**Алгоритм глобальной оптимизации, использующий деревья решений для выявления локальных экстремумов[[1]](#footnote-1)**

Д. И. Силенко, И. Г. Лебедев

Нижегородский государственный университет им. Н. И. Лобачевского

В работе рассматривается решение задач многомерной глобальной оптимизации с применением деревьев решений для выявления областей притяжения локальных минимумов. Локальный метод используется для локального уточнения и призван ускорить сходимость алгоритма. Искомая функция удовлетворяет условию Липшица с неизвестной константой. Для подтверждения теории были приведены вычислительные эксперименты, демонстрирующие ускорение при решении серии тестовых задач.

*Ключевые слова***:** глобальная оптимизация, локальная оптимизация, многоэкстремальные функции, деревья решений.

# Введение

При поиске глобального минимума функции можно придерживаться различных алгоритмов. Поскольку в реальных задачах глобальной оптимизации каждое вычисление значения функции представляет собой весьма трудоемкую задачу, необходимо уменьшить количество таких операций. Этого можно добиться целенаправленным выбором вариантов в процессе поиска оптимального решения. На этой идее основывается алгоритм глобального поиска (АГП) [**Ошибка! Источник ссылки не найден.**]. В данной работе мы объединим АГП и метод Хука-Дживса (метод локальной оптимизации) с помощью деревьев решений, с целью уменьшения количества проводимых итераций, а вместе с этим и количества вычислений целевой функции. В разделе «Эксперименты» можно найти результаты сравнения такого подхода и стандартного АГП.

# Постановка задачи

Рассмотрим задачу поиска глобального минимума функции в гиперинтервале . При этом будем предполагать, что функция удовлетворяет условию Липшица с априори неизвестной константой L.

Численное решение задачи () сводится к построению оценки , отвечающей некоторому понятию близости к точке на основе конечного числа вычислений значений оптимизируемой функции. Оптимизируемая функция  может быть задана не аналитически, а алгоритмически, как результат работы некоторой подпрограммы или библиотеки.

Поиск глобального оптимума сводится к построению некоторой сетки в области параметров, и выборе наилучшего значения функции на данной сетке. Вычислительные затраты на решение задачи растут экспоненциально с ростом размерности. В ННГУ им. Н.И. Лобачевского под руководством проф. Р.Г. Стронгина разработан эффективный подход к решению задач глобальной оптимизации. Решение многомерных задач сводится к решению серии вложенных задач меньшей размерности.

# Описание алгоритмов

* 1. **Многомерный алгоритм глобального поиска**

При решении многомерных задач применяется редукция размерности с использованием кривых Пеано. Они позволяют свести многомерную задачу минимизации в области к одномерной задаче минимизации на отрезке [0, 1]

где функция удовлетворяет уже равномерному условию Гельдера

Поэтому, не ограничивая общности, можно рассматривать минимизацию одномерной функции , удовлетворяющей условию Гельдера.

На начальной итерации алгоритма проводятся испытания на границах отрезка [0, 1] : и . Далее на каждой итерации мы имеем набор интервалов ограниченных точками испытания, для каждого интервала вычисляется характеристика, а в интервале с наибольшей характеристикой производится новое испытание. Итерации повторяются, пока не достигнут один из критериев остановки [2].

Схема алгоритма:

1. Упорядочить точки по координате:

1. Вычислить оценку  *μ* для неизвестной константы Липшица *L*,

1. Для каждого , вычислить характеристику

где − корень степени из длины интервала, , − параметр метода.

1. Определить номер :

Провести очередное испытание в точке из интервала ()

Алгоритм прекращает работу, если выполняется условие хотя бы для одного номера ; здесь есть заданная точность. В качестве оценки глобально-оптимального решения задачи (3) выбираются значения

Обоснование данного способа организации вычислений см. в [12]. Модификации, учитывающие наличие ограничений-неравенств в задаче, а также информация о производной целевой функции, представлены в [4, 3].

* 1. **Метод Хука-Дживса**

Метод Хука–Дживса – это комбинация исследующего поиска по направлениям и поиска по образцу [6, 7].

Суть исследующего поиска: задается величина шага. Если значение целевой функции в пробной точке не превышает значение в исходной, то шаг - успешный. Иначе, необходимо попробовать сместиться в другое направление. Когда перебор всех координат закончится, то считаем найденную точку базовой и заканчиваем исследующий поиск. Поиск по образцу реализует шаг из базовой точки вдоль прямой, соединяющей ее с предыдущей базовой точкой. Величина этого шага определяется задаваемым заранее параметром.

