**Совместное решение алгоритмами глобальной и локальной оптимизации, задач глобальной оптимизации[[1]](#footnote-1)**

Д. И. Силенко, И. Г. Лебедев

Нижегородский государственный университет им. Н. И. Лобачевского

Рассматриваются задачи одномерной оптимизации и локальные методы поиска минимума. Об оптимизируемой функции делается лишь общее предположение, что она удовлетворяет условию Липшеца с априори неизвестной константой. Локальные методы являются локальным уточнением для алгоритма глобального поиска и призваны улучшить его работу. Проведены численные эксперименты на тестовых задачах, подтверждающие что использование локального уточнения уменьшает число вычислений значений функции при решении задачи глобальной оптимизации.

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

# Введение

При поиске глобального минимума функции можно придерживаться различных алгоритмов: от основанных на идее случайного поиска [3, 4, 9], до детерминированных алгоритмов гарантирующих сходимость к глобальному минимуму [14, 8, 11]. 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) [12]. Но в данной работе речь пойдет об объединении АГП с локальными методами оптимизации.

Алгоритмы поиска локального экстремума предназначены для определения одного из локальных экстремумов на множестве допустимых решений, в котором целевая функция принимает максимальное или минимальное значение. При их построении могут использоваться как детерминированный спуск в область экстремума, так и случайный поиск. Среди детерминированных различают методы нулевого порядка и градиентные (1-го и 2-го порядка) [13, 15].

В данной работе используется метод Хука–Дживса (он будет описан ниже), а также построение аппроксимации точек параболой и нахождение ее минимума. Аппроксимация точек – это метод, основанный на построении функции, наиболее близко проходящей через исходные данные. Соответственно, существует два способа определения такой функции – построение интерполяционного многочлена n-ной степени (в нашем случае параболы), который проходит либо непосредственно через все заданные точки, либо в ближайшей окрестности от них. Для нашей задачи подойдет второй вариант.

Таким образом, нам необходимо найти уравнение параболы, проходящей в окрестности заданных точек, для его построения воспользуемся методом наименьших квадратов (МНК)

МНК – математический метод, основанный на определении аппроксимирующей функции, которая строится в ближайшей окрестности от точек из заданного массива экспериментальных данных. Близость исходной и аппроксимирующей функции определяется числовой мерой: сумма квадратов отклонений данных от аппроксимирующей кривой должны быть наименьшей [16].

# Постановка задачи

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

(1)

(2)

# Описание алгоритмов

* 1. **Алгоритм глобального поиска**

Основная идея данного подхода заключается в том, что минимизируемая функция рассматривается как реализация некоторого случайного процесса. Решающие правила алгоритма конструируются таким образом, что очередная итерация проводится в точке глобального минимума математического ожидания значений функции. Эта точка записывается в список известных значений и итерации повторяются. Так происходит до тех пор, пока не достигнут один из выбранных критериев остановки: расстояние между точками отрезка не становится меньше заданного значения или новые точки не попадают в окрестность истинного глобального минимума. Еще одним вариантом остановки работы алгоритма является достижение установленного максимума возможных итераций. [19]

Схема алгоритма:

Первые два испытания:и *.*

1. Упорядочить точки по координате:

(3)

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

(4)

1. Для каждого , вычислить характеристику

(5)



где Δ*i* − корень степени N из длины интервала, , − параметр метода.

1. Определить номер *t*:

} (6)

Провести очередное испытание в точке из интервала ()

(7)

Алгоритм прекращает работу, если выполняется условие хотя бы для одного номера ; здесь есть заданная точность. В качестве оценки глобально-оптимального решения задачи (1) выбираются значения

(8)

Обоснование данного способа организации параллельных вычислений см. в [17]. Модификации, учитывающие наличие ограничений-неравенств в задаче, а также информация о производной целевой функции, представлены в [1, 13, 6].

* 1. **Метод Хука-Дживса**

Метод Хука–Дживса – это комбинация исследующего поиска по направлениям и поиска по образцу [18, 10].

Исследующий поиск определяется следующим образом. Задаётся величина шага, которая может быть разной для разных координатных направлений и изменяться в процессе поиска. Если значение целевой функции в пробной точке не превышает значение в исходной, то шаг поиска рассматривается как успешный. В противном случае, необходимо вернуться в предыдущую точку и сделать шаг в противоположном направлении. После перебора всех координат исследующий поиск заканчивается. Полученная точка называется базовой.

Поиск по образцу заключается в реализации единственного шага из полученной базовой точки вдоль прямой, соединяющей её с предыдущей базовой точкой.

