстве, если начальное приближение близко к истинному значению. Однако, если матрица, используемая методом Пауэлла, становится близкой к вырожденной, метод может потерять эффективность и разойтись.

Метод Хука-Дживса более эффективным, когда функция ошибки не гладкая и имеет несколько локальных экстремумов. Этот метод применяет пошаговый поиск, перемещаясь по пространству параметров с использованием разных шагов, что позволяет находить глобальный минимум. Однако, если пространство параметров очень большое, этот метод может оказаться слишком затратным по времени.

Метод наискорейшего спуска хорошо работает в случаях, когда функция ошибки гладкая и имеет только один локальный минимум. Он использует информацию о градиенте функции ошибок для определения направления движения. Этот метод может быть более эффективным, когда пространство параметров большое и существует только один минимум. Однако, если функция ошибок имеет несколько локальных минимумов, метод наискорейшего спуска может остановиться в локальном минимуме.

Из результатов исследования можно сделать вывод, что метод Пауэлла в собственной реализации оказаллся менее эффективным, чем из библиотеки Scipy, что связано с потерей эффективности при вырожденных матрицах. Метод наискорейшего спуска работал эффективно только при критерии остановки N = 1000000, что делает его менее универсальным.

Метод Хука-Дживса, как собственная реализация, так и из библиотеки Scipy, показал стабильную работу при различных критериях остановки поиска. Однако, при малом количестве итераций он может потерять эффективность и разойтись.

Также была проведена аналитическая оценка сходимости методов и исследование их времени выполнения. В целом, реализации из библиотеки Scipy показали лучшие результаты в сравнении с собственными реализациями, это обусловлено тем, что методы оптимизации, реализованные в библиотеках, обычно проходят более тщательное тестирование и оптимизацию, чем собственные реализации. Библиотечные методы могут использовать более сложные и эффективные алгоритмы, оптимизированные под различные типы задач и их особенности. Кроме того, библиотеки обычно содержат дополнительные функции и возможности, такие как анализ градиента, управление шагом и т.д. В целом, использование библиотечных методов может значительно повысить эффективность и точность решения оптимизационных задач.

**СПИСОК ЛИТЕРАТУРЫ**

1. Bardsley, J. Inverse Problems: Tikhonov Theory and Algorithms / J. Bardsley. - Philadelphia: Society for Industrial and Applied Mathematics, 2001. - 150 с.
2. Рохлин М.Я., Катышева А.А. Оптимизация. Теория и практика / М.Я. Рохлин, А.А. Катышева. - М.: Издательский дом "Физико-математическая литература", 2011. - 304 с.
3. Нестеров Ю. Е. Методы выпуклой оптимизации / Ю. Е. Нестеров. - М.: Издательский дом "МЦНМО", 2010. - 304 с.

**Саранцева Алина Александровна**

Санкт-Петербургский государственный технологический институт (технический университет), г. Санкт-Петербург

Студентка кафедры системного анализа и информационных технологий

Тел.: +7(931)-239-88-81

E-mail: alina.logan2017@yandex.ru

**Боровинская Екатерина Сергеевна**

Санкт-Петербургский государственный технологический институт (технический университет), г. Санкт-Петербург

Д.т.н., доцент, профессор кафедры системного анализа и информационных технологий

Тел.: +7 (812) 494-93-02

E-mail: ekaterina.borovinskaya@daad-alumni.de