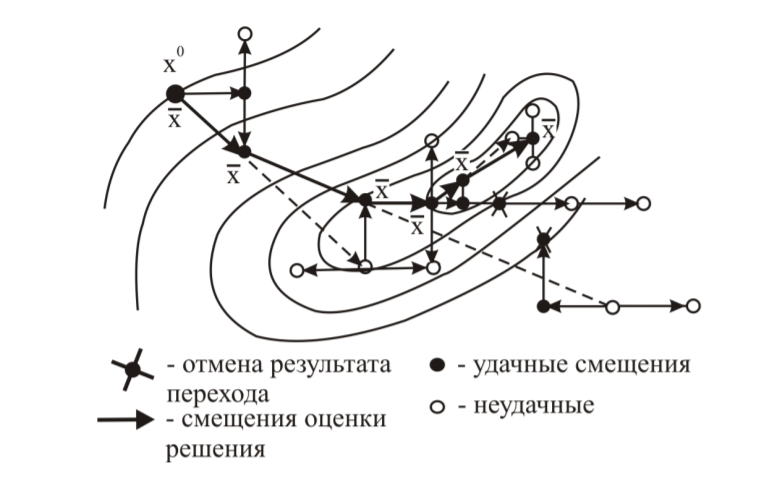


Рисунок 1 Пример итераций алгоритма Хука-Дживса

* 1. **Деревья решений**

Дерево решений — это метод автоматического анализа больших массивов данных, который применяется в машинном обучении. Оно представляет собой бинарное дерево и его можно использовать как для решения задач классификации, так и для регрессионных задач.

Подробное описание работы деревьев решений мы здесь опустим. Это связано с тем, что мы их не реализовывали, а использовали готовые из библиотеки OpenCV (это библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом). Подробнее с деревьями решений можно ознакомиться в [8].

* 1. **Объединение метода Хука-Дживса и АГП в случае многомерных задач**

Необходимо понять: запускать локальный метод из текущей точки или нет. Для этого можно осмотреть соседние точки и если среди них нет ни одной такой, значение функции в которой меньше, чем в текущей, то можно предположить, что мы находимся в области локального спуска и запустить локальный метод. Использование кривых Пеано здесь нецелесообразно, т.к. вид функции после отображения может сильно изменится, а кроме того, мы можем потерять информацию о взаимном расположении точек исходной функции. Поэтому мы будем работать именно с исходными точками.

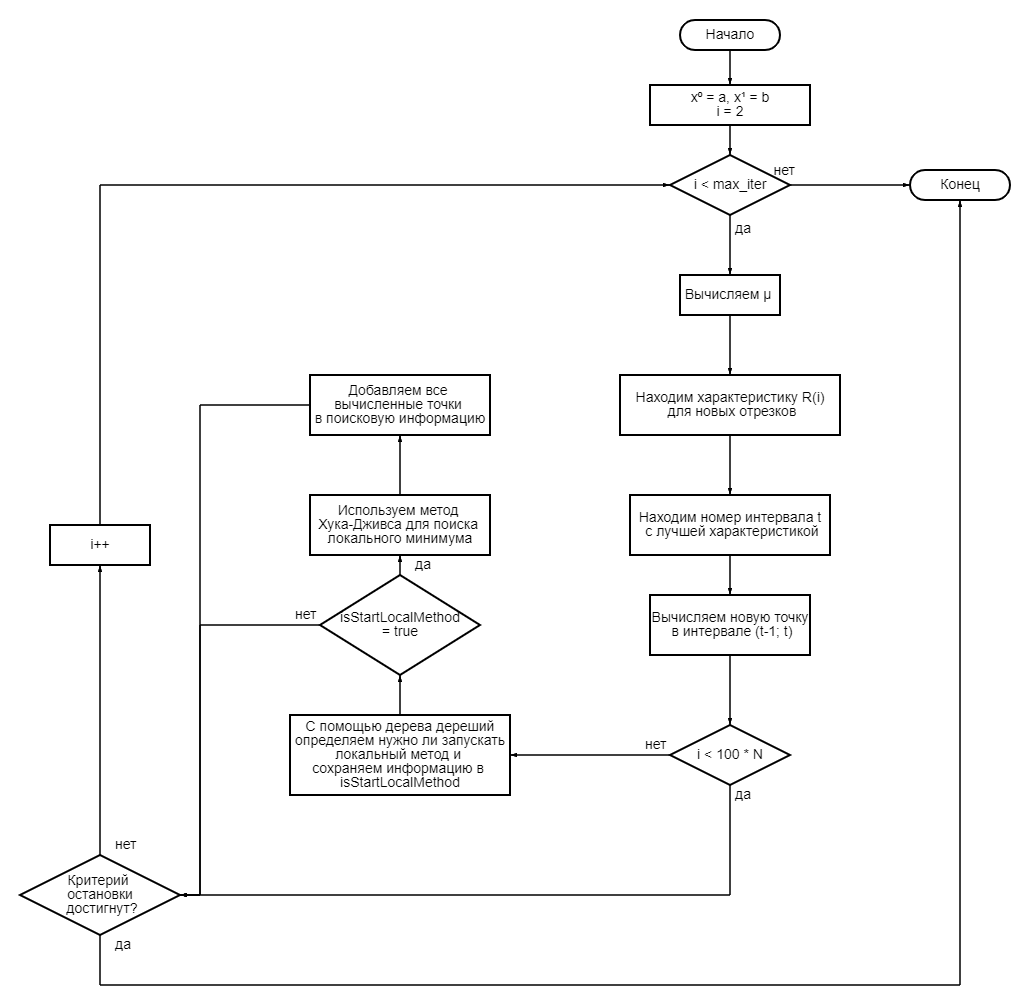
После проведения определенного числа испытаний, используем все накопившиеся точки для тренировки дерева решений. Далее построим равномерную сетку с определенным шагом по каждой из размерностей, а ее узлы подадим на вход предсказанию по дереву, чтобы получить значения функции в них. Теперь найдем ближайшую, с точки зрения евклидова расстояния, точку к исходной и определим ее соседей перебором индексов. Если нашли значение функции меньшее, чем в текущей точке, то продолжаем работу АГП; если нашли такое же значение функции, то необходимо аналогично проверить такую точку; иначе, запускаем локальный метод.