* 1. **Метод с использованием квадратичной аппроксимации**

Аппроксимирующая функция по МНК определяется из условия минимума суммы квадратов отклонений расчетной аппроксимирующей функции от заданного массива экспериментальных данных [16]. Записать это можно в следующем виде

(9)

Аппроксимирующая функция представляет из себя многочлен степени m (в нашем случае )

(10)

Заметим, что нам необходимо найти минимум функции (9), а необходимым условием его существования является равенство нулю ее частных производных по всем . Если расписать данное условие (предварительно подставив 10 в 9), получим

(11)

В данной системе раскроем скобки и перепишем ее в матричном виде

(12)

(13)

Здесь неизвестными являются значения . А решив систему мы как раз их и найдем. Напомним еще раз, что для нашей задачи , что позволяет нам легко найти минимальное значение построенной аппроксимации просто как вершину параболы. А также, поскольку мы рассматриваем построение параболы, то система линейных алгебраических уравнений, получаемая в методе наименьших квадратов, состоит лишь из трех уравнений. Для решения подобных задач не нужно производить много сложных вычислений, даже определитель считается относительно не сложно. Это, в свою очередь, позволяет найти новую точку локального минимума без больших вычислительных затрат.

Критерием остановки метода служит попадание новой найденной точки в эпсилон окрестность точки, вычисленной на предыдущей итерации. Есть два варианта, что делать с новой полученной точкой: заменять точку с наибольшим значением функции на эту новую и продолжать строить аппроксимацию все так же по пяти точкам или же просто добавлять это значение к текущим, тем самым увеличивая число точек в аппроксимации.

* 1. **Объединение локального метода и АГП**

Разобравшись с принципом работы локального метода, остается лишь понять в какой момент его запускать. Для этого у каждой точки, в процессе работы АГП, будем запоминать и при необходимости обновлять метку: является точка возрастающей, убывающей, верхним или нижним перегибом. Метки расставляются в зависимости от значений функции в соседних точках. Так, если у точки слева значение функции больше, чем у текущей, а справа – меньше, то точка убывающая и т.д. Запускать локальный метод будем если найдено 5 точек: точка, подозрительная на нижний перегиб, а так же две точки слева от нее и две справа.

В процессе своей работы локальный метод в любом случае вычисляет функцию в нескольких точках, а как говорилось ранее это трудоемкая операция. Чтобы избежать повторного вычисления в таких случаях, все значения, полученные в методе, добавляется в «поисковую информацию» [2].

# Эксперименты

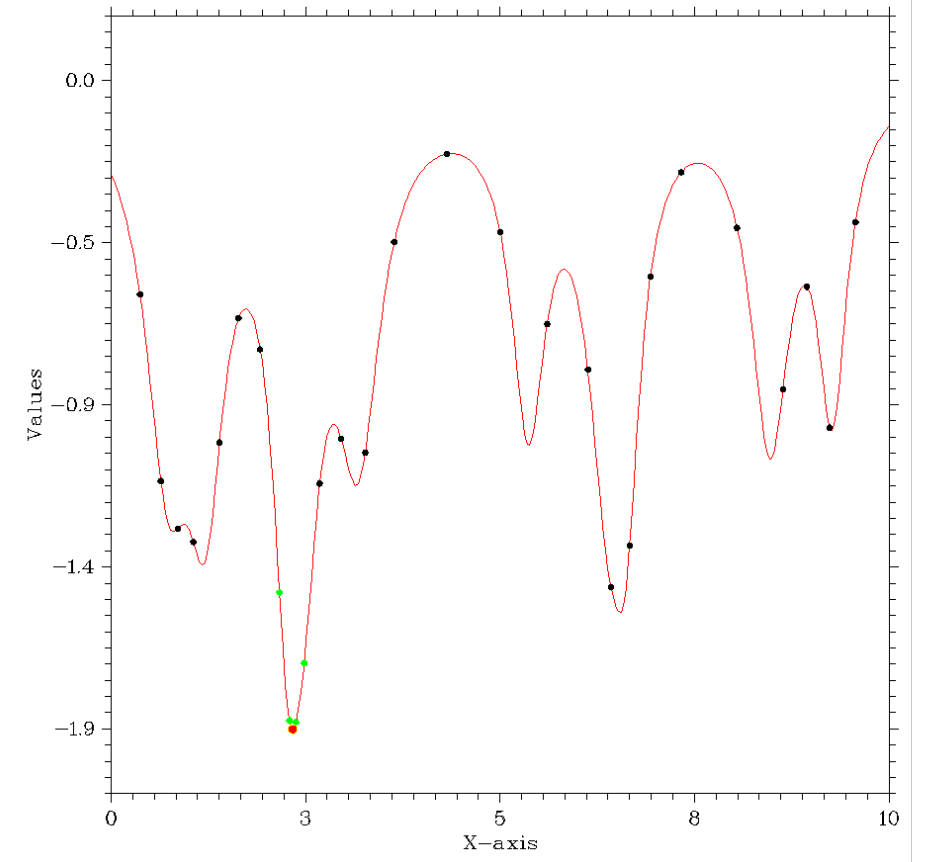
Эксперименты проводились на ПК с следующими параметрами:

Операционная система: Windows 10 Домашняя; Процессор: Intel(R) Core™ i5-8250U CPU @ 1.60 GHz; Версия Visual Studio: 2017.

В работе [5] описан GCGen-генератор, позволяющий порождать задачи многоэкстремальной оптимизации с заранее известными свойствами: размерностью задачи, точкой глобального минимума, значением функции в ней и т.п.

Испытания будем проводить на функции Шекеля. Тестовая функция Шекеля – это многомерная, много модальная, непрерывная, детерминированная функция, выглядящая, в нашем случае, следующим образом (рисунок 1)

(14)



**Рис. 1.** Пример вида функции Шекеля

В таблице 1 представлены значения среднего числа испытаний при решении 100 задач Шекеля при разных значения точности поиска. Задачи решались алгоритмом глобального поиска (АГП), объединённым алгоритмом глобального поиска с методом Хука-Дживса (Хука-Дживса) и с использованием квадратичной аппроксимации (Аппроксимация).

**Таблица 1.** Среднее число испытаний при разной точности

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **0,0001** | **0,0005** | **0,001** | **0,005** | **0,01** |
| **АГП** | 106,3 | 63,7 | 53,2 | 36,6 | 30,9 |
| **Хука-Дживса** | 39,3 | 38,1 | 35,3 | 33,5 | 32,0 |
| **Аппроксимация** | 38,4 | 35,4 | 33,9 | 31,4 | 29,9 |

# Заключение

В результате работы удалось успешно объединить алгоритм глобального поиска вместе с поиском локального минимума путем построения аппроксимации параболой и методом Хука-Джива. Это позволило сильно уменьшить количество вычислений целевой функции при необходимой большой точности поиска. С целью экспериментального подтверждения теоретических свойств рассматриваемого алгоритма проведены вычислительные эксперименты на серии из сотни тестовых задач.

# Литература

1. 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.
2. Barkalov K.A., Lebedev I.G., Kocheganova M.A., Gergel V.P. Combining local and global search in a parallel nested optimization scheme Communications in Computer and Information Science Volume 1263 CCIS, 2020, P. 100-112
3. Ferreiro A.M., Garcia J.A., Lopez-Salas J.G., Vazquez C. An efficient implementation of parallel simulated annealing algorithm in GPUs // Journal of global optimization. — 2013. — Vol. 57, No. 3. — P. 863–890.
4. Garcia-Martinez J.M., Garzon E.M., Ortigosa P.M. A GPU implementation of a hybrid evolutionary algorithm: GPuEGO // Journal of super-computing. — 2014. — Vol. 70, No. 2. — P. 684–695.
5. Gergel V.P. Barkalov, K.A., Sysoyev A.V., Rachinskaya M.A., Lebedev, I.G., A flexible generator of constrained global optimization test problems AIP Conference Proceedings Volume 2070, 12 February 2019, Номер статьи 20009
6. 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.
7. Gergel V.P. A method of using derivatives in the minimization of multiextremum functions / Gergel V.P. // Computational Mathematics and Mathematical Physics. – 1996. – Vol. 36, No. 6. – P. 729 – 742.
8. He J., Verstak A., Watson L.T., Sosonkina M. Design and implementation of a mas-sively parallel version of DIRECT // Computational optimization and applications. — 2008. — Vol. 40, No. 2. — P. 217–245.
9. Langdon W.B. Graphics processing units and genetic programming: an overview // Soft Computing. — 2011. — Vol. 15, No 8. — P. 1657–1669.
10. Nelder J., Mead R. A simplex method for function minimization. Computer Journal 7(4), 308{313 (1965)
11. Paulavicius R., Žilinskas J., Grothey A. Parallel branch and bound for global optimiza-tion with combination of Lipschitz bounds // Optimization methods and software. — 2011. — Vol. 26, No. 3. — P. 487–498.
12. Глобальная оптимизация: приложения и вычислительная сложность. URL: <http://hpc-education.unn.ru/ru/globopt/глобальная-оптимизация-приложения-и> (дата обращения: 10.10.2020).
13. Городецкий С. Ю. Методические материалы к лабораторной работе «Вычислительные методы поиска локальных минимумов функций», 2001. URL: <http://itmm.unn.ru/files/2016/09/MO_Lab2_LocOpt.pdf> (дата обращения: 09.10.2020).
14. Евтушенко Ю.Г., Малкова В.У., Станевичюс А.А. Параллельный поиск глобального экстремума функций многих переменных // Ж. вычисл. матем. и матем. физ. — 2009. — Т. 49, №2. — С. 255–269.
15. Захарова Е. М., Минашина И. К. Обзор методов многомерной оптимизации. Информационные процессы, том 14, №3, 2014, С. 256-274.
16. Моделирование в электроэнергетике: Аппроксимация опытных данных. Метод наименьших квадратов. URL: <http://simenergy.ru/math-analysis/digital-processing/85-ordinary-least-squares> (дата обращения: 02.10.2020).
17. 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.
18. Химмельблау Д. Прикладное нелинейное программирование. М.: Мир, 1975. 536 с.
19. 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).

1. Исследование выполнено в рамках государственного задания (0729-2020-0055), а также при финансовой поддержке научно-образовательного математического центра (075-02-2020-1483/1). [↑](#footnote-ref-1)