Модифицированная версия алгоритма глобального поиска выглядит следующим образом, в начале выполняются 4 этапа классического алгоритма, затем:

1. Если , то продолжаем работу АГП.
2. Если используем деревья решений не впервые, то переходим к пункту 13.
3. Инициализируем дерево решений и тренируем его.
4. Строим равномерную сетку, откладывая по каждой координате точек
5. Предсказываем значения функции в узлах сетки с помощью дерева решений.
6. Находим ближайшую с точки зрения Евклидова расстояния точку к текущей.
7. Обходим соседей.
8. Если не нашли точек со значением функции меньше, то запускаем метод Хука-Дживса.
9. Если запускаем алгоритм не впервые, то накапливаем массив из точек.
10. Все 100\*N точек обрабатывает аналогично пункту 7-12, но без пункта 8.

# Программная реализация

Все полученные результаты были добавлены в систему globalizer [10]. Была реализована объединенная схема с запуском локальных методов, для чего была внедрена система построения дерева решений, равномерной сетки и прочих сопутствующих вычислений. Для всего этого были добавлены новые методы в уже существующие классы: в классе TMethod метод UpdateStatus теперь выполняет проверку о запуске локального метода, а метод LocalS производит запуск выбранного локального метода. Новый метод UpdateStatusDecisionTreesMultiDims реализует всю логику, связанную с деревьями решений. Ниже приведена блок-схема того, как данная схема была объединена с АГП.



*Рисунок 2 Блок-схема объединения АГП и метода Хука-Дживса с использованием дерева решений*

# Эксперименты

Вычислительные эксперименты проводились на персональном компьютере со следующими характеристиками: Операционная система Windows 11, Процессор: Intel(R) Core™ i5-8250U CPU @ 1.60 GHz, 12 Gb RAM.

В работе [5, 9] описан GKLS-генератор, позволяющий порождать задачи многоэкстремальной оптимизации с заранее известными свойствами: количеством локальных минимумов, размерами их областей притяжения, точкой глобального минимума, значением функции в ней и т.п.

В таблице 1 представлено среднее количество итераций, и среднее число точек испытаний, проводимое двумя методами: алгоритмом глобального поиска (АГП) и объединением АГП и локального метода с использованием деревьев решений (Деревья решений) на задачах разной размерности (N) и разного класса (Класс задачи). Значение в скобках (если оно есть) показывает количество не решенных задач.

**Таблица 1.** Среднее число итераций и среднее число испытаний, проводимое разными алгоритмами

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | Класс задачи | Среднее число итераций | | Среднее число точек испытаний | |
| АГП | Деревья решений | АГП | Деревья решений |
| 2 | Simple | 2350.0 | 247.4 | 2348.0 | 390.0 |
| Hard | 4732.3 | 353.6 | 4730.3 | 1101.1 |
| 3 | Simple | 2115.5 | 579.5 | 2113.5 | 2441.4 |
| Hard | 5347.4 | 588.1 | 5345.4 | 2532.7 |
| 4 | Simple | 12168.9 | 768.2 | 12166.9 | 3582.2 |
| Hard | 25636.5 | 960.0 | 52634.5 | 5171.6 |
| 5 | Simple | 20633.5 | 978.0 | 20631.5 | 4840.4 |
| Hard | 161094.9 (4) | 1217.0 | 161092.9 (4) | 7115.5 |

Итак, ускорение по числу итераций неоспоримое, причем на задачах любой размерности. Среднее число точек испытаний тоже, практически везде, уменьшилось, но эти значения могут быть сильно выше, чем среднее число итераций. Это связано с тем, что в точках испытаний учитываются и те точки, которые в процессе своей работы ставил локальный метод, а их может быть довольно много.

# Заключение

В результате работы удалось реализовать схему объединения АГП и метода локальной оптимизации при использовании деревьев решений. Такой подход позволяет ускорить схождение к локальному минимуму, если это целесообразно, что позволяет уменьшить общее число проводимых итераций при работе АГП. Вычислительные эксперименты подтвердили, что это дает заметное ускорение на задачах разной размерности.

# Литература

1. Стронгин Р.Г., Гергель В.А., Гришагин В.А., Баркалов К.А. Параллельные вычисления в задачах глобальной оптимизации. М.: Издательство Московского университета, 2013. 280 с.
2. Шефов К. С., Степанова М. М. Реализация и применение параллельного алгоритма глобального поиска минимума к задаче оптимизации параметров молекулярно-динамического потенциала ReaxFF: URL: <http://crm.ics.org.ru/uploads/crmissues/crm_2015_3/15750.pdf>, 2015 (дата обращения: 13.10.2020).
3. Gergel V.P. A global optimization algorithm for multivariate functions with lipschitzian first derivatives / Gergel V.P. // Journal of Global Optimization. – 1997. – Vol. 10, No. 3. – P. 257-281.
4. Barkalov K.A. A global optimization technique with an adaptive order of checking for constraints / Barkalov K.A., Strongin R.G. // Computational Mathematics and Mathematical Physics. – 2002. – Vol. 42, No. 9. – P. 1289-1300.
5. Gaviano, M. Software for generation of classes of test functions with known local and global minima for global optimization/ M. Gaviano, D. Lera, D. E. Kvasov, Y. D. Sergeyev // ACM Transactions on Mathematical Software. – 2003. – Vol. 29. – P. 469-480.
6. *Химмельблау Д.* Прикладное нелинейное программирование // М.: Мир - 1975 - 536 с.
7. *Nelder J., Mead R*. A simplex method for function minimization // Computer Journal - 7(4) – 1965 - P. 308-313.
8. OpenCV (Open Source Computer Vision) documentation. URL: <https://docs.opencv.org/4.x/dc/dd6/ml_intro.html> (дата обращения: 10.07.2022)
9. Сергеев, Я.Д. Диагональные методы глобальной оптимизации / Я.Д. Сергеев, Д.Е. Квасов – М.: Физматлит, 2008. – 352 c.
10. *Sysoyev, A., Barkalov, K., Sovrasov, V., Lebedev, I., Gergel, V.* Globalizer – A parallel software system for solving global optimization problems. Lecture Notes in Computer Science. Volume 10421 LNCS, 2017, Pages 492-499
11. Sergeyev, Ya.D. Global search based on efficient diagonal partitions and a set of Lipschitz constants / Ya.D. Sergeyev, D.E. Kvasov // SIAM Journal on Optimization. – 2006. Vol. 16, No. 3. – P. 910–937
12. Стронгин Р.Г. Параллельные вычисления в задачах глобальной оптимизации / Стронгин Р.Г., Гергель В.П., Гришагин В.А., Баркалов К.А. – М.: Издательство Московского университета, 2013. 280 с.

1. Работа выполнена при финансовой поддержке Министерства науки и высшего образования РФ (проект № 0729-2020-0055) и научно-образовательного математического центра “Математика технологий будущего” (соглашение № 075-02-2021-1394). [↑](#footnote-ref-1)